ChairIO – The Chair-Based Interface

Steffi Beckhaus, Kristopher J. Blom, Matthias Haringer

1 Introduction

Chairs can be found everywhere, pervasively embedded into our daily life and barely noticed as tools with a function. Normally, a chair is just a well designed and robust device that supports sitting. From a static chair, the design has advanced to let us turn and roll (office chairs), to let us lean back and put our feet up (leisure chairs), and to help us in daily life (medical and supportive chairs). The chair's possibilities of shifting, tilting, rotating, rocking, and bouncing allow us to intuitively, often unconsciously, control the chair to support us in our work and leisure life. We felt that the same features could be used to control a computer, thus building a novel human computer interface, based on a chair.

The ChairIO is a chair-based computer interface developed in the interactive media / virtual environments (im/ve) group at the University of Hamburg. The motivation for the development of the ChairIO came from one of the group's interests, the search for alternative interaction devices and techniques. Of particular interest is the search for intuitive methods to control movement in immersive virtual environments. Experience, using the standard hand-coupled devices, had shown some of the disadvantages and deficiencies of those methods. A chair, especially one which allows movement in several directions, seemed like a potential candidate to support 3D motion, both directional and rotational. After some research into this idea, a new device and method was introduced in [1]: the chair-based interface to provide intuitive and hands-free navigational control of a virtual environment. The ChairIO's usage in games is a natural extension, providing a dynamic and intuitive interface for controlling the game and introducing new interaction metaphors in games [2]. Beyond those applications, operating a standard desktop environment is possible, giving the office chair extra meaning.

The ChairIO work to date has been based on a commercially available, innovative stool-type chair, the Swopper™. The Swopper is a product from aeris-Impulsmöbel GmbH [3], designed for ergonomics in an office environment. It has a number of unique characteristics that make it well suited for usage as an input/output device. Other types of chairs may also be suited to the ChairIO concept; They need only be able to either have sensors integrated into them, be extended to incorporate sensors, or have their motion tracked by other means.

Figure 1. The ChairIO used as gaming input device. Here, the Swopper™ from aeris-Impulsmöbel GmbH is used as the basis for the ChairIO. User playing a First-Person-Shooter with the ChairIO and gun interface (left). User playing a Jump'n'Run Game (right).

Chairs have been proposed and used as a computer input device in a few research projects. The first interface to control motion through a computer based scenario with a chair was in the "Virtual Museum" art installation, by Jeffrey Shaw in 1991 [4]. He used a chair to control the direction and rotation of view by tilting and rotating the chair, respectively. The SensorChair was developed by Paradiso et al. to perform music [5]. The chair was fixed in place, but featured capacitors that where influenced by the position of the feet and hand. Cohen proposed a chair to control internet applications [6]. The Sensing Chair by Tan et al. featured a pressure sensing mat to read the pressure distribution on the seat, thereby allowing detection of shifts in the user's weight [7]. A cushion based interface, with custom electronics built into the cushion, was published by Holleis et al. in [8].

Compared to the described works, the ChairIO using the Swopper has the advantage of putting the user in a position to operate the chair in full control of their movements. Supported by feet and chair, and using additional compensational movements of the upper body, using it is more of an intuitive, comfortable, full body experience than a deliberate, conscious operation. Through its stool-like construction, using a suspension system and the provided Degrees of Freedom (DOF's), two horizontal, one vertical, and one rotational, it is highly intuitive, very precise, and dynamic to operate, plus joyful to use.

The ChairIO has been well tested. More than 500 people used the ChairIO at the Virtual Reality 2005 conference in Bonn, at ars electronica festival 2006 in Linz, and in our lab. A version of the ChairIO was part of an exhibit at the International Automobile Salon 2006 in Genf. Several studies and tests investigated the ChairIO's features. They have tested the use of the chair, mappings of motion to behavior in the computer, and extended the ChairIO to include additional input and output devices. All tests and user responses so far have confirmed the intuitive use at first encounter and resulted in sometimes lengthy use of otherwise soon boring games. Figure 1 shows examples of the ChairIO for playing First-Person-Shooter and Jump'n'Run games. Videos, demonstrating the dynamic nature of the ChairIO and its use, can be found on the ChairIO website [9].

This chapter presents the ChairIO's potential, based on research and application projects, its use within several application domains, and discusses technical implications and practical solutions. The next section introduces the ChairIO as an input device, describing the features of the Swopper based version. Section 3 addresses the general application of the ChairIO for controlling motion in 3D environments. Section 4 describes controlling computer games with the ChairIO, focusing particularly on 2D and 3D games. The ChairIO's application as a desktop input device is presented in Section 5. Two studies of the ChairIO's preformance that were performed are presented in Section 6, which also reflects on the ChairIO's usage. Several possible technical implementations for sensing the underlying chair's movements are discussed in Section 7. The complimentary software side of the implementation is described in Section 8. Finally, Section 9 closes the chapter with some future directions for research.

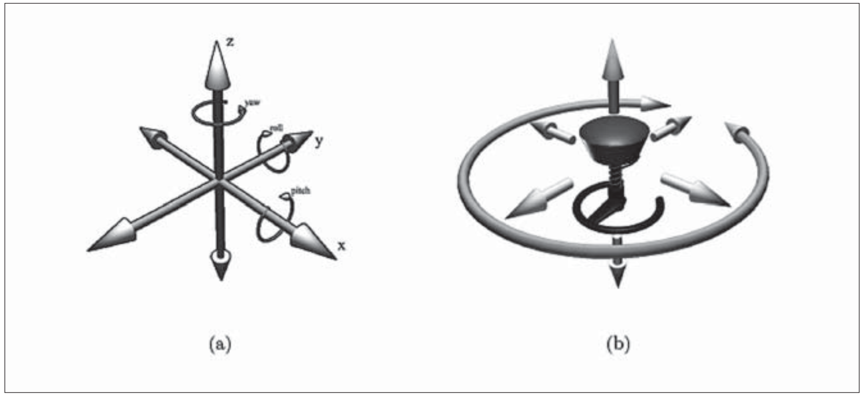


Fig. 2. Degrees of freedom (DOFs): (a) General 3D environment's DOFs (b) TheChairIO's DOFs using the Swopper as underlying chair.

