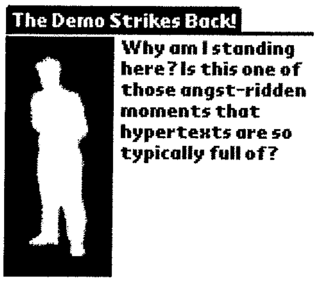
Palm Fiction Technical Documentation

Philip Tan '01 - 10 Dec 1999

Introduction

Although interactive fiction has made great strides on the desktop computer, one enduring criticism remains common: "You can't curl up in bed with a hypertext narrative". Readers use trimmed-down or full-blown HTML browsers on both desktop and palmtop computers, but authors have to use workarounds to bend HTML to narrative purposes. Portable digital assistants (PDAs) allow users to read material with greater freedom of location, but HTML is ill suited for the pixel-scarce screens of PDAs. Current software requires authors of multi-sequential stories to learn complex programming languages and to use structurally elaborate programming constructs to perform conceptually simple, repetitive tasks.



The Demo Strikes Back!

Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of?

HTML works with a *page* metaphor, which has its roots in a static screen display designed for printing, not reading on a screen. The popularity of HTML is partially due to the simplicity of the markup language. However, demand for certain applications has pushed HTML to perform tasks that it was not originally intended to excel at, such as dynamic screen updates, time-aware rendering, computation with persistent states and pixel-accurate layout.

Palm Fiction is capable of rendering pages such as the example displayed on the left, but is also capable of altering the appearance of the display without locking the author in a page metaphor. Furthermore, Palm Fiction addresses all the aforementioned criticisms leveled at HTML by having a feature set targeted at interactive narrative purposes. Palm Fiction encodes stories using its own scripting format, resembling a computer language (instead of a markup format) in order to further exploit the involvement of computation in portable digital narratives.

The Palm Fiction project includes the development of both authoring and browsing tools for multi-sequential stories on the 3Com Palm platform. Authors pick from a simple toolkit of commands to manipulate the 160 x 160 2-bit screen with repeatable precision. The authoring software discourages authors from creating long scrolling fields of in favor of the tap-and-push interface of the platform. Authors can override the hard buttons on the Palm for a more tangible form of interaction, and persistent Booleans allow stories to maintain states as the user reads. The reading software is also interruptible at all times, permitting readers to peruse "on the go" or during short breaks in a busy schedule. These creative constraints and demands require authors to take a thoughtful approach to content creation and navigational design of hypertext narratives.

The Palm Fiction Platform

The Palm Fiction application is available as freeware for authors who wish to experiment with different concepts in interactive fiction. Authors can download the software and sample stories from (http://redirect.to/palmfiction). Editing and browsing is rudimentary and allows experimentation with different interfaces. To provide for the construction of different types of interactive fiction, the Palm Fiction application supports hyperlinking, dynamic screen redraw, pixel-precise positioning of images and text, basic Boolean logic and animation via the use of user-interruptible delays. The application also allows authors to override the hardware buttons specific to 3Com Palm devices.

Palm Fiction is authored for the 3Com Palm III platform, although it should function equally well for the 3Com Palm IIIe, Palm V, Palm VII devices and unexpanded Palm IIIx and Palm Vx devices. The software requires PalmOS 3.0 (or higher) for its screen drawing calls and occupies approximately 30 kilobytes of memory. Authoring, additional stories and variable storage requires additional memory as required by the story code.

The Concept of the Lexia

"George Landau suggested the useful term *Lexias*, taking it from Roland Barthes, who invented it as a term for "reading unit" as part of his theory of texts. See Landau, *HyperText*, 4, 52-53, and Barthes, *S/Z*, 13." - Janet Murray, *Hamlet on the Holodeck*

The *Lexia* divides a Palm Fiction story into discernable chunks of code. In stories that divide the narrative into 'pages', like most interactive fictions written in HTML, a separate Lexia may define each page. Alternatively, a Lexia can denote a chunk of sequential code that executes certain functions. For instance, a Lexia may animate a specific corner of a screen, a Lexia may test and set a series of Booleans, a Lexia can perform a whole string of visible and invisible operations.

Most importantly, many commands can dynamically and interactively direct the story engine to 'jump' to another Lexia in order to perform a different chunk of code. Note that jumping *out* of a Lexia is possible from any command line in a Lexia, but jumping *into* a Lexia requires the story engine to begin execution from the first line of that Lexia. Unlike a subroutine, Palm Fiction does not return the program to a previous routine at the completion of a Lexia chunk; instead, Palm Fiction will halt operation and wait for user input to redirect it to a different Lexia.

