NAME

glutEstablishOverlay - establishes an overlay (if possible) for the current window.

SYNTAX

void glutEstablishOverlay(void);

DESCRIPTION

glutEstablishOverlay establishes an overlay (if possible) for the current window. The requested display mode for the overlay is determined by the initial display mode. glutLayerGet(GLUT_OVERLAY_POSSIBLE) can be called to determine if an overlay is possible for the current window with the current initial display mode. Do not attempt to establish an overlay when one is not possible; GLUT will terminate the program.

If glutEstablishOverlay is called when an overlay already exists, the existing overlay is first removed, and then a new overlay is established. The state of the old overlay's OpenGL context is discarded.

The initial display state of an overlay is shown, however the overlay is only actually shown if the overlay's window is shown.

Implicitly, the window's layer in use changes to the overlay immediately after the overlay is established.

EXAMPLE

Establishing an overlay is a bit involved, but easy once you get the hang of it. Here is an example:

```
int overlaySupport;
int transparent, red, white;

glutInitDisplayMode(GLUT_SINGLE | GLUT_INDEX);
overlaySupport = glutLayerGet(GLUT_OVERLAY_POSSIBLE);
if (overlaySupport) {
  glutEstablishOverlay();
  glutHideOverlay();
  transparent = glutLayerGet(GLUT_TRANSPARENT_INDEX);
  glClearIndex(transparent);
  red = (transparent + 1) % glutGet(GLUT_WINDOW_COLORMAP_SIZE);
  white = (transparent + 2) % glutGet(GLUT_WINDOW_COLORMAP_SIZE);
  glutSetColor(red, 1.0, 0.0, 0.0);  /* Red. */
  glutSetColor(white, 1.0, 1.0, 1.0);  /* White. */
  glutOverlayDisplayFunc(redrawOverlay);
  glutReshapeFunc(reshape);
} else {
  printf("Sorry, no nifty overlay (try an SGI workstation)!0);
}
```

If you setup an overlay and you install a reshape callback, you need to update the viewports and possibly projection matrices of both the normal plane and the overlay. For example, your reshape callback might look like this:

```
void
reshape(int w, int h)
{
  if (overlaySupport) {
    glutUseLayer(GLUT_OVERLAY);
    /* Setup overlay to have X style coordinate system. */
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
```

```
      glLoadIdentity();
      gluOrtho2D(0, w, 0, h);
      glScalef(1, -1, 1);
      glTranslatef(0, -h, 0);
      glMatrixMode(GL_MODELVIEW);
      glutUseLayer(GLUT_NORMAL);
    }
    glViewport(0, 0, w, h);
  }
```

See the glutOverlayDisplayFunc man page for an example showing one way to write your overlay display callback.

X IMPLEMENTATION NOTES

GLUT for X uses the SERVER_OVERLAY_VISUALS convention is used to determine if overlay visuals are available. While the convention allows for opaque overlays (no transparency) and overlays with the transparency specified as a bitmask, GLUT overlay management only provides access to transparent pixel overlays.

Until RGBA overlays are better understood, GLUT only supports color index overlays.

SEE ALSO

glutUseLayer, glutRemoveLayer, glutCreateWindow, glutPostOverlayRedisplay, glutShowOverlay, glutOverlayDisplayFunc

AUTHOR

Mark J. Kilgard (mjk@sgi.com)