

Continued from Part 1

Menu 2 - Entry Appearance

In Civ 5, a drop-down menu is a collection of UI components - a button to display the currently selected value and initiate the drop-down, a grid that forms the borders of the drop-down, a scroll panel to contain the stack of entries and a stack to hold the individual entries, and an instance to define an entry.

The standard "GenericPullDown" style defines all of those items for us, so if we just want the standard look-and-feel we don't need to define any of them ourselves.

However, if we want a non-standard look-and-feel we can over-ride one or more of the standard definitions to change how the menu looks. Let's start by adding the City State trait icon to the City State name in the drop-down.

In the "UI/Menu.xml" file change the <PullDown> element to

```
<PullDown ID="CSMenu" Style="GenericPullDown" ScrollThreshold="170"
Size="190,27" SpaceForScroll="1" Anchor="C,T" Offset="0,90">
    <StackData ID="CSMenuStack"/>

    <InstanceData Name="InstanceOne" >
        <Button ID="Button" Size="200,24" Anchor="L,C">
            <ShowOnMouseOver>
                <AlphaAnim Anchor="C,C" Size="140,24" Pause="0" Cycle="Bounce"
Speed="1" AlphaStart="1" AlphaEnd="0">
                    <Grid Size="140,26" Style="Grid9FrameTurns" />
                </AlphaAnim>
            </ShowOnMouseOver>

            <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
                <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
            </Label>
        </Button>
    </InstanceData>
</PullDown>
```

and change the "UI/Menu.lua" file to

```
local g_SortTable

function OnShowHide(bHide, bInit)
    if (not bHide) then
        UpdateCsList()
    end
end
```

UI Tutorial - Menus

```
ContextPtr:SetShowHideHandler(OnShowHide)

function OnOK()
    ContextPtr:SetHide(true)
end
Controls.OK:RegisterCallback(Mouse.eLClick, OnOK)

function SortByName(a, b)
    local sNameA = g_SortTable[tostring(a)].Name
    local sNameB = g_SortTable[tostring(b)].Name

    return sNameA < sNameB
end

function SortByTrait(a, b)
    local iTraitA = g_SortTable[tostring(a)].Trait
    local iTraitB = g_SortTable[tostring(b)].Trait

    if (iTraitA == iTraitB) then
        return SortByName(a, b)
    end

    return iTraitA < iTraitB
end

function OnCSSelected(iCS)
    local pCS = Players[iCS]

    Controls.CSMenu:GetButton():SetText(pCS:GetName())

    UI.LookAt(pCS:GetCapitalCity():Plot())
end

function SetCsDetails(pCS, entry)
    entry.CSName:SetText(pCS:GetName())
    entry.CSIcon:SetTexture(GameInfo.MinorCivTraits[
GameInfo.MinorCivilizations[pCS:GetMinorCivType()].
MinorCivTrait].TraitIcon)
    entry.CSIcon:SetHide(false)

    local __, csColour = pCS:GetPlayerColors(); csColour.w = 1
    entry.CSIcon:SetColor(csColour)
    entry.CSName:SetColor(csColour, 0)
end

function UpdateCsList()
    local iTeam = Game.GetActiveTeam()

    Controls.CSMenu:ClearEntries()
```

UI Tutorial - Menus

```
g_SortTable = {}

for iCS = GameDefines.MAX_MAJOR_CIVS, GameDefines.MAX_CIV_PLAYERS-1,
1 do
    local pCS = Players[iCS]
    if pCS:IsEverAlive() then
        if (pCS:IsAlive() and Teams[pCS:GetTeam()]:IsHasMet(iTeam)) then
            local sCsName = pCS:GetName()

            local entry = {}
            Controls.CSMenu:BuildEntry("InstanceOne", entry)
            g_SortTable[tostring(entry.Button)] =
{Trait=pCS:GetMinorCivTrait(), Name=sCsName}

            entry.Button:SetVoid1(iCS)

            SetCsDetails(pCS, entry)
        end
    end
end

Controls.CSMenuStack:SortChildren(SortByTrait)

Controls.CSMenu:GetButton():LocalizeAndSetText(
"TXT_KEY_TEST_MENU_CS_CHOOSE")

Controls.CSMenu:CalculateInternals()
Controls.CSMenu:RegisterSelectionCallback(OnCSSelected)
end
```

save the changes, rebuild the mod, restart Civ 5, re-enable the mod and start a new game. Dismiss the dialog, wander around a bit to discover several City States and then display the dialog again (via FireTuner)



The City States now display in their colour, prefixed by their trait icon, and ordered by trait then alphabetically.