The current implementation of Palm Fiction allows authors to separate their code with 'Lexia' commands. These define the beginning of a new Lexia (and the end of the previous Lexia chunk of commands). The *Link* and *Key* commands allow user interactions to direct the engine to jump to another Lexia. *GoTo* unconditionally directs story execution to a specified Lexia. *True?* and *False?* directs story execution to a specific Lexia if Boolean tests are met.

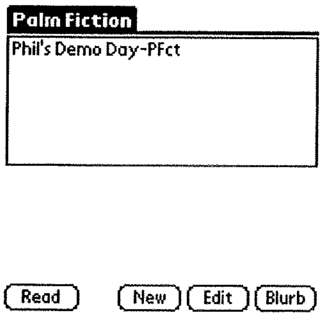
The current implementation of the in-application editing environment gives the impression of a story being a continuous array of code, much like assembly code. This is actually the case, but such a perspective does not help authors maximize the use of the Lexia concept. Furthermore, unlike assembly code, story command execution immediately halts and waits for user interaction when it reaches the end of a Lexia.

Indeed, future implementations may logically separate each Lexia into a separate resource for improved speed in Lexia access.

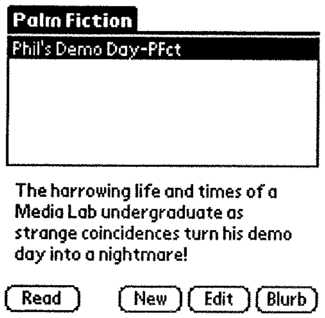
A more appropriate authoring environment should present Lexias having no sequential relationship with any other Lexia — only the code within a Lexia is necessarily sequential. In the most recent version of Palm Fiction, the in-application editing environment maintains the actual code representation of sequential Lexia, requiring the author to segregate Lexias mentally.

Palm Fiction Walkthrough

Palm Fiction is relatively simple to use without a formal instruction manual. The following walkthrough illustrates the range of capabilities by the software. The descriptions included here assume a familiarity with Palm OS GUI conventions such as tapping, selection lists, popup menus, application menus, text editing fields, buttons and so on.

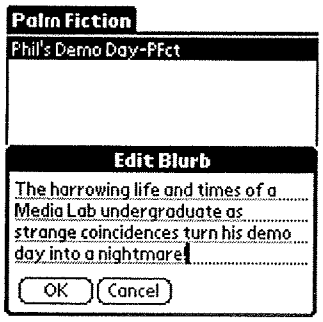


This is the Palm Fiction story list. It currently shows the user that there is one story, called 'Phil's Demo Day', installed in it.



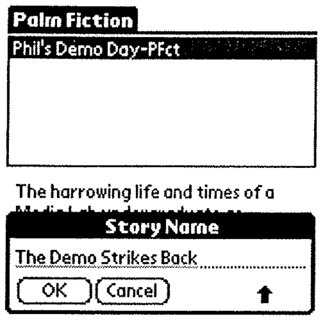
The harrowing life and times of a Media Lab undergraduate as strange coincidences turn his demo day into a nightmare!

Tapping on the story name reveals the blurb, which informs the viewer of the story contents before actually running the story code.



The harrowing life and times of a Media Lab undergraduate as strange coincidences turn his demo day into a nightmare!

The author can change the text of the blurb simply by tapping the 'Blurb' button. This is a standard text-editing box with Undo, Cut, Copy and Paste functionality.

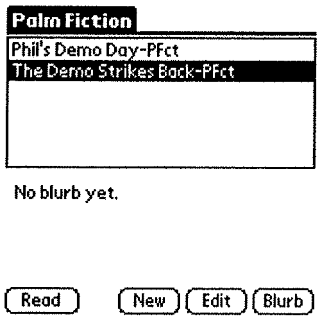


Phil's Demo Day-PFct

The harrowing life and times of a

The Demo Strikes Back

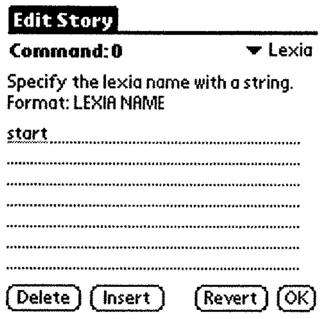
Authors can create new stories from scratch in Palm Fiction, simply by tapping 'New'. The author must enter a fixed story name before creating a new story.



Phil's Demo Day-PFct
The Demo Strikes Back-PFct

No blurb yet.

The new story name appears in the story list. New stories do not have a blurb, and authors can add a blurb by clicking the 'Blurb' button.

