

Rendering CAD/GIS Data on a Map Canvas

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.gui/2006-08/msg00422.html>

- *From:* redefined.horizons@xxxxxxxxxx
 - *Date:* 24 Aug 2006 09:00:33 -0700
-

I am working through the design of a disp[lay for a CAD/GIS program written in Java. I haven't decided for sure if I will use Swing or SWT for the GUI yet.

This program I am working on will have to display a large number of 2D geometric shapes. I'm want to make the rendering process as quick and efficient as possible. I had a couple of questions about how I can do this:

Let's say for the sake of this discussion that the area I will use to display my map, or the canvas, will be 600 by 300 pixels. Would it be possible to render the shapes that would fall within an area 20% larger than this to allow for quicker panning and zooming?

I'm trying to avoid a situation where everytime the user pans the display a minor amount that the whole screen needs to be rendered. In the above situation I would have that 20% buffer to allow some limited space for panning.

Is this method possible? Any ideas or suggestions on how I can learn the techniques I need to implement it?

Also, my program will allow the user to select shapes on the display using various interactions with the mouse, and will subsequently allow them to move or modify those geometries. Is it possible to only render the portion of the canvas that needs to be repainted, instead of repainting the entire canvas everytime a single shape is modified? Would I do this by splitting my display into a grid of different images, or can I just pass the rectangular bounds of the new area that needs to be rendered?

Rendering CAD/GIS Data on a Map Canvas

Is this method of partial rendering possible? Any ideas or suggestions on how I can learn the techniques I need to implement it?

I am interested in learning how this might be done in both Swing and SWT. The way each widget toolkit handles this area of graphics display may be the deciding factor in which is chosen for the program.

Thanks for the help.

Scott Huey

.