2 The ChairIO as Input Device

To take advantage of a chair's movements for controlling parameters in a computer, three requirements have to be met: a) the motion of the chair can be analyzed by means of sensors and that data is made available to the computer, b) the users are in a position that allows them to have control over the chair's motion, and c) the user is able to use the interface for continued lengths of time. The human body is highly capable in a seated position, particularly when using a highly articulate chair. In a seated position, the user's feet are generally on the ground, providing a means for exerting forces on the chair and stability to movements. Foot contact with the floor also provides stability for lengthy use of the chair, without overly constraining the movement of the hips. This is particularly important for chairs with multiple DOFs. These chairs rely on the fact that the human body is constructed such that the hips can be moved fairly drastically, i.e. shifted or rotated, while maintaining a specific position of the head, e.g. facing a computer screen.

In order to input the user's movement, into the computer with the chair as mediator, sensing systems must detect movements along any degree of freedom of the chair. This is described on hand of the Swopper chair, shown in Figure 1 and in the diagram of Figure 2(b). The Swopper's base supports the single legged stool at a single, centered point. The seat of the chair is connected to the base via a complex linkage. The connection at the base allows the seat to pivot freely in all directions, to a maximal angle of about 30°. The pivot's flexibility is adjustable, making the tilting movement easier or harder. Above the pivot joint is a suspension system, composed of a shock and coil spring. The spring stiffness is adjustable to account for users of different weight. The shock length is also adjustable, controlling the height of the seat at rest. The suspension system allows the Swopper seat to move up and down approximately 20cm. The piston of the shock forms the connection to the seat and allows the seat to rotate freely on the linkage.

The ChairIO can be moved and rotated along the DOFs shown in Figure 2(b). The chair allows control on 4 axes: tilt left/right, tilt/front/back, move up/down and rotate left/right. The two tilt axes can be either considered as rotational or translational axes. The movement of the seat on the linkage forms a rotational DOF (yaw in Figure 2(a)). The up/down axis is slightly limited in its usability, as changing the height requires placing more or less weight on the chair. Short movements, such as bouncing, are performed easily. However, even small up/down shifts are physically difficult to produce for more than a few seconds. Alternatively, the up/down axis can be used as a discrete, button-like input.

3 Controlling Movement in 3D Environments

In this and the following sections, the application potential of the ChairIO concept will be explored. The first area that was investigated, and one with a very strong potential, is that of controlling movement in 3D environments. Whether this is the context of Virtual Environments or that of gaming, the ChairIO's affordances, movements, and intuitive operation make it highly applicable for various movement metaphors. In this section, the general use of the ChairIO for controlling 3D motion is introduced. Two popular movement metaphors are then explored independent of application specifics. Those are 2.5D "ground following" and 3D flight.

3.1 Movement in 3D Landscapes

Control of movement in 3D environments is a foundational aspect of both Virtual Environments and many modern computer games. While movement, as a basic interaction task, would seem trivial, movement in a 3D world is more complicated than it appears. It remains an area of continuing research and development. Movement in 3D space allows a maximum of six degrees of freedom (DOF) each representing three translatory and three rotatory axes as shown in Figure 2(a). The translation axes are back to front (y (notation used in most input systems)), left to right (x), and bottom to top (z). The three rotation axes are rotation around z, which is also called heading or yaw (look left/look right), rotation around x, which is also called pitch (look up/look down), and rotation around y, which is called roll.

Most interaction methods that include all 6 DOFs are difficult to use. Among the greater problems with them, even when conducted well, is that they tend to be disorienting or lead to "simulator sickness." To combat this, applications commonly place restrictions on the movement mechanisms. One of the common movement methods is 3D movement restricted to the ground, also commonly called ground following, which is described in the following section.

The standard ChairIO mapping does not allow roll and pitch rotations. Roll orientations are not important for usual ground following applications. Additionally, they add to the user's disorientation, so avoiding roll rotations completely is an acceptable degradation. Pitch rotation, i.e. looking up and down, is more diffcult to ignore. In standard scenarios, a vertical view direction parallel to the floor may be suffcient. In front of high structures or in front of deep crevasse, it is helpful to be able to look up or down. Modern games often implement automated camera motions, for example Tomb Raider Legend, which moves the camera to a position that the application determines to be an interesting viewing angle. Those methods compensate the lack of input and simplify the interaction. The automated motions have to be implemented in the application. When the view is needed for aiming like it is done in First-Person-Shooters, the missing pitch angle is very limiting. Solutions to manually control this additional degree of freedom are presented in Section 4.

3.2 Ground Following – Movement in 2.5D

In virtual environments that are representations of environments similar to our physical environment, a classic restriction to the users movements is formed on the basis of gravity and called ground following. The user is affected by gravity

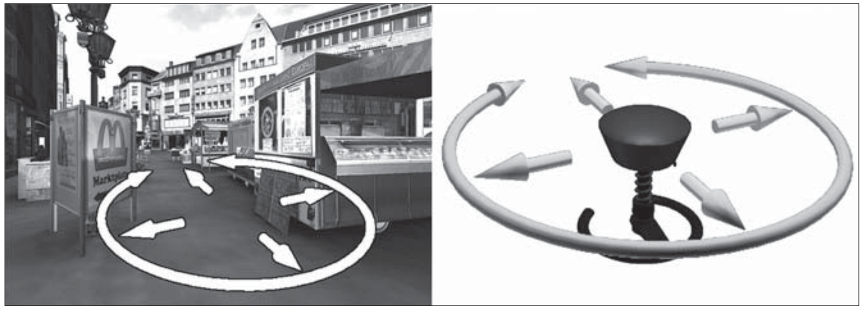


Figure 3. Diagram of 2.5D mapping and navigation of the ChairIO.

and is, therefore, always in contact with the ground plane. Since one moves through the virtual environment in essentially a 2D manner with the z axis only nominally changing, this method is sometimes referred to as 2.5D navigation of a 3D environment [1].

