MicroCosm^{TM} on Other Systems

A “LUCASFILM UNIVERSE” DOCUMENT
some comments on porting the **MicroCosm** *system to other hosts*
by
Chip Morningstar

Lucasfilm Ltd. Games Division
October 17, 1985

Introduction

This document discusses the issues associated with running the **MicroCosm** software on multiple different systems. This includes both porting the centralized host software to other host systems and putting the at-home interactive software on a range of different home computers.

Generally speaking, each component of the system requires certain things of the environment in which it resides. The degree to which a possible target environment meets or fails to meet these requirements will determine the degree of difficulty of porting a particular component into it. This document basically talks about what these requirements are for each portion of the system.

Porting The Host

The host must have, at a minimum, the basic computational resources necessary to support a reasonably large number of simultaneous on-line players. This includes having sufficient disk space to hold the current state of the complete object database as well as the **MicroCosm** system software itself. Much of the working system’s database needs to be memory resident, so lots of core is also needed. The exact extent of the memory requires (both core and disk) will, of course, not be known until the system is implemented, but it should not, in any case, be an unreasonably large amount as contemporary systems go. A similar line of reasoning applies for CPU requirements.

Virtual memory (VM) with a large virtual address space is a major plus though not an absolute requirement with the system as it is currently foreseen. It may be that the implementation will end up relying upon a VM environment, in which case VM of some sort will *become* an absolute requirement. Such a VM subsystem may, however, be implemented as part of the **MicroCosm** host software itself, if that is what is required to install it on another host. This would of course take longer to do than if the host already came with VM support in place.

Since the host’s job is to communicate with lots of **MicroCosm** inhabitants simultaneously, it obviously has to have the data communications facilities with which to do this. Presumably it has this capacity or it wouldn’t be in consideration as a potential host in the first place. A more fundamental issue is the *manner* in which it communicates with its various players.

The host will best support **MicroCosm** if it has a message-based protocol by which it interacts with its on-line players. This is because the **MicroCosm** design uses as message-based, transaction-oriented method of structuring its database accesses and its routing of communications between various players. Some potential host systems do not work this way. For example, some videotex systems rely on a screen or frame oriented model of interaction, in which they transmit fairly large batches of static or semi-static data to the players in predigested chunks. This would not be suitable for **MicroCosm**.

Any additional communications support that the system can provide will also be of great assist in readily getting **MicroCosm** up and running. In particular, communications routing and error recovery are two services that the host ought to provide. Communications routing services free the**Micr oCosm** software from having to know, for example, about the details of getting messages from itself to specific players’ computers. All it has to do is tell the system ‘‘send this message to player X’’. Error recovery services free the **MicroCosm** software from being concerned with the details of message packet construction, error checking, and retransmission, all of which are really none of its business. Many videotex systems, however, do not require such sophistication because most errors in message transmission simply result in a garbling of the display in some way, which is not fatal but merely unaesthetic. Such a garbling is easily remedied when the player sees it and asks for another copy of the screen, whereas errors in **MicroCosm** message transmission could result in all sorts of frightful and hard to detect inconsistencies in the system.

Another thing that the host system must do well is manage interprocess communications within its own operating system. The **MicroCosm** database server is a single, serial process that needs to talk to all of the players at once, each of whom presumably has a process of his own. Thus, we need to be able to support a large number of processes all talking to one common process. This seems obvious, but it is not always easy. For example, this would be quite difficult to do under Unix without some extra cleverness.

Porting The Home System

Requirements for the home system break apart into five rough categories: storage, graphics, sound, communications and input devices.

Input device requirements are, by design, quite minimal. The system requires a simple pointing device (the conventional 4-position joystick will do) and an ASCII keyboard.

Communications requirements hinge upon the ability to support a 300 baud modem. In particular, the home machine *must* be able to carry on exchanges over the modem as a background task (in some machines the modem requires the full attention of the CPU while in use). In other words, it has to be able to talk over the phone, update its display and handle player interaction, all at the same time.

Sound requirements are minimal. Sound is intended to enrich the experience rather than to be an essential part of it. More sophisticated machines with strong sound capabilities will have extensive sound effects associated with the behavior of objects and avatars and will be much the better for it, but systems with little or no sound capability will still be able to run the system without loss of any essentials.

Graphics requirements are significant but basic. The graphics for the home systems, as currently designed, do not rely on any esoteric hardware unique to a particular computer. What is required, rather, is an ordinary memory-mapped display that can be accessed quickly. Spatial resolution does not need to be spectacular, but should be at least comparable to that available on current low-end home machines such as the Commodore 64 or the Atari 800. Color resolution is similarly constrained. Special hardware such as sprites, bitblt processors, and so on, can be used to speed up the graphics performance and so are desirable, but they are not essential. The only rigid requirement is that the display access speed and CPU speed be sufficiently great to maintain the an acceptable frame rate. The previously mentioned home machines are also good examples of systems that minimally meet this requirement.

Storage: the more the better. Since the richness of the universe stems from the richness and variety of the set of objects that are represented within it, the more objects that the home system can hold the better things will work. Minimally, 64K of core and 150K of disk will have to hold the basic system and accouterments. More would, of course, be preferred.