Get Started Programming with Cyberdog

Description:

This online topic teaches you how to get started with Cyberdog programming. The topic has four sections: Setting up the Build Environment, Creating a CyberTextButton, Adding Drag & Drop and Adding the CyberServiceMenu. The step-by-step format of this topic illustrates how simple it can be to convert a basic OpenDoc part into a fully functioning Cyberdog component. The development is all done with MetroWerks CodeWarrior. Get Started Programming with Cyberdog is written entirely in HTML and is best viewed using Cyberdog 1.0.

Facts:

Web course
http://www.devworld.apple.com

Target Audience:

Experienced C++ programmers who wish to learn how to create Cyberdog parts.

Prerequisites:

Students must have a copy of CodeWarrior 8 or 9 and a copy of ResEdit. Students will need to install OpenDoc 1.0.4 and Cyberdog 1.0 to make best use of the topic. They will also need a working connection to the internet to try out their code. Students should have a solid understanding of C++. It is also helpful to have at least a basic understanding of OpenDoc.

Requirements:

Software:
System 7.5 or later
Metrowerks CodeWarrior 8 or 9
ResEdit

Course Outline:

I. CyberTextButton Tutorial - Intro
- A. Overview
- B. Using this course on the Web
- C. How to use this tutorial
- D. Conventions used

II. Setting up the Build Environmen

A. Introduction
B. Install OpenDoc and Cyberdog
C. Install Metrowerks CodeWarrior
D. Add the Cyberdog development folders to the CodeWarrior folder
E. Fix a bogus warning in the OpenDoc utilities
F. Create some commonly used aliases
G.Go build some sample code and run your part

III. Creating a CyberTextButton Part

A. Introduction
B. Save time and effort by having CodeSampler build your starter files
C. Include the CyberItem files and constants you will need
D. Include the files and constants you will need in the file "MyPart.cpp"
E. Initialize the MyPart's new member variables in MyPart's constructor
F. Create an initial CyberItem using a default URL
G. Initialize Cyberdog
H.Teach MyPart::ExternalizeContent how to write a CyberItem into a storage unit
I. Read in a CyberItem from a storage unit
J. Put the default URL into the STR# resource of the file MyPart.PPC.rsrc using ResEdit
K. Draw the URL in the frame
L. Add some basic code in the MyPart::DoMouseEvent routine to cause the CyberItem to open
M. Build your project and test your code

IV. Adding Drag and Drop support to CyberTextButton

A. Add Drag and Drop method prototypes and data members to you MyPart.h file
B. Update the SOM wrappers to support Drag and Drop
C. Add the OpenDoc interfaces and make our display frames droppable
D. Add code for handling DragEnter in the file MyPart.cpp
E. Add code for handling DragWithin in the file MyPart.cpp
F. Add code for handling DragLeave in the file MyPart.cpp
G. Add code for handling Drop in the file MyPart.cpp
H. Build your part and drop a URL on it

V. Adding the CyberServiceMenu to CyberTextButton

A. Adjust your class definition to deal with ODMenuBar and CyberServiceMenu
B. Include the CyberServiceMenu interface and add a variable to MyPart
C. Modify MyPart::ReleaseAll to dispose of the menus
D. Initialize the CyberServiceMenu in MyPart::Initialize
E. Make sure the CyberServiceMenu gets adjusted when your AdjustMenus method is called
- Notify the CyberServiceMenu when you lose the menu focus
- Notify the CyberServiceMenu when you gain the menu focus
- Modify HandleMenuEvent so that CyberServiceMenu has first shot at handling the event
- Add the MyPart::CreateMenus method to the end of your file
- Add the MyPart::CheckMenus method to the end of your file
- Build your project and try out your new menus