This method of movement maps well on the ChairIO. The 2D movement is controlled by the translatory axes of the seat and rotation of the viewpoint is performed by rotation of the seat, which also can be seen in Figure 3. This mapping was used in our initial study with the ChairIO, presented in [1], and proved to be highly intuitive. The environment of our initial study was that of a German market place (featuring the city of Bonn), as shown in Figures 3 and 11. In Section 6, we describe the application in more detail and discuss the results of our testing in that environment.

3.3 Flying – Movement in 3D

A full 3D translational movement with heading rotation can be implemented with the ChairIO, i.e. flying without pitch and roll rotations. In order to achieve this, the additional DOFs of the ChairIO, the upwards/downwards direction in Figure 2(b), can be used.

A number of different methods have been explored. The most obvious metaphor is that of a flight metaphor. How exactly to control this, however, is not so clear. Using a direct metaphor, where the controls are very similar to those of a real plane is possible. In this metaphor, the tilting of the seat is used as pitch and roll. This seems to be fairly natural to those who are accustomed to how flying works and how the physical movement directly maps to the steering input for flying. However, the direct flight metaphor is for many people unintuitive. An alternative is to use the rotation as a heading input in the way that it was in 2.5D movement. The tilt represents then the movement in the according direction and up/down on the chair leads to gaining or loosing height. This mapping is more intuitive for most users.

A combined 3D and 2.5D method proved successful, which performed a mode switch between the two modes. In the 2.5D mode, the user moved as described in Section 3.2. When, however, the user desired to switch to the 3D flying modus, they simply “bounced” on the seat to break free of the ground. This leads to a flight modus that is similar to that presented in the last paragraph. To return to the 2.5D modus from the flight modus, the user simply lands, the modus changing as the user comes close to the ground again.

4 Gaming

The first comments many users enthusiastically made, was a desire to use the ChairIO in games, especially in First-Person-Shooters. The ChairIO has a rather playful component to it when compared to moving a mouse. This playful nature of the ChairIO additionally increases the desire to play. In this section, we discuss the gaming applications of the ChairIO. The discussion is divided into 2D, 3D, and other gaming, as the areas employ different types of metaphors for control. The discussed metaphors may of course be used in any kind of game environment, if suitable. The section on 3D games builds on the previously explored movement in 3D in the last section. It includes the use of additional tools for gaming, as many games rely heavily on the existence of multiple buttons for game control.

4.1 2D Games

One area of potential for the ChairIO is the classic gaming area of 2D games. Many of those games have a digital joystick input and require one or two buttons, making the mapping to the ChairIO very natural and easy. However, the games take on

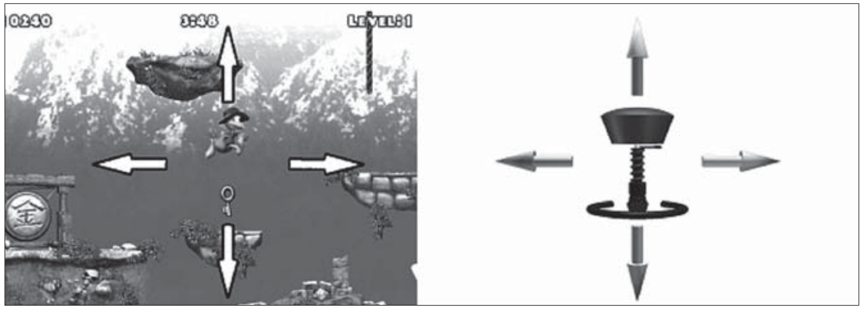


Figure 4. The ChairIO's bounce motion is mapped to the jump action of the Jump'n'RunPlatform game.

a new character when played with the ChairIO. Beyond the new fun factor, we believe the user's sense of agency is increased, as their physical movement is then reflected in the player's character. See Laurel [10] for good insight into agency.

The simplest group of games to adapt are those that require only a joystick for translation. Classic games, such as PacMan and Frogger, receive a new and enjoyable touch, when using the ChairIO. The games also have a new challenge, as the physical movement of the user's entire movement is naturally a greater (and lengthier) movement than moving the wrist.

The Jump'n'Run games of the early computer gaming era combine with the ChairIO input extremely well. These games naturally require the steering of the avatar direction as in PacMan et al. However, they incorporate a single button, or sometimes additional aspects, which controls the characters ability to jump. Classics of this genre include Donkey Kong and the Mario Brothers series. With the ChairIO, these games combine the direct physical connection of movement with the enjoyment of "bouncing" on the seat (see Figure 1). This is mapped in a way that, when the users want to jump, they simply bounce down. This was realized in our experimentation by making the height change into an digital "button" press, using a simple threshold function.

In the following, two 2D games with different control mappings are presented. Moorhuhn - Schatzjäger2 (Phenomedia) is a Jump'n'Run, platform style game. The basic movements are run left, run right, and jump. On a PC, the left and right movements are usually mapped to the arrow buttons of the keyboard or the D-pad of a gamepad. The jump action is mapped on a button that can be reached with the other hand, as the gameplay requires jumping to be used while running. The ChairIO mapping for Schatzjäger2 is straight foreward; Run left and run right is mapped to the tilt left/right motions of the ChairIO and the jump action is mapped on to the downward bounce motion. This can be seen in Figure 4. This mapping is highly intuitive; Playing such platform games, many people have been watched making similar left to right motions with their upper body when playing with gamepads. The direct physical movement, especially the bouncing, was reported to give additional gaming depth and better involvement in the game.