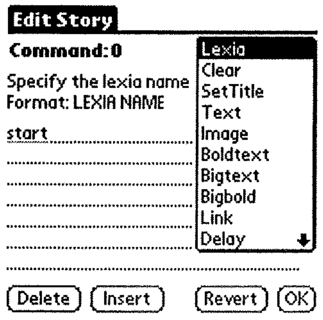


Command: 0

Specify the lexia name with a string.
Format: LEXIA NAME

start

Tapping on 'Edit' will reveal the editing environment. The screen shows which command line the author is currently at, a menu of possible commands, a reminder of the syntax of the selected command and a space to fill in parameters.

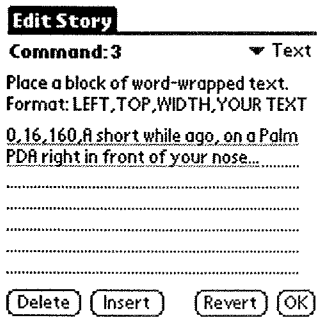


Command: 0

Specify the lexia name
Format: LEXIA NAME

start

The author chooses the command of the command line by tapping on the menu in the upper-right corner, and selecting the desired command.



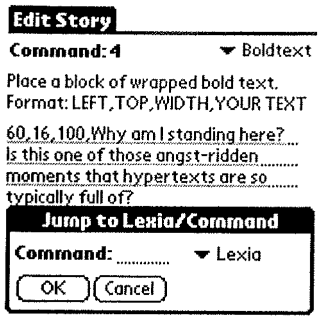
Command: 3

Place a block of word-wrapped text.
Format: LEFT,TOP,WIDTH,YOUR TEXT

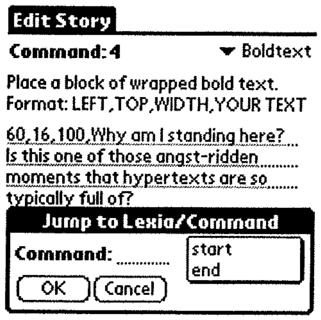
0,16,160,A short while ago, on a Palm PDA right in front of your nose...

In order to look at other command lines, the author uses the hard up and down buttons on the Palm PDA. The 'Insert' and 'Delete' buttons add and delete command lines from the story.

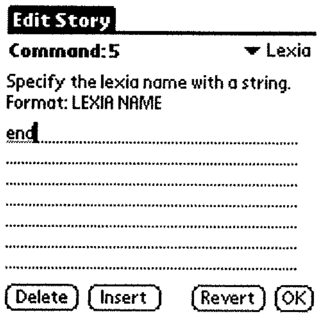
Note that the syntax reminder changes as the author selects a different command. In this case, the 'Text' command needs parameters as a comma-delimited string: left pixel position, top pixel position, width of text block in pixels, text


Edit Story
Command: 4 ▾ Boldtext
Place a block of wrapped bold text.
Format: LEFT,TOP,WIDTH,YOUR TEXT
60,16,100,Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of?
Jump to Lexia/Command
Command: ▾ Lexia

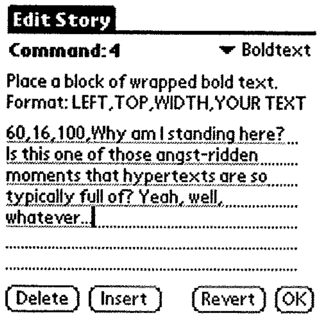
Instead of using the hard up and down buttons, tapping on the 'Command:' caption near the top of the screen brings up this dialog box. The author can enter in a specific command number to jump to.


Edit Story
Command: 4 ▾ Boldtext
Place a block of wrapped bold text.
Format: LEFT,TOP,WIDTH,YOUR TEXT
60,16,100,Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of?
Jump to Lexia/Command
Command:
start
end

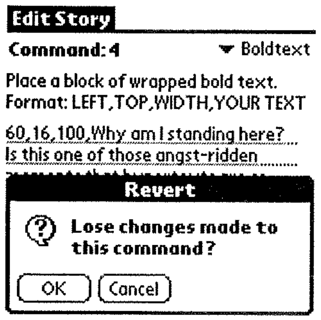
Alternatively, the author can choose from a pop-up menu of all the Lexia in the code of a story, and jump directly to those locations in the code.


Edit Story
Command: 5 ▾ Lexia
Specify the lexia name with a string.
Format: LEXIA NAME
end

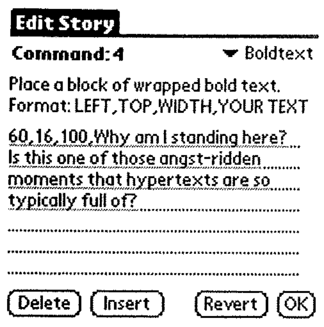
Lexia names are not only important for directing the story engine's flow of control, descriptive names are also helpful for authors in managing stories with many Lexia.


Edit Story
Command: 4 ▾ Boldtext
Place a block of wrapped bold text.
Format: LEFT,TOP,WIDTH,YOUR TEXT
60,16,100,Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of? Yeah, well, whatever...

