NAME

glutPostOverlayRedisplay - marks the overlay of the current window as needing to be redisplayed.

SYNTAX

void glutPostOverlayRedisplay(void);

DESCRIPTION

Mark the overlay of current window as needing to be redisplayed. The next iteration through glutMainLoop, the window's overlay display callback (or simply the display callback if no overlay display callback is registered) will be called to redisplay the window's overlay plane. Multiple calls to glutPostOverlayRedisplay before the next display callback opportunity (or overlay display callback opportunity if one is registered) generate only a single redisplay. glutPostOverlayRedisplay may be called within a window's display or overlay display callback to re-mark that window for redisplay.

Logically, overlay damage notification for a window is treated as a glutPostOverlayRedisplay on the damaged window. Unlike damage reported by the window system, glutPostOverlayRedisplay will not set to true the overlay's damaged status (returned by glutLayerGet(GLUT_OVERLAY_DAMAGED).

EXAMPLE

If you are doing an interactive effect like rubberbanding in the overlay, it is a good idea to structure your rendering to minimize flicker (most overlays are single-buffered). Only clear the overlay if you know that the window has been damaged. Otherwise, try to simply erase what you last drew and redraw it in an updated position. Here is an example overlay display callback used to implement overlay rubberbanding:

```
void
redrawOverlay(void)
{
  static int prevStretchX, prevStretchY;

  if (glutLayerGet(GLUT_OVERLAY_DAMAGED)) {
    /* Damage means we need a full clear. */
    glClear(GL_COLOR_BUFFER_BIT);
  } else {
    /* Undraw last rubber-band. */
    glIndexi(transparent);
    glBegin(GL_LINE_LOOP);
    glVertex2i(anchorX, anchorY);
    glVertex2i(anchorX, prevStretchY);
    glVertex2i(prevStretchX, prevStretchY);
    glVertex2i(prevStretchX, anchorY);
    glEnd();
  }
  glIndexi(red);
  glBegin(GL_LINE_LOOP);
  glVertex2i(anchorX, anchorY);
  glVertex2i(anchorX, stretchY);
  glVertex2i(stretchX, stretchY);
  glVertex2i(stretchX, anchorY);
  glEnd();
  prevStretchX = stretchX;
  prevStretchY = stretchY;
}
```

Notice how glutLayerGet(GLUT_OVERLAY_DAMAGED) is used to determine if a clear needs to take place because of damage; if a clear is unnecessary, it is faster to just draw the last rubberband using the transparent pixel.

When the application is through with the rubberbanding effect, the best way to get ride of the rubberband is to simply hide the overlay by calling glutHideOverlay.

SEE ALSO

glutPostRedisplay, glutEstablishOverlay, glutLayerGet

AUTHOR

Mark J. Kilgard (mjk@sgi.com)