

A First Person Shooter with Dual Guns

Using Multiple Optical Air Mouse Devices

Young-Bum Kim , Min-Sub Shim, Chang Geun Song and Yu-Seop Kim *

Department of Computer Engineering, Hallym University
 39 Hallymdaehak-gil, Chuncheon, Gangwon-do, 200-702, Korea
{stylemove, connecting , cgson, yskim01}@hallym.ac.kr

Abstract

We proposed a first person shooter for both hands with multiple mouse devices, which are not used in existing games. Players control their movement and shooting separately using dual guns for both hands. For two-hand usage, we designed a display method for multiple mouse pointers. First, we loaded typical physical driver into the memory and registered our developed multiple mouse device driver instead of the existing MS Windows device driver. We then executed both message loop procedures for Windows mouse event handling and those for multiple mouse event handling. We could display multiple mouse cursor images and utilize both mouse devices separately. After that, a prototype game using both hands was developed for an experimental study.

Key words: Multiple mouse devices, Game, First person Shooter

1. Introduction

First-person shooters (FPS), like Doom and Quake [1], form a genre of video games which is characterized by an on-screen view that simulates the playable characters' perspectives and a focus on the use of range weapons such as guns[2]. Normally, a player uses one hand to shoot a gun and the other hand to manipulate the other movements of the player. However, if two guns can be handled to shoot several targets simultaneously, amusement from the game can be increased. Dual guns have been implemented in some games [3], but they use only one mouse. We tried to solve that point by using multiple mouse devices which can control pointer moves just by moving each mouse. We referred to and modified CPNMouse's [7] open API technology in order to apply them to our FPS. Additionally, we incorporated optical air mouse devices to improve the mobility of our game.

2. A Filter Driver

Multiple mouse technologies, which have been utilized previously in the computer-aided education area [4, 5], are implemented by using a filter driver concept [6]. A filter driver is a kind of intermediate driver. The driver snatches I/O requests transferred to the commercial drivers, like a file system driver and a disk driver, and complements the driver functions or provides opportunities to add new functions. In this paper, a

mouse USB filter driver was developed by using the Windows mouse USB driver to monitor mouse event data transferred to USB ports in order to display mouse pointers on the screen.

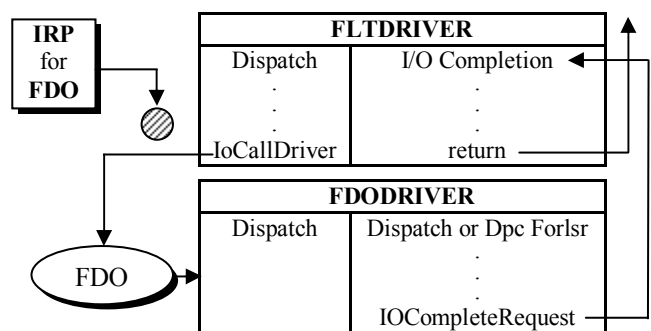


Fig. 1 A Filter Driver

The main difference between the filter driver and other layered driver is the device objects generated by the drivers. The object of the filter driver has no name opposed to the layered drivers which export objects with unique names. The filter drivers operate by attaching one of their device objects to objects on the lower layered drivers. In figure 1, a filter driver (FLTDRIVER) attaches its filter driver object to the object (FDO) from a layered driver (FDO DRIVER). All IRPs (I/O Request Packets) transferred to FDO are automatically routed to the dispatch routine in FLTDRIVER.

3. Multiple Mouse Pointer Display

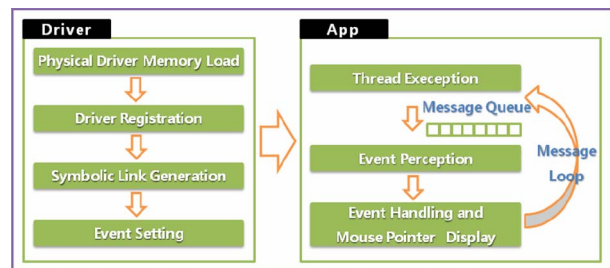


Fig. 2 The process of multiple mouse pointer display

Figure 2 shows the whole process of multiple mouse pointer display. Below the figure each phase is described.

