Adapting 3D Controllers for Use in Virtual Worlds

Pedro Cardoso[1], Marco Melo[1], Nuno Gomes[2], Aidan Kehoe[3], Leonel Morgado[4]

[1] Universidade de Trás-os-Montes e Alto Douro, Departamento de Engenharias, Vila Real, Portugal

[2] 3Dconnexion, San Jose, CA, USA

[3] Logitech, Fremont, CA, USA

[4] GECAD – Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão,
UTAD – Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
Phone: +351 259350369, Fax: +351 259350356, e-mail: leonelm@utad.pt

***Abstract* — One of the current usage issues regarding 3D virtual world environments, especially for children and elders, has been the complexity of the physical methods by which a user can interact with them. The typical arrangement involves relatively complex combinations of keyboard shortcuts with mouse movement and clicks. 3D controllers provide a possible answer to this problem. We present a user-configurable piece of software under development, which will enable the use of 3D controllers from 3Dconnexion within various 3D virtual world environments.**

I. Introduction

With the appearance of the 3D virtual world environments, people were given the possibility to travel into different worlds and perform different types of journeys without leaving their homes, just using their computers. Virtual worlds or virtual-world systems like Second Life [3], Active Worlds [4], and Open Croquet [5] are being used around the world by millions of people, through the Internet [6]. To move around in these virtual worlds, users must have the ability to use a keyboard and a mouse, often in concert, something which can be difficult for some users – especially children, elders, or people with physical disabilities, who may lack the ability to accurately coordinate keyboard and mouse actions. The keyboard actions may themselves require a significant level of two-handed coordination. For instance, if a Second Life user wants to perform a strafe movement (moving sideways while looking forward), it is typically necessary to make a combination such as *SHIFT + LEFT ARROW*.

We are developing a software application, whose overall settings are presented here, which will allow users to perform mappings between keyboard and mouse events and actions of 3Dconnexion's 3D controllers (described in the next section), which can reduce the level of complexity of movements, and allow the user to navigate the virtual world with the use of just one hand.

For this development, we are using the SpacePilot and SpaceNavigator controllers [7], which have the capability of performing 3D rotations and 3D translations and possess various programmable buttons. With these 3D controllers, a user sees diminished the amount of coordination needed: the user can navigate using just one hand, by pushing, pulling, or pressing the controllers' cap.

We would like to thank 3Dconnexion for providing the controllers and technical background supporting this development effort.

II. 3Dconnexion controllers

A. The controller cap

3Dconnexion's 3D controllers where designed to eliminate the tedious changeover between mouse and keyboard when performing 3D actions, focused on the needs of engineers, designers, and architects. With a simple touch of the device, the user can pan, zoom and rotate in 2D/3D, in one fluid motion, simplifying navigation. To achieve these using mouse and keyboard combinations requires separate steps, clicks and interruptions, to perform all maneuvers. 3D navigation devices thus are likely to present an advantage over mouse and keyboard combinations, regarding the ease of performing intricate adjustments to camera views and models.

The key element of these 3D navigation devices is the **controller cap**. Pressure-sensing technology allows the cap to be manipulated in various ways, as shown in Figures 1 & 2.

Figure 1: controller cap rotations

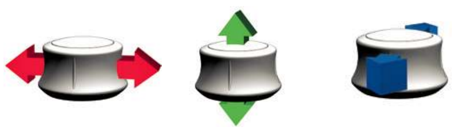

Figure 2: controller cap translations

By pushing, pulling, twisting or tilting the cap a fraction of an inch, one can simultaneously pan, zoom, and rotate in 3D. It is also possible to increase pressure to produce faster movements, or decrease pressure, for intricate adjustments.

B. SpacePilot controller

The SpacePilot (Figure 3) has a central controller cap amidst a large, steady base, and 10 programmable speed keys, reachable without having to take the hand off the controller. This helps minimize hand movements, supporting an increase in 3D control coordination within 3D-world environments.

