The Manhattan Project

Time available -- From October 1, 1985 to March 15, 1986, there are 166 days, with 111 work days. From March 16, 1986 to June 1, 1986, there are another 78 days, with 55 work days. This leaves us effectively 166 work days (244 if we work weekends and holidays) to build the WHOLE thing. Urp. Going from June 1, 1986 to September 1, 1986 adds another 64 work days (or another 92 with weekends and holidays) for a total of 230 work days (336 with weekends -- hey, that’s almost a whole year!).

Staffing required -- I see six positions to do the job. The work is distributed as follows (personnel start dates are given in parentheses):

1) Architect/choreographer/project manager (Oct. 1)
2) Commodore 64 graphics programmer (Oct. 15)
3) Commodore 64 internal simulation programmer (Oct. 15)
4) World/object designer/illustrator (Nov. 1)
5) World database coder/communications programmer/system integrator (Dec. 1)
6) Host database programmer (possibly a QL person) (Nov. 1)

Possibly it could also be done with five people, but we should regard that as a fall-back position as it would be more than a little tight.

Timetable --

1985 Oct 1 - Project start; staff selected; hopefully no pre-project paperwork logjams! (please, Steve...)

Oct 15 - Host<-->remote interaction protocol specified; C64 graphics, simulation internals designed; remote graphics format defined; C64 graphics & internals implementation begins

Nov 1 - Host database object internal representation specified; host database server implementation started; C64 graphics & internals implementation underway

Nov 15 - Serious debugging of host database server begins; C64 implementation continues; preliminary world object set & object capabilities specified

Dec 1 - Debugging of host<-->remote communications begins; C64 graphics implementation continues; object set definition continues

Dec 15 - Host<-->remote communications reliable; C64 graphics & internals implementation continues; host database begun; user documentation started

1986 Jan 1 - Implementation proceeds; lots of time wasted during holidays

Jan 15 - First serious attempt to interface between C64 graphics & C64 internals; object set/host database setup continues; user documentation work continues

Feb 1 - C64 graphics & internals working together in preliminary version; minimal host<-->remote interactions working; object database effort proceeding apace; first draft documentation finished

Feb 14 - C64 system working to first approximation; polishing and fixing of inevitable last minute glitches starts; beta-test host system complete and ready for full scale testing with phase I C64 software; documentation revisions abound

Mar 1 - Beta-test object database complete; more polishing and massaging; panic ensues; documentation for beta-test system complete

Mar 15 - Beta-test release; panic subsides, anxiety resumes

Apr 1 - Analysis of first beta-test feedback; bug fixing, feature improvement, and polishing of details continues; object database continues to expand (from here on out, everyone who is not otherwise occupied is busy adding things to the world); April fools' day

Apr 15 - Adjustment & alteration of documentation to reflect recent changes; more polishing, debugging and detailing; income taxes due

May 1 - Polish & debug; debug & polish; more world/object stuff; commies take day off (hopefully no commies here)

May 15 - More of the same; panic resumes; final documentation finished

Jun 1 - Final release of initial system; panic retreats; sleep

Jun 2 - Are we rich yet? No? OK, well I guess we can wait a little while longer...

Jun 15 - Fixing of inevitable bugs that we didn't have time for proceeds in both the C64 and host systems; universe improves

Jul 1 - Fixing of bugs found as a result of thorough thrashing by user community; even more stuff in the universe

Aug 1 - Host support effort downgraded to "maintenance" (i.e., host database programmer goes away; one of the two C64 programmers can go away too, as can the database coder/system integrator)

Sep 1 - Now it really has to be finished; everybody goes away except for those involved with follow-on development of universe (see comments below under "Ramp down")

Sep 2 - *Now* are we rich yet?

Yes, I realize that the schedule gets more and more vague as time passes.

Ramp down -- Programming effort will make a gradual transition from development to maintenance during the period from June to September 1, 1986. This schedule does not take into account possible future work. Such work can come in several forms:

- enhancement of C64 user interface (low probability)

- adaptation of home system to other machines (Amiga, Atari, Apple), probably with enhancements that take advantage of those machines' greater capabilities (high probability assuming the base system is successful)

- expansion of the universe itself

The latter development can itself come in several flavors. The three most notable are:

- addition of new elements by Lucasfilm (i.e., more objects, more object capabilities, new and better graphics and imagery, more places to go/things to do in the world)

- addition of new elements by third parties (the same as the previous item in ultimate effect but very different in terms of the work organization and business arrangements)

- addition of gateways from the universe into otherwise unrelated software & systems which are provided by third parties (the basic mechanism to do this will be built into the system, and we will quite possibly have some such attachments in place when we first go on the air)

By September 1, 1986, the process of adding elements to the universe should be fairly cut-and-dried. We will have developed tools to do all the hard parts, since the technical side of this will be largely a matter of internal book-keeping. Addition of elements will then be primarily an artistic and design problem, as it rightfully should be. One consideration that will be important will, of course, be system load, since each new element consumes disk and memory space and potentially can soak up CPU time. If the system is a success, in time we will have to consider measures to expand its capacity beyond the limits planned for by Q-Link. I do not know at what point this would happen.

