

Diary of a Game 1

URIDIUM 2

Andrew Braybrook, famed for such Amiga classics as Paradroid 90, Rainbow Islands and Fire & Ice, has recently started work on his latest project - Uridium 2. A sequel to his blockbuster Commodore 64 blaster, Uridium 2 looks set to be his best Amiga work yet. This month The One EXCLUSIVELY begins serialising his no-holds-barred development diary.

Month by month and in his own words, Andrew will take you through every step of the game's development, from conception this month to completion - hopefully - at the end of the year. As the work progresses, you'll be privy to the latest screenshots, along with Andrew's most intimate thoughts, feelings and programming tricks. You thought the Princess Di serialisation was big? You just haven't seen ANYTHING yet! Take it away, Andrew...

PART ONE - THE STORY SO FAR

The initial foundations for Uridium 2 were laid down in early 1991 during Fire & Ice's development. Having got a smooth-scrolling system working, which allowed easy background animation and didn't take up too much time per frame, it seemed a good idea to save a copy of the basic system to another project directory. I could then carry on adding routines specific to Fire & Ice's needs and not have to remove them later for Uridium 2. This, as it turned out much later, was nearly a good idea.

The idea of doing Uridium 2 on the Amiga had been around since 1987 when I bought my first Amiga, an Amiga 1000, from Gary Liddon, erstwhile journalist and programmer extraordinaire (at least that's what he tells me). My first naive dabblings with DPaint resulted in some fairly crude mock-ups of what the game might look like on the Amiga. Since then

Graffgold has expanded from a two-man outfit to an eight-man one, and my graphics ability pales into insignificance against the expectations of Amiga owners today. I shall thus refrain from doing too many pixels for this game but any that I do 'lay down' I will sign personally.

I spent a few evenings changing the playing window size and testing out the 32-colour sprite plotting routines. I try to write all routines to be as flexible as possible so I can just tell the assembler how many bit-planes, i.e. colours, I want to display and the routines will still work. I really do want to use 32-colour backgrounds in this game and, without going into too much detail about why all Amiga games aren't in at least 32 colours, I'll probably be cursing that decision later.

All work totally ceased on Uridium 2 a short while afterwards when we decided that



our choice of publisher for this game was limited by the publishing rights to the original Uridium being in the hands of an administrative receiver. This has since been resolved and Graftgold now has all publishing rights to future variants of Uridium. It has taken another six months to actually restart the project as the pace on Fire & Ice heated up (excuse the pun) and all my time was taken up finishing that off.

It also seemed appropriate to start the project proper on a sensible PC, rather than the heap of scrap metal that my old 8088-based PC has become these days. Let's be honest here, it's down tools brothers until we get some realistic tools to work with - I'm fed up with emptying the lake with a teaspoon! When it takes about four minutes to change one line of code and try it out, you've forgotten what it was that you wanted to test. Coding these days involves very, very large programs and you need professional tools and high-speed computers to deal with it all.

Take Fire & Ice. The actual game code, excluding our operating system, must run to about 60,000 lines of code (and I don't mean the inflated figure that the assembler spits out after expanding all the macros). Just 60,000 lines of handwritten code would take about a whole box of 1000 sheets of fanfold paper to print out, if the printer lived that long!

Anyway, the story has a happy middle at least, as I'm typing this on a 486DX turbo-nutter machine with a 100MB hard disk whose only desire in life is to do my bidding but yesterday, instead of making an appointment for a week's time. The old machine sits under another desk in shame, full up, burnt out and redundant. Graftgold is dragged kicking and screaming back into reality from its little world of slow motion.

Work has continued now at a cracking pace, and the first thing that happened was a total rewrite of the scrolling system



so carefully removed from Fire & Ice at the beginning. I said it was only nearly a good idea. That system was good for Fire & Ice because I wanted lots of background animations using 16 by 16 pixel super-characters. Uridium 2, as I see it, requires little or no background animation - no boiling porridge, rippling sea, rising bubbles, waterfalls or spinning pick-ups you see.

Instead I want to be able to reflect background characters and have alternate palettes on individual 8 by 8 pixel characters. This is more akin to the C64 and certain other newer computer-type machines that are best left unmentioned. The scrolling rewrite took about two days and did everything that I wanted and it was good BUT it was a little on the frame-time expensive side when scrolling at high speed, and Uridium needs high speed like Nigel Mansell does.

It's one thing scrolling at a couple of pixels a frame; you can build up newly scrolled data over the next 'n' frames before you cross a 16-pixel boundary, which is the resolution that the Amiga coarse scrolls by. This beast requires scrolling at up to 8 pixels per frame, leaving only two frames to build a stripe of new characters on the leading edge of the scroll.

(Above) Andrew's first working version of Uridium 2 has the new smaller, Manta fighter flying over a background made up by characters that he had to click that everything is functioning properly. The ship can already thrust, turn and fire as in the original.

(Left) One feature currently in the experimental stage is to have a second fighter, either controlled by another player or as a robot drone that copies the Manta's actions.

(Below) These two shots, which look almost exactly like the C64 original, were mocked up by Andrew on DPaint a few years back. He's confident that the finished version will look far more sophisticated.



THE GAME

For those of you not familiar with the original Uridium, here's the deal:

There are these fifteen giant alien 'super-dreadnoughts' lumbering towards the solar system, sucking the mineral resources from planetary cores as they go. You've been dispatched in a Manta-class space fighter to put a stop to all the nastiness. Basically what this entails is super-low flying across the horizontally-scrolling tops of the cruisers (each one is so big it's a level in its own right), weaving through the metallic superstructure and doing battle with dreadnought's squadrons of drone fighters. Once you've landed on the runway at the other end of the ship and set the self-destruct sequence, you must escape back the way you came as the ship explodes around you. It's super-fast, super-frenetic - and super-difficult.

When released in 1986, Uridium pretty much swept the awards board, picking up Best Arcade-Style Game, runner-up Game of the Year and Programmer of the Year for Andrew at the Golden Joystick Awards. Andrew also scooped Programmer of the Year and Best Shoot-'Em-Up at the Newsfield Awards. Oh yes, and it got to the top of the charts as well. Hardly surprising he's doing a sequel, is it?

"Uridium 2, as I see it, requires little or no background animation - no boiling porridge, rising bubbles, waterfalls or spinning pick-ups like in Fire & Ice."

The upshot of all this is that 16-bit computers don't much like working in bytes; it's no quicker than working in words (that's two bytes nailed together), so I'm back to working in 16 by 16 pixel lumps which I can reflect or change colours of. This reduces the scrolling overhead so that I can run a sensible number of sprites around on the screen. So what do we actually have on the screen?

Well, graphics work actually commenced at the beginning of June 1992 with some background blocks. I haven't put any into the game yet

as I don't have a suitable mapping program that can do what I need, and I don't want to spend two months writing one on the Amiga. Steve (Turner), Graftgold's boss-man, has promised to write a generalised mega super mapper on the PC, so I'll wait for that. I don't need backgrounds just yet.

So what's actually on the screen, then? Well, I've got a couple of rough Manta frames done on DPaint and a load of numbers where all the frames of animation should be so I can check whether the program is working, and I've got a few test background

blocks, again with numbers on so I can see if they're working. Oh, and I've done a score panel with the game name on.... 'Uridium 2' in large friendly letters! As from tomorrow I'll log

things a bit more formally so we don't get sued by the Equal Rights for Diaries brigade. I've already got the dates done, all I have to do is fill in the rest...

NEXT MONTH!

The pace hots up as the Uridium 2 super-diary begins in earnest.

URIDIUM 2

Last month Andrew Braybrook, coder extraordinaire, described the laying of the the first binary foundation stones to his latest opus - Uridium 2, the long-awaited sequel to his classic Commodore 64 blaster. Having got his super-sexy, 32-colours, 50-frames-per-second scrolling routine up and running, Andrew continues his saga here - and remember, it's only in The One!

PART TWO - JUNE/JULY

Uridium 2, in my mind's eye, has to start where the first Uridium left off. All the playability from the original C64 game has to be there - the speed, the dynamics, etc - but that must be the starting point, not the finish. I have to add to that. Certainly the graphics can be brought up to date but new features must be incorporated too, and overall the game must be much bigger.

The old adage that a sequel has to be at least 237 times better than the original to be perceived as better at all is certainly true. I'll not pretend that this is going to be easy. In order to give me the best chance of mak-

ing this game stand out, decisions must be made now as to the hardware requirements of this game. In order to get speed out of the Amiga I don't want to be limited by thoughts of how to make this run on other, less capable machines. Only the Amiga version matters.

Also, I think I'm going to need more than 512K, so it may be a 1MB-only game. I know that many of you have upgraded your 512K machines with extra RAM - anyone who hasn't yet ought to seriously consider it as many more games will be coming out for the larger machines. After all, an extra 512K can be bought for around £20 - less than the cost of one game!

Monday June 22nd

Today has been a day of optimisation and consolidation. Rather than developing any new code I wanted to speed up certain functions within Uridium 2. For example, every time a bullet is fired, the fire routine asks our operating system for some memory to store the bullet's position, speed, etc, in.

As with all operating systems, no matter how well written, this is a bit of a slow process. It's much better to have forty or fifty blocks of memory ready and waiting in a list so that when you want to fire a bullet it just says "NEXT!" rather than "Excuse me, Mr Operating System, but would you

mind awfully sparing a little of your valuable time..." etc, etc. It may not seem much now but it could mean I can run a couple more objects around if I keep the code as lean as possible.

I've also put in the score printing routines using multi-tasking so that even if both players' scores need updating, the actual printing of the score is done whenever there's a spare moment rather than immediately, which could cause a crisis if the game is getting busy. This would manifest itself as a 'glitch' or shudder as the game failed to complete all the processing required within one fifth of a second.

Uridium 2 is also the first ever game I've done with a simultaneous two-player option, so it's a bit of a tangle working out which score to update. One Manta ship leads while the other plays wing-man, but the lead Manta could be either Player One or Player Two, each with a separate score. The bullets fired by each must be tagged to say who fired them so that when an enemy ship is destroyed, it knows whose bullet did it.

Philip Williams, the graphics artist, has just finished all 46 of the Manta's animation frames, in 16 colours only, on Cyber-

paint 2. I now have to convert the graphics to IFF format and load them into DPaint so that I can convert them to 32 colours. Our palette at the moment consists of four colour series in six shades. Trying to get all 32 colours onto any one sprite will be difficult but at least we can get different combinations onto various objects. We have a green series, a blue series, some red/orange/yellow explosions and some garish purples. Whatever happened to good old grey? I'll talk him into it somehow.

Tuesday June 23rd

No work on Uridium 2 today, it was a fiery, icy day.

Wednesday June 24th

One of the problems with the Amiga's 32-colour mode is that I've never done anything with that many colours before. And, although I've written the plot routines to run in 16- or 32-colour mode ('cos I like to think ahead), we have no graphics for 32 colours and only one art package that can cope, i.e. one copy of DPaint.

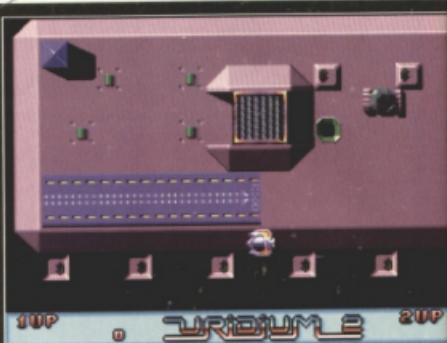
I drew a font in 16 colours using the ever faithful font book for Fire & Ice that never got used. This seemed like a good opportunity to add a few

font in as a series of sprite images so that I don't have to write a new print routine. This should allow for a more dynamic title sequence.

Thursday June 25th

In order to cut down on the work done every fifteenth of a second I'll have to be careful not to move too much at once when printing text. Once a letter has stopped moving I can fix it in the background and then forget about it.

The closest analogy I can think of is that if you have twenty police cars patrolling a town and you want to make it look like more, you drive them to specific locations, drop off your policemen, and get out the full-size cardboard cut-out police car. You then drive the real car away to the next site. This gives the illusion of



(Above) Gosh, look at those explosions! These are what you'll see when the Manta crashes, so though you may be sulking, you'll have something nice to look at.

(Left) The big mothership, originally to be the craft that dropped off the Manta at the start of the level, may now be used as an end-of-level baddy.

(Left) The new graphics have already made the world of difference to Uridium 2's physical appearance - it's already looking like a polished game. Note the tank and Uridimine launcher in the centre of the screen, two nasties that are already functioning.

I'm one of those people that likes to get each bit of code working perfectly before going on to the next bit, so if something goes wrong there's a good chance it was down to the code that I'm working on at the moment. Sometimes though you need a lot of code to test a new feature out. Let me elaborate...

I've just finished the score printing routine which I need for debugging information (I just get it to print any number of interest rather than the score) but... In order to test that I need to be able to really score points, so... I need bullets to fire (which are done) and I need a target to shoot at. Rather than code all of

“ The old adage that a sequel has to be at least 237 times better than the original to be perceived as better at all is certainly true.

”

the enemy ship formation routines I have set up my cursor to be an enemy bullet and an enemy ship, so I can shoot it for points or run into it to test the explosion routines.

Monday June 29th

Took delivery of some explosion graphics and a large mine, which I used as a Uridimine, a device used to chase unwanted players away from the nice friendly alien motherships. Having three different sets of explosions means that we can make the explosions more varied, mixing frames, choosing random sets and seeding small explosions from bigger ones.



more colours to it using a procedure which goes something like this:

- (1) Load the .NEO format picture into CyberPaint on the 4 Mega ST.
- (2) Save the picture out as a .PIF file which is nearly the same as a .NEO.

- (3) Move the sloppy disk to the Amiga and load it into DPaint.
- (4) Move the sloppy disk to the Amiga and load it into DPaint.

Finally I get to play with real pixels! I added about another eight colours onto the font. Of course, everybody complains that they can't read it. I'm just putting the

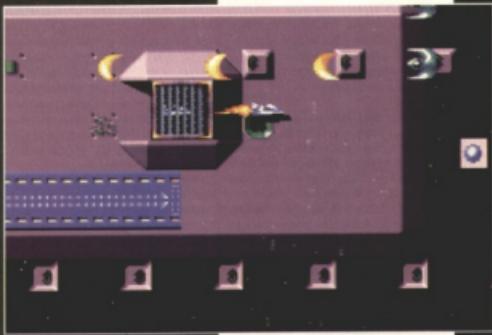
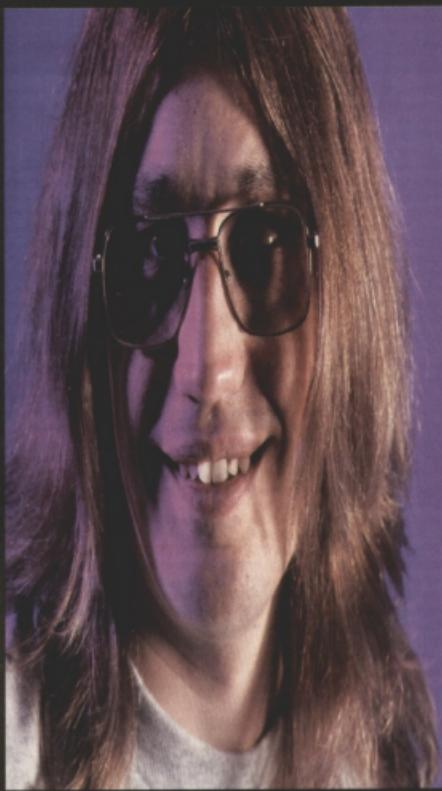
having more than twenty police cars. Of course, hot pursuit in the cardboard cut-out is not a good idea. In fact the idea is so far-fetched it could never happen in real life.

