HyperText Limited

CS 1706 Project III

Philip Isenhour

Integration Plan

Integration Point I

HyperBook Data Structure

HyperBook Data Structure will be created using a linked list to store the link/file name pairs. The following procedures/functions will be created to handle this data type:

CreateBook	Initialize the book structure
ReadNewBook	Read a new HyperBook from a data file
ValidFileName	Returns true if the string passed to it is the name of an existent file and ends with ".htl"
DestroyBook	Destroy the HyperBook linked-list(s)
BookOpen	Will return true if a book is currently defined (open)
FindFileName	Given a book and a link name, this procedure returns the matching file name
GetFirstChapter	Get the link name of the first chapter in the book
ClearString	(Internal to Implementation) Fill a string with the null character
ConvertString	Convert a string to the String type as defined in the strings package

HyperChapter Data Structure

HyperChapter Data Structure will consist of a "textdoc" (as defined in MFV) to store the text of the chapter and allow for use of MFV's display and scrolling routines. A doubly–linked list of strings will be used to hold the links associated with the chapter. The integer number of HyperNotes will be stored, along with an array of records, each record with one field containing the name
of the note and another containing a linked-list of strings. The link–name of the chapter will also be stored, along with two

"PopMenuType" data structures containing the Links and Notes menu data. The following procedures/functions will be provided:

CreateChapter	Initialize the chapter structure
ReadNewChapter	Read a chapter from the data file
DestroyChapter	Destroy a chapter
GetChapterName	Get the link–name of the chapter
DisplayChapter	Display the text of the chapter in a window
ScrollChapter	Scroll the chapter in the display

Expected Behavior

At this point HTL should be able to successfully read both a HyperBook and HyperChapter file (the first link specified in the book). It should be able to display the file in a window and scroll the file. Scroll errors will be handled.

To accomplish this the following procedures/functions will also be provided:

GetFileName	Get and check a file name
OpenNewBook	Open a book and its first chapter–link
GetKeyCommand	Get a command from the keyboard (will not be complete)
HandleCommand	Check and process commands (will not be complete)
HandleError	Write an appropriate error message (will not be complete)

Integration Point II

PopMenu Data Structure

A "Pop–Up" menu data structure will be created. It will create the smallest possible menu for the items it contains (i.e., no blank lines or blanks at the end
of the longest item in the menu. The horizontal and vertical position of the upper left corner of the menu will be stored, along with a linked list of strings containing the menu items. The following procedures/functions will be provided:

CreateMenu	Create a menu at a specific location on the screen
DestroyMenu	Destroy the menu structure
AddMenuItem	Append an item to the end of the menu
GetMenuItem	Show the menu, wait for the user to choose an item, and return the item name and the ordinal value of the item in the list (or

–1 if no selection)

DrawMenuItems (Internal to Implementation) Redraw the menu,
hilighting the “marked” item

Horizontal Menu

The horizontal menu will be implemented with the following procedures/ functions :

SetUpHMenu Put the menu names into an array of strings
ShowHMenu Display the menu with nothing hilighted
HiliteHMenuItem Redraw the menu, hilighting one of the items
GetHMenuCom Activates the menu, reads the arrow keys, return
key, space bar, and delete key and returns
a command code (enumerated type) based
on the user’s actions

Expected Behavior

All standard HTL windows will be defined. Additional links can be opened (although the link selection menu may not be implemented) and closed, with the stack functioning properly. The horizontal menu will be functional (although all menu items may not be implemented). The current book can be closed and another one opened. The following procedures/functions will be complete:

DefineWindows Define the standard HTL windows
CloseChapter Close the current chapter (do not store it)
CloseCurBook Close the current book and all open chapters
OpenNewChapter Open a new HyperChapter
CheckCommand Check for invalid menu selections
UpdateTopWin Update the “Title Bar” on the screen, adding the
book, link, and program names

Integration Point III

Expected Behavior

The program should be fully functional. Missing sections of HandleCommand and HandleError will have been completed. All horizontal menu items will be implemented, including the pop–up menus. On–line help system will be complete, along with facilities for

displaying the HyperNotes. The title screen will appear at startup and users will be asked to verify the Exit command. Additional procedures/functions will be:

ShowTitleScreen	Show and then clear a title screen
DisplayNote	Show the notes menu, get the user's selection, then display and clear the note
GetLinkName	Show the links menu and return the link–name of the user's selection
SetUpMenus within a chapter	Set up the links and notes menus structure, adding all the names of the chapter's links and notes. (Internal to HyperDoc unit)
ShowHelp	Display a help screen. May display a pop–up menu to allow user to choose help topic.
VerifyQuit	Ask the user if he wants to quit HTL