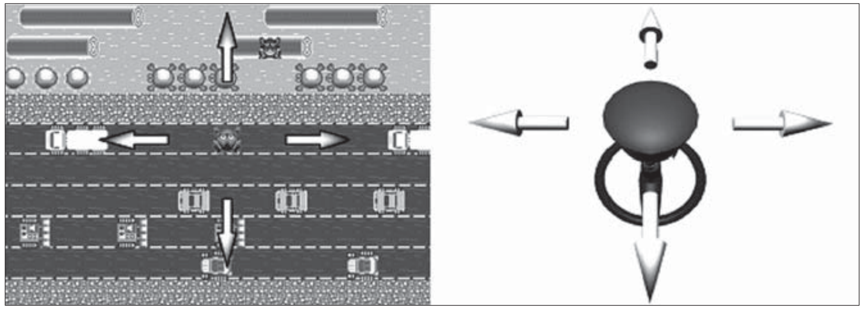


Figure 5. 2D ChairIO mapping of a remake of the classic arcade game Frogger.

Using the ChairIO in classic joystick based arcade games is also an obvious application. As an example from this genre, a Frogger remake is controlled with the ChairIO, as shown in Figure 5. The games forward, backward, left and right motions are mapped onto the according ChairIO tilt motions as with a joystick. Effective gameplay requires precise step by step control; With the ChairIO this requires quick motions in the intended direction coupled with a quick motion back to the center position. This behavior requires some practice for precise movement in the game. The extra bit of concentration that is needed to achieve the necessary body control is an additional component of enjoyment according to people who played with it.

4.2 3D Games

Controlling movement in 3D worlds is possibly the application area that has shown the most potential and the most enthusiasm from users, probably due

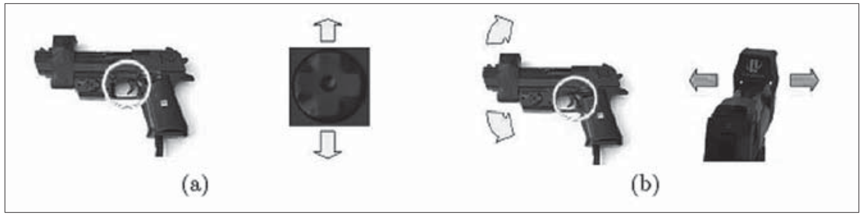


Figure 6. First-Person-Shooter extensions to the ChairIO: Viewpoint control possibilities with an additional device: (a) Controlling the up/down View-Direction with the Joy-Pad - ChairIO mapping as in Figure 3 (b) Controlling the up/down, left/right View-Direction with the Gun - ChairIO mapping as in Figure 4.

to the complex, but highly intuitive nature of performing the task with the ChairIO. This encompasses many potential usages in 3D worlds, each of which has particular advantages for using the ChairIO. Section 3 described the typical mappings used for movement in 3D environments. In this section, the ChairIO's usage in 3D games is explored, particularly the factors beyond movement.

The first 3D gaming environment explored was that of a First-Person-Shooter (FPS) or Ego-shooter, in our case Unreal Tournament™ 2004 (Epic Games) (presented in [2]). FPSs have a number of additional required and possible interactions beyond motion control. Normal usage of an FPS requires numerous keys of the keyboard and the mouse, including buttons and the scroll wheel. In a FPS, the main requirement, beyond running and looking around (with speeds that are exceedingly fast), is firing various weapons.

Different mappings and interaction methods were explored with the FPS. The navigation was performed using the ChairIO plus using an additional interactive gun device both for game control, firing, and for supporting rotational movement. We equipped the gun with an orientation sensor to detect the gun's movement. Figure 1(a) shows a user playing with this interface, while Figure 6 shows the gun interface in detail. In [2], various different mappings are compared and tested in a user study. Two of the mappings can be seen in Figures 6(a) and 6(b). In all mappings, the gun's trigger is used for shooting. Problematic is the look up/down axis which is needed for aiming. Using the gun joypad for up/down (Figure 6(a)) breaks the intuition of the other interfaces. Using the ChairIO only for movement and the gun for viewing (Figure 6(b)) is better for aiming, but does not use the full navigation potential of the chair. The best solution, therefore, would be to decouple aiming and view orientation, which is not supported by most FPS. More details on mappings and results can be found in the according paper [2].

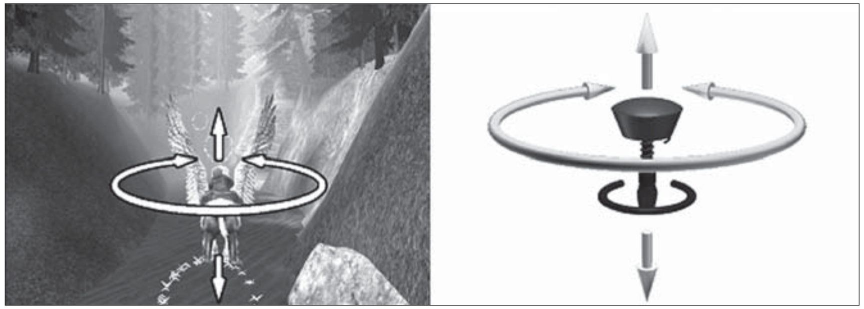


Figure 7. Flying a Hippogriff through ring targets. The downward seat bounce maps to a beat of the Hippogriff's wings.