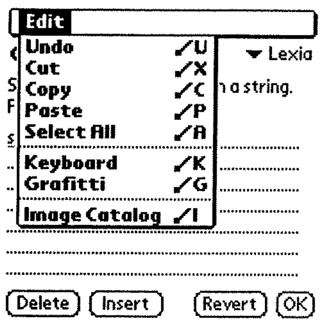
If the author makes an amendment to existing code, the author can restore the original code with the 'Revert' button.


Edit Story
Command: 4 ▾ Boldtext
Place a block of wrapped bold text.
Format: LEFT,TOP,WIDTH,YOUR TEXT
60,16,100,Why am I standing here? Is this one of those angst-ridden
Revert
Lose changes made to this command?

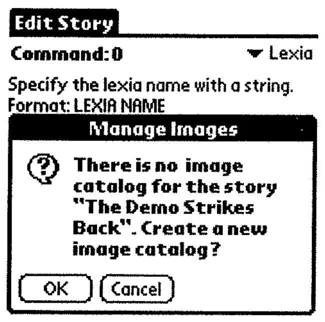
A warning dialog prevents accidental loss of amendments. Note that Palm Fiction saves amendments every time an author views a different line or exits from the main editing environment.


Edit Story
Command:4 ▼ Boldtext
Place a block of wrapped bold text.
Format: LEFT,TOP,WIDTH,YOUR TEXT
60,16,100,Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of?

There are 4 different fonts available in the Palm OS. Even when using 'Boldtext', 'Bigtext' and 'BigBold' commands instead of 'Text', the author places text blocks in the same manner as before.



A closer look at the Edit menu of Palm Fiction demonstrates all the standard text-editing functions, as well as a means to access the image catalog for the story.


Edit Story
Command:0 ▼ Lexia
Specify the lexia name with a string.
Format: LEXIA NAME
Manage Images
There is no image catalog for the story "The Demo Strikes Back". Create a new image catalog?

Since text-only stories do not need image catalogs, the author needs to add an image catalog in order to use images. Selecting 'Image Catalog' from the Edit menu brings up this dialog box that will automate the creation of an image catalog.

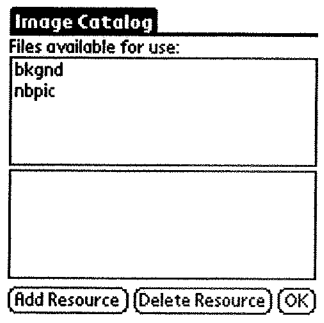


Image Catalog
Files available for use:
bkgnd
nbpic

This is the Image Catalog editing environment. The top rectangle shows every Image Viewer III format file currently in the Palm PDA. The author can import these images into the story's image catalog.

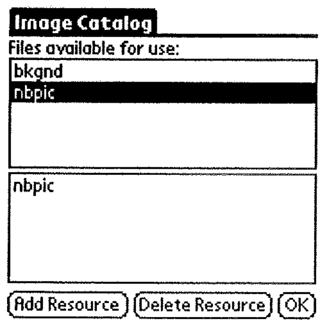
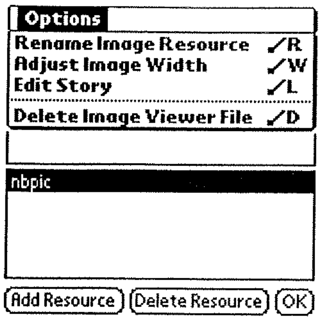


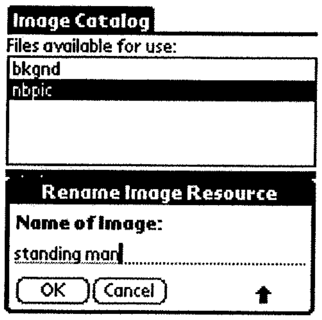
Image Catalog
Files available for use:
bkgnd
nbpic
nbpic

Tapping on the 'Add Resource' button will add a selected image file to the image catalog.



nbpic

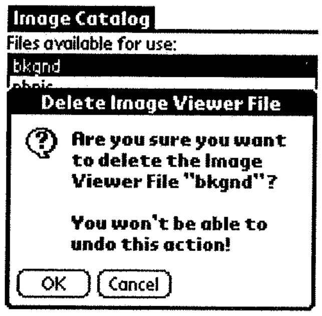
The author has additional functionality available in the Image Catalog Options menu.



Files available for use:
bkgnd
nbpic

Name of Image:
standing man

For instance, the author can rename image resources for easier reference when editing code.

