Habitat Mail

by
Chip Morningstar

Lucasfilm Ltd. Games Division
November 17, 1986

Introduction

This document describes the present design of the *Habitat* mail system. The mail system provides a means for Avatars to communicate with one another even though they may not be logged in to the system at the same time. It also provides a means for players to send messages that may be saved away by the receiving Avatars for future reference.

How It Works: The Player's View

Mail is written and sent using pieces of paper. Any piece of paper can be used to send a letter. A player composes a letter simply by writing his or her message on a piece of paper using the conventional paper/text editor interface. The only special actions that the player needs to take are to write ''To: *whomever*'' on the top line of the page and then to exit the text editing mode by selecting the **MAIL IT** option at the bottom of the screen. The first action provides an address so the *Habitat* Post Office knows who to deliver the letter to. The second action is the means by which the player indicates that this is, in fact, a letter that is now to be mailed out.

When the player selects **MAIL IT** he exits the text interface and sees his piece of paper turn into a winged letter and go flying away from his hands, through the air and off the screen. The player is left empty handed.

Mail is delivered automatically to a special slot in the receiving Avatar's pockets. If the Avatar receiving mail is online when it is delivered, he sees a winged letter come flying in from the edge of the screen that disappears when it reaches the vicinity of his pockets.

To actually read received mail, the Avatar reaches into his pockets using the **GET** command, just as he would to extract an ordinary object. On the ''pick-from'' screen, the player sees a winged letter object sitting in the reserved slot. This icon represents all pieces of received but unread mail. Pointing at it and pressing the joystick button selects the letter for reading. The Avatar winds up with a piece of paper in his hands that is the first unread letter. This piece of paper can then be read like any other, and either saved for future reference or discarded. Further **GET**s retrieve further unread mail, until all pending letters have been retrieved.

If there is no mail waiting, instead of the winged letter object, the player sees a blank piece of paper in his pockets. **GET**ing this retrieves a fresh, blank sheet. Thus, if there is no unread mail waiting, a player always has immediate access to an unlimited supply of paper. In order to alleviate the potential litter problem, **PUT**ing a blank sheet of paper down anywhere causes it to disappear automatically. (By the way, if there is unread mail waiting and the player feels a desperate need to send a message or leave a note, all he needs to do is read his mail until he finds a letter he doesn't need to keep. Its piece of paper can then immediately be erased and reused. He need not wait until all mail is read and he can get at the paper supply.)

How It Works: The Internal View

We create a special graphic style of paper that has two visually distinct graphic states: a blank pad of paper and a winged letter. These are*dif ferent graphic states* of the *same graphic style* of object. This means that the appearance of the object can be toggled back and forth between the two forms without the Commodore 64 having to go to disk or rearrange its heap. A slot is reserved in each Avatar's pockets and

an instance of this type of object is permanently placed there. Initially (i.e., when the Avatar is hatched), the piece of paper is in the blank state.

When an Avatar receives mail, the paper object in his pocket is set to the winged letter state. When he retrieves the last piece of mail awaiting him at any given time, it is set to the plain paper state. The object in his pocket is not otherwise touched.

When an Avatar reaches into his pocket and tries to **GET** the blank piece of paper, instead of handing the object to the Avatar, the host behavior code for the **GET** operation treats this as a special case and clones the piece of paper, handing a new object to the Avatar using a HEREIS message. The Commodore 64 behavior code for this **GET** operation will need to be modified to recognize a ''successful failure'' return from the host, so that it doesn't beep at the player when it is not allowed to remove the item from the pocket.

When an Avatar reaches into his pocket and tries to **GET** the winged letter, the operation is again intercepted as a special case by the host behavior code for the **GET** operation. In this case, however, it works virtually identically to the present **GET** from mailbox operation: a new paper object is generated in the Avatar's hand containing the text of the next message in the player's mail queue. The only notable difference between this and the current mailbox **GET** is that instead of putting down the mailbox flag when all mail has been retrieved (changing the graphic state of the mailbox) the code must turn the winged letter in the player's pocket into a blank pad of paper (changing the paper's graphic state). In other words, the only difference is the object whose state is changed when the mail runs out.

Actually, the **GET** from pocket behavior for both the winged letter and the blank pad of paper are essentially the same. The only difference is the text that is on the piece of paper that is generated: in the case of the letter it is the text of a mail message and in the case of the blank pad it is a blank page.

The send-mail operation is triggered when the player exits the text interface by choosing the **MAIL IT** button. The Commodore 64 sends a special message to the paper object on the host that is a request to send mail. The host behavior code for this request operates like the present mailbox **PUT** behavior.

The arrival of new mail requires an asynchronous notification from the host, so that the Commodore 64 can change the state of the paper in pocket and so it can animate the arrival. This notification will be an asynchronous behavior of the paper object on the Commodore.

What Needs To Be Done

Building on top of the software we already have, here is what we must do to make this mail system design work:

- *Add* **ERASE** *button to text interface.* The paper handling routines already have the code to clear the page. We simply need to bring this function out to the user level. Time estimate: ??
- *Add* **MAIL IT** *button to text interface.* This will drop out of the text interface, send a message to the host similar to that presently sent when dropping a letter in a mailbox and animate the winged letter flying off the screen. Time estimate: ??
- *Create graphic of winged letter.* This is a simple piece of artwork. Time estimate: 2 hours.
- *Add asynchronous behavior code for paper in the Commodore 64 to animate incoming mail.* This is similar to the outgoing mail animation, but the travel path is reversed and action is triggered by an asynchronous message from the host. Time estimate: ??
- *Augment the pick-from-pocket behavior code in both the host and the Commodore to get mail and clone paper.* Other than the triggering action (pointing at something in the pocket rather than **GET**-ing from a mailbox), the new get-mail function can work just like the present one.
- *Add a special case in the* **PUT** *routines so that blank pieces of paper vanish when put down.* This can be handled entirely by the host behavior code. Time estimate: ??
- *Alter the present mailer to deliver to the Avatar rather than to the Avatar's home mailbox.* This entails updating the graphic state of the special paper object in pocket and the sending of a notification message to the player when new mail arrives. Time estimate: ??