The Lua code adds the trait to `g_SortTable`, defines a `SortByTrait()` function and uses that to sort the drop-down list - nothing in that code should be a surprise by now.

The `SetCsDetails()` function performs the actual work of setting the City State name, trait icon and colours into an individual entry - it does what it needs to do, you don't have to understand it.

To over-ride the default Instance in the PullDown we use the `<InstanceData>` element

```
<InstanceData Name="InstanceOne" >
  <Button ID="Button" Size="200,24" Anchor="L,C">
    <ShowOnMouseOver>
      <AlphaAnim Anchor="C,C" Size="140,24" Pause="0" Cycle="Bounce"
Speed="1" AlphaStart="1" AlphaEnd="0">
        <Grid Size="140,26" Style="Grid9FrameTurns" />
      </AlphaAnim>
    </ShowOnMouseOver>

    <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
      <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
    </Label>
  </Button>
</InstanceData>
```

The instance contains a button, which has some mouse over animation and also a label with an icon - this was what we were getting for "free" from the standard drop-down, but now we have to define it all ourselves!

Having changed the drop-down entries to include the trait icon, we'd better add it to the main button as well.

Menu 2b - Button Appearance

In the "UI/Menu.xml" file change the <PullDown> element to

```
<PullDown ID="CSMenu" Style="GenericPullDown" ScrollThreshold="170"
Size="190,27" SpaceForScroll="1" Anchor="C,T" Offset="0,90">
  <ButtonData>
    <GridButton Style="SquareButton" Anchor="L,C" >
      <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
TextureOffset="0.0" Offset="3.3" Size="16,16" Anchor="R,B" />

      <ShowOnMouseOver>
        <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
TextureOffset="0.16" Offset="3.3" Size="16,16" Anchor="R,B" />
      </ShowOnMouseOver>

      <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
        <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
      </Label>
    </GridButton>
  </ButtonData>

  <StackData ID="CSMenuStack"/>

  <InstanceData Name="InstanceOne" >
    <Button ID="Button" Size="200,24" Anchor="L,C">
      <ShowOnMouseOver>
        <AlphaAnim Anchor="C,C" Size="140,24" Pause="0" Cycle="Bounce"
Speed="1" AlphaStart="1" AlphaEnd="0">
          <Grid Size="140,26" Style="Grid9FrameTurns" />
        </AlphaAnim>
      </ShowOnMouseOver>

      <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
        <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
      </Label>
    </Button>
```

UI Tutorial - Menus

```
</InstanceData>
</PullDown>
```

and in the "UI/Menu.lua" file change the OnCSSelected() function to

```
function OnCSSelected(iCS)
    local pCS = Players[iCS]

    SetCsDetails(pCS, Controls)

    UI.LookAt(pCS:GetCapitalCity():Plot())
end
```

and the UpdateCsList() function to

```
function UpdateCsList()
    local iTeam = Game.GetActiveTeam()

    Controls.CSMenu:ClearEntries()
    g_SortTable = {}

    for iCS = GameDefines.MAX_MAJOR_CIVS, GameDefines.MAX_CIV_PLAYERS-1,
1 do
        local pCS = Players[iCS]
        if pCS:IsEverAlive() then
            if (pCS:IsAlive() and Teams[pCS:GetTeam()]:IsHasMet(iTeam)) then
                local sCsName = pCS:GetName()

                local entry = {}
                Controls.CSMenu:BuildEntry("InstanceOne", entry)
                g_SortTable[tostring(entry.Button)] =
{Trait=pCS:GetMinorCivTrait(), Name=sCsName}

                entry.Button:SetVoid1(iCS)

                SetCsDetails(pCS, entry)
            end
        end
    end

    Controls.CSMenuStack:SortChildren(SortByTrait)

    Controls.CSName:LocalizeAndSetText("TXT_KEY_TEST_MENU_CS_CHOOSE")
    Controls.CSIcon:SetHide(true)

    Controls.CSMenu:CalculateInternals()
    Controls.CSMenu:RegisterSelectionCallback(OnCSSelected)
end
```