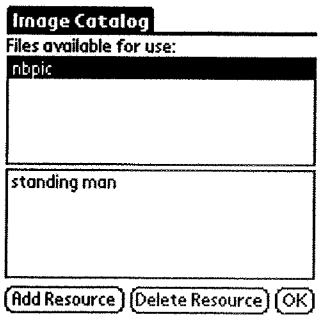


Files available for use:
bkgnd
nbpic

Are you sure you want to delete the Image Viewer File "bkgnd"?

You won't be able to undo this action!

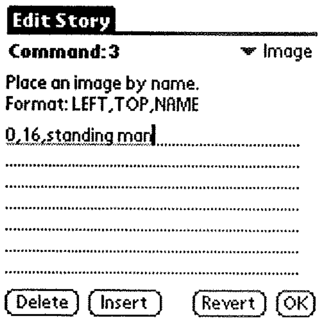
Once the author imports images into the image catalog, the author can remove Image Viewer III files with the 'Delete Image Viewer File' menu command.



Files available for use:
nbpic

standing man

The Image Catalog editing environment updates to reflect the deletion of the Image Viewer File.

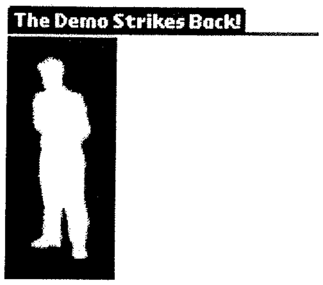


Command: 3

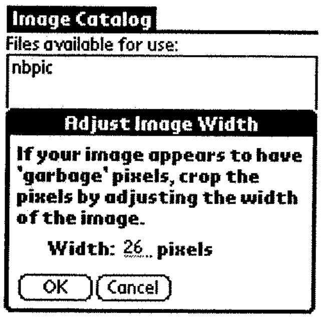
Place an image by name.
Format: LEFT,TOP,NAME

```
0,16,standing man
```

Tapping 'OK' in the Image Catalog environment returns the author to the main editing environment. This code refers to the desired image by its new name and specifies an upper-left pixel position.



Just for reference, the above code will place the image in this manner when interpreting the story code.

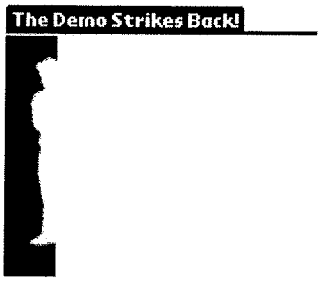


If your image appears to have 'garbage' pixels, crop the pixels by adjusting the width of the image.

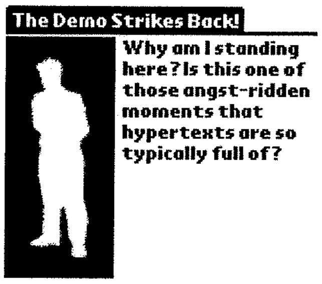
Width: 26 pixels

The Image Viewer III file format only saves files with pixel widths in multiples of 8. If the desired image width is not a multiple of 8, the author can specify an exact image width using the 'Adjust Image Width' command.

The actual width of this image is 56 pixels, but to show the effect of this command, here the author sets it to 26 pixels.

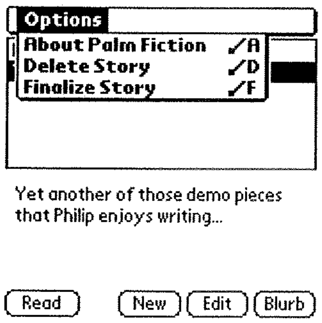


This effectively truncates the image by 30 pixels. However, the invisible image information is present in the Image Catalog. If the author chooses to reset the image width to 56 pixels, the full image will be visible again.



Why am I standing here? Is this one of those angst-ridden moments that hypertexts are so typically full of?

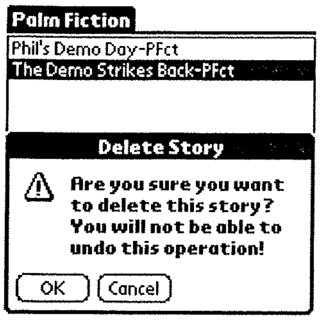
In this case, we can see the effects of sequential 'SetTitle', 'Image' and 'Boldtext' commands.



Yet another of those demo pieces that Philip enjoys writing...

The story list menu allows users to delete or finalize stories.

Deleting stories will allow users to reclaim precious memory in their Palm PDA for more stories or applications.


Palm Fiction

Phil's Demo Day-PFct

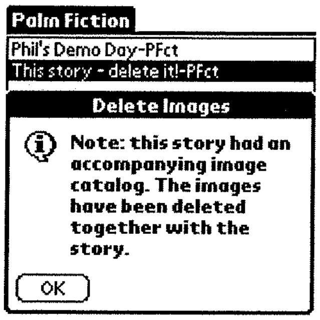
The Demo Strikes Back-PFct

Delete Story

Are you sure you want to delete this story? You will not be able to undo this operation!

