I've managed to hide the cursor in my windowed OpenGL app using this code:

```c
glutSetCursor(GLUT_CURSOR_NONE);
```

but since my app is a window, the cursor is only hidden while it is over the window. If you move the mouse away from the window, it shows up again, and my glutPassiveMotionFunc can no longer get the mouse co-ordinates.

How can I make it so the mouse is invisible even if it is not over the OpenGL window, allowing me to always get the coordinates of the mouse.

If it makes things clearer, I basically want to implement a mouse-look input system like quake/half-life/insert_fps_name_here. I need to be able to get the coordinates of the mouse every time it is used, so i can see what direction the user is moving the mouse, and how fast.

---

If you're using Win32, ShowCursor will solve your problem. Simply pass TRUE to see it, and FALSE to hide it. However, if you're aiming for a mouse-look system which mimics Quake and Half-Life, you need some other Win32 functions too: GetCursorPos and SetCursorPos.

The problem you'll occasionally run up against if you leave the mouse free, but hidden, is this: when the mouse is at the side of the screen, it doesn't move. Sounds obvious, but you won't get any movement at all, so once the mouse is at the side of the screen, you can't look around any further with it. You can solve this by using GetCursorPos/SetCursorPos. These two functions get and set the mouse's position, relative to the upper-left corner of the Windows desktop. You put the mouse in the centre of the screen at the start of the program, get it's position every frame, then reset back it to the centre of the screen. This way, the mouse never reaches the edge of the screen, and you get flawless movement...

---

Windows 95 - 32 bit extensions and a graphical shell for a 16 bit patch
to an 8 bit operating system originally coded for a 4 bit microprocessor, written by a 2 bit company that can't stand 1 bit of competition.

User Rating: 1111 | Rate This User PM Profile Journal Report this Post to a Moderator | Link

AndreTheGiant  Member since: 8/14/2002  From: Canadia

Posted - 1/19/2004 9:02:54 PM

I am using Win32, but I'm trying to keep my program from relying on windows for anything. Is there any way to mimic the getCursorPos / setCursorPos functions you mentioned using only GLUT? Or maybe another way around that avoids reliance on windows?

If not, what header do i need to include for those 2 functions?

thanks,
Andre.

User Rating: 1020 | Rate This User PM Profile Journal Report this Post to a Moderator | Link

Ostsol  Member since: 12/20/2002  From: Edmonton, Alberta

Posted - 1/19/2004 9:06:35 PM

windows.h, I think. . .

User Rating: 1000 | Rate This User PM Profile Journal Report this Post to a Moderator | Link

Anonymous Poster

Posted - 1/29/2004 11:37:34 PM

In GLUT you can set the mouse position by using

glutWarpPointer(int x, int y);

This will allow you to set the position to the middle of the active screen. In order to find out which direction the mouse is being moved you just subtract the curr x,y (in PassiveMotionFunc) from the mid_x, mid_y, like this:

int dirY = mid_y - y; // If they are moving up dirY will be positive
int dirX = mid_x - x; // If they are moving right, dirX will be negative

This can also be used to get the speed they are are moving the mouse in, dirX/dirY will be greate as they move the mouse faster.

Hope this helps.

User Rating: 950 Report this Post to a Moderator | Link

AndreTheGiant  Member since: 8/14/2002  From: Canadia

Posted - 1/30/2004 3:49:55 PM
Thanks, AP

But the problem is, after I read the current mouse position, and the middle of the window to find out what direction the mouse moved, I move the mouse cursor back to the middle to get ready for next time, and that calls the passiveMouseFunc() again. So I get infinite recursion as soon as I move the mouse, and that's obviously a bad thing.

Ok I found a temporary solution and would like to know if it's the 'standard' way to do it: basically I only call warpPointer to put the cursor back in the center of the window IF it's not already there. This seems to stop the infinite recursion, but I noticed a different problem:

If I move the cursor too fast, the cursor 'pops out' of my window and my app no longer has the mouse focus. I'm guessing this is caused because I move the cursor farther than the width of the window before the passiveMouseMotion() function gets a chance to be called, and therefore it is not called. Hopefully I'm explaining this clearly. How can I prevent my app from losing the focus like this?

Anonymous Poster

Sorry I forgot to add that you need a line in there to just return if mid_x == x and mid_y == y. I have never ran into your other problem. How big is your window? Most windows I have developed (and I am somewhat new at this as well) have been at least 640x480. Are you calling the glutWarpPointer as soon as possible in your motion handler? I have always called it immediately after my test to see if the mouse even moved or not and this seems to have worked so far.

Anonymous Poster

I should also mention I program mainly in Linux (Debian) so the frequency of calls to PassiveMotionFunc may be different in Windows.