Friday June 26th

Found out most of how to get a PAL Amiga 500Plus to think it's an NTSC machine. You can do it by doctoring the circuit board with a couple of well-placed wires and an optional switch. An easier way is to phone Tony Crowther, who will tell you to hit the register at SDF11DC with... Well something. A bit of experimentation soon finds out at least one value that works.



(Left and below) Andrew's recently added a hangar roof to the background graphics - the Manta can pass under the grille for a neat 3D effect. The result's certainly impressive - in the two-player shot below, you can see the second ship inside the hangar - but getting it to work has caused Andrew no end of aggravation. Notice also the first enemy ships, which are attacking in firing to preset patterns. Currently, this is all there is of the background to see - it's basically just a short testbed used to make sure everything's functioning correctly. The proper background designs will be implemented at a later stage.



(Left) The man himself - when Andrew came down for the photo-shoot, he thought he'd squeeze in a quick plug by wearing a Fire & Ice t-shirt. Unfortunately we decided to cut out the actual logo - if you want an ad in this mag, you'll have to pay like everyone else. So there.



(Above) The Manta is hotly pursued by the game's first set of flying meannies. Note the craters in the background ship's hull - in the final version there will be dozens of destructible surface features.

Tuesday June 30th

Put in the drop-off ship which Phillip has been working on. It's 96-pixels square and the player's ship or ships are launched from a rear hatch, ready for cover. It looks best in darkish, moody greys so I think I've finally talked him into getting the greys into the game. Nag, nag, nag.

Wednesday July 1st

The problem with wanting to just try things out is that you write the code quickly because you want to see results. Then you think of things that you can't do with what you've written so you have to rewrite it. A game with five different game modes and three different control methods per

player, able to detect possibly both devices while playing a demo game, amongst other things, becomes a right tangled mess. And it's watching for a change of those options as well. Coupled with the editor on the PC crashing at an important moment, all this made for a confusing day. And the game doesn't look any different apart from an animated high score display.

Thursday July 2nd

Put in an option select screen so I can see if the game is playing the same game as I think I've selected, with all the right control modes. It is. Wow! Spent the rest of the morning designing the data layouts for the enemy ship formations. All features from the original Uridium must be incorporated plus enough room to add extra features. By the end of the day I had two formations, one of three ships and one of five, created, moving, firing, being shot, and purging if too far away from the player(s). All that took at least a fortnight to write and debug in the original game. Wonder if the boss will give me the next two weeks off?

Diary of a Game

21

Got my new keyboard to play with. No, not a Peavey DPM3SE plus super synthesizer but a 287,000-key PC keyboard, with sensible function keys down the side where you can get them SHIFTed or ALTed with one hand. What a good idea. I'm sure our original PC XTs could do that five years ago, then some bright spark thought: "Let's move all the F-keys to the top where they're really awkward to get at!" Brilliant!

Friday July 3rd

Black Friday. Today the first hardware sprite graphics got drawn - the enemy ship bullets. But can I get this brute of a computer to display them? All I get is a strip of garbage down the screen. These routines all worked in Fire & Ice. What is it that I don't understand about this? I'll have to take this slowly and just test one image at a time. I'm sure it's a hardware problem. I hate leaving problems to solve overnight, let alone over a weekend. My mind just works overtime and I get cross.

Monday July 6th

As I suspected - and I'd been planning the test to prove this all weekend - the sprites displayed get their size and position from the first two words of the display data. These are, in fact, nothing to do with size and position, which it was told about earlier. This all worked in Fire & Ice so why doesn't it work here? Well, I'm fed up with it. And what do I do with code that I'm fed up with? Throw it away and write something better.

So, for the first time I now have a sprite multiplexor on the Amiga that allows me to re-use the three 15-colour hardware sprites that aren't already busy. "Not much spectacular about that," I hear you say. But this routine has a contingency plan. Since the hardware sprites are being used for enemy bullets it would be a blow if some flickered, or worse, didn't get displayed at all, i.e. the Shot-By-Invisible-Bullet syndrome.

What my routine does is use the hardware sprites as best it can and any that don't fit which would disappear are converted to normal blitter objects. I never thought I'd ever do something as ridiculous as that; after all, it's a technique used by the AmigaDOS graphics library. For an encore I've put in some background collision detection so that player bullets and the player ship stop when they hit walls or other tall objects.

Tuesday July 7th

Worked on our disk system to make it more AmigaDOS friendly. Failed. Went to see the great rock band Spinal Tap at the Albert Hall.

Wednesday July 8th

What a band! Great atmo-

sphere, had a marvellous time. Go, Nigel, Go! In the absence of our wayward graphics artist and going by the old saying "If a job's worth doing, it's worth doing yourself" I got out the old DPaint disk and had a play with some spinning circles for a coming-out-of-Hyperspace effect. I thought I'd write an eight-line BASIC program to generate some co-ordinates using sines and cosines but could I get it to work? Fallback mode consisted of some paper, a pencil, a calculator, and a Braybrook-can-round-a-negative-number-better-than-a-grotty-BASIC-program frame of mind!

Thursday July 9th

Still no graphics artist. Apparently he's on holiday. Looks like I'll have to do the graphics myself again today. I nearly shuffled all the colours in the palette and remapped all the graphics to that palette, just to get better shadow effects, then I thought better of it. The shadows cast in the game at the moment are fairly subtle. Had I gone ahead with the palette changes they'd have been about as subtle as Rambo catching butterflies.

Friday July 10th

I've been thinking about Monday's sprite multiplexor all week. The data to plot is stored in hardware sprite format, so if it gets overloaded it has to convert the data to blitter format and plot it the slow way. So just when it's busy it has to do even more work. This is not clever. Against that the data is nice and compact. If I store the data in blitter format, when it gets overloaded the data is all ready to plot, and when it's not busy it has time to unravel the data and convert it to hardware sprite format, which has to get copied to another place for display purposes. It's more efficient in my head - whether it is in the game is more difficult to tell.

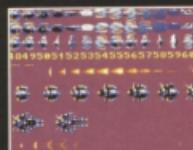
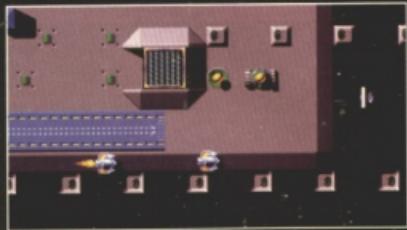
Monday July 13th

Had us all staring at the screen watching for what he said was a background graphic momentarily appearing and then changing back again. This is, of course, impossible. Only he could see it, and despite many times saying "Look! It did it again!", nobody else spotted it. Time to get him certified, I reckon.

The main changes today have been in darkening the colours in the palette and doing the shadows in black. This is by no means elegant of code, fleet of foot or pretty as a picture but it is more striking and is in style.

Tuesday July 14th

Something was stomping all over some important data tables in memory. It took only a few seconds to realise what was happening, but the



(Left) A selection of the latest Manta sprites - it's possible the final ones will have less bright colour schemes. The frames in the bottom half of the screen show the ship turning into a Transformer-like robot - it's possible this feature will be included in the game for when the ship lands at the end of a level.

real question was "Why?" It's one of those situations where any feature added that afternoon could have caused it, which was a blow because I'd only added one new feature and that hadn't been executed yet. Desperation says that a line of code somewhere has gone AWOL so restore it from yesterday's backup and it'll go away. It didn't!

Just by luck on one test run the game lived a little longer, time enough for me to spot that the shadow plotter wasn't working and a blob appeared on the control panel in the shape of a shadow.

My new feature required a change to an important data table that causes all routines to be re-assembled, and the shadow plotter was a new one that missed that process. But, although I now know why it was going wrong, I don't really know how, I suppose it doesn't matter, 'cos it's fixed now.

The new feature was a system for fast targeting by homing missiles. Say you had a tin-can fetish and wanted to find all the different ones in Tesco's. Rather than search through all the aisles, it would be nice if there was a map by the door showing where the cans were. So Tesco's is the list of all objects in the game, the cans are enemy targets, and the fetishist is the homing missiles being launched. Easy, eh?

The other major addition today is an animation for player bullets hitting walls, another use for the hard-

ware sprite multi-plexor. And I've got it doing the Manta jet bursts as well. Glad I wrote that one.

Wednesday July 15th

Discovery of the day: "move.w ccr,d4" is not a real 68000 instruction at all, contrary to my reference book. It's only implemented on 68010 CPUs or better. Getting hold of something as simple as the carry flag for later reference without suddenly destroying it becomes a major problem. All I want to do is see if the player has tripped over a 10,000-point boundary to award a bonus life. How difficult can it be?

Put in a hangar roof for the player to fly under to game the a 3D look. This led to problems - the homing mines must also go under the roof (easy) but not take short cuts through walls (tricky). The homing mines were not doing any background collision detection. Now, at least, you can give them the slip by hiding behind walls.

Thursday July 16th

Bad day for the interleaved plotter (the routine plotting most of the objects using the blitter). I found two cases where an object is partially off screen and the background doesn't get restored where it was.

A kludge fix can isolate that case and compensate but the whole thing needs a rewrite. It's bad news because nearly all the other specialised plot routines are based on it.

NEXT MONTH!

Will Andrew manage to fix his plotting routine or will a kludge suffice? Hold on to your potatoes - you'll find out next issue!

URIDIUM 2

Only three months into its creation, and Andrew Braybrook's Amiga sequel to his all-time C64 classic is already shaping up with the potential to be the greatest 16-bit shoot-'em-up of the decade. And the only place you'll hear the FULL story behind its creation is here, in Andrew's own words in our exclusive serialisation of his on-going development diary. The control mode is working, some of the aliens are in and bullets are flying everywhere - but there's still a long, long way to go...

PART THREE - JULY/AUGUST

Tuesday July 21st

Following on from the minor crisis reported last time that all of the plot routines were slightly flawed in a couple of obscure situations, I've been re-writing them all. Not from scratch, you understand - that would take ages - just the meaty bits that talk to the blitter. The required end result is a set of routines that are more efficient to run and amend to create new plot routines. That process took all morning. At the moment the game consists of a mish-mash of all the different types of enemy that we've created so far. This is so that I can test them all by blowing up the ones I don't want to test. It also means that I can see if all the general routines are working. The objective in the early days of a game is to create a library of routines that are all thoroughly tested so that when complex stuff goes in later I know that any problems that occur cannot be blamed on what I call the 'core'. Once all the core routines are

working and they're as efficient as I can make them I can just forget about them, knowing that they're reliable. That's the theory anyway. New additions to the actual game are a slightly changed palette to give more different colour combinations and a mine-laying meanie that drifts about, attempting to deposit static mines around the screen. The meanie tries to stay on-screen by selecting a new direction that sends it right across the screen. With a fast scrolling game it's important that the bad guys get seen, rather than lagging off-screen behind the action, vainly trying to keep up with the player's movement. This one's only a simple meanie but it's beatable, which gives the player a reward in out-sussing it. There's no point in having mega-intelligent meanies with the latest missiles and cloaking devices such that they are totally invincible. I might as well just print 'Game Over' on the screen in the first two seconds and not bother to write any more.

Wednesday July 22nd
This morning's tasks were two-fold. One was to get the homing missiles nice and slinky so they get drop off from both sides of the Manta ship, then power off forwards before selecting a target and chasing after it. The second task was to think of something easier to do in the afternoon than code in the robot control mode. Unfortunately, I couldn't. Rather than have the simple 'fruit machine' sub-game of the original *Uridium*, I want a more complex arrangement whereby the Manta flies over the runway, transforms itself into a giant robot and then drops to the surface, breaking through the hull to the deck below. An alternate control mode then sees the player battling it out with the dreadnought's occupants *Paradroid* '90-style, only much simpler and quicker. And a right barrel of fun sorting it all out was. Grab routines from *Paradroid* '90 and *Fire & Ice*, add some new ones, smooth them out to make them faster and watch

them not do what I wanted them to do. Background collision detection never was my idea of fun, but if everything in the game obeys the physics of solid objects properly the overall effect is better. It's one of those things you only notice if it's not there. I hate meanies that move through solid walls and floors that I can't go through.

Thursday July 23rd

A final bit of tweaking on the robot control mode allows me to walk it backwards while firing. Great for that dignified strategic withdrawal. Now to tie everything together. A lot of things need doing at once: the Manta-to-robot transformation sequence, setting up a second level with the surviving robots on it, triggering departure from the robot shoot-out, and finally getting to another airborne sequence. Such things as starting co-ordinates, a different sized scrolling window and setting up surviving robots from the previous level rather than the players' lives all need to be arranged.

Friday July 24th

Put in a control mode for the drone robot. Unlike in the main game where the drone follows the constantly-moving main ship, the drone robot can collide with the player's main robot. It needs a cunning algorithm to keep it out of the way. I've rigged it to move away from the player if it is very close, and walk towards the player if far away. This currently works in

open spaces but it remains to be seen how well it copes in confined spaces. Of course, the drone can be backed into a wall and approached and is unable to do anything about it. Also put in a robot weapon that, when fired, releases a bullet casing from the gun. This falls to the ground with a shadow before coming to a stop. Looks quite swishy.

Monday July 27th

Phillip supplied me with the robot animation frames, three in each of the eight movement directions. That's not really enough to show convincing walking, even from above. So, faced with either adding at least another 16 animation frames or finding another solution, my money's on thinking of a way out. How difficult can it be to control two feet separately, he says, remembering the 'Justice' episode of *Red Dwarf IV*, when the lads had mechanical boots on to escort them. I tried nearly all the possible combinations of add and subtract instructions before arriving at a workable system involving a leading foot which spawns a trailing foot. The idea of two independent free-thinking feet is nice but they'd probably go off on their own separate ways, rather like seven leavened boots when not properly supervised. So the leading foot does all the work, setting up positions for itself and the trailing foot. This system is incredibly simple now that it's done, and the feet look great as one of them is always locked to the floor, just like real life. How many video games have managed that? Most tend to 'moonwalk'. I also slowed down the robot speed to show off the walking a bit better and it's a 500% improvement, and as a beneficial side-effect the drone robot now behaves much more life-like as it fidgets about if you chase it. There's no stopping me now, I'm on a roll.

Tuesday July 28th

'Twas again a Firey Icy sort of day. I only mention this because otherwise you might think I'm taking a sly day off and also it illustrates that a programmer hasn't finished with a game just because it has been released. I spent much time writing some playing tips which would be finished but for the fact that the editor crashed on me while saving out the final version. Luckily I had most of what I'd written still on the screen so I wrote it all on paper and retype it tomorrow.

Wednesday July 29th

Bit of a wondering-what-to-do day. The game is now waiting on ideas and for fatty Phillip to draw the robot feet, which he's trying to use as blackmail to get me to



(Left and below) With the game's ten power-up weapons now fully implemented, *Uridium 2* is starting to look a lot more impressive graphically. The variety of the action is still taking place over a blank experimental backdrop. The alien waves are becoming more intelligent, too - all good early pointers towards a top-quality blast-off for the game is finally finished.



(Left) The *Paradroid '90* style walking-robots section, where players can earn top bonus points by destroying the big internal reactors before bugging out and exploding. And the little "thugs" that constantly attack our heroes. Fortunately, the robots can leave at any time via one of the level's emergency exits.



(Left) The graphic frames for the Manta's intelligent homing missiles; showing how they turn to track a moving target. The strange mine-like objects at the bottom, though finished, are still waiting for Andrew to decide what he wants to use them for.



(Left) And here are the graphics for the robot that the Manta transforms into for the end-of-ship sub-game. You can't see the feet here, as they are drawn as separate objects and added by the programme. Note how the frames haven't simply been flipped - each one has been painstakingly drawn to ensure a realistic lightsourcing effect.



(Left) The famous ioniser 'toothpaste' lasers, originally used in *Defender* and lifted by Andrew for *Morpheus* make a comeback in *Uridium 2*. Look at those zingy beams go!