OK Cancel

If a user chooses 'Delete Story', this warning dialog box ensures that users have a chance to cancel their decision.


Palm Fiction

Phil's Demo Day-PFct

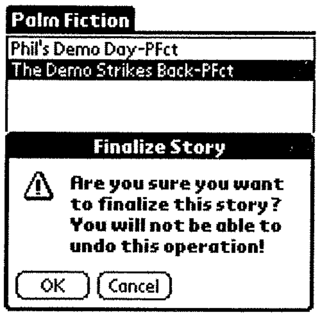
This story - delete it!-PFct

Delete Images

Note: this story had an accompanying image catalog. The images have been deleted together with the story.

OK

If you delete a story that has an accompanying image catalog, the 'Delete Story' command will also automatically remove the image catalog from memory.


Palm Fiction

Phil's Demo Day-PFct

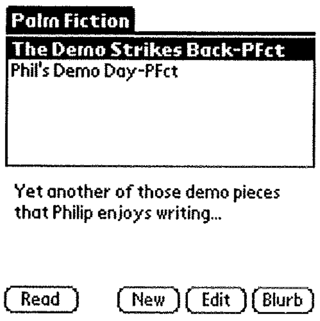
The Demo Strikes Back-PFct

Finalize Story

Are you sure you want to finalize this story? You will not be able to undo this operation!

OK Cancel

Finalizing a story prevents anybody from editing the story. Authors should backup their stories before committing to a Finalize, in case they wish to alter the code later.


Palm Fiction

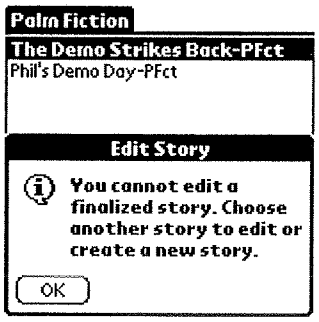
The Demo Strikes Back-PFct

Phil's Demo Day-PFct

Yet another of those demo pieces that Philip enjoys writing...

Read New Edit Blurb

Finalized stories appear in bold and appear above unfinalized stories in the main story list. Blurbs appear in the lower part of the screen as normal.


Palm Fiction

The Demo Strikes Back-PFct

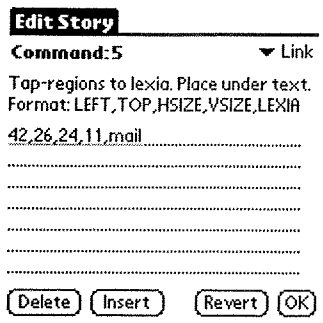
Phil's Demo Day-PFct

Edit Story

You cannot edit a finalized story. Choose another story to edit or create a new story.

OK

If a user taps 'Edit' or 'Blurb' while a Finalized story is selected, this dialog box will appear.


Edit Story

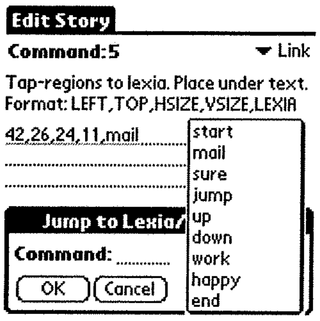
Command:5 ▾ Link

Tap-regions to lexia. Place under text.
Format: LEFT,TOP,HSIZE,VSIZE,LEXIA

42,26,24,11,mail

'Link' commands allow readers to control the flow of a story. It creates a rectangular region on a screen that will direct the engine to jump to another lexia when clicked.

If a 'Link' command specifies a region that overlaps any text blocks, the story must execute the 'Link' command before the text is drawn, or the 'Link' command will not work properly.


Edit Story

Command:5 ▾ Link

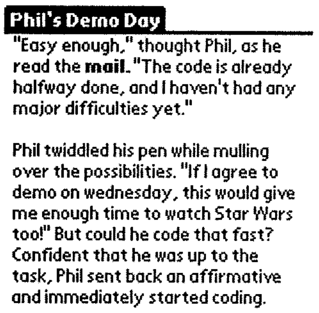
Tap-regions to lexia. Place under text.
Format: LEFT,TOP,HSIZE,VSIZE,LEXIA

42,26,24,11,mail

Jump to Lexia

Command:

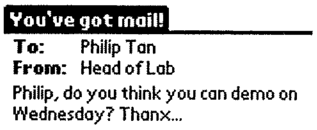
The author must also be careful to specify the parameters without unnecessary spaces or misspellings. In this case, a user tapping on the region starting at pixel (42,26) with a width of 24 pixels and height of 11 pixels will direct the story to Lexia 'mail'.


