

HabitatTM Extensions

further directions for Habitat development
by
Chip Morningstar

Lucasfilm Ltd. Games Division
October 14, 1986

Introduction

There is a wide variety of different directions that we could plausibly take *Habitat*. The purpose of this document is to outline the options available for extension, both for the short-term and the long-term.

The directions that *Habitat* could be extended fall into four general categories:

- ☐ Enhancements to the fundamental *Habitat* software to improve performance and ease of use.
- ☐ Addition of new objects to the basic object set to allow new activities that were previously not possible.
- ☐ Extension of the present *Habitat* world with the addition of new realms and new adventures.
- ☐ Creation of entirely new worlds using the same underlying software. This involves the creation of whole new object sets and realms to go with them.

I will discuss each of these in turn.

Enhancements to existing software

Since the *Habitat* effort began, we have learned a good deal more about the construction of such systems than we previously knew. Our experiences suggest a number of areas in which the existing software, both on the host and on the Commodore 64, could be improved or extended.

Better graphics

We would like to add a few additional graphics primitives to the present rendering engine to allow greater flexibility and improved aesthetic appeal in the backgrounds we generate. The major thing we would like to attempt is spline curves. These would allow us to display arbitrary, idiosyncratic shapes. This would be useful for shorelines, caves, and other natural features that do not lend themselves readily to portrayal with the orderly, geometric primitives we currently support. We would additionally like to add circles and lines, memory space permitting.

Rewrite the choreography engine

The present choreography engine (the software component that sequences the animation for anything that an Avatar's body does) is a compromise in several respects, having evolved in an environment of constantly changing design demands. We would like to rewrite it from scratch, now that we have a clear understanding of what it must do. This would provide us with cleaner and more flexible Avatar motion sequencing. The new choreography engine would definitely be smaller than the old, freeing up program memory space for other enhancements. The Avatar object instance would also be smaller, so that the amount of "heap" memory available for objects within the region would also be increased.

Improved spatial model

The present *Habitat* world is what is colloquially known as "2½-D". In other words, objects, though they have a position in three dimensions, are simply flats, like pieces of cardboard cut and painted like the

things they represent and stood on edge. While this is ideal for many of the items found in the world, it is not quite right for others. In particular, there are a number of objects which represent things that cover the ground, such as ponds and other bodies of water. It would be nice to have a somewhat more sophisticated spatial model that enabled these things to be represented without resort to the sort of excessive (and error prone) trickery that we currently must use. This entails some changes to the host database representation of all objects (though no changes to the way the host database itself operates) to add a sense of depth. The host collision-detection and movement restriction routines would, of course, also have to be updated to use this additional information.

Better coordination of objects with choreography

There is currently a single motion that is used to illustrate an Avatar bending over to pick something up. Although this is usually just right, as when the Avatar picks something off of the ground, it is sometimes wildly inappropriate, as when the Avatar picks something off of a table. Avatars sometimes don't position themselves ideally with respect to other objects for certain actions, such as opening or closing doors. This is due to the method by which we determine where to position an Avatar with respect to another object. The solution to all of this is to include with each object's imagery a couple of bytes that specify how the Avatar should stand and move with respect to that particular object. This is a relatively simple change, but requires a lot of attention to the painstaking details of each particular kind of object.

Support for additional memory hardware

In particular, the extra 64K of memory available in the Commodore 128 and the extra 512K of memory available in the Commodore 1750 Ram Expansion Module. These could substantially enhance the performance of the home system by drastically reducing or even eliminating the need for disk access (and consequent delays) after initial program loading. Though the market for such an extension is relatively small, this enhancement would be relatively easy (a week or two of effort for one person). If nothing else, it could improve the appearance of demonstrations enormously.

Music driver

The original design called for a music driver to implement such things as jukeboxes, tape players and so on. The music driver in the present system, however, is rather primitive and does not provide the sort of flexibility we would like in order to create music in a very general way. We would like to implement a more sophisticated musical mechanism along with such objects as would make use of it.

Additions to the basic object set

There are a number of classes of objects that would be nice to add to the existing set, since they would greatly enhance the space of possibilities within the world at relatively low cost. These are appealing because they can be added to the running system using the field-update mechanism, since, unlike the enhancements proposed above, they only require additions to the object disk.

Knobs, dials, switches and buttons

These would be a collection of semi-magical objects. "Magical", in the context of *Habitat*, simply means that the host has a large leeway in determining what they do, rather than having their behavior embedded in the Commodore 64 software. These objects would let us construct control panels and other machines for various purposes. For example, we could build a complicated alchemical apparatus, a nerve center for a nuclear power plant, spaceship control panels, mysterious alien machines of unknown purpose, and so on. There is a wide range of puzzles and situations we could set up with some general building blocks of this sort and a little custom code on the host end to glue it all together.

