Subject: Re: Rainbow, first iteration
Posted by mirek on Tue, 12 Jul 2011 16:02:08 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 12 July 2011 09:46especially FBUpdate and FBFlush, when exactly are they called and what exactly are they supposed to do?

Well, unfortunately, meanings are not yet 100% fixed, as I am still solving some issues with mouse cursor fluidity, but actual broad meaning is:

FBUpdate - tell that some portion of framebuffer is to be moved to the screen. At the moment, it is not required nor prohibited that the update happens immediately.

FBFlush - at the end of function, all previous FBUpdate should be visible on the screen. If they are moved by FBUpdate, FBFlush can be NOP.

Quote:
Framebuffer is performing its drawing of the controls to a BufferPainter, whoose content can then be bitblitted to the real framebuffer.

Yes.

Quote:
(what are the other ImageDraw

Provides means for widget to draw on its view area directly (without Refresh/Paint).

Quote:
, BackDraw etc.)

Not used for Framebuffer; on system with direct painting, it is used when BackPaint mode is active (to provide backbuffer).

Quote:
Framebuffer expects / calls some functions to help finish that process.

Yes.

Quote:
it also expects the real backend to generate / derive the events/messages from your underlying hardware.

Yes, as part of event processing.

Quote:
FBEndsession(): is this the means to signal to the Framebuffer package that the app wants to quit?

No, it is basically means that GUI is to be shut down (e.g. computer switched off).

Of course, if we run "windowed fb", then equivalent is closing the host window.

BTW, dealing with end-session event is somewhat unfinished bussiness in U++...

Quote:
FBSleep: ideally, this should sleep a fixed granularity of time, say 10ms, but be 'cancelable' or 'expireable' on arrival of new events to process.. if this is not possible, simply Sleep(10)?

Yes.

Quote:
FBIsWaitingEvent: should determine in a nonblocking manner, if there are messages or events to be processed. this is called in advance, prior to FBProcessEvent, which is called if messages/events to process really do exist. if there is no means to determine if events are there, simply return always true?

Well, that would be bad - it has to return false sometimes, as some processing is tied to "empty input queue" condition, e.g. painting or timer.

Quote:
FBProcessEvent: here, *one single* backend message/event per call is dequeued and dispatched to upp understandable messages/events, using some custom translation mechanism..

Yep.

Quote:
but there are more things one needs to implement:

```
bool GetShift()       { uint8* ka = SDL_GetKeyState(NULL); return ka[SDLK_LSHIFT] ||
ka[SDLK_RSHIFT]; }
bool GetCtrl()        { uint8* ka = SDL_GetKeyState(NULL); return ka[SDLK_LCTRL]  ||
ka[SDLK_RCTRL]; }
bool GetAlt()         { uint8* ka = SDL_GetKeyState(NULL); return ka[SDLK_LALT]   ||
ka[SDLK_RALT]; }
```

```
bool GetCapsLock()    { uint8* ka = SDL_GetKeyState(NULL); return ka[SDLK_CAPSLOCK]; }
bool GetMouseLeft()   { return (SDL_GetMouseState(NULL,NULL) &
SDL_BUTTON(SDL_BUTTON_LEFT)); }
bool GetMouseRight()  { return (SDL_GetMouseState(NULL,NULL) &
SDL_BUTTON(SDL_BUTTON_RIGHT)); }
bool GetMouseMiddle() { return (SDL_GetMouseState(NULL,NULL) &
SDL_BUTTON(SDL_BUTTON_MIDDLE)); }
```

Careful here. The values MUST be frozen at the moment of input event. Same for GetMousePos.

Quote:
the Keys.h assignments, for K_* of upp, caution, it uses some special structure, K_ALT and K_ALT_KEY are not the same..

Yep.

Mirek