Phil's Demo Day

"Easy enough," thought Phil, as he read the **mail**. "The code is already halfway done, and I haven't had any major difficulties yet."

Phil twiddled his pen while mulling over the possibilities. "If I agree to demo on wednesday, this would give me enough time to watch Star Wars too!" But could he code that fast? Confident that he was up to the task, Phil sent back an affirmative and immediately started coding.

This region happens to be directly underneath the bold word in the text.


You've got mail!

To: Philip Tan
From: Head of Lab
Philip, do you think you can demo on Wednesday? Thanx...

Thus, when a user taps on the word 'mail', the story engine jumps to the 'mail' Lexia, which includes a number of commands that draw the screen as shown on the left.

Description of Commands

Command	Parameters	Description
Lexia	Lexia name (text string)	Specifies the ending of the previous Lexia and the beginning of a new Lexia.
Clear*	None	Clears the screen of all elements except the title, resets the title of the screen to 'Palm Fiction'.
SetTitle	Title (text string)	Changes the title of the screen to the specified string.

Text	Left position (number 0-160), Top position (number 0-160), Width in pixels (number 0-(160-Left)), Text (text string)	Defines a word-wrapped text block by top-left corner and width in pixels, and places the text string into the text block. Note that the text string can have carriage returns.
Image	Left position (number), Top position (number), Image name (text string)	Draws an image from the image catalog on screen. Note that the image can be partially or fully off the screen, unlike 'Text'.
Boldtext	Left position (number 0-160), Top position (number 0-160), Width in pixels (number 0-(160-Left)), Text (text string)	Similar to 'Text', this command draws text in bold.
Bigtext	Left position (number 0-160), Top position (number 0-160), Width in pixels (number 0-(160-Left)), Text (text string)	Similar to 'Text', this command draws text in a large font.
BigBold	Left position (number 0-160), Top position (number 0-160), Width in pixels (number 0-(160-Left)), Text (text string)	Similar to 'Text', this command draws text in a bold, large font.
Link	Left position (number 0-160), Top position (number 0-160), Width in pixels (number 0-(160-Left)), Height in pixels (number 0-(160-Top)), Lexia name (text string)	Specifies a rectangular region on screen. When the reader taps this region, Palm Fiction jumps and continues executing from the specified Lexia.
Delay	Time in milliseconds (number)	Waits the specified amount of time before proceeding to the next command line. The user can still tap on 'Link' regions or press hard keys in the middle of a 'Delay'. Useful for animation.
Keyup Keydown	Lexia name (text string) OR blank	Redefines the operation of the hard up and down buttons on the Palm PDA, redirecting Palm Fiction to the specified Lexia when a button is pressed. A blank parameter returns the operation of a button to its original function.
Keydate Keyphone Keytodo Keymemo	Lexia name (text string) OR blank	Redefines the operation of the hard application buttons on the Palm PDA, redirecting Palm Fiction to the specified Lexia when a button is pressed. A blank parameter returns the operation of

		a button to its original function.
Goto	Lexia name (text string)	Unconditionally redirects Palm Fiction to the specified Lexia when the story reaches the 'Goto' command.
NewVar	Boolean name (text string)	Creates a new Boolean variable with the value TRUE. A Boolean must be created it can be changed or tested. Note that repeated calling of this command with the same parameter will waste memory in the Palm PDA.
True?	Boolean name (text string), Lexia name (text string)	If the Boolean variable is TRUE, redirects Palm Fiction to the specified Lexia.
False?	Boolean name (text string), Lexia name (text string)	If the Boolean variable is FALSE, redirects Palm Fiction to the specified Lexia.
SetTrue*	Boolean name (text string)	Sets the specified Boolean variable to TRUE.
SetFalse*	Boolean name (text string)	Sets the specified Boolean variable to FALSE.
SetOppo*	Boolean name (text string)	Sets the specified Boolean variable to the opposite value of what it was.
Comment	Comments (text string)	Ignores the parameters. Useful for author's notes or debugging by disabling commands without deleting parameters.

* If a user exits in the middle of reading a story and returns to Palm Fiction, the story executes from the last 'Clear', 'SetTrue', 'SetFalse' or 'SetOppo' command and restores all the variables to its last known status. This usually redraws a screen properly, unless a variable change occurs after the last 'Clear'. Thus, in order to maintain proper variable integrity and screen redraw, 'SetTrue', 'SetFalse' or 'SetOppo' commands should occur directly before 'Clear' commands whenever possible.

To test screen redraw, the 'Screen Refresh' command from the Options menu (available while reading a story) will simulate exiting and reentering Palm Fiction. Alternatively, the hard up and down buttons will have the same effect when pressed while reading a story, if a Keyup or Keydown command has not altered the operation of the buttons.