Using the MicroCosm as a bridge between the user and other systems will be somewhat more complex, but should be as well understood by the time we hand the system off to full-time manager types.

Availability of the system (and associated tools) to third-party developers can happen anytime between February 1 and the end of the project. Of course, the later it happens, the more stable the environment they will have to work in will be. The timing of third-party involvement will depend upon the degree of their enthusiasm for the project and our assessment of their prospects for self-sufficiency. If they're enthusiastic and really resourceful, they can get on board early. If they're only moderately interested or if they will require lots of hand-holding, they will have to wait until the later stages. Of course, if nobody is interested then it doesn't matter.

Other resources required --

Computers:

We'll need C64s for everyone with the usual complement of disk drives, monitors, etc. They will all also require modems with which to talk to Q-Link. The modems will then need phone lines.

I presume that Q-Link will provide the host computer and any associated hardware that is required on their end, that being their business and all.

We will also need development host machines on our end of the sort we use now: Suns and the Vax. 2 Suns plus the Vax would be ideal. We'll need these machines to run Macross on, to program the C64. We'll need them for text editing & documentation too. Also, I intend to produce a set of development software tools for the definition of the components of the universe. These will largely be very simple preprocessors that will output C or PL/I code to run on the Q-Link host. In all likelihood these tools will be run on our machines to produce output that will be run on their machines.

Some tool for Gary to create graphic imagery for the C64 will also be needed. The presently available tools are crude but workable. It appears that the stuff we already use with the Atari may be adaptable for these purposes. Gary says we can use the Atari screen editor without alteration even though the images created will be going onto the C64. I'm thinking of using an adaptation of Charlie's cel animation driver to render the avatars, and Gary swears he won't use ACE again until some additional capabilities are added to it. I will have to confer with Charlie to see how much work these additional capabilities will require and whether we can get him to do it.

I am presently making heavy use of Ron's Macintosh for architectural and conceptual diagrams. This will continue. We currently have Macintosh Smalltalk-80 on order from Apple. I hope to be able to use this to prototype bits and pieces of the object interactions needed to model the contents of the universe. However, Smalltalk needs a 512K Mac to run, so at present I'll have to steal time on Levine's or Steve's Mac during the evenings when they're not around. It may prove desirable to add a fat Mac for my desk to the shopping list. (Or you might consider just buying a memory upgrade for Ron's machine, which would certainly not be inappropriate compensation given the use that I've already gotten out of it. The only risk is that if Ron should leave for some reason or just decide he wants his computer back, we are stuck.) Getting the Lisa fixed and operational again would also fill this need, though MacDraw on the Lisa is a bit squirly because the pixels are the wrong shape.

Communications:

Since Q-Link is in Virginia and we are here, communications will be a major concern. Once we are tapped into their system we can use *it* for electronic mail service. We won't (I suppose) have to pay for use of their system, but we may have to pay for time on the packet-switching communications network through which our computers will call them. Possibly Q-Link will just absorb this expense. The business types can figure that out. We may want to look into some sort of interface so that our electronic mail system can talk to theirs. I don't have any idea how doable this might be. The existing UUCP software is so fancy it may already have the hooks needed, but I'm not the expert on that. We will also undoubtedly end up spending a small fortune on long-distance telephone calls.

Also, it seems inevitable that there will have to be some travel betwixt here and there, both us visiting them and them visiting us. I don't know how much such travel there will be for sure, but figuring on one or two people making monthly trips in each direction would probably be a good minimum guesstimate.

Depending upon the level of participation of third-party software vendors either as providers of additional universe elements or as clients using MicroCosm as an access mechanism to sell their product, additional travel may be required. This will depend upon who they are, where they are, and how many of them there are, none of which we know. I suspect that third-party participation will be minimal in the beginning phases, but after we open up for beta-test it's anybody's guess.

Miscellaneous:

The usual overhead to cover the cost of light bulbs, thumbtacks, paperclips, ball-point pens, lawyers, printer paper and other trivial but necessary items will no doubt apply.

Can we find it in our hearts to end the tyranny of the coffee chauvinists and get some Coca-Cola in here for these guys? How do you expect people to program under these conditions?

Undoubtedly there are other things, but I can't think of any. It's kind of like packing for a trip to Europe. You feel like you've forgotten something, but you'll never know what it is until you need it.