
Subject: Re: Rainbow, first iteration

Posted by [kohait00](#) on Tue, 12 Jul 2011 13:46:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

especially FBUpdate and FBFlush, when exactly are they called and what exactly are they supposed to do? i could guess from the names but it's not specific, i'll try to summerize in brief what i have found out so far:

Framebuffer is performing its drawing of the controls to a BufferPainter, whose content can then be bitblitted to the real framebuffer. (what are the other ImageDraw, BackDraw etc.)

Framebuffer expects / calls some functions to help finish that process. it also expects the real backend to generate / derive the events/messages from your underlying hardware.

FBUpdate: reports the area that should directly be repainted / transfered to the underlying hardware framebuffer corresponding area, or should it schedule some kind of writeback..?

FBFlush: should it transfer everything (the entire area) from Ctrl::GetFrameBuffer to the underlying framebuffer section? is this not obsolete and could be done with FBUpdate(EntireSize)? Or does it work as a FBCommit after several FBUpdate calls? in this case, when FBUpdate directly memcpy's to the real framebuffer, no commit is needed, and FBFlush can remain {}

FBEndsession(): is this the means to signal to the Framebuffer package that the app wants to quit? (especially when there is one bare application, means no SDL or sth. there is no other means)? since SDL has the SDL_QUIT, which could map to the bool *quit flag from ProcessEvent.

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)?

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?

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

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)); }
```

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

one does normally NOT need to define the starting point of the application (done in Framebuffer), but can override it (see SDLFb)

```
#define GUI_APP_MAIN \
```