Many current 3D games use a mouselook, plus foreward/backward/sidestep left/sidestep right, keyboard mapping. For those games, the 2.5D control mapping used for the virtual environment or for the navigation in Unreal Tournament™ is appropriate. From the vast number of 3D games with different control mappings, a small selection has been evaluated and are presented here. The Harry Potter and the Prisoner of Azkaban™ (EA Games) adventure game uses the standard first person control mapping as above for most of the game. It, however, has special controls for a few mini-games. For example, in the Hippogriff minigame, the player flies a Hippogriff through a series of targets. Targets are rings at different heights and orientations, included in a 3D landscape. Normally, the Hippogriff is steered left and steered right through standard controls. The control of height and speed is performed by triggering the Hippogriff to flap its wings. To descend and slow down one just lets the Hippogriff glide, without using controls. The challenge of this control style lies in timing the subsequent wing flaps appropriately and rhythmically to achieve the correct height for flying through the rings. Using the ChairIO, the trigger of the wing flap has been mapped onto the downward bounce motion of the seat and the left and right steering onto the left and right rotation of the chair, as shown in Figure 7. The downward bounce is mirroring effectively the wing flap with the own seat motion and gives an authentic feeling. Users of this setup found this easy to grasp and enjoyed the mirrored motion of the Hippogriff with their own physical movement.

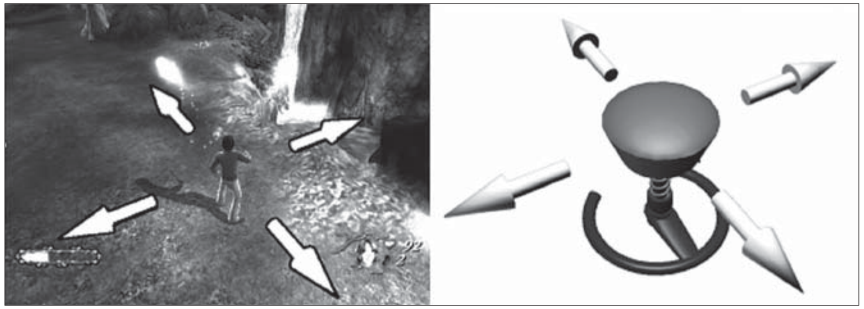


Figure 8. Harry Potter™ walk mapping. All directions inbetween the arrows are also possible. Bounce down maps to throwing a curse on an enemy.

In Harry Potter and the Goblet of Fire™ (EA Games), the characters move in the direction of the arrow keys, independent of the current orientation. Pressing an arrow key turns the character in that direction and moves him in this direction. Keyboard input is limited to 45° directions of travel (one arrow or two arrows simultaneously pressed). Using the ChairIO's analog output, one can move freely in the full 360° direction range, as can be seen in Figure 8. Interaction in the game is in the form of casting spells, which are mapped to the downward bounce motion. Moving through the landscape, pursuing and hitting enemy monsters with this active gesture is a quite joyful and satisfactory experience. Accessing the other spells and actions needed in the game can be accomplished by incorporating a small button device in a similar manner, as the gun in the First-Person-Shooter example above.

The racing game genre has very different control methods and is the one area where special controllers are often used. We tested steering Need for Speed™ Most Wanted (EA Games) with the ChairIO. The most important actions in a driving game are steering left and right, accelerating, and braking. Steering left and right was mapped to the ChairIO left/right rotate motion, as in Figure 9. Tilt left and right was also tested, but the rotation appeared more intuitive, probably because of the similarity to a steering wheel motion. The acceleration and braking was mapped to the forward and backward motion of the ChairIO, with even switching into reverse happening in "automatic" mode. Most users were able to quickly adapt to the interface. The largest issue users had with the ChairIO control was oversteer, which resulted in fish-tailing for some players. With practice this effect was largely overcome.

Figure 9. Need for Speed™ drive mapping. Forward maps to accelerate; Backward to brake.

A yet unexplored potential for the ChairIO is as (external) camera control for various games. In different game genres, where a third person perspective is used, the user is able to position the camera. Occasionally, this is an important aspect for the game itself, as in Blade Runner™ (Virgin Interactive), and can also be of importance in strategy games.

Many more mappings of the ChairIO motion to parameter control in a game are possible. With the 2D planar input and the potentially 4 DOF input of the chair, it is possible to make specific gestures with the lower body and map those onto functions in a game. This is analogous to the 2D gesture input in the game Black and White (EA Games). The potential here goes further than just gaming. Connecting a game like application with back strengthening body gestures may result in new kinds of "wellness-games" that may prove useful in everyday work life.

5 Replacing the Mouse – Desktop Input

A more general application area for the ChairIO concept is as a replacement for the ubiquitous mouse as a general input device both in the desktop environment and for games. The mouse provides a direct movement metaphor for moving a cursor on the 2D plane of the screen. In combination with 1 to 3 buttons integrated into the mouse, several modes can be operated in standard usage: selection of the underlying object (left button click), selection of a context menu of the underlying object (right button click), movement and release of the underlying object (Drag

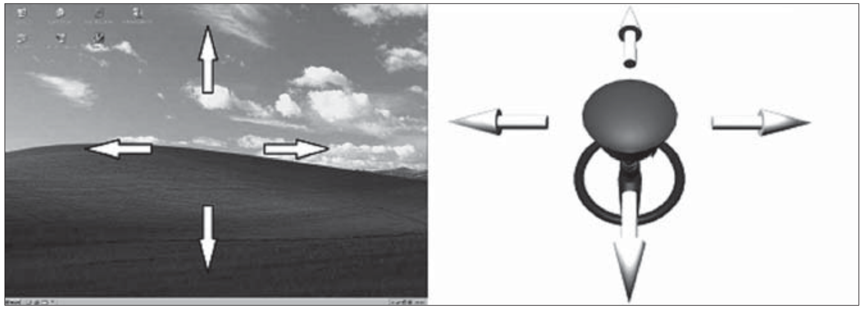


Figure 10. Diagram of the desktop control mapping of the ChairIO. Click actions are mapped to external buttons.

and Drop). Tasks like sketching a path, modifying numeric values, and typing text are supported by mouse control via specific mappings.

