Part 4 - colors, fonts and status window

In this part of the tutorial we will look into text and background colors, fonts and the use of the status window in the Glk version of the interpreter.

Colors

By default the interpreter window shows white text on a black background. XVAN has built in functions to change text and background colors:

- background(<color>) sets the background color
- text(<color>) sets the text color.

Possible values for <color> are blue and black. Note that blue and black must be defined as words in the vocabulary file.

Setting the background color to either blue or white automatically sets the text color to white.

Setting the text color to either blue or white automatically sets the background color to black.

For our sample story we want white text on a blue background, so we change the player's t_init trigger as follows:

o_player.t_init

```
t_init

background(blue) # blue background with white text

printcr(d_init)

printcr("")

entrance(owner(o_player))
```

Fonts (Glk interpreter)

XVAN has print functions that can print **boldface** and *italic* text:

- printbold() and printcrbold()
- printitalic() and printcritalic()

These functions work only in the GLK version of the interpreter. You do not have to make separate game source files for Glk and non-Glk. The non-Glk interpreter will handle these functions but will print normal text.

In our tutorial story, we want the location names to be printed in boldface. In the common t_entrance trigger and the various location t_entrance triggers we change the text "printcr(d_shortdescr)" to "printcrbold(d_shortdescr)". We won't copy it all here, you'll find it in the final story file part4-end.xvn.

Status window (Glk interpreter)

The Glk version of the interpreter has a three line status window on top of the game window. Following functions are available to manipulate the status window:

- clearstatus() clears all text from the status window;
- printstatus() prints text from current cursor position in status window;
- printcrstatus() same as above but adds a carriage return;
- setcursor() positions the cursor at the given position in the status window.

In the status window, we want to continuously display the player's location, number of moves and score. We create a trigger in the player object:

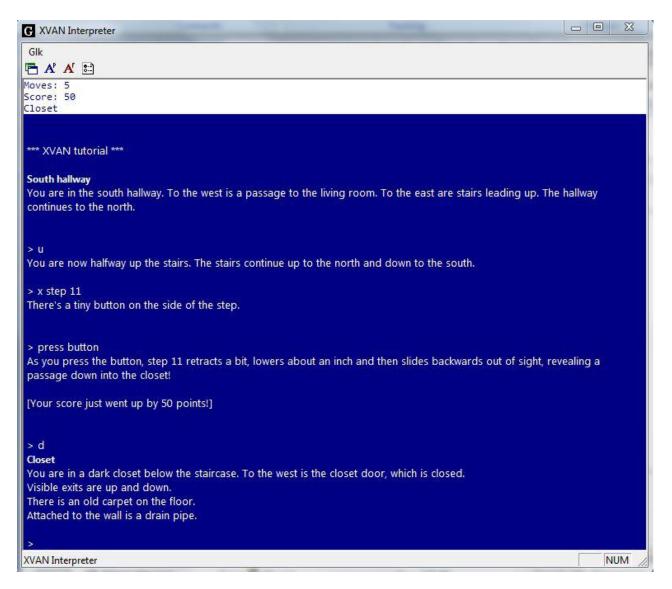
```
t_status_window
clearstatus()
# print the number of moves
setcursor(0,0)
printstatus("Moves: [m_init]")
# print the score
setcursor(0,1)
printstatus("Score: [r_score]")
# print the location's name
setcursor(0,2)
if islit(I_location) then
printstatus(|location.d_shortdescr)
else
printstatus("Darkness")
```

The non-Glk version of the interpreter will accept the commands but do nothing.

We want to refresh the status window at the end of every move, so we create the following timer:

```
m_status_window
init 0
step 0
direction up
interval 1
state go
trigger_at 0
execute o_player.t_status_window
```

This is what the status window looks like (the white part above the blue window).



This is the end of part 4. Everything we did is in the files part4-end.xvn and part4-end.lib (the last file is identical to part3-end.lib).