Figure 3: SpacePilot controller from 3Dconnexion

C. B. SpaceTraveler controller

The SpaceTraveler (Figure 4) is a smaller controller. It features the same technology (it is also possible to push, pull, twist, and tilt the cap of the SpaceTraveler) and provides the same 3D navigation of the SpacePilot controller, but doesn't have such a large base, and only 8 programmable buttons, located around its base.

Figure 4: SpaceTraveler controller from 3Dconnexion

III. Interfacing application

A. Overall description

3Dconnexion creates and provides its own drivers for the controllers, and builds software libraries (DLLs), which directly interact with specific applications. The company also includes with its drivers several COM+ services for developers to use when developing their applications. Our aim is to use these elements to create an application that will act as an interface between the device and the existing interfaces of 3D virtual worlds. A first reason for this approach was the lack of open access to the code of client applications of 3D virtual-world environments: while Second Life and Open Croquet clients are distributed with open source, many others aren't. However, in retrospect, this approach has the added benefit of being readily configurable, in an independent manner of the code of the client application. This is a significant advantage, because client applications are updated quite frequently, but the methods for user interaction through the mouse and keyboard seldom change. And only in the rare event of such a change will the user have to configure our interfacing application again.

The interfacing application is being programmed in C#, and in its final form it will allow the user to specify mappings between movements of the controller cap and keyboard or mouse events. These will be stored as **client application profiles**, which the user can then enable to use the 3D controllers for navigating within specific virtual-world environments. The interfacing application will have the option for saving and loading client application profiles for specific programs. This way, the configuration of mappings doesn't have to be performed every time.

When the user is finished making all choices in the application, he/she can minimize it to Windows' System Tray (the small taskbar section near the clock display), running unobtrusively along with the controller drivers originally installed with the device.

The application interface itself will be divided into several sections (Figure 5). An **Axes Configuration** tab will contain two types of binding for cap movements: keyboard events and mouse events; a **Button Configurations** tab will also contain two types of binding, but for the device's buttons. This way, the users will have the ability to choose if they want to use the controller's cap to control the virtual world actions or the controller's buttons, or even both at the same time.

So for instance, an action such as *ALT + Left Mouse down* (*e.g.*, in Second Life, this indicates that the mouse controls the camera-panning around the clicked element) can be assigned to a button, for on/off use;

and an action such as *SHIFT + Left Arrow* (*e.g.*, in Second Life, strafing to the left) can be assigned to a cap movement such as translation to the left, for a more interactive response.