The ChairIO intuitively provides a mapping for the positional tasks usually performed by the mouse pointer movement on the screen. Tilting the chair in forwards/backwards direction or to the sides maps to up/down or left/right cursor movement as in Figure 10. With a suitable adjustment to the mapping function between chair movement and mouse movement, selection of 2D screen coordinates is possible. However, all other standard "mouse" functions require buttons to be clicked, double-clicked, or clicked-held-released.

Various methods of coupling such inputs with the ChairIO are possible. Simply, using a keyboard or mouse click is one method. An alternative straight forward solution to this is to use a foot control, while keeping the hands free. If buttons are positioned in a comfortable place near the resting place of the feet, a foot can control the same functions as mouse buttons normally provide.

Active research at im/ve is testing the addition of two foot controls next to the chair's base to expand the ChairIO's capabilities, in order to make to a full mouse replacement. Even though the feet need to be in good contact with the ground in order to move the chair in a controlled way, this grounding does not have to comprise the whole foot. Clicking buttons with the foot or operating an analog foot control is possible and is already commonly used in some areas, for example with sewing machines and racing game controllers. This feature is adaptable for control of desktop environments; However, using a foot control in a highly dynamic environments like a FPS is difficult. The user's feet are in too unstable of a position for the fast movements required when used for button presses. Additionally, experience says that users change their foot positions often to support the fast motions. This behavior is illustrated in the videos of users playing FPSs [9]. Using the bouncing function of the seat as button click is another possible solution. However, for precise selection, this method has proven to be too imprecise. Bouncing is a course movement and slightly modifies the position of the seat in the plane, even when care is taken. That makes especially double clicking and drag and drop actions very difficult.

A potential advantage the ChairIO has over the mouse is that the user no longer has to move between using the keyboard and pointer device (mouse). Typing and fluid mouse movement can occur simultaneously. Toggling between typing and readjusting the cursor position requires only mental toggling, as the hands can stay on the keyboard, while the chair operates the mouse movement.

Similar to learning to operate the mouse, the chair plus foot control input method involves a short learning phase and requires personalization of parameters. Beyond the physical chair adjustments, the most important user settings is that of the mouse speed. In standard mouse usage, this adjustment is largely ignored, as the mouse's usage is largely user independant. However, with the ChairIO this is a much more critical point. With properly setting parameters and some learning time, the ChairIO can be used as a mouse replacement for desktop application.

While standard mouse movement can be easily mapped from the ChairIO translation, the full potential of the ChairIO is not used. Incorporating the other DOFs, the rotation of the seat and the bouncing/hopping motion, is expected to lead to other interaction metaphors and methods that are not presently possible with the traditional input devices, keyboard and mouse. The support of nonstandard metaphors for controlling the desktop can be imagined fairly easily. For instance, a cycle menu has a natural mapping with the rotation of the seat, allowing the user to have a physical movement that is directly related to the metaphor. The rotation of the seat could be mapped to increase and decrease numeric values in a way that is familiar from turning knobs. For example, turning the volume up and down maps intuitively onto turning the seat.

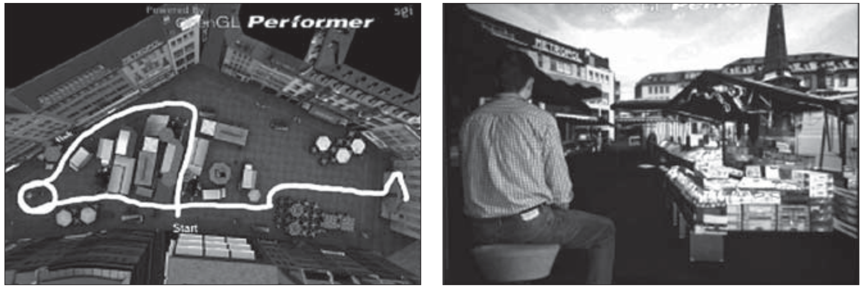
6 User Tests, Observations, and Implications

Developing new interaction techniques and metaphors always requires user testing. User preferences, differences, and their interface requirements are diverse and can only be known by testing. Even with standard interfaces like a mouse, small changes in the placement of buttons can make an interface usable or useless. Predicting, if an interface is intuitive – usable without further explanation and nearly instantaneously – is hardly possible for a developer. Therefore, especially in the area of human computer interfaces, testing is necessary. Unfortunately, while standards exist for testing desktop interfaces, no standard methods exist so far to evaluate more unconventional interfaces in a formal way.

This section describes two of the studies, specifically designed for testing the ChairIO for travel in Virtual Reality and for First-Person-Shooter style gaming. The first study tested the ChairIO's usability for navigating a fairly complex virtual environment. The task set included various travel tasks which are classically very difficult to perform [1]. With this, also potential standard test tasks for navigation methods in virtual environments are presented. In the second study, the ChairIO's applicability for usage in gaming, specifically for First-Person-Shooter style games, was investigated [2]. In both studies the defined tasks are general enough to allow comparison of different hardware interfaces and interface metaphors, i.e. the mapping of hardware interface to behavior in the application. In addition to these informal studies, various opportunities to test out other ideas were exploited. Some of those are reported on here, as they are of general interest.

6.1 3D Motion – User Study

In [1], the results of an initial user study on the ChairIO are presented. A number of goals for this study existed, the foremost being to determine the validity of the ChairIO and the 2.5D mapping (described in Section 3.2) for navigating virtual environments. The secondary goal was to determine, how intuitive the method was, as it was expected that the ChairIO's usage would be self-explanatory. Since a pre-study trail indicated that the usability goals would be easily achieved, a number of tasks were included in the task set that were more difficult and of extreme difficulty with traditional interfaces. Each of the tasks tests a particular aspect of general navigation and some were specifically designed to test navigation with the ChairIO in more depth. The four tasks, in the order performed, were:



(a) Typical User Path

(b) Entering the Passage Way

(c) Column Task

(d) The Stairs

Figure 11. User Study tasks in the Bonn Market Place environment.