(Below) At the end of each dreadnaught, the Manta craft(s) transform like those famous 'Robots in Disguise' and crash through the hull. Note the tank with two independent turrets. This is a programming headache if ever there was one...

take out something that probably won't stay anyway. How cheap can you get? So I wrote the high score and initials entry routines.

Although boringly mundane tasks, they are still tricky to do because I like to present them differently in every game which means I can only nick tiny bits out of my old games. It has to be done sometime and it makes people think I've nearly finished if the presentation side of things is finished.

With possibly two players playing the game it has to work out if either player has just lost their last life, if so are they on the high score table and then what input device are they using to enter their initials. I've got the letters of the alphabet swinging round in a large oval with a Manta ship pointing to one of the letters. You just move the stick to rotate the letters and hit FIRE to select the letter which then slides into place.

Thursday July 30th

Drew some robot feet myself. By having one frame with a low shadow and one with a higher one it looks like the robot is lifting each foot off the ground in turn. Had a go at drawing a robot body too, which wasn't too bad but I can't face rotating it in eight directions with all the lighting done properly. Finished off

the high score routine bar some text to say well done.

Friday July 31st

Had another go at drawing a simpler robot body, and threw in as many colours as possible. This helps to separate the different parts. Still can't face rotating it in eight directions. Although DPaint gives as accurate a rendition as is possible rotated at 45°, it has no idea what I wanted to draw in the first place, so the maths takes liberties with my straight lines. What I need is one totally unit frame drawn at 45° which can then be spun to get the three diagonals and then lit. Human ray-tracing, no less. Also created another plot routine to display an object in one colour for showing that an object has been hit. It plots in this colour for one cycle before reverting to its original colours, giving a nice flashing effect.

Monday August 3rd

Phillip has yet again come up with something outside the scope of my software.

Not satisfied with a tank with a turret, he wants a tank with two turrets, each with limited rotation. So a quick re-working of the turret-facing routines to allow limits to be specified on the amount of rotation allows me to put his tank in.

During this process I also hit upon another snag with the targeting-on-player bits. The tank turret needs to get position and rotation data from the tank body, with other data coming from the player to decide where to point. So I need another pointer to the player (as there could be two players) for the turret to remember. This is all helpful the other way round as the homing bullets of the players may need multiple pointers for their target and their parent, because a new weapon I developed today involves launching a smart missile which selects a target, fires at it and then moves on to the next.

Tuesday August 4th

Finished off the double-turreted tank which is suitably deadly, and put in a pair of ships that fly past lobbing a bullet between them, an old trick from a vector game circa 1982. You can destroy either ship to make them stop or just don't get in their way. Working out the direction and speed for the lobbed bullet involved a bit of Pythagoras to get it to disappear in the right place, and getting the communication between the ships right was a pain.

Wednesday August 5th

I've noticed some weirdness going on every now and again when using homing missiles. Occasionally one will come out and chase nothing in particular or another homing missile. This is, of course, impossible. Tracking it down involved putting in little tests in various places in the code to identify when the mistake was being made. I thought that maybe if a tar-

get was destroyed by something other than the homer on its tail, the homer would still have a lock on that object's data record, so if a new homer was fired, it would use the old target data and be chased by the first homer.

I shored up that bit so that targets dying hold on to their data record for one game cycle so that the homer always spots that the target is destroyed. This did not fix the problem. The homing missiles are smarter than that! If their target is destroyed they are supposed to select a new target and chase that.

So... scenario 2: A homer chasing its target, the target dies but not by being hit by the homer, the homer spots that the target dies but is not able to select a new target as there isn't one. The homer then dies naturally by timing out and hasn't cleared out its pointer to its original target, so it says "Well I didn't kill my target so someone else should give it a go". Thus it designates its target as an unselected target for new homers unaware that the target has long gone and is now a homer itself. So all I needed to do was clear out the target from the homer so it forgot about it once the target had died. Problem solved. You're nine tenths of the way there if you can just identify the problem but sometimes you only see knock-on effects and it can take a while to sort out which haystack the needle is in.

Thursday August 6th

Today is a rare day indeed - there are no known bugs in the program. That





(Left) Yes, you guessed it... it's the high-score table. Erm... and that's about it, really.



It may look very pretty, but *Uridium 2* will almost definitely NOT be looking like this Deluxe Paint mock-up. Andrew isn't sure about the organic feel of the backgrounds, and will probably end up reverting to the more familiar shiny steel constructions that original *Uridium* players will be familiar with.

is, it's not embarrassing itself due to mistakes in my code! That means that I can write some sparkly new code and know what to blame when something goes wrong. Today's new features then are: a new graphic and algorithm for the chaser weapon that now sits on meanies until they die rather than fire at them from a short distance which was a bit unreliable for moving targets. Also added some little meanies that I'm calling thugs, which run around inside the ship for the robots to shoot at. Also a fast laser shot in one of three snazzy colours, no less. Get 'em while stocks last. Jason, our sound super-hero, says he's written a bar of the title tune. A whole bar, mind. And not just any bar either, but one from the middle!

Friday August 7th

Day off due to Hairy Paul thinking it's a good idea laying a carpet at 3am. It wasn't, but it was fun while it lasted.

Monday August 10th

Put in some new weapons for the Manta, namely an orange laser, a bomb to drop onto the dreadnought to shake the enemy up a bit, a Defender-style ioniser

gun, and a wide-spread twirly thingy. With all these new weapons a method of obtaining them is required, and the hooks were all ready to allow the releasing of an object when a wave is destroyed. A bit of tampering produced a pod, looking very much like a homing missile as no graphics are yet available. Collect it and the weapon changes. One little unanticipated side effect is that in drone mode if a weapon is collected both ships get the weapon. Not sure whether I like that or not, let alone whether I can stop it. Two-player mode is unaffected so either player can pick up the pod for a weapon change.

Tuesday August 11th

Drew the graphics for the ground bomb which, if it misses the deck as it falls, heads off into the void. Changed the chaser bomb to slow it down as it nears its target. That stops it from buzzing around as it damages its target, then once destroyed it selects a new target and off it goes, at speed. Put some copper fading into the control panel to get some more colours on it and got it up to 39. It's in a bit of a wonky format so getting it from DPaint to the game involved writing a small

routine to re-arrange it. I also speeded up our age-old collision detection routines to get them running faster, which could easily save a couple of raster lines. All game routines are measured now in the number of instructions or micro-seconds to execute but the depth of a border colour bar. You set the border colour to a real sexy purple at the beginning of a routine, then back to black when you finish, and see how big a bar it makes, i.e. the distance the monitor scans as it's building up the picture each frame.

Wednesday August 12th

Put in some more test attack waves of ships to have a go at shooting and producing pods to try out all the weapons available. It's no good waxing lyrical about how good the super-smart gargantua-bazookas are if no-one can actually see them. Changed the control panel colouring to a dark green and implemented the remaining ships or robot hit points display. These are interchangeable as they are never required at the same time. The panel update is running under multi-tasking so that it doesn't waste valuable time. If the game is busy then the score display might not be updated immediately, it will get done when things aren't so busy.

Thursday August 13th

Just for presentation's sake I put in the text for the high-score entry screen and also for the player ready screen, which has to say which player is in control or, if both players are playing as a team, which one is controlling the lead ship. I've also been carrying up some of the *Paradroid '90* background graphics to use temporarily for the interior battles. Then I can check out the background collision detection routines a bit further and try and set up a deck to see how difficult the game is. Must put in a main title screen soon. I've cut out the *Uridium 2* logo so I can drive it about the screen as sprites. Of course the game name might change so there's no point in getting too carried away.

Friday August 14th

Having had a further play with the robot control mode, especially with the drone robot attempting to follow the player, it is clear that the drone is going to get itself caught on narrow bridges and do its "I'm completely stupid" impersonation and try to home in on the player and plough into the bridge sides. Let's see if I can fix it. This is the fun bit because I know the control mode well so I can adjust it fairly confidently and know that I won't wreck it (but I made a backup just in case!) After some investigation I decided to

rig the drone 'intelligence' routine so that the drone ignores homing in on the player if he didn't move last time and wait until he's walked his way out of trouble before resuming. It looks very clever and handles the background better than a player who's not-really-concentrating-too-hard-at-all actually.

Monday August 17th

Put in a title screen to get the game logo in. This needs to be bigger really, but it'll do for now. I've rigged up the inside sequence to allow the player(s) to be destroyed by the thugs rather than escaping by taking off after the last point of energy is removed. The main task will still be to destroy as many generators as possible within a time limit, but I don't want to put that in yet as there's nothing more annoying than getting to an important bit to test when a time limit suddenly destroys everything. By the way, thanks to Steve Allan for his kind words in September's *The One*. *Graffiti Gold* have no immediate plans to put *Morpheus* on the Amiga. *Uridium 2* will take a few months yet and after that I'd like to tackle another original project. It might be fun to do, though.

Tuesday-Thursday August 18th-20th

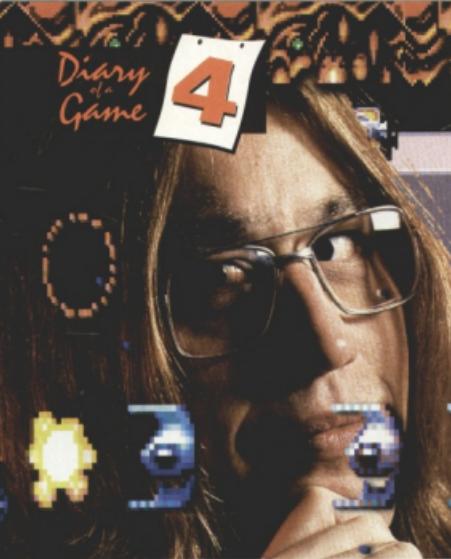
Another Fiery Icy session.

Friday August 21st

I wanted to put in a meaty mass-devastation weapon which has a slow reload time to ensure that not too many go off at once. That's the easy bit, and I can stop the two-player team from getting one each by only releasing one at a time, but the drone mode gives the new weapon to both ships. Each has its own reload time but they can both fire on consecutive frames. Time to do this properly and set it up so that each Manta gets its own weapon. It's not too tricky to do, but it is a bit messy, with pointers to weapons tables, object and player data all flying about at once. Picking up weapons needs to be smarter too, as there's no difference between the player and a drone, so the weapon exchange from a pod to a player must be done by the player, not the pod. All the pod knows is that it has hit a player's ship.

NEXT MONTH!

More graphics! More game design secrets! More exclusive info! Can you handle it? We can't!



PART FOUR - AUGUST/SEPTEMBER

Tuesday August 25th

Great way to start the new diary. Met Gary Liddon at EMAP Towers while delivering the previous diary part and had a few bevvies with him and The One lads before catching the train home to Witton. Had I stayed awake I might have got off at the right station. As it was I ended up in Harwich, which was quite a shock. You don't want to know how much a taxi costs at that time in the morning.

Had a further play with the weapons systems. A flame-thrower for the robot seemed like a good idea. No graphics for it yet but we're spraying out the ubiquitous explosion graphics that get used whenever the proper ones aren't available. Also changed the mass devestation weapon as that was a little slow to take effect; usually what I was aiming at had long gone by the time the bullets came out. Now it's a rocket that gets produced and sprays bullets (also currently looking like explosions) from top and bottom.

Spent the afternoon experimenting with plasma - it's a graphic effect we've seen on a demo. It's quite fun trying to figure out what they've done. We may not come up with the exact same method that they're using but we'll get similar results. If you only need to get one effect on the screen you can throw a lot of memory at the problem, but we have to get a game in there as well. Finally I got some red plasma on the screen. It tends to do really strange things to your eyes if you keep watching it. Weird.

Wednesday August 26th

Now to build a title screen out of my plasma. There's no point in having something pretty if you can't use it. The first thing was to create some nice looking plasma to work with. The plasma is now more ordered and curvy so it's easier to see what's happening. Since these things like being left/right symmetrical and there's no point in wasting memory, I wrote a routine to reflect half the picture across the screen.

The reflection routine didn't quite work but the effect was really wacky - the screen was stretching up and down and I still haven't sussed out why. Part of the optimisation involved only plotting the plasma in sixteen colours, leaving me the other sixteen to add text over the top. Then I got to thinking - since there isn't much CPU time left after the plasma is plotted, why not write Uridium in giant size letters in another plasma colour? All I have to do is teach a cursor to drive-around the screen, printing up big letters. This turned out to be just like writing a turtle program in Logo. Tedious or what?

Thursday 1st September

Really did go on holiday this time. Returned to find Phillip drawing full 32-colour background, contrary to instructions as usual. Still, if you can't beat 'em put in all the routines to display the full 32-colour graphics.

I've been changing the colours around on the block font to make it even more readable, since the whole world and his dog say "Yeah, it's really pretty but what does it say?". Everybody, that is, except Phillip, who just said "Yeuck!" - but then he says that about anything that he didn't draw him-

URIDIUM

2

The epic saga of the creation of a super-sequel reaches its fourth instalment. At last the dreadnought designs are beginning to take shape, and they're a far cry from those seen in the old C64 classic. Andrew Braybrook, the programming genius coding the game, tells all...

self. I've had a go at redrawing one or two of the letters to make more readable shapes. The 'R' is causing trouble as most of the letters are capitals, but the 'R' is actually an 'r'.

Wednesday 2nd September

Spent nearly all day battling with different linker output formats. All I want to do is load the game in relocatable format so I can store the code wherever the machine's extra RAM is, but I want my variables to be short-word addressed in the top 32K of the Amiga's memory for extra speed. ALink format seems to support 16-bit relocation data but I can't persuade SNLink to produce the right output. Can anybody help? I just want one section, namely my variables, to be linked absolute in word-addressable memory.

Thursday 3rd September

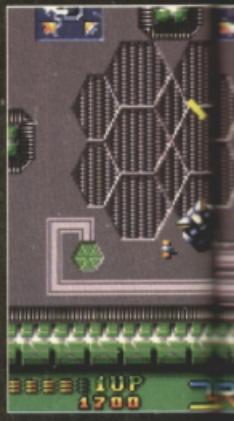
We need to put together a demo for the European Computer Trade Show next week. Jason has started converting his music into our format and hopes to have some token sound effects ready. We still have no backgrounds but the mapper is nearly ready to at least try to put a map in. I've commissioned some more graphics from another of our graphics artists, Mark Bentley, so I've not got any dummy sprites in the game. If only we had an extra 24 hours we could put so much more in.

Friday 4th September

Getting close to show-time and the mapper is nearly finished for its first test - no guarantees, no nothing. The music and some trial sound-effects have been set up by Jason. Unfortunately the graphics didn't get finished and I was left with

(Below) For the sake of the ECTS demo, Andrew re-used some old *Paradroid* 90 graphic blocks to flesh out the backgrounds inside the dreadnought.

(Below right) Having destroyed all the generators inside the ship the mission is complete. The droids must then find the lift-off pad, transform back into Mantas and make good their escape.



a sheet of really pretty organic graphics that I had no idea how to slot together.

The mapper held up fairly well until I tried to put its output into the game. I have had to resort to using my old graphics so I can see what is going on. The mapper is somewhat to say the least, but this is its first test. Normally I'd wait for it to be fixed but with the show looming I'll have to program my way out of this by reformatting the map and character set data within the game itself.

Monday 7th September

Show day. Uridium 2 gets its first showing to the press, unfortunately without any real backgrounds in it. It's also the first time that Renegade, the publishers, have had a good look at it. Had a Cuban dinner in the evening with the Bitmaps (name-dropper) to celebrate the completion of another Graffgold project, Gods, on the Sega Megadrive. If you'd like to play this game, write to Sega Europe and tell them so. It's rather good.

