
Subject: Rainbow Framebuffer - theide running
Posted by [mirek](#) on Fri, 22 Jul 2011 13:44:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

TheIDE running on framebuffer backend:

File Attachments

1) [Clipboard01.png](#), downloaded 602 times

Subject: Re: Rainbow Framebuffer - theide running
Posted by [cbpporter](#) on Fri, 22 Jul 2011 13:54:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wow!

Great!

I brought up the subject of window borders (simulating windows on a flat drawing canvas) and the support Rainbow has for this. I didn't quite understand the details of this, so now is a good time to ask again. I see that the windows have titlebars and buttons. These are managed by the specific Rainbow backend, in this case WinFB? There is code in the backend to draw window borders and handle mouse events on these borders? Simulating what normally is done by Windows or X11?

Subject: Re: Rainbow Framebuffer - theide running
Posted by [mirek](#) on Fri, 22 Jul 2011 17:46:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

cbpporter wrote on Fri, 22 July 2011 09:54Wow!

Great!

I brought up the subject of window borders (simulating windows on a flat drawing canvas) and the support Rainbow has for this. I didn't quite understand the details of this, so now is a good time to ask again. I see that the windows have titlebars and buttons. These are managed by the specific Rainbow backend, in this case WinFB?

Yes.

Quote:

There is code in the backend to draw window borders and handle mouse events on these borders? Simulating what normally is done by Windows or X11?

Exactly. Now, it is possible that if ever FB is used for embedded devices, all of this effort is a little bit overkill, as I think that on embedded, there will be no windows.

But FB is also "proof of concept" for rainbow. So I wanted to have it complete...

NOTE: Implementation-wise, those window borders are somewhat hacked solution - FB Ctrl code basically implements borderless windows with owners and TopWindow is implemented as "sort of" owned (only for internal implementation) by TopWindowFrame (but it is not visible outside of Framebuffer Ctrl implementation).