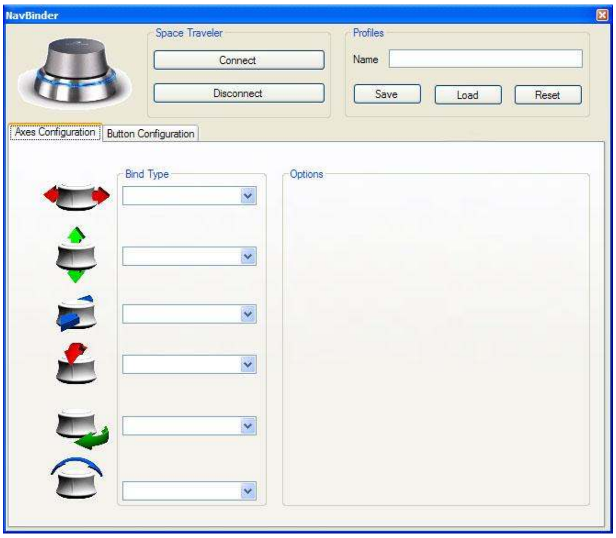

Figure 5. Interface prototype

B. Configuration of axes and buttons

As mentioned above, the user will have the option to choose which type of mappings (bind types) he/she will want to assign to the six possible combinations of axes available in both the SpacePilot and the SpaceTraveler. In Second Life, for instance, the walking movements of the user's avatar involve keyboard actions. So, if the user wants to use the controller cap translation axes to specify walking movements, he/she must choose "Keyboard" for the bind type and then in the option panel choose the category type of keys. The final application will have three categories available: normal input characters (including alphanumeric characters and punctuation marks), functions keys (F1 to F12), and control keys (CTRL, SHIFT, ALT, etc.). For Second Life, the user will choose the arrow keys and may elect to assign them to the translation axes: left-right translation for left-right arrow keys, front-back translation for up-down arrow keys, and possible up-down translation for page down-page up keys, which in Second Life control avatar's jumping and flying movements.

He/she may then decide to use the rotation axes of the 3D controller to perform observe the surrounding environment without actually moving. In Second Life, such actions are performed with the mouse. To achieve this, the user would simply choose "Mouse" in the bind type. The application will also have the option for the user to perform sensibility adjustments, both for keyboard and for mouse bindings.

The process of button configuration under development is identical to the process of axes configuration. The user will be specifying bindings for controller button events instead of controller cap movements. One should note that the user will be able to perform simultaneously both buttons actions and axes movements, if so desired, and in this way produce complex combinations of keyboard and mouse events.

C. Sample control configurations for Second Life

While not everyone will have identical preferences or personal accessibility requisites, we believe the following sample list of bindings may help clarify the scope of the interfacing application.

Move Forward = Push controller cap forward
(translation axes). Bind type = Keyboard.
Key options = Control keys (*up arrow*).

Move Backwards = Pull controller cap backwards
(translation axes). Bind type = Keyboard.
Key options = Control keys (*down arrow*).

Strafe Left = Push controller cap left
(translation axes). Bind type = Keyboard.
Key options = Control keys (*shift+left arrow*).

Strafe Right = Push controller cap right
(translation axes). Bind type = Keyboard.
Key options = Control keys (*shift+right arrow*).

Rotate Left = Rotate controller cap left
(rotation axes). Bind type = Keyboard.
Key options = Control keys (*left arrow*).

Rotate Right = Rotate controller cap right
(rotation axes). Bind type = Keyboard.
Key options = Control keys (*right arrow*).

Fly up = Pull controller cap up
(translation axes). Bind type = Keyboard.
Key options = Control keys (*page up*).

Fly down/Land = Push controller cap down
(translation axes). Bind type = Keyboard.
Key options = Control keys (*page down*).

G. Symbols and Acronyms

Use of symbols and units of the International System (SI) is recommended. All acronyms should be defined at the time of their first occurrence.

IV. Final Thoughts

By creating this application, our hope is that people who face difficulties using keyboard combinations or combined keyboard and mouse action can better interact with 3D virtual-world environments, and better benefit from their use.

REFERENCES

[1] 3Dconnexion, *SpaceTraveler™ – The portable solution*. Web page retrieved from http://www.3dconnexion.com/products/3a1c.php on June 30th, 2007.

[2] 3Dconnexion, *SpacePilot™ – The intelligent solution*, Web page retrieved from http://www.3dconnexion.com/products/3a4b.php on June 30th, 2007.

[3] Linden Research, Second Life®, Web page retrieved from http://www.secondlife.com/ on June 30th, 2007.

[4] Activeworlds, *activeworlds*, Web page retrieved from http://www.activeworlds.com/ on June 30th, 2007.

[5] The Croquet™ Consortium, *The Croquet™ Consortium*, Web page retrieved on June 30th, 2007, from http://www.opencroquet.org/

[6] B. S. Woodcock, "An Analysis of MMOG Subscription Growth – Version 21.0", in the Web site *MMOGCHART.COM*. June 29th, 2006. Retrieved from http://www.mmogchart.com on May 31st, 2007.

[7] 3Dconnexion, *3Dconnexion*, Web page retrieved from http://www.3dconnexion.com/ on June 30th, 2007.