Interacting With a View

• Instead of reviewing the way we have the last few classes, this class we will begin by looking at some code that should help you with your final project. The following script is the View.SelectPoint script (It’s activated with the tool).
View.SelectPoint

theView = av.GetActiveDoc
r = theView.ReturnUserRect
theThemes = theView.GetActiveThemes
if (r.IsNull) then
    p = theView.GetDisplay.ReturnUserPoint
    if (System.IsShiftKeyDown) then
        op = #VTAB_SELTYPE_XOR
    else
        op = #VTAB_SELTYPE_NEW
    end
    for each t in theThemes
        if (t.CanSelect) then
            t.SelectByPoint(p, op)
        end
    end
else
    if (System.IsShiftKeyDown) then
        op = #VTAB_SELTYPE_OR
    else
        op = #VTAB_SELTYPE_NEW
    end
    for each t in theThemes
        if (t.CanSelect) then
            t.SelectByRect(r, op)
        end
    end
end
av.GetProject.SetModified(true)
Programming Themes

- Introduction to Themes
- Object Model Diagrams
- Adding Themes
- Legends
Themes

• A theme is a set of geographic features in a view. A theme represents a source of geographic data including:
  – Arc/Info Coverages and Shape Files
  – A CAD Drawing
  – An Image Source
  – A Database that can be geocoded
Theme Object Model Diagrams
Themes in a View (OMD)
Shapes in A Theme (OMD)
Theme Data Sources (OMD)
Feature Themes (OMD)
Database Themes and Image Themes (OMD)
Adding a Theme to a View

'* Getting the active document from the application (assume it's a view)
myView = av.GetActiveDoc

'* create a new file source
aSrcName = SrcName.Make("a:\newtheme.shp")
'* create a new theme using aSrcName
newTheme = Theme.Make(aSrcName)

'* Add the new theme to myView
myView.AddTheme(newTheme)
Modifying a Legend

themeList = myView.GetThemes
n=1
for each aTheme in themeList
  aTheme.SetVisible(true)
aLegend = aTheme.GetLegend
if (n = 1) then
  aLegend.SetlegendType(#LEGEND_TYPE_COLOR)
aLegend.Quantile(aTheme, "Id", 4)
aThemeSetName("Theme" + n.AsString + " -- Quantile")
myLegendFile = "a:\quant.avl".AsFileName
aLegend.Load(myLegendFile,#LEGEND_LOADTYPE_ALL)
elseif (n = 2) then
  aLegend.SetlegendType(#LEGEND_TYPE_COLOR)
aLegend.Interval(aTheme, "Id", 4)
aThemeSetName("Theme" + n.AsString + " -- Interval")
myLegendFile = "a:\int.avl".AsFileName
aLegend.Load(myLegendFile,#LEGEND_LOADTYPE_ALL)
else
  aThemeSetName("Theme" + n.AsString)
aTheme.SetLegendVisible(false)
end
myView.InvalidateTOC(aTheme)
n = n + 1