Game playing pieces

Dice, checkers, chess pieces, and so on would allow us to carry on all manner of conventional games within *Habitat*. We already have the geometric graphics primitives to lay out such things as chess or checker boards, backgammon fields, and so on. All we need to add are the playing pieces to use them.

Animals

There seems to be a strong demand for additional sorts of animals than house cats and anonymous dead squished rodents. It would be relatively simple to add dogs, parrots, and the like, giving them the simple to non-existent behavior that our present, limited menagerie possesses. However, a much more interesting idea is to take this as an opportunity to give them all some more sophisticated and interesting behavior. In addition to the imagery and (relatively simple) software for the Commodore 64, this would require some enhancements of uncertain scope to the host, in order to allow it to control its own assortment of autonomous automata.

Robots

Robots could fill a number of potential niches within *Habitat*. The most notable of these is shopkeeper: vending machines have such inhumanity that even the addition of a fairly stupid, obviously automated but nonetheless anthropomorphic shopkeeper would go a long way towards alleviating. Robots could also interact with players in other interesting ways, especially if they could obey simple commands or move around. As with animals, the Commodore 64 software required is relatively simple, but the control mechanism on the host end would need to be carefully crafted in order to operate effectively without overly taxing the resources of the Stratus.

Elevators

Elevators are a special class of TelePort booth that only carry you to a limited number of places and don't cost any money. They can be used to construct apartment buildings, which in turn can be used to enable us to generate large numbers of turf regions without significant human intervention. It will be important to be able to do this when the new-player enrollment rate gets high.

Horizon objects

These would be background scenery to be placed on the horizon of a region to indicate what lies in the distance. These objects would be strips about $\frac{1}{4}$ the width of a screen representing objects in the distance, such as mountains, skylines, treelines, and so on. These would substantially enhance the depth of outdoor scenes. They require virtually no programming time, though they do require a lot of artwork to be generated.

Other graphic styles of existing object classes

We would like to add a lot of additional imagery to vary the present mix of scenic objects. In particular, we would like to add a large number of additional styles of trees, plants, rocks, bushes, doors, windows, buildings, mountains, and so on. These do not require any additional programming at all, since they are merely new faces for types of objects that are already implemented. However, they will require the time and effort of one of our artists to generate them.

Extension of the existing world

There is virtually no limit to the ways that the present *Habitat* world could be extended — expandability was a primary design goal. The principle unfinished task is to define a complete global framework for future expansion. In other words, to create a “map” of the entire world with places reserved for future towns, wilderness areas, playing fields, and other functions yet to be determined. Along with this map there should be a plan for the orderly growth of the world, particularly the residential zones, and for reclamation of inactive areas. Once this plan is formulated, we can begin an orderly program of expansion and extension.

New worlds

The existing *Habitat* software is very general in its structure. It can be used with little or no change to realize other worlds with different visual style and different game play than the original *Habitat*. The first extensions of this sort would want to use the existing main *Habitat* program as is, thus it would not be possible to change certain fundamental things, such as the set of choreography available for Avatar

movement or the set of graphics primitives that scenes are rendered with. However, it is quite practical to create an entirely new object disk that realizes a different world with different objects, different characters, and different possibilities.

Since the original *Habitat* world is designed to be a flexible and general place that allows all sorts of things to happen, the object set has also been designed to be very general and flexible. However, it seems reasonable to adopt a different approach when developing add-on object sets. New object sets could be adapted to a small number of specialized adventures or activities. This would restrict the range of things that could be done with the new object sets, but would also let them be more focused on delivering a particular gaming experience. The number of directions to go in this genre is almost unlimited. However, let's suggest a couple of the more obvious directions here as a starting point for further discussion:

Space World

The *Space World* object disk lets us create realms that are the interiors of spaceships, space stations, and other futuristic structures. Space World is a trading and adventure game. It is basically an adventure game, but more cerebral than violent. The object is to build up an interstellar trading fortune, travelling around the galaxy in your space ship, stopping at various worlds to trade, hire (or get rid of) crew members, have your ship upgraded, and so on. Naturally there are the usual sorts of hazards that are found in such universes, such as Space Pirates, unscrupulous business dealers, shady characters found in sleazy spaceport bars, and the like. But, if you're smart, you can get rich. If you're not, you can get killed.

Hack-N-Slash

Hack-N-Slash corresponds to the more traditional "Dungeons and Dragons" sort of role playing game. The setting is Hollywood Medieval, i.e., not the way the middle ages really were but the way various romantic imaginations would like to think it had been. The level of technology is low, but there is lots of magic. Avatars have numerous specialized attributes, since that is what seems to appeal to the D&D types. The emphasis is on combat, magic use and treasure gathering. Life in the *Hack-N-Slash* world tends to be violent, focused on killing nasty monsters (of which there are plenty) and trying to accumulate a larger hoard of plunder than the next guy.