(1) Loading of a physical driver's memory
 In this phase the memory of a computer system's

physical driver is loaded. The physical driver is a driver which manages the connection between the hardware and the computer system.

(2) Device registration

Additional multiple mouse drivers are registered in the system using the memory of the physical driver loaded in the phase (1). The registration includes the mouse drivers' generation and the addition of the mouse drivers into a device stack.

(3) Symbolic link generation

A symbolic link connecting the mouse devices registered in (2) to a user visible device name is generated.

(4) Event setting

This phase sets events that occur through the mouse devices. The events set in this paper are as follows: left mouse button down, left mouse button up, right mouse button down, right mouse button up, up/down mouse move, right/left mouse move, mouse wheel up, mouse wheel down, mouse wheel scroll down, mouse wheel scroll up, etc.

(5) Thread execution

Threads for the mouse devices are executed in this phase and are ready to perceive the events set in (4).

(6) Event perception

In this phase drives applications using the multiple mouse devices are executed and perceive events derived on the applications. In this case, a perception event is one of the events described in phase (4). Event perception is executed whenever the operating system of the computer perceives the mice event of a registered logical driver.

(7) Event handling and mouse pointer display

Using the events perceived in phase (6), the sizes of the full window and the game window are dynamically calculated, and the mouse pointers are displayed by outputting their cursors as images in the game window.

4. A prototype FPS game

A kind of first-person shooter (FPS) game is developed. Targets appear on the screen continually, and players shoot at the targets using dual guns belonging to each hand. Figure 3 shows screen shots of the game in which several targets can be seen on the screen, but players cannot know which targets to shoot at because only the back sides of the targets are shown initially. When any target is turned over a sound is generated to inform the players, who get their points for shooting at a proper target. The goal of this game is to obtain the highest points by striking targets within 3 minutes. The user's input part handles the user input events generated by two mouse devices. Clicking the right button of each mouse changes the guns, which are then fired with left-button clicks.



Fig. 3 Screen shots of the game

5. Conclusion and Future Works

We developed a first-person shooter game for PC using multiple air mouse devices to provide players with the functionality of using dual guns for both hands. Moreover, to improve the reality of the game, we use air optical mouse devices instead of the conventional ones. We implemented a filter driver concept to add a secondary mouse device driver to the existing device driver system. For the future works, we now have a plan to improve our game. For example, we will integrate a direction detection device into the game to make the target screen surround the user. Of course, we did not build the game as good as a commercial game in quality, but we can suggest prototype games utilizing multiple mouse device technology.

Acknowledgement

This work was supported by the New University for Regional Innovation (NURI) project of Korea Research Foundation.

References

1. id Software, "<http://www.idsoftware.com/>", 2007
2. Wikipedia, "http://en.wikipedia.org/wiki/First-person_shooter", 2007
3. ValuSoft, "<http://www.valusoft.com/products/nitrofami-ly.html>" 2007.
4. Pawar, U., J. Pal, R. Gupta, and K. Toyama, "Multiple Mice for Retention Task in Disadvantaged Schools" ACM Conference on Human Factors in Computing Systems(CHI'07), San Jose, California, April 28–May 3, 2007.
5. Pawar, U., J. Pal, and K. Toyama, "Multiple Mice for Computers in Education in Developing Countries," 1st International Conference on ICT and Development, UC Berkeley, May, 2006.
6. Lee, B., "Windows File Systems for the Drive Developers," Cyber Press., 2006 (in Korean).
7. CPNMouse, "<http://cpnmouse.sourceforge.net/>".