Tuesday 8th September

Now is the time to re-organise things in the machine and to free up some of that valuable chip RAM by moving the game code to fast RAM. That means that I have to tangle with Amiga relocatable link format, which is pretty horrible, and I have to sneakily redefine my variables because Amiga link format does support short word addressing but SNASW won't actually produce any short word addressed sections. Why not, chaps? Not everyone wants to run system legal stuff, you know.

Still, a wily programmer can find ways round this. I've written a scatter-loader that locates the Amiga's fast RAM, puts my code there and then relocates it. It's like picking your house up and moving it down the street, then changing the number on the house to fit in with the houses on either side of it. Then it has to tell the post office where it is, but it still knows where the post office is because that hasn't moved.

What am I drivelling on about?

Wednesday 9th September

Taken the whole program apart today to define my variables in a different way. The main objective is to split the program into two lumps, one that can be relocated into fast RAM, and one that can't be relocated because it's too messy. Then I have to set up a communications area to allow the two halves to communicate. This will mostly be one-way traffic. Both halves get bolted back together again while debugging to speed things up.

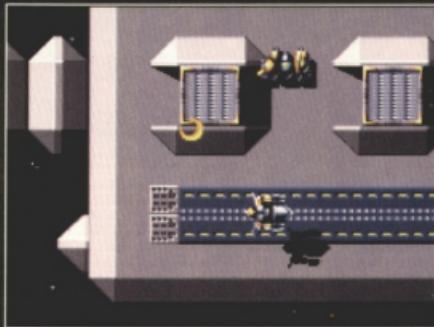
All this monkeying around took about five hours. Now I need to come up with a way to test it. I don't think that just letting rip with a code relocator is a good strategy. It needs to be traced slowly to identify places



Now that some of the new ship graphics are in place Uridium 2 is really beginning to shape up. The organic look is very 'Alien', don't you think? Here you can see the pretty bubble laser in action.



(Above and below) These shots are from the demo Andrew knocked together for the ECTS using old graphic blocks. Note the Manta transforming into the robot in the shot below.



No-one who played the game today noticed the tweaks that I had applied to the control modes. They nag like little kids for changes and then don't notice when they happen.



Diary of a Game 4

where it may go wrong. That will be tomorrow's job. Mustn't do this too quickly - I told Renegade it could take three weeks! Of course, it still might.

Thursday 10th September

Got the code relocating what now officially frees up about 80K of chip RAM for more graphics and sound, which is good news because it wouldn't all fit before. Jason wants more memory for samples because he wants to put speech in.

On the game front I've changed some of the parameters of the control modes to make them more responsive. The Manta turns quicker, as do the robots which also move a little faster too. That helps to squash the thugs. Also in is a new mass destruction weapon that fires lasers in an arc around the ship.

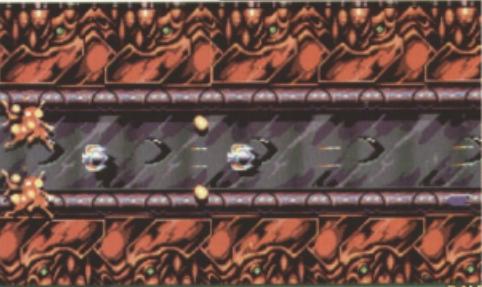
Friday 11th September

Sorted out some niggling bugs. Bad initialisation caused weird things to happen as some flags weren't being cleared out to start with, including a pause flag so the first game automatically paused itself. It's a case of studying the symptoms and deducing what can possibly be the cause of them, and then figuring out what has caused the cause. Knock-on effects can have a programmer chasing the wrong thing for days.

Needless to say, no-one who played the game today noticed the tweaks that I applied to the control modes. They nag

the mapper already. Still no mouse to use with it, so cursor keys it is. Not very efficient.

Jason has produced some speech to put in, expanding the size of the sound samples from 90K to around 140K. This is going to be a tight squeeze indeed.



Great care is needed when flying around near the dreadnought's mighty engines - even brushing the white-hot exhaust flames will melt the Mantas into scrap.

The Mantas on a bombing run down a trench in the dreadnought's surface. Although the bombs look nice they have one big drawback at the moment - the Manta can't fire forward, leaving it open to enemy attack.

The speech seems to be centred around picking up bits of a few key words, 'Playern' being one of them. For example, by missing out the 's' and the 'n' you get 'Player', and by dropping the 'P' and the 'n' you get 'laser'.

Added a grenade weapon for the robot which is equivalent to the bombs

had loads of new features added and all the sound but what comes across in pictures? Graphics, and they're still the same. Better crack the old whip on the graphics department. Also on the agenda is making up a demo to take to EMAP towers. It would be a really bad idea to wait until the day to try and make up a demo considering what I've just done to the game regarding its internal arrangements.

My initial strategy of making up a hybrid debug 'all-in-one' version but with a live load was flawed. I couldn't get the thing to load in and survive, and it took ages to make up the disk. The trouble with the live version is you get no help if it crashes. Next plan is to make up a full live version with relocating code and all. This also failed. Getting desperate now, so start putting border colour changes into the code to see where it gets to. Didn't even hit the first one. This is bizarre.

After two or three hours of head scratching I decided to just download the first part of the game to the Amiga and have a look at it. "That's not my code!" I cried. I had linked the game together in the wrong mode, so I got a Cross Products Executable code instead of pure binary. Realisation of the problem caused so much relaxation I could have gone into hibernation.

Mid-afternoon and the demo is made

up, loading code and music in on its own. Lucky that as I'd already wrecked the show demo disk trying to put the new version on it. Now I can concentrate on making up a new demo with new graphics knowing that I have a fall-back disk.

Wednesday 16th September
Took delivery of the first part of the new ship layout with new graphics. This map is huge and I really mean HUUUUUGE. Phillip has gone mad on the scale. I've never had a map this big before and driving around it highlighted a new bug. The scrolling system blew it going downwards after about a screen and a half. Rather a fundamental error in there somewhere and finding it means remembering how it all works. Those bits of code assumed to be working do tend to get forgotten about.

Thursday 17th September
Last minute thrashing about to get a demo ready to have some pictures taken. The first map is nearly complete and at least you can fly over it. The extra size has highlighted a need to generate the attack formations near the player. Trying to destroy a wave of fighters spread over the whole map width would be too difficult.

Last minute panic problems included a renegade tank that drove off the screen because its control data had been corrupted by our operating system looking for extra chip RAM. The program is already more than 512K long. That's just too big to manage. I'll probably drop the sound out now that it is working.



like little kids for changes and then don't notice when they happen. Further tweaks included making the homing missiles turn slower while they accelerate away from being fired (that makes them look more realistic) and a smart-bomb that hits all targets on-screen with the appropriate stroboscopic.

Monday 14th September

Set Phillip up with a PC and the mapper to create the first real map of background graphics. A few teething troubles needed to be eradicated and we have a list of changes for

of the Manta. Great for taking out the generators, not so good at hitting the thugs. Discovered a deep-rooted bug which was giving infinite use of some weapons which were supposed to be limited. That one could, and I emphasise could, have been the cause of a mysterious crash one time that has been sitting at the back of my mind, nagging away.

Tuesday 15th September

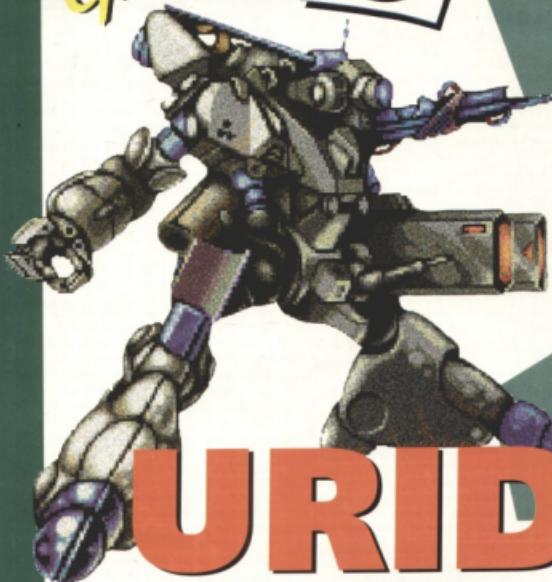
Getting close to diary delivery time and the game doesn't LOOK that much different from last month. Of course, it has

NEXT MONTH!

Uridium 2 enters its fifth month of development, and once again we'll be there with the very latest graphics and white-hot game design secrets, all direct from the horse's mouth! And remember, this is the ONLY place you'll read it...

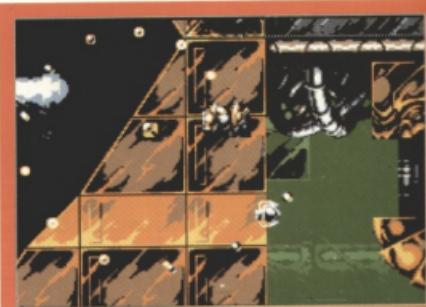
*Diary
of a
Game*

5



URIDIUM

Will Andrew Braybrook's latest be his greatest? The Amiga sequel to his C64 megablaster is now in its fifth month of development and getting better all the time. Every month in his exclusively-serialised diary, Andrew delivers a blow-by-blow account of how the potentially blockbuster game is coming along. THIS MONTH! The game reaches an important crossroads and there's personal drama to rival any soap opera...



(Above) This is the new dual-playfield version of Uridium 2, which has two layers of background graphics scrolling independently in parallax. The orange plate is the top layer, with the less colourful green plate on the bottom. You can see how the graphics are less colourful, but where this version shines is in the scrolling - duller it may be, but it's jaw-droppingly impressive and compelling once you see it move.

"It's getting tricky to tune the game now because I'm so used to it. That's why it's good to get outside help from time to time. A stranger to the game can be a lot less subjective about things."



PART FIVE - SEPTEMBER/OCTOBER

Friday 18th September

The Uridium 2 program has now got so massive that in order to develop it further I will have to temporarily chop something out. First thing to go is the sound. Now I remember why I don't normally put that in until near the end of the project. That saves about 150K which keeps things to sensible sizes. Jason can still work on the sound with his sound editor and I can get the sound back by making up a 'live' version of the game if I need to. While I was out

yesterday he created some new SFX for me, including two really strange effects that aren't really appropriate to the game but are so good that I want to put them in somewhere.

Got the fighter attack waves to be produced near the player wherever he is, as there's no point in going to all the trouble of producing a great wave of fighters when no-one is looking.

Monday 21st September

In order to squash things down still further to give me as much chip RAM as possible I've separated off the robot sprites as they're not needed

most of the time. When inside the ship the robot sprites are swapped in as the enemy ships are swapped out. There's a bit of monkey business going on to keep the assembler happy about which graphics are which and it took most of the day just to get the game working again.

Tuesday 22nd September

I'm sure we fixed that once. The first dreadnought of the first fleet is now almost complete but for adding destructible bits and other features like bunkers and missile silos.

Spent the rest of the time rearranging the memory map in order to

be able to load the game into any 1Mb Amiga, not just ones with 1Mb of chip RAM. This task is nearly complete now. With all the preparations being made for loading files from disk, it seemed appropriate to add in the routine that asks for the right disk should it discover the wrong one, or none at all, in the drive.

At this stage I've nearly worked out completely what data gets loaded in, what is always there but in a packed format, and what is always present unpacked. I want to avoid irritating delays where possible so it'll do just the one load from disk to pull in a whole fleet of dreadnoughts. It should also have room enough to cache the first one in memory all the time.

Wednesday 23rd September

Put in a feature to darken down the control panel where it says '1 UP' and '2 UP', so it only lights up when the player in question is in control. This also has to take into account the two-player mode and light both up. I hate games that remove the waiting player's score from the screen - I call that player the 'chess player'. You need to know how both are doing all of the time for comparison, and because the only time you really can take your eyes off what you're doing is when the other player is 'on'.

Thursday 24th September

Went through the old C64 Uridium flight patterns to familiarise myself with what they do. My new code is much smarter than the C64's so a lot of the patterns are unnecessary. For example, I can vertically reflect a flight plan to get two ships mirroring each other's movement just by specifying upside-down mode. One thing I had missed out though was chaser mode, where an enemy ship tries to line up with the player before squirting Bonobos of Doom at him. I want this to be a bit smoother so I'm working on an algorithm to line up behind the player more intelligently. This is not going well at the moment.

Friday 25th September

Cured the last long-term niggle which involved the homing missiles. They have a minimum turning circle and if fired very close to their target they can circle it until they time out. What they need is a contingency plan. First they have to recognise that they are in that situation. This can be identified by keeping track of consecutive moves at full turn. Any turn less than full means that they have successfully lined up with their target. After turning 270° on full lock they fly straight for about six moves before resuming targeting. By this time they are far enough away that they have room to turn and hit the target every time.



(Left) The latest version of the 32-colour Uridium 2 that Andrew has been working on up until now is likely to be accepted in favour of the new dual-play-field format. Which one do you prefer - super-sexy scrolling or the multicoloured swapshop on offer here?

Tuesday 29th September

Did some end-game-screen designing. Obviously I'm not at liberty to tell you precisely what I've been doing, so I'll just say that I've come up with a new golden palette for the title screen, rather than plain yellow.

Wednesday 30th September

More meanies are the order of the day. Cargo ships that take off from runways, concealed laser towers that spring up and spray doom and disaster before hiding away again and such-like. I'm constructing meany algorithms in our Alien Manoeuvre Program which allows me to make things happen more quickly.

Thursday 1st October

Test-pilots Robert Orchard (the inventor of the name 'Uridium') and Richard Harvey came in to give the game the once-over. Within seconds they had the Manta flying backwards, something that hasn't occurred for months, and I can't reproduce the effect. Users! Bah!

Friday 2nd October

I want to get some shadows on the enemy ships. That's a bit of a hefty overhead but I've thought of a way of persuading the hardware sprite multi-plexor a fancy name for a routine which sorts the objects to be displayed into vertical position sequence and re-uses the hardware sprites to show them, by the way. Had another juggle about with the



robot speeds to make it feel more - you don't have to have two play-controllable. Just about anything that can be changed is defined as a variable so that I can change them easily and immediately see the results. It's getting tricky to tune the game now because I'm so used to it. That's why it's good to get outside help from time to time. A stranger to the game can be a lot less subjective about things.

Monday 5th October

It's one of those days where you start thinking "What if...?". What if I did the game in dual-playfield mode? How long will it take to find out? Never tried dual-playfield before. How difficult can it be? Only had a couple of hours at it but I got the bare bones of a system up and running. I copied all the game files to a new directory so I don't wreck any code, and started converting it.

Tuesday 6th October

Carried on with Operation Dual-Playfield. Made an astounding discovery

robot speeds to make it feel more - you don't have to have two play-controllable. Just about anything that can be changed is defined as a variable so that I can change them easily and immediately see the results. It's getting tricky to tune the game now because I'm so used to it. That's why it's good to get outside help from time to time. A stranger to the game can be a lot less subjective about things.

The actual gameplay will be largely unaffected by the choice but the effect on the game is rather drastic, to say the least. The appearance will be completely changed. Anybody who suggests running both methods is likely to be severely damaged as all the graphics would need to be stored in two different formats.

Wednesday 7th October

Graeme from Renegade paid us a visit so I showed him the two versions of the game. When he saw the dual-playfield version I think he was quite impressed as he uttered a number of unprintables! So, how can I remove some of the cons of this mode? Well, following on from the statement running both 32 colours and dual playfield in the same game, how about if all the titles and presentation is in 32 colours and the game is in dual-playfield? That way I keep my plasma and my lovely font.

It took an afternoon of machine crashes to do it, but it's done.

Thursday 8th October

More work getting the dual-playfield version running. It's now all working so all we have to do is decide which version to go with. The CPU use on both versions is about the same, so it doesn't make much difference to me. But can anyone draw sensible graphics in seven colours, of which one must be black?

