Continued from Part 1

World 1b - City View Exclusion

Add the following to the "UI/World.lua" file

```
function OnEnterCityScreen()
  ContextPtr:SetHide(true)
end
Events.SerialEventEnterCityScreen.Add(OnEnterCityScreen)

function OnExitCityScreen()
  ContextPtr:SetHide(false)
end
Events.SerialEventExitCityScreen.Add(OnExitCityScreen)
```

save the changes, rebuild the mod, restart Civ 5, re-enable the mod, start a new game and found your capital.

Our greeting is still there, but this time when you enter City View it disappears

and will reappear when you exit City View.

The Lua code simply hooks the events that the game generates as the City View is entered/exited and hides/shows the greeting respectively.

```
function OnEnterCityScreen()
  ContextPtr:SetHide(true)
end
Events.SerialEventEnterCityScreen.Add(OnEnterCityScreen)

function OnExitCityScreen()
  ContextPtr:SetHide(false)
end
Events.SerialEventExitCityScreen.Add(OnExitCityScreen)
```

Normally we don't just want to place a single piece of static text on the World Map, but we want to place many from Lua - usually as the result of an event.

World 2 - Battle Flags

To illustrate this we will place the "crossed swords icon" on the map every time one of our units is killed.

Change the "UI/World.xml" file to

```
<?xml version="1.0" encoding="utf-8" ?>
<Context>
  <Container ID="FlagContainer"/>
```

```
  <Instance Name="Flag">
    <WorldAnchor ID="Anchor">
      <Label ID="Flag" Size="24,24" Anchor="C,C" String="[ICON_WAR]"
/>
    </WorldAnchor>
  </Instance>
</Context>
```

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

```
include("FLuaVector")
include("InstanceManager")

local g_FlagManager = InstanceManager:new("Flag", "Anchor",
Controls.FlagContainer)
local g_WorldOffset = {x=0, y=-15, z=0}

function OnEnterCityScreen()
  Controls.FlagContainer:SetHide(true)
end
Events.SerialEventEnterCityScreen.Add(OnEnterCityScreen)

function OnExitCityScreen()
  Controls.FlagContainer:SetHide(false)
end
Events.SerialEventExitCityScreen.Add(OnExitCityScreen)

function GetWorldPos(pPlot)
  return HexToWorld(ToHexFromGrid({x=pPlot:GetX(), y=pPlot:GetY()}))
end

function PlaceInWorld(control, world)
  control:SetWorldPosition(VecAdd(world, g_WorldOffset))
end

function OnUnitSetDamage(iPlayer, iUnit, iDamage, iPreviousDamage)
  if (Game.GetActivePlayer() == iPlayer) then
    local pPlayer = Players[iPlayer]
    if (pPlayer:IsAlive()) then
      local pUnit = pPlayer:GetUnitByID(iUnit)
      if (pUnit ~= nil and pUnit:IsDead()) then
        local instance = g_FlagManager:GetInstance()

instance.Flag:SetToolTipString(Locale.ConvertTextKey(pUnit:GetName()))
        PlaceInWorld(instance.Anchor, GetWorldPos(pUnit:GetPlot()))
      end
    end
  end
end
```

```
Events.SerialEventUnitSetDamage.Add(OnUnitSetDamage)
```

save the changes, rebuild the mod, restart Civ 5, re-enable the mod and start a new game. Found you capital and either play (badly) so that your units die or use FireTuner to set up a combat (remember you are trying to lose!)

Every time one of your units dies, the "crossed swords icon" will be place on the map.

Hopefully most of the XML is familiar

```
  <Container ID="FlagContainer"/>

  <Instance Name="Flag">
    <WorldAnchor ID="Anchor">
      <Label ID="Flag" Size="24,24" Anchor="C,C" String="[ICON_WAR]"
/>
    </WorldAnchor>
  </Instance>
```

We create a Container to hold all our "flags" and wrap the WorldAnchor in an Instance element so we can create many of them from Lua.

The City View events now hide and show the FlagContainer (we could hide the context, but this seems to be the "de-facto" approach within the core game code)

```
function OnEnterCityScreen()
  Controls.FlagContainer:SetHide(true)
end
```

```
Events.SerialEventEnterCityScreen.Add(OnEnterCityScreen)

function OnExitCityScreen()
  Controls.FlagContainer:SetHide(false)
end
```

and the rest of the Lua is concerned with detecting a unit's demise, creating an instance of the flag, setting the tooltip for it and placing it on the World Map.

```
include("FLuaVector")
include("InstanceManager")

local g_FlagManager = InstanceManager:new("Flag", "Anchor",
Controls.FlagContainer)
local g_WorldOffset = {x=0, y=-15, z=0}

function GetWorldPos(pPlot)
  return HexToWorld(ToHexFromGrid({x=pPlot:GetX(), y=pPlot:GetY()}))
end

function PlaceInWorld(control, world)
  control:SetWorldPosition(VecAdd(world, g_WorldOffset))
end

function OnUnitSetDamage(iPlayer, iUnit, iDamage, iPreviousDamage)
  if (Game.GetActivePlayer() == iPlayer) then
    local pPlayer = Players[iPlayer]
    if (pPlayer:IsAlive()) then
      local pUnit = pPlayer:GetUnitByID(iUnit)
      if (pUnit ~= nil and pUnit:IsDead()) then
        local instance = g_FlagManager:GetInstance()

instance.Flag:SetToolTipString(Locale.ConvertTextKey(pUnit:GetName()))
        PlaceInWorld(instance.Anchor, GetWorldPos(pUnit:GetPlot()))
      end
    end
  end
end
Events.SerialEventUnitSetDamage.Add(OnUnitSetDamage)
```

The "PlaceInWorld()" function is the only bit of all that that needs some explanation.

```
include("FLuaVector")
local g_WorldOffset = {x=0, y=-15, z=0}

function PlaceInWorld(control, world)
  control:SetWorldPosition(VecAdd(world, g_WorldOffset))
end
```

If we convert a plot to a world position and place our control there, it appears in the middle of the tile - which is probably the worst place to put it as it's where the user is most likely to click. Which in itself is not an issue unless our control happens to be a button - not uncommon - and then the user will inadvertently click the button.

The PlaceInWorld() function uses the VecAdd method from the standard FLuaVector library of functions to add an offset to the converted plot location. In this case the offset places the "battle flag" at the bottom middle of the tile. Given that {x=-30, y=15, z=0} is the top-left vertex of the tile (see ResourceIconManager.lua) you can work others out from that as needed.

Summary

This tutorial has covered the Civ 5 User Interface (UI) elements and techniques for placing controls on the World Map, and they are

- World Placement
    - WorldAnchor
    - :SetWorldPosition

All of the XML and Lua code for the examples in this tutorial can be downloaded from the Mod Hub as "Test - UI Tutorial - 6 World Placement" (in the Other category). Each world step is included separately and can be displayed from the FireTuner Lua Console tab by selecting the World context and then entering "ShowN()" in the command line (where N is the step to display, eg "Show1b()", "Show2()", etc)

Part 7 of this series of UI Tutorials wraps up by covering the infrequently used elements that haven't been discussed elsewhere.