1. navigate through an aisle of the market place
2. circle an object, while constantly holding this object in the center of view
3. navigate to the other side of the market as fast as possible
4. climb the stairs of the City Hall, including a 180° turn

The complete course layout for a typical run (the user could pick, which side of the market to travel on for the third task) can be seen in overview in Figure 11(a).

The study proceeded for each user, by explaining only that he or she was testing a new interface for navigating 3D worlds. At this point, the users were given their first task, to navigate through a relatively small aisle way. Note, the users were told nothing about how to use the interface outside of being asked to sit down and that it was a navigation interface. Typical users needed no more than a second to grasp, how to use the ChairIO. No user needed more than 5 seconds to get moving through the aisle, shown in Figure 11(b). This task was simply a "warm-up" for the user and was focused on seeing if the user could perform the navigation through a fairly tight space immediately.

After the aisle, the user was asked to turn and to proceed to the next goal, a large advertising column as in Figure 11(c). The user was asked to circle the object, while keeping the view on the object the entire time. This was the first difficult task, one that is extremely difficult to conduct with standard interfaces. This task combines translational movement with a rotational component at the same time, making it more difficult than any single motion action. All users were able to accomplish the task, but showed differences in the required lengths of time for finding the correct way to perform it. Most users managed to have smooth, isocentric circles around the cylinder. For some users, coordinating the movement required a bit of trial and error, but was achieved in a short period of time.

The third task was to move to the other side of the market as fast as they could. This was to test their ability to control the movement at high speed and, by this point, this task was no problem for any user. The final task was more challenging, as the users were asked to move up the stairs (modeled as a ramp) of the "City Hall," shown in Figure 11(d), including to perform a 180° turn on the landing. This task required fine control of the ChairIO so that the user did not fall off the staircase. Some of the users managed this without difficulty; Others had to make a couple of attempts before they could accomplish this task. In the users defense, it is important to note that this task was made exceedingly difficult in that no collision detection was enabled, meaning that falling off the staircase was very easy – even through the walls.

Among other things, the study showed that the ChairIO was highly intuitive and functional for navigation in a virtual environment. As most navigation devices for virtual environments are hard to use, especially for first-time users, the ChairIO makes navigation, a main feature of virtual environments, available to everyone. Other results were the possibility to achieve fine and fast movements without mode change, and the effective combination of rotation and sideward motion shown in the advertising column task.

6.2 Game Play – User Study

In a second study, it was investigated, how well the ChairIO would perform for playing first person games, where the user has to do more than just move through the world [2]. The First-Person-Shooter genre was chosen because it represents a sort of stress test for the ChairIO usage, as the speed and reaction times that users need to play competitively are high. The user study featured Unreal Tournament™ 2004. The goal of the testing was not only to see, if users could play, but to get insights into the level of mental load that the user experiences using the ChairIO for navigation. It was also conducted to compare view mappings with an additional device as described in Section 4.2. Combining navigation of the world with other interaction methods forms a barrier for many innovative interfaces, as the user's concentration is divided between the two. The study hypothesized that the ChairIO would create little to no impact on the user's cognitive processes, allowing the other interactions to be focused on. The study additionally compared various ChairIO configurations with the standard mouse/keyboard interface and with a joystick.

The study included a wide variety of users, ranging those who had never played games to those who were experienced FPS players and including both male and female users. All of the users reported enjoying playing, including a few users who had expressed a dislike of the game genre before. While all of the experienced users commented that they were slower than with the "twitch" mouse/keyboard interface, they all reported enjoying the more physical game play and expressed the desire to have such an interface (noting that they wanted to play against others using the ChairIO). Several participants, who normally do not play FPS's, found the game very interesting with the chair input and, for the first time, liked playing the game. In those cases, it can be assumed that the intuitive nature of the ChairIO interface, compared with the difficult to learn keyboard mouse control, lowered the initial barrier to be able to play the game. Also, the body motion component, leading to a deeper engagement into the game, drew all users to keep on playing.

6.3 General Observations and their Implications

In our studies, tests, and exhibitions we have been able to extract a number of issues with the interface. While the overall response to the ChairIO has been overwhelmingly positive, there are a few users who had difficulties. In one case, an older gentleman was unable to successfully use the ChairIO; He seemed not to trust moving around on the interface, possible fearing falling off or not physically being able to operate the chair. Other than this, the typical problems encountered usually occurred due to improper calibration of the system or the user's sitting posture on the seat. The calibration is critical for the interface, as users have different body sizes, weights, and sitting preferences. Those differences lead to changes in, among other things, the ChairIO's "neutral position," which is the position where no motion in any direction is triggered. If the user had been sitting tilted forward on the seat while the neutral position was defined in the calibration process, they typically complained of not being able to move quickly forward. Others had small problems with balance, typically because the seat was not set to the proper height, thereby not allowing them to use the muscles of the legs in strong positions. How to overcome these issues and how to calibrate the ChairIO is described in Sections 7 and 8.

6.4 Standing Still – Using Zero Zones

One specific issue with using the ChairIO as a navigation device is standing still in the virtual world. First, it is hard to stay exactly still on a dynamic chair like the Swopper and, second, the sensor noise (sensor dependent) always adds small jumpy motions to the navigation. Because of this, a zero zone mechanism was tested for movement and rotation, disabling any motion within a specified threshold of the centered position. This can also be interpreted as a special nonlinear mapping. The zero zones allow one to stand exactly still in the environment. However, a major disadvantage arose for slow and combined movement. The usually completely intuitive chair behavior became inconsistent when one zero zone was starting to act. Even with a smooth drop off and several other mappings, the original level of intuitive movement without the zones could not be reached. For this reason, most of our applications do not use the zones. The small movements that potentially arise from this actually may enhance the feeling of being immersed and more naturally mirror our own behavior in the real world, hardly ever being completely still.

