HOME TWITTER FACEBOOK RSS ATOM FORUM

Using gluUnProject

Like 8 +4

gluUnProject converts Windows screen coordinates to OpenGL coordinates. This means that you can get the position of your mouse on an OpenGL Window and use this method to find the x, y and z coordinate of where you clicked.

You need to pass several variables to the function, these are:

1. Viewport Origin And Extent
2. The Modelview Matrix
3. The Projection Matrix
4. The Windows Screen Coordinates
5. Variables Where The Output OpenGL Coords Will Be Stored

So how do you get them?

1. Viewport Origin And Extent

We need to grab the current viewport. The information we need is the starting X and Y position of our GL viewport along with the viewport width and height.

Once we get this information using glGetIntegerv(GL_VIEWPORT, viewport), viewport will hold the following information:

viewport[0]=x
viewport[1]=y
viewport[2]=width
viewport[3]=height

?
```
GLint viewport[4];                   // Where The Viewport Values Will Be Stored
glGetIntegerv(GL_VIEWPORT, viewport);           // Retrieves The Viewport Values (X, Y, Width, Height)
```

2. The Modelview Matrix

Once we have the viewport information, we can grab the Modelview information. The Modelview Matrix determines how the vertices of OpenGL primitives are transformed to eye coordinates.

?
```
GLdouble modelview[16];                  // Where The 16 Doubles Of The Modelview Matrix Are To Be Stored
glGetDoublev(GL_MODELVIEW_MATRIX, modelview);       // Retrieve The Modelview Matrix
```

3. The Projection Matrix

After that, we need to get the Projection Matrix. The Projection Matrix transforms vertices in eye coordinates to clip coordinates.

?
```
GLdouble projection[16];                 // Where The 16 Doubles Of The Projection Matrix Are To Be Stored
glGetDoublev(GL_PROJECTION_MATRIX, projection);     // Retrieve The Projection Matrix
```

4. The Windows Screen Coordinates

After we have done all of that, we can grab the Windows screen coordinates. We are interested in the current mouse position.

?
```
POINT mouse;                         // Stores The X And Y Coords For The Current Mouse Position
GetCursorPos(&mouse);                    // Gets The Current Cursor Coordinates (Mouse Coordinates)
ScreenToClient(hWnd, &mouse);

GLfloat winX, winY, winZ;                // Holds Our X, Y and Z Coordinates

winX = (float)mouse.x;                   // Holds The Mouse X Coordinate
winY = (float)mouse.y;                   // Holds The Mouse Y Coordinate
```

Now Windows coordinates start with (0, 0) being at the top left whereas OpenGL coords start at the lower left. To convert to OpenGL coordinates we do the following:

?
```
winY = (float)viewport[3] - winY;            // Subtract The Current Mouse Y Coordinate From The Screen Height.
```

You may have noticed the missing z coordinate, well here is how to get it:

?
```
glReadPixels(winX, winY, 1, 1, GL_DEPTH_COMPONENT, GL_FLOAT, &winZ);
```

5. Variables Where The Output OpenGL Coords Will Be Stored

All that is left to do is calculate our final OpenGL coordinates.

?
```
GLdouble posX, posY, posZ;              // Hold The Final Values
```

Now here is the completed C code that will return the correct OpenGL coordinates if you pass it the mouse coordinates:

?
```
CVector3 GetOGLPos(int x, int y)
{
    GLint viewport[4];
    GLdouble modelview[16];
    GLdouble projection[16];
    GLfloat winX, winY, winZ;
    GLdouble posX, posY, posZ;

    glGetDoublev( GL_MODELVIEW_MATRIX, modelview );
    glGetDoublev( GL_PROJECTION_MATRIX, projection );
    glGetIntegerv( GL_VIEWPORT, viewport );

    winX = (float)x;
    winY = (float)viewport[3] - (float)y;
    glReadPixels( x, int(winY), 1, 1, GL_DEPTH_COMPONENT, GL_FLOAT, &winZ );

    gluUnProject( winX, winY, winZ, modelview, projection, viewport, &posX, &posY, &posZ);

    return CVector3(posX, posY, posZ);
}
```

Also here is the Delphi code for the same function, donated by Sander Koopmans:

?
```
// The Type Declaration Should Be Placed In The Header Of The .pas File
type T3D_Point = array [1..3] of Double;

function GetOGLPos(X, Y: Integer): T3D_Point;
var
    viewport:   array [1..4]  of Integer;
    modelview:  array [1..16] of Double;
    projection: array [1..16] of Double;
    winZ: Single;
begin
    glGetDoublev( GL_MODELVIEW_MATRIX, @modelview );
    glGetDoublev( GL_PROJECTION_MATRIX, @projection );
    glGetIntegerv( GL_VIEWPORT, @viewport );

    // In Delphi A Y Value Of 0 Returns An Unknown Value
    // I Discovered This While I Was Testing A Crosshair
    if( Y = 0 )then Y := 1;

    glReadPixels(   X, -Y, 1, 1, GL_DEPTH_COMPONENT, GL_FLOAT, @winZ );
    gluUnProject(   X, viewport[4]-Y, winZ,
            @modelview, @projection, @viewport,
            Result[1], Result[2], Result[3]);
end;
```

If you are rendering to a window, use the ScreenToClient method (before passing to the method) to convert the mouse co-ordinates given by GetCursorPos to window co-ordinates. Very special thanks to Sander for pointing out and fixing some bugs in the code and the addition of a Delphi version.

Additional info for Delphi users from Sander:

In the Delphi example the fullscreen / window bug isn't fixed, because it will cost a lot of extra code if you don't use the normal Delphi window TForm. If you want to fix this bug your self then Delphi has the exact same command. The command is stored in the class TControl from the VCL library.

* Additional Commenting By NeHe