Friday 9th October

Walking home last night I'd just talked myself out of doing the dual-playfield version. How about saying that for Uridium 3? I hadn't seen a satisfactory palette to do the job. Two sets of three colours just doesn't give the nice shading and one set of six colours looks like the TV isn't working properly. Just to add

a bit of excitement, Philip the graphics artist has resigned, which should cut down on the arguments a bit, but it leaves a shortfall in the graphics department. Over to Mark Bentley, one of the new recruits, with a last-ditch first task - find me a palette I can live with!

I now have a palette of near-grey shades with a red and an orange for explosions. The greys are neutral enough that they can be mixed with the red and orange, and there are enough similar colours for good shading effects. The backgrounds need massaging but all the sprites converted well with little touching up. One version has to go soon as I can't maintain both forever. It's like juggling three chainsaws at once.

Monday 12th October

Added a new bonus feature, a score-doubler. Needed a new icon for that so I set about drawing a new icon block to put all the bonuses in. The old ones didn't stand out too well.

Tuesday 13th October

Swaying towards dual-playfield mode. Only worked on that today. I've built in some variations in the display, like being able to swap the playfields over so that the three-colour one is on top. I can also free up the second playfield so it doesn't parallax at all. I now have the capability of running a giant meany, almost for free, although it only appears in three colours. How useful that will be remains to be seen.

Also worked on an end-of-dreadnought sequence, where the Manta(e) speed away from the doomed dreadnought before it blows up. Mark has been working on the backgrounds for the dreadnought proper; he tells me he's nearly got enough to go for a test map.

Thursday 16th October

Mark remapped the 32-colour backgrounds into seven colours to create a palette in sympathy with the backgrounds. From a distance you wouldn't spot the difference, and that's without any touching up. The sprites needed reworking to get them looking good in the new palette, which has three red/oranges and only two blues. The explosions should now look even better. Definitely swaying 65%/35% in favour of the dual-playfield version! See what you think from the pictures here.



(Above) Flying the Manta over the vertical strip of flashing lights triggers the landing procedure. Here the ship is caught halfway through its process of transforming into a robot.



(Left) The deadly 'toothpaste' lasers in action in the controversial dual-playfield mode.

NEXT MONTH!

What will Andrew do next? Is Dex really Tinsel's illegitimate lovechild? Who shot Brad? All this and much, much more, only in your Number One One

Diary
of a
Game

6

URIDIUM 2

It's the game that everyone's talking about - and in the case of programmer Andrew Braybrook, it's normally things like "That Uridium 2 is a real pain in the backside, I can tell you." Each month in *The One*, you can read how Renegade's latest arcade opus, due out next Spring, is coming along, with daily updates and top gossip straight from the horse's mouth as we serialise AB's development diary, now in its sixth month. Take it away, o bespectacled one...

**PART SIX - OCTOBER/NOVEMBER****Wednesday 21st October**

We're pretty much decided on going ahead with the dual-playfield version, I think. With that decision out of the way we can steam ahead with producing graphics and more game-play items. Having made this momentous decision, I've cut all ties in the game with the original 32-colour background graphics and converted one set of them to seven, with Mark [the graphic artist] converting and redrawing where necessary. Had some fun organising the ground bombs so that if they miss the top playfield they fall some more and will detect the back

playfield, if any, and blow up or drift off.

Jason has added a new tune for the inside-ship bit; and managed to save 60K or so by resampling some of his instruments. That gives us buckets more space to put better speech in, and a better lead instrument for the title tune. He still doesn't seem keen on the stereo Rickenbacker bass sample, though. Can't think why!

Thursday 22nd October

Tried to get the dreadnought to melt away today so as the reactors are destroyed, the ship breaks up. The robot then has to steer clear of the spreading hole or die. Spent all afternoon chasing a nasty bug caused by the process of updating the background. Still haven't nailed

it. Every time I put in a test to see if something is getting corrupted, the bug just happens anyway so I'm not on the right track yet.

Friday 23rd October

Tried a number of methods of melting the ship, all of which use horrendous quantities of CPU time and so cause glitches all over the place. The last method I've tried gives me most time, but only the top layer of the ship melts at the moment. I've just written the routines to hit the other layer but I've had enough for today.

Monday 26th October

Experimented with ways to melt the background layer of the dreadnought before giving up. All the coordinate systems were getting tan-

gled up and it was taking up much too much time. As all the blocks melt, each one spreads onto four bordering ones which quickly generates a lot of melting explosions. Doing this on two levels would have looked pretty, I'm sure, but would also give the processor a seizure. Settled for one layer melting and the other brightening up and then darkening.

Tuesday 27th October

Mark has nearly finished work on the background character sets that were in 32 colours. He has added loads more detail to them to make up for the lack of colour. It does look good now, and with most of the sprites now redone we can get on with producing some new graphics.

Jason wanted to add a new facil-

scrolling three-colour contour map.

Monday 2nd November

The small map needs the following tasks to be done:

- (1) Translate the large blocks in the map into single coloured pixels by examining the heights.
- (2) Place all of these pixels into hardware sprite images.
- (3) Cut a hole in the display panel for the map.
- (4) Get the copper list setup routine to work out which part of the map to show and display the appropriate hardware sprites.
- (5) Draw a new panel display with a cut-out in the middle to show the map through.

I blanked out the panel temporarily so I could see where the sprites were. There's no point in getting the calculator out and trying to figure out where the sprites are going, just get in there and throw some figures at it. As long as you have a fair chance of seeing where things are, you can move them a bit at a time until you get it right.

Tuesday 3rd November

Managed to throw away a few hours work when I restored a file from a backup because I'd made a total botch-up of it. I hadn't done a backup quite as recently as I had hoped so I lost some of the more recent updates to the world data.

Brought in a pitch-shifter pedal for Jason to create some more speech with. Now that I've freed up some more memory we can spend some more on the sound, and Jason doesn't have to worry about his voice being heard because that pedal is quite amazing. You, too, can be Darth Vader.

Wednesday 4th November

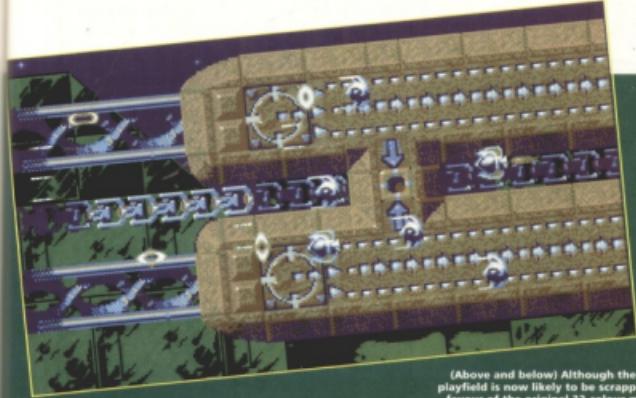
Not a great day for the history books. Just about everything that could go wrong did. It knows I'm in a hurry. The mapper can't produce four-colour output so I had to reformat its character set. The map was in the wrong position so I had to move that over. Then the FAX machine ran out of paper just as I wanted to send one long distance.

Thursday 5th November

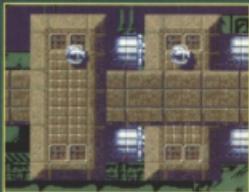
Paid a flying visit to Paris to see how the French put on a computer show. Congratulations to Sensible Software for Soccer getting the Tilt Best Computer Sports Game award.

Monday 9th November

Went to another computer show at the week-end. I'm not allowed to mention which one as Renegade didn't have a stand there. Got Undium 2 running on an Amiga 4000. No incompatibility problems yet.



(Above and below) Although the dual playfield is now likely to be scrapped in favour of the original 32-colour mode, these new dreadnought designs give a better idea of what the game should look like. These steel and concrete constructions are more reminiscent of the original game than the organic designs of the later ships.



(Above) The simultaneous two-player option is still in operation, with the highest scoring player from each level controlling the game's scroll, with the other ship having to keep up as best it can.



when playing it. Actually, it's a bit grim at the moment.

Friday 30th October

Rationalised the attack waves to remove unnecessary ones that don't need to be there with the new system. Also fixed the chasing patterns as the new chaser ships are a bit smarter but need to be coaxed along a bit more. Everything from the original can be done, but with bells and

whistles and plenty more options to boot.

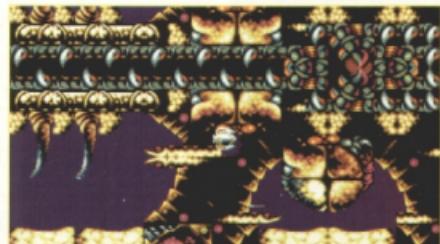
I want to build a small map display into the control panel to give the player a bit more look-ahead. If I use the hardware sprite software that I used in Fire & Ice then I should be able to take over the sprite DMA channels at the bottom of the game screen and get them to display a

Wednesday 28th October

One minute it's hot, the next it's cold. Must be some Fire & Ice somewhere.

Thursday 29th October

Lifted all the old ship attack patterns and waves from the C64 version and dropped them into the Amiga code. They took a gratuitous amount of chopping about on the editor as I now define these things in a slightly more civilised manner. I can now gauge the game's difficulty a bit better by seeing how long people last



(Right) The latest, all-singing, 32-colour dreadnought in action. The game now features a scrolling radar display showing your position over the giant ship.



(Above and right) The new dreadnoughts are becoming more surreal and Giger-esque by the minute - notice the skeletal motif here. The game also now boasts a funky in-game track and loading screen music inspired by the famous C64 work of Martin Galway.



Got a second set of fleet graphics from a freelance artist, Stephen Rushbrook, who is doing what should become the first fleet. The style is more along the lines of the original C64 but much better. We managed to put the graphics into the game after a struggle with the packer, which crashed repeatedly, and the mapper that steadfastly refused to produce anything remotely related to the correct output.

Tuesday 10th November

Defined all the destructible lumps of the new deck and all the Uridimine launch ports. This highlighted two new problems. For one thing, too many destructibles in one locale tend

to cause the machine to have a seizure when they all blow at once. That isn't actually too bad as it just appears as camera shake, which I simulated in software anyway. The other problem is that the Uridimines get generated and left behind a little too easily. We've got to give the bad guys a chance, haven't we?

Tidied up quite a few routines that had been pretending that they were working (these computers will try anything to sneak a bug through customs) and put in three new bonus blocks to give the player points or an extra ship pod for destroying a full wave of fighters.

Thursday 12th November

Put in some new bad guys. There's a slow flying meany which I did call a helicopter until some bright spark pointed out that they wouldn't work in space. Everyone's a smart alec. This helicopter-type thing gets lifted up from its hangar below-deck and lifts off to chase after the player.

Mark wanted the parallax layer to not parallax on his latest ship design. There's no pleasing some people. In not-parallaxing mode ('cos I'd already considered this possibility, being a smart kinda programmer) I had set it up to run the three-colour layer over the seven-colour one to act as overhead gantries or some such. Needless to say, this wasn't good enough as he wanted the three-colour layer below. A quick wave of the magic keyboard and we now have TOTAL flexibility.

Monday 16th November

Graeme paid us a visit from Renegade. He's not tremendously keen on



Because the sprawling maps are far more sprawling than the original C64, it's not always easy to plot where the enemy attack waves should appear. It's this part of the game that is taking much of the tweaking time - if the enemy appear too close to the player's ship the game becomes too difficult, if they appear too far away then the game becomes too easy.

the seven-colour graphics and since Mark is hinting that he'll go on strike if he has only seven colours to play with then we have a big problem. Most of the screenshots I've taken fail to impress as they are limited in colour. It's only when you see the game moving that it looks great.

Tuesday 17th November

Since the dual-playfield version is losing support rapidly I'm going to return to the 32-colour version. I've neglected that for a few weeks now and I've been finding bugs in the dual-playfield one which will still be waiting to be discovered again. This is a great memory test. Also I've been improving the gameplay overall and will have to redo all the improvements that I've made. I had to go through the whole lot file by file. I loaded up both versions and paged through them simultaneously. If you switch the editor quickly from one version to the other and back then the differences show up like an elephant in a very small garden indeed. Then I just had to decide whether that difference was because of the two playfields, in which case I don't want it, or because it's an improvement. About three hours later and I got it all linked together.

Wednesday 18th November

Mark is now battling with a new palette for his 32-colour versions of all the ship graphics. This is fun.

Thursday 19th November

Well, Mark's new palette is a real bundle of laughs. I can't do a thing with it. No black, nine browns (which I can't tell apart from the greens) and they're all over the place. He seems able to draw with it. I guess that's what they mean by job security. No one else can use the palette. This makes life a bit difficult for my 3D block font - I suppose I'll have to do a new one.

Friday 20th November

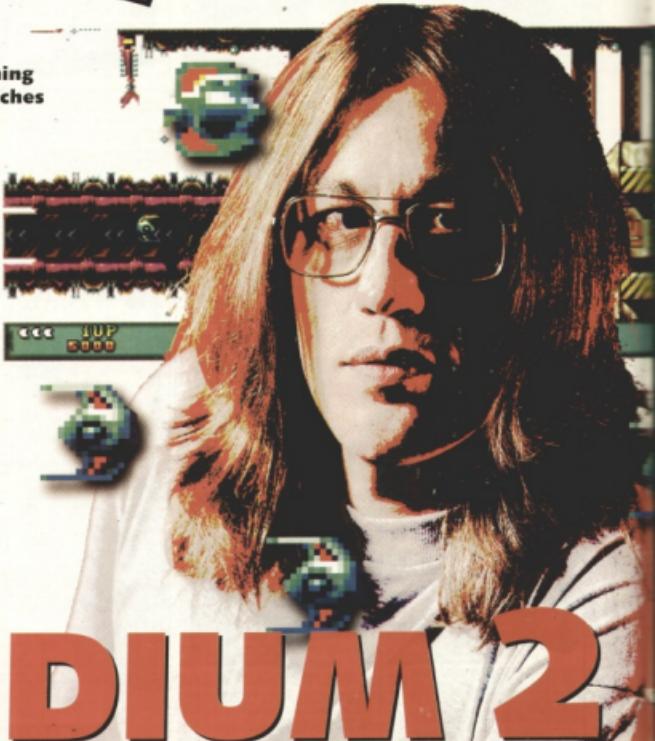
As diary delivery date approaches, I have to figure out what I can reasonably put into this month's demo disk to take to EMAP towers. All the sound and music is just about sorted, it's just the graphics that are currently in limbo.

Mr. Rushbrook has been working on an animated intro sequence. None could accuse him of being anything less than ambitious.

Diary of a Game

7

While the rest of you were probably snoozing in front of the telly, polishing off endless Turkey sandwiches and wandering around 'confused' as a newt, Andrew Braybrook has been busting his hump over the festive period to ensure that you can be playing his latest shoot-'em-up blockbuster as soon as possible. And by now you're bound to know that all the gory details can only be found here, in Andrew's exclusive monthly diary serialising the creation of Uridium 2. This month we're in heaven 'cos we've reached Part Seven, and the end is nearly in sight. Well, almost...



URIDIUM 2

PART SEVEN NOVEMBER/DECEMBER

Wednesday 25th November

Had a further visit from Mr Rushbrook who has converted most of his background graphics from seven colours to thirty-two. Let's just say I'm not Mr Popular at the moment. The extra colours certainly help though. Just a bit more tweaking and they'll be OK. I've also been hassling Mark for all the game graphics to be converted to his new palette. This requires re-mapping and, being a fussy person, some re-drawing of certain areas.

Thursday 26th November

Got some of the graphics through from Mark today. Just the central Manta ship to go and then we can concentrate on creating new graphics. Added a new torpedo weapon to destroy ground targets. Only the bombs and this new torpedo can knock out walls and what I call 'destructibles'. That way you have to destroy waves of fighters to gain the weapons with which to take out the rest of the dreadnought. Landing may then only be possible after the player has destroyed and knocked out a specified percentage of the enemy dreadnought.

