Subject: Re: AGG and Upp Draw integration...
Posted by mirek on Thu, 13 Apr 2006 12:19:41 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 13 April 2006 01:17quote from AGG...
and just some thoughts...

```
//The image buffers
    // are not displayed directly, they should be copied to or
    // combined somehow with the rbuf_window(). rbuf_window() is
    // the only buffer that can be actually displayed.
    rendering_buffer& rbuf_window()        { return m_rbuf_window; }
    rendering_buffer& rbuf_img(unsigned idx) { return m_rbuf_img[idx]; }

//=======

    void copy_img_to_window(unsigned idx)
    {
        if(idx < max_images && rbuf_img(idx).buf())
        {
            rbuf_window().copy_from(rbuf_img(idx));
        }
    }

    //-----------------------------------------------------------------
    void copy_window_to_img(unsigned idx)
    {
        if(idx < max_images)
        {
            create_img(idx, rbuf_window().width(), rbuf_window().height());
            rbuf_img(idx).copy_from(rbuf_window());
        }
    }

    //-----------------------------------------------------------------
    void copy_img_to_img(unsigned idx_to, unsigned idx_from)
    {
        if(idx_from < max_images &&
           idx_to < max_images &&
           rbuf_img(idx_from).buf())
        {
            create_img(idx_to,
                    rbuf_img(idx_from).width(),
                    rbuf_img(idx_from).height());
            rbuf_img(idx_to).copy_from(rbuf_img(idx_from));
        }
    }
```

If I understand correctly, one simple approach could be:
1. to use some of agg image manipulation functions even outside Ultimate's Draw on one image as
2. agg's so called "rendering buffer" and simply ...draw that image! Just a question of pixel formats...

More difficult would be to bind graphic objects with events...
and 1/3 pixel precision...

Sure, in fact this is what will happen in output to the screen, but this does not solve Drawing/printing...

The scenartio that has to be supported:

Image is drawn at certain size to Drawing, then whole drawing is resized and drawn elsewhere (printer).

Therefore virtualization has to support storing drawing ops into Drawing and also printing somehow.

Mirek