Unicode to UTF-8 missing characters range

Posted by Jorge_Luis - 2013/02/22 14:07

I'm making a window to retrieve some data from the net and show it on a list. Many lines of text have Unicode characters, in the format \uxxxx (where x are hexadecimal values). I changed them to &#xxxx; to made the conversion.
But then, I found the converted characters weren't right. After several test, I finally found the range of the missing characters.
The 0x0120 (G) character is correct, but 0x0121 and 0x0122 aren't right. The two other characters (s and T) are in fact 0x161 and 0x162 (this means a difference of 64).
The most annoying thing is that the DecodeString() function (used in RSS widget) show the right values.

Here's a picture of the test:
http://www.rainlendar.net/cms/images/fbfiles/images/rss_unicode.png

Here's the code I use to populate the list:

```
  Rainlendar_SetItemValue(window, "test.list", "list", "")
  for i=0, 2 do
    data = Test_DecodeString("&#x012" .. i .. ";")
    Rainlendar_SetItemValue(window, "test.list", "list." .. i+1 .. ".type", "1")
    Rainlendar_SetItemValue(window, "test.list", "list." .. i+1 .. ".text", "&#x012" .. i .. "; " .. data)
  end
  for i=288, 290 do
    data = Test_DecodeString("&#" .. i .. ";")
    Rainlendar_SetItemValue(window, "test.list", "list." .. i-284 .. ".type", "1")
    Rainlendar_SetItemValue(window, "test.list", "list." .. i-284 .. ".text", "&#" .. i .. "; " .. data)
  end
  for i=0, 2 do
    data = Test_DecodeString("\\u012" .. i)
    Rainlendar_SetItemValue(window, "test.list", "list." .. i+7 .. ".type", "1")
    Rainlendar_SetItemValue(window, "test.list", "list." .. i+7 .. ".text", "\\u012" .. i .. " " .. data)
  end
  Rainlendar_Redraw(0, window)
```

==========

Re:Unicode to UTF-8 missing characters range

Posted by Jorge_Luis - 2013/02/22 14:35

I found the error. Previously, I changed string.char(val) to Test_Convert(val), with Test_Convert() as:

```
function Test_Convert(value)
  value = math.floor(value)
  if(value==0) then return "\0" end
  if(value==1) then return "\1" end
  ...
  if(value==255) then return "\255" end
end
```

Then, I realise the problem is that in many cases "192 + val / 64" isn't a integer value, and, for example, string.char(196.51) isn't the same as string.char(196). Changing it to string.char(math.floor(192 + val / 64)) solve this issue.

==========