Friday 27th November

Did anyone spot the Renegade/Grafton gold-sponsored Lombard RAC rally car, number 1207? Congratulations to Lloyd Shelley driving and Ray Berries navigating for starting 120th, finishing 64th and coming 8th in class N2. Discovered how to do snakey-type multiple-sprite things today. Chris Wood has been here converting Fire & Ice to the PC and explained how to do it, and it's definitely cheating! We gave up after we had got to university-level mathematics using hyperbolic co-tangents and the like. Not the sort of thing you ask a 68000 chip to sort out at high speed.

Anyway, it turns out that as long as you keep at least one end of the snake moving it looks quite realistic just by each letting link of the snake take the average position of the ones on either side of it. Since all the links are updated sequentially then this averaging process is actually using half out-of-date positions as one link won't have been moved when its position is read. That actually produces a 'lagging behind' effect which looks more realistic. Now all I have to do is think of a place to use this in Uridium 2! Why is it that the simple solutions are often the best?



(Above left and right) The latest new dreadnought design in action. The game's getting harder now as the level designs are becoming cluttered with objects that the player must steer around. On later levels, the Manta must be turned on its side in order to squeeze through narrow gaps.

(Left and below) As new graphics arrive and are implemented, parts of Uridium 2 are beginning to look more reminiscent of the original. Note the tall towers, which the player's ship can crash into - all collisionable objects have yellow and black warning surrounds.



Tuesday 1st December

Just time to make up a couple of demo disks to take to London as we are talking to certain parties regarding the production of some music especially for Uridium 2 by some well-known chart personalities. All top secret, hush-hush, say-no-more at the moment.

Wednesday 2nd December

Came up with a new method of controlling all the destructibles. Up till now for every destructible block on the background there has been an invisible object sitting there waiting to be hit by a bomb. When it does,

updates the map and the screens, produces an explosion, gives away some points and finally makes a note that it has been destroyed so it doesn't come back. That fits nicely with our system but is a bit wasteful as most of the time there are no bombs flying about.

A structure re-arrangement needs to be made so that a bomb going off has a look for itself on the map to see if any blocks are destructible. The only processing that needs to be done is therefore by the bomb. The fly in the ointment is that something must keep track of destroyed blocks, for both players, so that they stay

destroyed if the players swap over due to pilot error. There is still need for a list of destructibles and I need a fast search algorithm to move through the list and find the relevant entry to tick it off. A sequential search is NOT very efficient; there is a better way.

Thursday 3rd December

Wrote the necessary binary chop search algorithm which was a useful exercise at least in proving that finding the average of two numbers is not particularly fast in 68000. Although the binary search always gets to the required entry in a few

attempts, each attempt is a lot slower than just hacking through all the entries sequentially - about five times slower to be precise. Can you be precise when using the word 'about'?

Things had just got to the stage where I'd got everything working silky-smoothly when Mark comes in and asks for it to be able to blow up a four-block square object as one, which I could accommodate (and I only put that word in 'cos I can spell it), and then I realised that all of today's and yesterday's work has been a bit of a waste of time.

There's a considerably easier way to do all this keeping track of destructibles that I would expect just about any other programmer to do, and that's to just save away all 32K of the map (one for each player - making 64K) each time the player is destroyed. That way all the destructible blocks are just there waiting in the map for next time. No fancy search algorithm, no having to list all the destructibles, no having to go through that list destroying all the ones already hit and all it costs is 64K of fast RAM. Got buckets of that - don't know what to do with all the damn stuff!

Friday 4th December

In went the map preserve and restore routines and out went the binary search routine. Shame, that, because it was rather clever. Trouble is that it wasn't blantly clever - not that preserving the map is blantly clever either, but it's certainly faster than the old method so it stays.

Having two graphics artists working on a game starts creating file-naming problems. I was numbering the fleets of dreadnoughts in the sequence that they were being created in, and Mark is numbering them in the order that they will end up in the game. Result: Fleets go missing as they get copied onto my hard disk as the same number gets used twice. A massive renaming exercise was therefore



(Left) The two Mantas, one in computer-controlled 'drone' mode, skim the surface of one of the dreadnoughts from the first fleet, whose biomechanical, skeletal structure probably owes a lot to the work of H.R. Giger.

undertaken to organise everything so that no more accidents occur.

On a lighter note, my desk has moved seven feet to the left to fit even more graphics artists into the room. Question: How many graphics artists does it take to change a light bulb? Answer: One, but you have to impress upon him the importance of not changing the light bulb at all; it's really good as it is.

Monday 7th December

Another re-vamp of all the weapons occurred today. The torpedo weapon is officially in, to enable the destruction of certain weaker wall sections; the chaser weapon has been tightened up as it has trouble when it is on the trail of moving objects; the bombs were affecting too wide an area causing too much mayhem too quickly so they've been cut down a bit; and the ioniser weapon has been changed from two thin bursts of toothpaste to one thicker one. Miscellaneous fixes also included making the last smart bomb fired actually work and the bonus score pods give points to the correct player, both howlers in their own rights.

And did I say last Thursday that I'd got buckets of fast RAM? Well, the cupboard is bare and the doggie is going to have to eat cake, or - to put it another way - whoops, I've used all the memory up.

Tuesday 8th December

Got a new map, Ship Two of Fleet One to be precise, so I had to put in all the meany generation points, activating the lifts that bring ships to the runways and all the Uridimme ports. Also put in collisionable walls for the first time, which highlighted an ongoing problem, i.e. you just can't see the high walls until it's too late. Put some black and yellow warning stripes round the edge of the wall, and then tried putting some garish patterns on the wall tops. Tried various wallpaper designs before settling on a throbbing colour pattern. More 'art deco' than military cam-

ouflage, but playability is at stake here, you know.

A concession to one of Jason's moans is that you can now fire other bullets while the shield is up so you don't get impatient waiting for it to run out. Also is a fade to all white as the dreadnought melts away for dramatic effect. It also speeds up the proceedings somewhat.

Wednesday 9th December

The latest addition is a 'gunsight' for the torpedo weapon so you can see where the torpedo is going to land more easily. The calculation to find the right landing position is rather hairy as the torpedo is fired from a moving platform so it takes the speed of the firer, adds a constant speed for the action of firing and then arcs down to the ground, and the sight has to know where it's going to hit.

Gun turrets that appear from inside the dreadnought have been added. A small hatchway opens and the guns raise up, fire one shot and then retreat underground.

Friday 11th December

Blowing hot & cold on the A1200.

Monday 14th December

The game has started running out of memory all over the place so it's getting quite tight in there. First plan was to cut down on the amount of map that I saved for remembering all the destructibles. Then it hit me... Why save all the map? Why not scan it for character

codes that have been destroyed and just remember where they are? Even preserving 1,000 possible locations for each player only uses 4K, not 64K. The restore map then just reads the list of destroyed locations, updates the map before the player restarts and it's all ready to go. This is the third method tried and also the simplest, smallest and quickest.

Tuesday 15th December

Spent all day working on some routines to convert RGB colours to HSV colours. Anyone who has played with DPaint and changed the palette will probably have noticed that you get two sets of three sliders to adjust the colours. RGB colours is how the hardware expects you to supply colour information and HSV is just another way of specifying colours, like using Matabele Gumbo Beads for money. It's just that the translation is a bit tricky, you see.

No manuals really go into details about how to do it so, as it's an artistic matter, why not ask a graphic artist? Why not then ask them how to nail jelly to the ceiling? They'll be grateful I worked this out one day. By the way, the reason for doing this, other than that it's been something I've wanted to know how to do for some time, is that when you're fading between certain colours using RGB values you get undesirable colours in between. The old Atari 800 uses a system more like HSV, it had sixteen colours, and sixteen brightnesses of those colours, great for metallic effects. Some of those colour fades aren't even reproducible on the Amiga.

Anyway, the maths was a bit tricky, not complicated but messy, but now I can convert from RGB to HSV and back, the horrendous bit turned out to be fading between two HSV values. It all hinges on defining a circle of ninety hues (the V in HSV), then deciding how much white is NOT mixed in (the V in HSV, for saturation of colour, lack of white) and finally the brightness of the resulting colour (the V in HSV,

for... luminosity, or something!)

Then to fade you just rotate around the circle of colours by the shortest route and move in and out of the circle for brightness, with a bit of up and down for saturation. Marvelous, eh? How does this affect me. I hear you ask? Well, maybe in a game or two I will use these routines to more effect but for now the Manta glows prettier colours and that's it!

Wednesday 16th December

Added shadows to all the walls on Mark's new deck and put in some appropriate meannies, like space ships on the runways and Uridimaks on the surface. Spent much of the rest of the day marvelling at all the sexy new instructions on the 68020 chip, as featured in the Amiga 1200. They (Motorola) seem to have thought of all the things that are awkward to do on a 68000 and added new instructions and addressing modes to handle them. Wow! By the way, I still don't know if the A1200 has any more sprites than previous Amigas. Probably, as the adverts do.

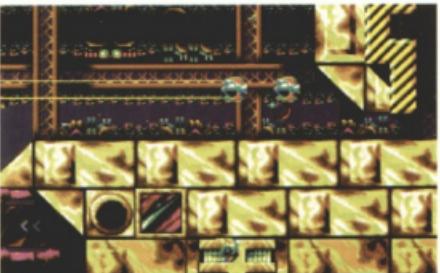
Friday 18th December

It's a 'Thinking about the game design day'. That's my excuse, anyway, for writing a grand total of six lines of code. But as we programmers say, as long as they're a good six lines of code...! Had a drink of coloured water at lunchtime and it turns out that one of our graphics artists is a kleptomaniac. I promised John I wouldn't say who it was though. The very fabric of society is coming apart at the seams. It's a sad old world.

Monday 21st December

I'm making the reactors more difficult to kill by allowing them to shoot back, and they also have satellites to absorb shots. I have to really sort out the underground bit as we only have some test graphics and no real layouts as such.

Work stopped rather abruptly for the Craftgold Christmas lunch at The Victoria which got an all-round thumbs-up. Unfortunately for Uridim 2 not a lot of work got done in the afternoon. Some of our Christmas stash of drinks bit the big one and everyone zig-zagged off home into the sunset. More thinking and designing will need to be done over the festive season ready for some gratuitous programming in the New Year.



(Right) More action above that Giger-inspired dreadnought. Here the collisionable walls are really obvious - just look at those chevrons! Doesn't stop them being easy to smash into, though...

NEXT MONTH!
Duh... Part Eight.



URIDIUM 2

Oooh, we're almost there! Andrew Braybrook's super-blaster for the 90s is oh-so-nearly finished, with his ongoing development diary now in its penultimate month. The game's not due in the shops until around June, but AB is beavering away like a little programming demon to cram in all those wowzer features and explosions. In fact... well, we'll let him tell you all about it...

PART EIGHT - JANUARY 1993

Wednesday 6th January
Spent the last couple of days and some of the Christmas hols checking out various bits of our software on the Amiga A1200, as well as checking out other people's. Well, it's fast, but a lot of stuff won't run any more. In order to determine just

exactly how much faster an A1200 is and to monitor how much spare time Uridium 2 has at its disposal, I've written a routine to see how much time is left after each frame is built.

That's not quite as easy as it may seem because we run a multi-tasking environment in our games so the score panel update is also eating some time but it gets shut out when

the game gets busy. So just measuring the time that is left over after the score process has had its fill won't tell me when it's really in trouble, but it will let me know when it's getting close.

The next tricky bit is to display this information. This process mustn't take too long as it effectively distorts the results. Printing the answer in 32 glorious colours is therefore out of the question. I was just about to key in a table of display colours to change the score panel colour according to the percentage of spare time when I remembered all of the trouble I went to in working out how to do HSV colours last month, which, apart from anything else, required me to type in a table of 90 colours, from red, through yellow and green to blue. Ideal! As long as the spare time doesn't exceed 90% then I can just pick up a colour from this table.

I knew all that HSV stuff would be useful one day!

Thursday 7th January

The A1200 showed itself to have about three times as much spare time, but then I realised that the tiny routine measuring the spare time will all get cached by the 68020 and therefore will run much quicker anyway, making it look like there's more spare time than there would be using the usual mix of blitter, copper and CPU. I'll have to try it with the cache turned off.

We've been considering alternative bonus games, more complex than the old 'slot machine' emulator of the C64 version, but less complex than a game of Chess against someone who's really good at Chess. Lastest plan is a game of intergalactic bar-billiards to get the reactor core to self destruct. Not

quite sure how we're going to explain this in the game scenario. Blame Mark and Simon for this one, not me. The only real way to find out if this is going to work is to program it and try it out on a few people.

Tried loading Uridium 2 on the A1200 through a stereo TV via its inbuilt modulator. Quite a good picture but the sound is rather muffled. Sounds fine through a normal monitor or the hi-fi, which is rather puzzling. Jason is looking into it. Why isn't the stereo output of the Amiga sent to a stereo TV actually in stereo, then? Is there a problem here?

Friday 8th January

It took a bit of thinking to work out what new routines I needed to do this sub-game, and a lot more thinking to work out where in the game to put the test-bed. Being a sub-game it would normally only occur after a couple of minutes playing time, but naturally I don't want to have to waste all that time just getting to the bit that I want to test, so I put that part of the game in front of the titles sequence.

It all gets rather weird at this stage because the joystick is switched off except the fire button at this time, so I couldn't control anything. The joystick is off because I may be running a demo mode and joystick input comes from another routine rather than the real piece of hardware.

All this is now integrated into all the various joystick polling, options key press, demo record/playback routines, etc, so I ended up just having to remove a few lines of code to fool the titles sequence into thinking it was playing a real game. As Scotty said in Star Trek IV: "The more complex the plumbing, the easier it is to block up the drains". Anyway, after debugging the bouncing routine so that I can simulate billiard balls, the sub-game is basically up and running. I can quickly try out different speeds and controls on a number of people and the game does appear to be rather hard and quite slow. More thought required.

Monday 11th January

Gone dotty today. Building rotating throbbing patterns out of dots in an attempt to gather inspiration for the sub-game. My dot particle processor is quite slow when it comes down to it. By the time it has checked the position to see if the dot is on the screen and worked out where it has to go and finally worked out what colour you want it to be it might as well have used a single pixel hardware sprite. Now there's an idea!

Tuesday 12th January

Wrestled, nay grappled, with AmigaDOS for the second time

ever. Jason's 'Amiga Programmer's Guide' won through in the end and told me where to find the magic flags that tell me what CPU is being used. I felt that this is the sort of information that the game should have available and since I don't know how to determine which CPU is which then one may as well ask the operating system.

This is all done at boot time, when the disk is first inserted into the machine, and is communicated up to our operating system, the Kernel, just as it takes over the running of the machine. Now to try it out on the A1200 to see if it spots a 68020 CPU. Having correctly identified a faster CPU then I have the option of switching in 68020 specific assembler instructions (if SNASM will let me) or just running a few more sprites around on the screen. I think it's unlikely that I'll take much advantage of the new chipset at run-time, but the extra memory and faster CPU will all be useful.

Thursday 14th January

SNASM didn't let me. "That's not a real instruction", it said. Never mind. There's more than one way to spoil the broth. For now though, the immediate problem is: Why won't the game run at all? Someone had put an old piece of code back into my program that gulped a hefty 32K save map buffer that I had changed to a 4K buffer ages ago. How can this happen? I had already deduced that the game was crashing because it had run out of memory so I have been looking at the major areas of the game to see where memory could be saved.

Compacted two lumps of plasma down from 7K apiece to 1K, that was quite satisfying. Now at least the program is shoe-horned back into a 1 Meg machine but I'm not sure how much memory I have left. The sub-game is still causing me no end of headaches. I have come up with something fast, simple, exciting and a bit different. Designing the game is just so much harder than programming it. I've done enough programming now that I can code myself out of hefty paper bag without too much difficulty but designing things requires a constant flow of inspiration and new ideas. You can't buy that in shops!