For reasons of predictability, it is advisable to keep the first command in a story (Command: 0) a Lexia command, and to leave the last command in a story as a Lexia command or a Comment.

Appendix: File Format Design

This section assumes a familiarity with standard Palm C programming terminology.

Palm Fiction stories are Palm Database (.PDB) files distinct from the application. To read a story, the user must upload the story file into the Palm device and use the Palm Fiction application to open the file. The user can also create stories from scratch using the Palm Fiction application, which includes a rudimentary editing mode. The name for each Palm Fiction story possesses the suffix '-PFct' in order to distinguish it from other .PDB files in the same device.

PFct files consist of lines of commands stored as entries in the PalmOS basic database format. The Palm Fiction application accesses each line sequentially by means of an integer line counter. The first character in each line is a byte that represents a specific command. The following commands are available for the author to use, according to the byte representation of each command:

First byte in line	Command name	First byte in line	Command name
00	Lexia	0C	Keydate
01	Clear	0D	Keyphone
02	SetTitle	0E	Keytodo
03	Text	0F	Keymemo
04	Image	10	GoTo
05	Boldtext	11	NewVar
06	Bigtext	12	True?
07	Bigbold	13	False?
08	Link	14	SetTrue
09	Delay	15	SetFalse
0A	Keyup	16	SetOppo
0B	Keydown	17 or higher	Comment

Immediately following the byte will be the appropriate parameters for each command. Some commands, such as Clear and Comment, will ignore any parameters. Commands that begin with 'Key' have different behaviors depending on the presence or absence of parameters. Stories store each set of parameters as a null-terminated string.

The Application Info block for each Palm Fiction story stores a 150-byte null-terminated string known as a *blurb*. This blurb can contain copyright information or a description of the story to aid readers in differentiating stories stored in the same Palm device.

Images for Palm Fiction stories are stored in Palm Resource Database (.PRC) files, distinct from both the application and the story file. A Palm Fiction image catalog possesses the suffix '-PPct'. If a story uses images then both the story file (-PFct) and image catalog (-PPct) must be stored in the same Palm device. Authors can import images from uncompressed 2-bit (4 gray level) Image Viewer III files stored in the same Palm device.

The imported images are either compressed using the PalmOS' internal compression algorithm or not compressed at all, depending on which is smaller. Images are then stored as tBMP bitmap resources as defined by the PalmOS. Because Palm Fiction does not run on black-and-white Palm devices, imported images do not need a corresponding black-and-white version of the same image. Bitmap resources are stored with sequentially ascending resource IDs.

Each bitmap has a corresponding name, stored as tSTR resources with identical resource IDs as the images. When the Palm Fiction application opens a story with a corresponding image catalog, it generates arrays to store the images and the names. These arrays form a lookup table that allows authors to refer to images by name rather than by ID number.

Note that image catalogs are .PRC files, much like Palm applications. This is because Palm applications, like Macintosh applications, are also collections of resources. Although this may lead to some minor confusion, storing images as resources provides very quick access and redraw of images via PalmOS system calls.

Palm files also have a 'creator ID' and 'file type, much like Macintosh files. All Palm Fiction stories and applications have the creator set to 'PFct', allowing the PalmOS to delete both the application and associated files when the user chooses to remove Palm Fiction from the Palm device. 'PFct' is registered in the Palm Computing Creator ID database.

Image catalogs have 'PPct' set as their file type. There are two file types for story files. The 'PFsk' type is for stories that are *fixed*, i.e. the standard Palm Fiction authoring environment can no longer edit the story. Editable story files have the file type 'PFst'. Note that both fixed and editable stories still have '-PFct' in their names. For the most part, authors will not have to worry about file types. They only need to consider adding an image catalog if images are required, or fixing their stories once authoring is complete.

Appendix: Problems with the File Format

The separation of story code and images has proven to be troublesome, although effective from a performance standpoint. Future versions of Palm Fiction should incorporate story code together with images in .PRC files, taking advantage of the increased bitmap drawing speed and reducing the confusion of having to store two files for a single story.

Furthermore, the advantages of separating the stories from the application is suspect. Integrating the story engine with each story will allow users to easily exchange stories via infrared communications. Furthermore, users do not need to download and install a separate application for the story. Also, future changes to the command set or file format will still permit early stories to work with an integrated engine. Finally, the Palm bundled applications specifically hide the separation of application and data for simplicity — the datebook does not appear to have a 'datebook application' and 'datebook files', even though the PalmOS actually separates data in this manner. The convenience of having an additional 30K per story would probably outweigh the miniscule storage efficiency and would better fit the usability standards of the PalmOS.