save the changes, rebuild the mod, restart Civ 5, re-enable the mod and start a new game. Dismiss the dialog, wander around a bit to discover several City States and then display the dialog again (via FireTuner)



And now the main button also has the trait icon.

The Lua changes are minimal but rely on the label and image in the main button having the same names as those in the instance.

In the same way that we used <InstanceData> to over-ride the default Instance, we use <ButtonData> to over-ride the default (main) button

```
<ButtonData>
  <GridButton Style="SquareButton" Anchor="L,C" >
    <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
    TextureOffset="0.0" Offset="3.3" Size="16,16" Anchor="R,B" />

    <ShowOnMouseOver>
      <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
      TextureOffset="0.16" Offset="3.3" Size="16,16" Anchor="R,B" />
    </ShowOnMouseOver>

    <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
    FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
      <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
      Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
    </Label>
  </GridButton>
```

```
</ButtonData>
```

The button has some mouse over animation and also a label with an icon - this was what we were getting for "free" from the standard drop-down, but now we have to define it all ourselves!

Menu 3 - Scroll Area Appearance

So if we can change the appearance of the entries in the drop-down and the main button, can we also change the appearance of the actual drop-down grid. The answer is "yes, sort of".

In the "UI/Menu.xml" file change the <PullDown> element to

```
<PullDown ID="CSMenu" Style="GenericPullDown" ScrollThreshold="143"
Size="220,27" SpaceForScroll="1" Anchor="C,T" Offset="0,90">
  <ButtonData>
    <GridButton Style="SquareButton" Anchor="L,C" >
      <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
TextureOffset="0.0" Offset="3.3" Size="16,16" Anchor="R,B" />

      <ShowOnMouseOver>
        <Image Texture="Assets/UI/Art/Controls/9 Grids/Corner16.dds"
TextureOffset="0.16" Offset="3.3" Size="16,16" Anchor="R,B" />
      </ShowOnMouseOver>

      <Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha"
String="TXT_KEY_TEST_MENU_CS_CHOOSE">
        <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255" Hidden="1"/>
      </Label>
    </GridButton>
  </ButtonData>

  <GridData ID="CSMenuGrid" Anchor="L,T" Offset="0,-42" Size="220,230"
Style="Grid9DetailTwo140" Padding="5,5" />

  <ScrollPanelData ID="CSMenuPanel" Offset="4,2" Vertical="1">
    <ScrollBar Style="VertSlider" Anchor="L,T" Offset="0,18"/>
    <UpButton Style="ScrollBarUp" Anchor="L,T" Offset="0,0"/>
    <DownButton Style="ScrollBarDown" Anchor="L,T" Offset="0,122"/>
  </ScrollPanelData>

  <StackData ID="CSMenuStack" Offset="10,0"/>

  <InstanceData Name="InstanceOne">
    <Button ID="Button" Size="200,24" Anchor="L,C">
      <ShowOnMouseOver>
        <AlphaAnim Anchor="C,C" Size="140,24" Pause="0" Cycle="Bounce"
Speed="1" AlphaStart="1" AlphaEnd="0">
```