Friday 15th January

Got a pretty swirly star display throbbing in and out but I can't think of a use for it at all. It displays four rings of about a dozen stars in each and is simulating 3D by effectively moving the stars in and out from the centre of the ring and making them glow brighter when further out. The rings also spin at different speeds to give a spiral effect.



Nope, spent most of the afternoon trying to think how I can use that effect but I've failed.

It's about time I could load in smaller maps than the maximum size for the dreadnoughts and get them displayed in the correct position. Managed to crash the data compactor program by feeding it one of the early dreadnoughts which is quite small. Since it's not actually my compactor then I have no chance of fixing it so I'll have to change the maps to get them smaller to start with. At least I can say I've done something useful today.

Monday 18th January

Right, I feel a sub-game coming on, so I had a quick dabble on DPaint to get a rough idea of graphics sizes and colour usage. I'm going to have a fairly extensive palette change to do this display, using six colours to glow various stages of a giant defensive shield. I don't know exactly how I'm going to justify the game in the scenario but that's half the fun, trying to think up a cohesive plot for the whole thing later. Does anybody read the game plots?

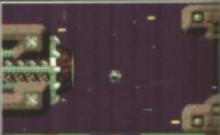
Perhaps I ought to write a snooker game. Mind you, anybody who had never seen the real thing might wonder what the plot is all about. This white sphere is the agent of the alien stick creatures used to destroy the majority red sphere creatures by projecting them down into six bottomless pits. The ethnic minority spheres are then picked off one by one until none but the white traitor remain.

I've relented over the landing the Manta on the final runway. That's now back in as the robot section of the game is to be replaced. It is no longer appropriate for a transformation sequence and a robot bursting through the hull of the dreadnought. Idea cancelled due to lack of interest.

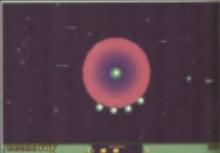
Tuesday 19th January

Tried out a new plot routine that I just feed a shape and it will display it in any of the 32 colours. I spent some time defining circles of different sizes so I can do an expanding red circle of debris when the Manta

Walls marked with yellow and black chevrons are deadly - will you be good enough to get through this tunnel?



(Above) Some of the new-look Uridium ships are broken into two or more pieces.



(Above) The all-new sub-game in its early stages - note Andrew's dodgy scratch GFX.

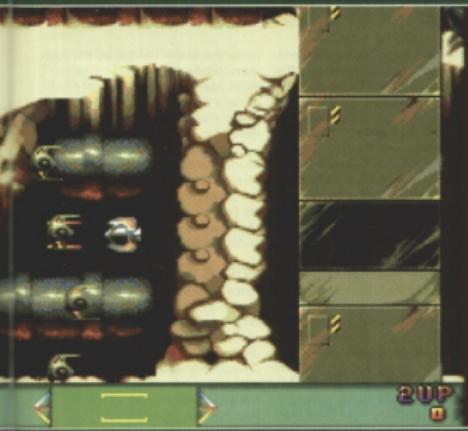




(Above and right) The super dreadnoughts get weird and weird - these two monsters (volcanic above and wood-effect on the right) are currently being toyed with.



(Right) Aaaah! Both Manta ships are in mortal danger here, as a horde of bomb rocks onto the ships' heat signatures - serves them right for hanging around too long.



blows up. The biggest circle I did was 96 pixels across, which seemed huge on DPaint but appears quite insignificant in the game. Bigger circles are required.

The new sub-game control mode fortunately takes the same parameters as the old robot mode so it didn't take long to tune up a workable new control routine. But can anybody cope with it? This bit is now more like an asteroid control, with loads of momentum. How will people get on with this?

Wednesday 20th January

Graeme paid us another visit. This time he had one arm longer than the other because he had dragged an Amiga 4000 over so we can have a look at it, and, more importantly, a listen. Our sound routine is a little unhappy on an A4000; our sound samples are not always coming out right. They sound fine on a 500+ and a 1200, and switching the cache off on the 4000 rectifies the problem so we are talking about a timing discrepancy. The 68040 is so amazingly fast normally that the Amiga sound hardware isn't getting enough time to react. Jason is looking into it.

Mark is busy scanning pictures of rocks, asteroids, moon craters and the like for our asteroid level. He now has a flatbed scanner and a picture publishing package on the PC which allows him to digitise pictures to a very high resolution and in loads of colours. He then has to reduce the number of colours and the resolution to that of the Amiga, and re-touch the images to get them to all fit together.

Thursday 21st January

Whole day spent mulling over the sound routine. Just what is that 68040 data cache doing to our sounds?

Friday 22nd January

Another day on the sound routine. Two of us, now. A frustrating week is marvellously rounded off by getting the phone slammed down by who-ever mans the phones at Commodore Technical Support. First call at 14:20 finds 'The Man Who Knows All' at Commodore still at lunch, ring back later. I figured I'd give him a while as I had some more investigating to do. Ring back at 15:12, leave it ringing for a full four minutes only to be met

by someone saying "All the lines are closed now, ring back Monday at 10 o'clock" and the phone gets slammed down. Someone's in need of a holiday.

Mind you, with a short working day like 10 'til 3 perhaps I should apply for a job there. I mean, here am I, on the front line, trying to get our code working on one last machine, it works on all others, getting no support whatsoever. We've wasted three days trying to figure this out and if this is a known problem, or anyone suggests switching off the data cache, I shall be most disappointed.

Thanks to Toby Simpson and Rob Northen for being considerably more helpful than shedding some light on the problem, under the veil of secrecy that surrounds yet another Commodore product. Just what is it about this total secrecy thing? If I had a new, wonderful computer I'd be shouting about it from the rooftops as soon as it was ready. So what if the opposition find out about it, if it's good they'll be quaking in their boots, and the more people who have correct detailed information about the machine, the more products will get written quickly for it. Any new machine, no matter how wonderful, will surely die without a constant rich supply of software. I'll scratch your back, you scratch yours as well.

Monday 25th January

More test backgrounds for the sub-game. Seconded six extra colours for in-flight modification. I want to represent protective shielding on the ship's power generator with concentric circles of glowing colours. Decided to ditch the mask plot routine. I was using it to generate an expanding circular explosion, but it takes too many circle definitions, and there seems little point in going to all the trouble of running a 32-colour game only to plot explosions in one colour.

Tuesday 26th January

Coaxed two test maps of two new graphics styles out of the graphics artists, the asteroid dreadnoughts from Mark and the battle station from Simon. These will become fleets three and six respectively. Just time to put them into demo format for this month's piccies.

NEXT MONTH!

Experience the joy, the tears and the - hey! - feelings, as the Uridium 2 diary reaches its conclusion. Keep a box of Kleenex handy. Sniff.

URIDIUM 2

There are tears in our eyes here at The One for this is - yes! - the ninth and final instalment of Andrew Braybrook's no-holds-barred programming diary, describing the ins and outs of creating the Amiga-ised sequel to his C64 smash hit. Bye-bye Andrew, don't forget to write...

PART 9 JANUARY/FEBRUARY

FRIDAY 29TH JANUARY

Jason has written a large piece of music for the game intro, taking about 160K of sample space and incorporating, of all things, a didgeridoo. He is somehow blackmailing the rest of the staff to say how wonderful it sounds, but I can't see it myself. The tune itself is very good, but I'm not too sure about Australian ethnic instruments. No one has used one in a computer game before to our knowledge, so at least it's aboriginal! One of Jason's puns, sorry.

Got some graphics in for the guys dropping into the dreadnought from their landed Mantas and much to my surprise Mark suggested taking out the bullet sprites so only a muzzle flash remains, followed by an explosion wherever the bullet hits. That's the first time he's actually suggested anything which saves me some CPU time. I'll get him properly trained one day.

MONDAY 1ST FEBRUARY

Preparing demos to take to France for an exhibition. This is where it would be nice if the whole game was finished in a day, but realistically I have to create demos that are bullet-proof and with as few sticking plasters holding them together as possible.

The sub-game is the weakest-looking area so I concentrated on that. I got two different coloured suited-up pilots facing left and right with gun-recoil frames and tidied up the generator shield rings. I also needed a way of getting out of the sub-game so I had to put in an exit (and entrance) for each player, based on the hyperspace sequence used earlier in the game. Also written is the drone control routine for the sub-game, which attempts to keep the drone on the opposite side of the generator. This works reasonably well, but I haven't had time to smooth out the movement yet. At least it all hangs together.

THURSDAY 4TH FEBRUARY

Returned from an awards ceremony in Paris, empty-handed once again, to more problems and solutions. Rob Northern has been on the dog and bone to Motorola only to discover that someone thought it would be amusing to move the cache-control bits on a 68040 chip, which is why we're having so much trouble switching the cache off. Why not use the same bits as the 68030? What ever happened to compatibility? This, and other questions like 'Why does toast always fall butter-side down?', will probably never be answered.

Serious memory re-organisation required now; the game won't yet run on a humble A500 with half a Meg of fast RAM and half a Meg of chip RAM. That doesn't really surprise me, I have to say, but Mark is starting to ask awkward questions like

"How much memory do I actually have for the graphics, then?". Does he really expect me to know the answer to that? More squeezing required to get the game to fit into all 1 Meg machines.

FRIDAY 5TH FEBRUARY

Interesting fact about Amiga 4000's No. 637: Did you know that the 68040 CPU in the A4000 runs slower than a 68000 in an A500 with all its caches turned off? That fact prompted further investigation into our sound player as we can't just shut the caches off and run the game because we have to have the caches on to even get a sensible speed out of the thing.

So, Jason and I set about finding the difference between our new sound routine which doesn't work on an A4000 and our old sound routine which does. The only possible problem difference we discovered was that the old routine waits for ages after it sets the audio DMA on before it posts the next sample pointer to the chip. That sound chip really does have a mind of its own. Anyway, the sound routine now works, hooray, hooray. So the new Fire & Ice master disk can go for duplication and I feel comfortable about it working on just about any Amiga, and Uridium II will also now work correctly on all Amigas.

MONDAY 8TH FEBRUARY

Two meetings today. One was with Red Cloud to start the ball rolling on the artwork, box design, disk labels, posters and all the other publicity paraphernalia. The second was with another famous music personality to maybe organise some music for our intro sequence. I now have to produce a video of the game so far to provide a bit of inspiration. If only Tom at Renegade hadn't got all my demo disks.

TUESDAY 9TH FEBRUARY

Back down to Earth and some real programming. More work required on the sub-game. Now I can try out some more elements to hassle the players while destroying the reactor core. The problem with the invisible bullets fired by the player is that you can't see the damn things. Obviously really, but it makes checking the collision detection a mite tricky. Got the particle processor involved in a large explosion when the reactor blows, at which time I don't really mind if the processor slows down a bit. It won't on an A1200 so there's a great reason for going out and buying one right now!

THURSDAY 11TH FEBRUARY

It's really hard to find somewhere to have a sensible design meeting in this place. There's twelve of us in three rooms, so you're never alone. Mark and I decided to have an external design meeting, and adjourned to the Archery Bar for a dreadnought designing session. It proved to be fairly fruitful as we got various ideas sorted out and

(Left) Look! There they are! Three of 'em, in a row near that power-up icon! What an talking about? Andrew's new Victory Point symbols, of course!



some rough plans for one of the fleets. The trick with any game is not to make it too difficult. Making a game harder is very easy, making a game easy enough to start with is the tricky bit.

FRIDAY 12TH FEBRUARY

Couple of new features created for Mark today. First off, a searchlight which roves the landing strips and secondly, a cunning chaff system. This detects whether either Manta is firing homing missiles and, if so, generates loads of false targets for them. The missiles thus go haring off at the chaff and do not attack the real targets. Had to play with the number of chaff particles and firing rate to get maximum confusion out of the minimum objects.

Made a fifteen minute video of the game as mentioned on Monday. You'd think that videoing an Amiga would be the easiest thing in the world. Not so. Amiga SCART output doesn't do anything to excite our video, and the video only plays through my monitor in black and white with no sound. Technology, eh? Who needs it?

MONDAY 15TH FEBRUARY

Another successful days programming. A mountain of code (well, a small one anyway) has been created and slotted in at the appropriate points to give what we are calling 'Victory Points' for certain aerial and ground targets. Landing will not be possible until enough victory points have been collected. So, all the code to count them, give them, tell you when you've got enough, generate a 'V' symbol, etc, has all been inserted and the program still runs. But... hardly any of the new code is actually being used as I neglected to include any special targets that could generate victory points. Oh dopey me! Still, I can go home in the knowledge that the game doesn't crash, and I drew a nice graphic for the victory point icon.



(Left) It's easy to tell what's a wall and what's not a wall. For example, here we have big yellow and black chevrons on them and for another your shots splat into them when they hit, as here.



(Left) That big white splash is Andrew's spotlight. It may not look too good here but in play it flickers, giving a pleasing transparent effect.



TUESDAY 16TH FEBRUARY

Tested all the bits that I didn't yesterday, and now I have victory icons generated by fighters, aerial meanies, ground meanies and destructible backgrounds. Any or all of these potential victory point generators are selectable so on different ships I can specify which type of target the player must go for. In reality we'll probably have that fixed

from fleet to fleet, so ship one is always on a timer, ship two might be ground destructibles, ship three might be destroy certain fighters, etc.

I set the victory icon to move down the screen accelerating upwards, a sort of reverse gravity. That does a couple of things: (1) it creates a moving object that catches the eye; (2) it gets the icon clear of the target destroyed so it doesn't get con-

Today I changed the air filter on Brooke Shields' Porsche...

(Left) If you look carefully in that area where the missiles are you can just make out some tiny pieces of chaff. They work a lot better when the game's moving, believe me!

fused with explosions or other generated objects; and (3) it ultimately directs the object off the screen. Just for fun I drew the icon myself (quite pleased with it actually) and shrunk it by one pixel each way at a time to create eight images. Now the icon grows from its generation point.

The new mapper is being road-tested by the graphics department and seems quite quick to generate new maps. A few teething troubles with actual data output formatting have thus far prevented me from trying Simon's new map. Still, these bugs are there to be found.

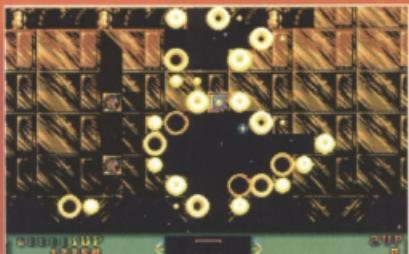
WEDNESDAY 17TH FEBRUARY

Still no mapper output. It has got it into its mind to shuffle the character codes in the map, thus re-arranging them into something not particularly pretty.

Installed a system to change the flashing colour from blue through green, yellow,



(Left) When sufficient Victory Points have been collected the Manta is summoned to battle. The pilot gets to do battle with this thing. The trick is to blast away at the glowing sphere while dodging the homing missiles. This is all played for points and the player can leave at any time if things get hairy, but a big bonus awaits for anyone who can wear the sphere down and blast the thing in the middle.



(Left) Once the Manta pilot has done battle with the sphere, he's invited to take this nice sequence of the dreadnought blowing up bit by bit. Andrew is keen to point out that the explosion graphics were drawn by him, so they may well change in the final game.

orange, then red to show the radiation factor in the subgame. As long as the graphics for the background use the two designated colours then it should be clear to the player when it goes red. Question: Is anyone still using a black and white TV with their Amiga? Better use sound to indicate the radiation level too. That only leaves the people with black and white TVs with broken speakers to worry about.

