Region Memory Loading

by
Chip Morningstar

Lucasfilm Ltd. Games Division
March 18, 1987

Introduction

The purpose of this document is to discuss the constraints on Habitat region design based on the memory load in the Commodore 64. We will first discuss the basic problem and give our working assumptions in analyzing it. Then we'll talk about what this means for region design. Finally, we'll say a few words about the present world design and how it meets (or fails to meet) the constraints described here.

The Commodore 64 Memory Model

The Commodore 64 has 14,590 bytes of "heap space" available. This space is used to store all of the objects found in the region currently occupied by the player. These objects include the region itself, all Avatars and everything they are carrying, and any scenic items that may be present. Each of these objects has associated with it imagery, behavior code and sound effects data, in addition to the state information for the object itself. Of course, if two objects share a given sound effect, behavior, or image, we only need to keep one copy in memory.

There is a hard limit of six Avatars per region. In general, we would like to be able to fit at least this many, so we must design our regions so that there is sufficient space left, after the basic scenic objects are taken into account, to hold six Avatars plus a typical assortment of items that they might be carrying.

To compute the memory load for six Avatars in a region, we have to make some assumptions about what those Avatars will be bringing into the region with them, since any objects they are carrying around also contribute to memory consumption. Our assumptions are reasonably generous, in order to be conservative in our planning. For our estimates here, we assume that each Avatar will be carrying a paper object, a Tokens object and a container of some sort, such as a box or a bag (we are assuming that these objects will be out in the open, i.e., not in pocket — one Avatar reading mail, another putting stuff in his bag and another buying something from a vendo, for example). We further assume that each Avatar has a unique head of average size. Finally, we assume that each player is carrying two unique objects of random class in his pockets. Given these assumptions, a single Avatar with his basic resources occupies approximately 10,750 bytes, and each additional Avatar instance costs an additional 400 bytes or so (the largest variable in this number is the size of the Avatar's head). Thus, an empty region with six Avatars requires around 12,750 bytes total. This leaves a little under 2K bytes for the region itself.

Implications

No region can be completely immune to overloading of memory, since there is no way to restrict the amount of stuff that players can bring in and deposit. However, as long as at least one player is able to fit, it will be possible for the players themselves to clear things out in crowded places, assuming that the overload is player-originated and not intrinsic to the region design.

In high traffic areas (i.e., most public areas), we wish to allow the full complement of six Avatars to fit. However, less travelled places can be more crowded. These include backwoods areas that will not tend to be visited by many Avatars at once, activity areas, such as the arcade, where the number of players is already limited by the activity itself (e.g., chess with two players), and private regions (such as turves) where the owner may not care if anyone else can fit in. The choice of typical capacity for any given region will have to be made by the designer on the basis of the intended purpose and location of the region.

The Current World

The 2K working limit for six Avatars actually represents quite a bit of space, since objects are not generally very large (the single largest object is the Avatar itself). Typical region loads found in the present world vary from about .5K to about 3K. Fortunately, most public areas easily fit under the 2K limit. Notable exceptions are the stores with multiple vending machines and the some of the bars (the worst offender seems to be Kelly's Bar and Grill).

The stores are readily fixed by giving them multiple rooms, with one vending machine in each room. This redesign process is already under way. The bars are more tricky, since they are intended to be social spaces. Spreading things out is not the best solution. These will simply have to be redesigned to be more space efficient. Fortunately, one of the major space consumers in Kelly's is a vending machine selling cups and glasses. Now that we have a way of restricting certain objects from being removed from a region (something that was necessary to implement the library), we can eliminate this vending machine.