UI Tutorial - Menus

```
<Grid Size="140,26" Style="Grid9FrameTurns" />
</AlphaAnim>
</ShowOnMouseOver>

<Label ID="CSName" Anchor="L,C" Offset="40,0" Font="TwCenMT20"
FontStyle="Shadow" ColorSet="Beige_Black_Alpha">
    <Image ID="CSIcon" Anchor="L,C" AnchorSide="O,I" Size="32,32"
Texture="CivSymbolAtlas32.dds" Color="201,248,255,255"/>
</Label>
</Button>
</InstanceData>
</PullDown>
```

and in the "UI/Menu.lua" file change the UpdateCsList() function to

```
function UpdateCsList()
    local iTeam = Game.GetActiveTeam()

    Controls.CSMenu:ClearEntries()
    g_SortTable = {}

    for iCS = GameDefines.MAX_MAJOR_CIVS, GameDefines.MAX_CIV_PLAYERS-1,
1 do
        local pCS = Players[iCS]
        if pCS:IsEverAlive() then
            if (pCS:IsAlive() and Teams[pCS:GetTeam()]:IsHasMet(iTeam)) then
                local sCsName = pCS:GetName()

                local entry = {}
                Controls.CSMenu:BuildEntry("InstanceOne", entry)
                g_SortTable[tostring(entry.Button)] =
{Trait=pCS:GetMinorCivTrait(), Name=sCsName}

                entry.Button:SetVoid1(iCS)

                SetCsDetails(pCS, entry)
            end
        end
    end

    Controls.CSMenuStack:SortChildren(SortByTrait)

    Controls.CSName:LocalizeAndSetText("TXT_KEY_TEST_MENU_CS_CHOOSE")
    Controls.CSName:SetColorByName("Beige_Black_Alpha")
    Controls.CSIcon:SetHide(true)

    Controls.CSMenu:RegisterSelectionCallback(OnCSSelected)

    Controls.CSMenu:CalculateInternals()
```

```
Controls.CSMenuGrid:SetSizeY(230)
Controls.CSMenuPanel:SetOffsetVal(4,2)
end
```

save the changes, rebuild the mod, restart Civ 5, re-enable the mod and start a new game. Dismiss the dialog, wander around a bit to discover several City States and then display the dialog again (via FireTuner)



In theory, all we should have to do is use <GridData> and <ScrollPanelData> to over-ride those standard components

```
<GridData ID="CSMenuGrid" Anchor="L,T" Offset="0,-42" Size="220,230"
Style="Grid9DetailTwo140" Padding="5,5" />

<ScrollPanelData ID="CSMenuPanel" Offset="4,2" Vertical="1">
  <ScrollBar Style="VertSlider" Anchor="L,T" Offset="0,18"/>
  <UpButton Style="ScrollBarUp" Anchor="L,T" Offset="0,0"/>
  <DownButton Style="ScrollBarDown" Anchor="L,T" Offset="0,122"/>
</ScrollPanelData>
```

However, there is a wide variation in the usable space inside different grids and the game engine seems to make some assumptions, which are not true for all grids. You therefore spend a lot of time anchoring, offsetting and sizing the various components to get the required layout only to discover that "Controls.CSMenu:CalculateInternals()" undoes all your work by using its assumptions and changing all your offsets, etc.

Which is why

UI Tutorial - Menus

```
Controls.CSMenuGrid:SetSizeY(230)  
Controls.CSMenuPanel:SetOffsetVal(4,2)
```

occur after the "Controls.CSMenu:CalculateInternals()" call, as they reinstate the required layout!

It's really not worth the effort.

Summary

This tutorial has covered the Civ 5 User Interface (UI) elements and techniques for creating drop-down menus, and they are

- Menus
 - PullDown
 - StackData
 - InstanceData
 - ButtonData
 - GridData
 - ScrollPanelData

All of the XML and Lua code for the examples in this tutorial can be downloaded from the Mod Hub as "Test - UI Tutorial - 3 Menus" (in the Other category). Each menu step is included separately and can be displayed from the FireTuner Lua Console tab by selecting the Menu context and then entering "ShowN()" in the command line (where N is the step to display, eg "Show1b()", "Show3()", etc)

Part 4 of this series of UI Tutorials will cover other types of user input controls - check boxes, radio buttons, edit areas, etc.