THURSDAY 18TH FEBRUARY

Actually got some useable output from the new mapper. The character set is now the correct length, in the right order, and looking exactly like it started out. Tremendous. Shuffled the occasional reflect bits about on the map and now the map I can fly over is the same one that Simon created on the mapper. Could it be that the mapper is now bug-free? I wish!

FRIDAY 19TH FEBRUARY

I've finally crumbled and put a cheat mode into the game. Don't get excited though, it'll be removed for the production version with a simple assembly switch: "CheatMode equ No". For now, it allows me to switch in any weapon I want by pressing a function key, and allows me to collect the required number of victory points to land at any time.

I've made the ioniser gun destroy ground targets. That was one of Mark's suggestions from a while ago to allow more choice of weapons

to attack destructible backgrounds. I haven't put that in before because I didn't really believe in it and now that I've tried it I think I'll remove that feature as they are suddenly so powerful that you wouldn't even want to use a different weapon.

MONDAY 22ND FEBRUARY

Day trip to Pinewood film studios with Uncle Tom to talk about a new flatbed model animation sequence. This could be used in still or motion form within certain versions of the game and promotions relating to the game. I rather like the idea of having a detailed model of the Manta ship available. It actually involves creating a real object from an idea rather than an arrangement of magnetic particles on a disk.

TUESDAY 23RD FEBRUARY

I want to simplify the options screen as it's a bit cluttered at present, what with displaying the game mode and two control methods. It would be a lot more Euro-friendly if it used icons. First lesson in graphics: joysticks are easier to draw than keyboards! Too many little fiddly bits on keyboards. Now I need some player icons.

Learned how to use the flatbed scanner and associated software. Amazing is the word for two reasons: (1) great quality computer graphic versions, in full colour, of the printed page; and (2) enormous quantities of hard disk space gobblin' up storing said pictures. I wonder how easy it will be to convert these to HAMB mode on the Amiga.

WEDNESDAY 24TH FEBRUARY

This is going to be the last week of the last instalment of the Uridium 2 diary. Basically I want to have a few surprises left for you when the game is actually complete and I can't do that if we show you a set of pictures of the game every month. It also relieves a bit of the pressure off of me to actually concentrate fully on the game.

It's sometimes a bit tricky remembering something interesting from the day's events to write about, rather like a car mechanic saying "This morning I changed the gearbox on a Metro and this afternoon I put a new set of tyres on a Fiesta..." Whereas if you'd just changed the air filter on Brooke Shields' Porsche then it'd be an altogether different story.

THURSDAY 25TH FEBRUARY

Today I changed the air filter on Brooke Shields' Porsche. No, they'll never believe that. Alright, try... today I helped Mike Montgomery with the sound and keyboard interrupt generators on The Chaos Engine. Now that sounds a bit more believable - just.

The weekend sees Graftgold moving to a new office just down the road so I got a look inside the new place today. Haven't sorted out my 'spot' yet, but I think it's about time I had my own room. Quite uncanny really, but a number of people here agree with that. What can they mean? Anyway, everybody has been given a cunningly marked set of sticky labels to slap all over their computers and desks so that they can find all their gear next week.

FRIDAY 26TH FEBRUARY

Not an overly large quantity of Uridium work being carried out today as the drawers and cupboards get emptied and packed away and the desks and chairs are dismantled. Volunteers required to lift the arcade machine down the stairs. That's definitely going to live on the ground floor in the new place!

So what's left to do on the game? Well, mainly graphics. Lots of work is required to get all seven fleets in and looking great. Plenty of data to be done deciding what means start where, what attack waves to use and general playtesting to make sure that the game is a challenge, but completable.

Well, that's about it for the diary. I hope it's been at least a little bit interesting to read, although I haven't managed to stir up much controversy in Letters. Perhaps I'm getting mellow from talking to our cool drummer, Alfie. I hope it won't be too long before the game is completed and that you'll remember all the hard work and detail that went into writing it. Then, if you like it, please go out and buy it. If you don't like it, then remember that doesn't entitle you to go out and steal it just to see how much you don't like it. And if you wouldn't have bought it under any circumstances then you're not entitled to play it anyway.

Thanks for your attention, maybe we'll meet up at a computer show sometime, or who knows, in a train to somewhere. Remember the game, Uridium 2, and thanks to David Upchurch, Gary Whitta and Simon Byron at The One for putting up with my monthly visits to EMAP Towers (although some put up with me better than others!). Cheers all, this is AB signing off until the next time.



FEATURE

Come on, come on - it's the moment you've all been waiting for, featuring the games you've all been waiting for! Simon Byron probes into the depths of the computer games industry and shouts out once again...

WHERE

...URIDIUM 2?



"Cheers all, this is AB signing off until the next time."

Sniffle. Boo-hoo. Pass the hankies. Just thinking about those poignant words that Andrew Braybrook chose to round off his nine-part, no-holds-barred Uridium 2 diary makes me want to weep (and I'm not scared to, being a 90s man and all that). But hang on a minute. Let's think about what he said just before he signed off. It's all full of damned lies designed to fool you, the viewing nation, into wetting your collective under-wear in anticipation. All this "It won't be too long before the game is completed" business - rot! Tsch.

You can't trust anyone these days.

It turns out that things are just as bad for Andy - he'd spent the night before I'd called him propping up a bar close to home. "Part of my therapy," he said. What about us, then? It's okay for you, you get to see the game day in, day out. Tell us and tell us true - where is Uridium 2?

"Things are getting a little bit behind," he admits. "We're realised a few months back that we weren't going to hit the May release date so Renegade took the decision to hold back until September, which bought a me a little more time but things haven't been going

to plan here."

So what's different since we last spoke, I mean, you last wrote? "Most of the background graphics have changed, mainly because they've been under constant development and things are continually evolving. I've got four graphic artists working on and off and they all keep coming up with different styles and arguing about what palette we should use -

that sort of thing."

It's obviously taken this amount of time for a particular reason. What's been the hardest part so far? "Just managing it all," he laughs. "It's frustrating when you don't do your own graphics any more and

you're at the mercy of other people. But then again it's also nice in some respects because I can concentrate solely on the programming and game design. Sometimes it's a case of knowing what I want to see but not being able to explain it without physically drawing it and if I could draw it then I wouldn't need graphic artists.

"We're spending a lot of time getting the playability just right. What we're finding is that we can't make it easy enough. Because we're so close to the game we don't really have an idea how difficult or easy it is but we're getting messages back

from people like the Bitmaps [name dropping, do you think?] who have been playing it and saying things like 'We can't get past this bit' and so on. It's actually very difficult to make things easy."

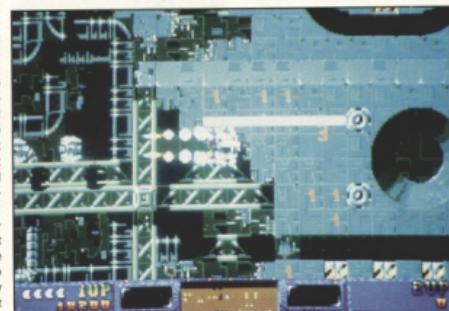
"We have cut down some of the firing rates on the early stages. A turret might stick its head up and sit there for a second saying 'Shoot me' rather than letting off a stream of bullets as it originally did. You've got to be a little bit gentle on the player because the game is much more free-flowing than something like R-Type which forces you through its levels at its own speed - in U2 you can basically go wherever you want whenever you want at whatever speed you want."

"Because there are no speed limits, you've got to ensure that the player doesn't become overwhelmed too early on - although if you do fly around at a breakneck speed then you're likely to end up a copper."

How has this almost finished version compare to how Andy imagined the game would turn out when you

began coding? Is it exactly the same? "Sort of. It doesn't look exactly how I imagined it. It's playing a lot better, mainly because we've made quite a few improvements to the control mode - in some ways it plays even better than the C64 version. It's taken a lot longer than I thought, though. We've had the basic bones up and running about nine months ago and it seems that very little has changed in that time apart from the graphics, but there have been little improvements going in all the time. I certainly haven't been sitting here twiddling my thumbs all day long."

Okay then. When-oh-when will the game be finished? "That's the \$64,000 question. We're redoing the speech and hopefully including two different types of sound effects - one for machines with at least 1Mb of chip memory and one for machines which haven't. That's a major problem - I keep running out of chip memory, but that's another story. Anyway, we're hoping to have all the data in and begin play-testing in the next two weeks. The game should be released in September."





Tricky things, sequels. Especially sequels to well-known classics. And you'd think it'd be so easy, wouldn't you? Give 'em a bit more of what they liked before with a few minor changes and they'll go wild. At least, that's the theory.

And it often works, especially with game sequels. After all, they're limited by their structure and gameplay anyway so there's not really a lot of leeway to do something totally different. If you bought, say, Uridium 2, and when you loaded it up it turned out to be a cutsey platform romp you'd probably be a little miffed I expect (unless it was very good).

Sure enough, then, that's what you get with Uridium 2 - more of the same but much, much better. However, as the original Uridium

appeared on the C64 aeons ago I suspect most of you have only heard about it in fable; a long-distant dream of times gone by when computers loaded from cassette.

As before, six fleets of alien dreadnoughts are slowly approaching the Earth, leaving charred husks of once-colonised worlds in their wake. A lone Manta fighter is launched to intercept them, the theory being that a small ship might just be able to slip past its mighty defences, find their weak spots and destroy them.

The theory works; the dreadnought defences are geared to deal with assaults by cruisers and battleships. However, there are still each dreadnought's on-board fighters to deal with. The scene is set for the Mother Of All Space Battles...



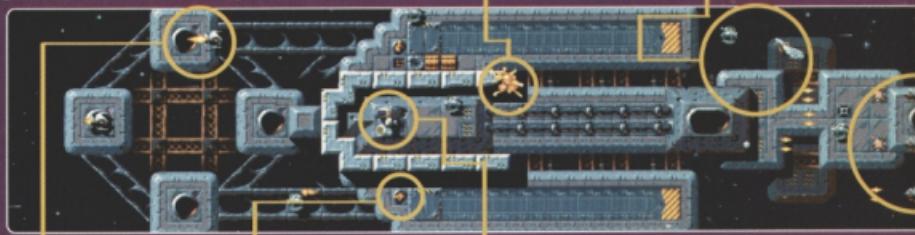
Power-ups appear if you manage to wipe out a complete wave of enemy fighters and offer various point bonuses, temporary shields and weapon upgrades, each with their own advantages and disadvantages. Most enjoyable are the bombs, allowing you to sweep across the ship, blowing the crap out of any deck installation! You can only have one weapon on the go at a time, but if you get bored with it you can revert back to your standard double laser at any time with a waggle of the joystick.



Once you've landed on a dreadnought your pilot will enter its reactor and a bonus sub-game. You task here is to blow away the shields surrounding the core and then ram it. This is made tough by the core's defence systems, the weird gravity, your pilot's inertia and the recoil from your gun! Do it, though, and bonuses will start to appear as rubble pours from the ceiling. You can summon a warp back to your waiting Manta at any time if your shields are taking a battering.

LOOK AT THE SIZE OF THAT THING!

Unlike the first game, which only allowed you to fly left and right along the dreadnoughts, Uridium 2 introduces vertical movement as well. This here piece-together shows you about a third of one of the medium-sized dreadnoughts so, as you can guess, some of the later ships are HUGE!



If you see a hatch or an opening on the dreadnought's hull then take care. While most of the alien ships appear in waves, zooming in from left or right, some may pop up from the dreadnought's interior docking bays and launch surprise attacks.

Each dreadnought is festooned with tiny gun turrets and they can be easily overlooked as you concentrate on downing the incoming alien fighters. You can be sure, however, that they won't overlook you. A quick blast from your lasers will sort them out.

These weird-looking deck features are jammers. They cripple your long-range scanner, making it impossible for you to see if walls or aliens are approaching. If you pick up any torpedoes or bombs take them out ASAP.



IUM 2

Andrew Braybrook returns to familiar territory with his latest, a 90s updating of his Commodore 64 classic. David Upchurch, who was a big fan of the first game, goes positively ga-ga over the triumphant result.

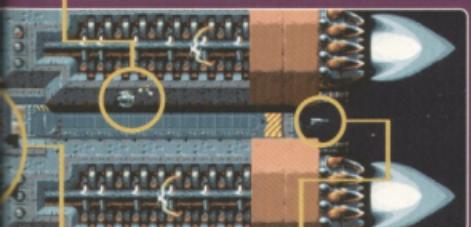


Your Manta can do a lot of damage but even so it's not powerful enough to blow up a dreadnought. So your aim is to find a suitable place to land, allowing you to plant bombs in the reactor core and hopefully nip off before they explode. You're only allowed to land once, however, so a set number of alien attack waves have been dealt with. Certain ships and installations release special 'Victory Points', however, which reduce the number of waves you have to face and thus allow you to land sooner.



Uridium 2 offers a whole host of one- and two-player game options. You can choose to fly alone, with a drone co-pilot or a human co-pilot. In the latter case one player is the lead Manta and controls both players' spent gun, generating movement over the dreadnought, while the other is a wingman who is limited to vertical movement around the lead's position. You can also play a more traditional two-player game where you alternate turns between lives, flying with or without a drone co-pilot.

The dreaded Uridimine! If you hang around one part of the ship too long these will be fired at you. Like the missiles, they home in, but they're faster and much smarter. Run away!



Turn, dammit, turn! If a wave of fighters is sneaking up behind you, the best tactic is to turn back to face them. While you flip over you're temporarily above their line of fire and thus can't be shot down.

Your Manta can be made to spin onto its side, presenting a smaller target to the enemy and allowing you to slip through gaps in the dreadnought's walls. This dreadnought is relatively 'open plan', but later on space can get really tight so learn the skill now.

THE VERDICT

The beauty of Uridium 2 is that, unlike most modern shoot-'em-ups, progress relies purely on the player's arcade skills and not simply on learning where and how the aliens will attack you or amassing a battlecruiser's-worth of power-ups. This makes Uridium 2 a bit of a sod to play at first, and you'll lose many a Manta as you slowly but surely learn how best to pilot it. Once you've acquired that ability, however - and it doesn't take that long, to be honest - you'll find Uridium 2 one of the dreamiest, most satisfying blasts you've ever had the privilege of playing. It's a show-off's delight, in fact, and good players will have any on-lookers oohing and aahing in appreciation as they zip nimbly between narrow gaps on the dreadnought's hull, casually flipping over to zap an incoming alien squadron then spinning back to take out a few gun turrets and launch pads. All the same, gaming weeds ought to be warned that this isn't the easiest of shoot-'em-ups, and even those geezers who boasted that they beat Project-X two hours after buying it will spend many a long night pulling out their few remaining tufts of hair as they mistakenly plough straight into a wall for the fiftieth time. It's worth all the heartache, though, as the sense of achievement you earn from defeating each dreadnought is beyond description. My only moan (and it's a small one) is that the 'exploding dreadnought' sequence isn't a patch on the original one and doesn't quite give you the pay-off you feel you deserve when you complete a level. That aside, Uridium 2 is without doubt a true classic and an essential addition to your software collection. Buy. Now.

A500/600



Publisher: Renegade
Developer: Graftgold

£26.99 Out Now

Not Hard Disk Installable

Joystick/ Mouse

Memory 1Mb

Disk 2

G R A P H I C S

89%

S O N I C

97%

P L A Y A B I L I T Y

90%

L I S T R A B I L I T Y

92%

E T C E R A L
91%

A1200

CD32

Uridium 2 takes great advantage of the A1200's superior power and memory. Graphical tweaks have been added here and there - smoke trails from the homing missiles, spent bullet cases falling away from the pilot's gun, that sort of thing - and there's a special A1200-only 'Mayhem' game mode for those who reckon themselves real star players. Also, as each fleet of dreadnoughts is loaded in, they are stored in memory, eliminating any further disk accessing time. All in all, very nice. A1200/CD32-specific versions are being considered but there are no definite plans as yet.