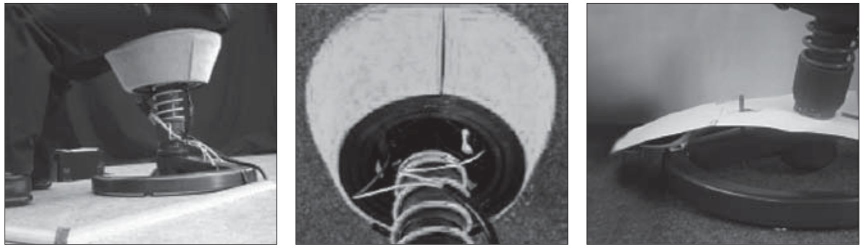

(a) (b) (c)

Figure 12. The tracking setup used at ars electronica festival 2006 is shown. The chair automatically moves back into the standard position using rubberbands. b) Two electro-magnetic sensors are attached in different locations in the chair. c) The ChairIO using a mouse to detect movements.

7 Technical Implementation – Hardware

The conceptual basis of the ChairIO is taking a chair and, through some means, detecting its motions in order to control a computer application. Several applications and mappings of movements of such an extended chair were presented in the previous sections. This section deals with the technical details of implementing the ChairIO device, using different tracking methods, and adding further controls.

7.1 Motion Sensing Technology

This subsection discusses several possible methods of extending existing chairs, particularly the Swopper, to track their movement. Ultimately, the best option would be to have suitable sensor systems included into the chair's design. As this would be highly dependent on the manufacturer to build the system, those solutions will not be discussed here. The discussion focuses on three technologies applicable for existing chairs, using the Swopper as an example.

The basic construction and degrees of freedom of the Swopper were introduced in Section 3. The Swopper is built very compact, such that it is difficult to place sensors inside the suspension system. The free rotating seat hinders the use of cabling, as the cables wind around the chair while rotating. Also, the design of the seat makes mounting of external sensors difficult. This is all compounded by the fact that the tilting arm of the Swopper is not fixed in the base. That means the complete linkage, including the suspension system, can rotate arbitrarily during usage. These factors complicate the fixation of any cable bound sensors.

Three possible sensor solutions for measuring the motions of the different axes are presented here:

- Virtual Reality (VR)-tracking (high cost)
- Marker based tracking (low-cost, but imprecise)
- Laser Mice (low cost, fragile, but relatively precise)

Implementations based on other technologies are conceivable, e.g. wireless solutions, mechanical sensors, or optical solutions. In the future, as more specialized tracking solutions become available for lower price, further options, such as gyroscopes and acceleration sensors, will provide excellent means of measuring the tilt of the chair. using a mouse to detect movements.

VR Solution: In the initial ChairIO work, technology available in VR laboratories was used to quickly prototype the ChairIO. The basis for this solution is the Polhemus Patriot, an electro-magnetic tracking system. Electro-magnetic trackers are popular devices in the VR field, as they deliver precise information about the position and orientation of small tracking units, receivers, in 3D space. The Polhemus Patriot is a low-end desktop solution, capable of tracking two cable bound receivers at maximal range of 1.5m from the emitter.

The tracker could be used in a number of ways to detect the movements of the ChairIO. Here, a method is presented, which proved its robustness over the course of weeks in three installations in public spaces. The two receivers are fastened to the underside of Swopper's seat, positioned so that they are 180° from each other and equidistant from the seat center, as seen in Figure 12. The positions of the receivers are used for the computation of the chair's movement.

A number of parameters – specific to each setup and user – are required for calculating the chair's movement. The first parameter is the orientation of the seat in the initial position, represented by a ray between the two receivers. From the ray, the center point of the seat can be found as the midpoint of the ray. The pivot point of the Swopper linkage can then be found. It is located some distance below the center of the seat. In our initial work, this offset was manually set to a physically measured distance. Unfortunately, the true distance is dependent on the height adjustment of the seat, requiring it to be interactively determined for exact usage.

With those values, the program has all required initialization values. The calculation of the seat position during usage follows a similar flow:

1. Calculate the ray between receiver units
2. Find the center point of the seat
3. Find the x/y translation of the seat by taking the difference between the current seat center point and the initial seat center
4. Using the ray from step 1, determine the angle of rotation of the seat by taking the dot product of the initial and current rays
5. Calculate the height change as: abs((initialSeatCenter - pivotPoint)-(currentSeatCenter - pivotPoint))

The direction of tilt can be derived from the resultant vector of step 3 as with how far the seat is tilted. The distance recovered using this method is slightly incorrect as the height and tilt motion are dependent. The correct distance of travel would be the distance on the surface of the sphere centered on the pivot point. However, as this is small, it can safely be ignored here or solved in similar manner to a method described in Section 8.2. The rotation directly results from step 4 and the height from step 5.

Marker-Based Tracking Solution: Another technology investigated for determining the ChairIO's movement was marker-based visual tracking. Markerbased tracking solutions use various types of markers and cameras to detect the position and orientation of the markers relative to the camera. In the area of Augmented Reality, a popular low-cost solution uses printed 2D fiducial markers and commodity video cameras, such as web-cams. Based on the design of the markers, a single camera is sufficient to perform calculations to determine the 3D position of the markers (although the depth axis is of a higher granularity). By placing markers surrounding the seat, for example as part of the chair's seat cover, it is possible track the ChairIO.

While this solution is cheap and based on easily available tools, webcams and standard printers, it has certain drawbacks. Camera based methods suffer from latency and update rates issues. Here, one of the biggest issues is that of fast movements; To operate quickly, the system assumes that recognized markers only move a small distance every frame, which is potentially a problem for the fast movements necessary for games. The largest problem with this method is the occlusion problem. The markers, or at least a subset of them, must always be visible to the cameras. Mounting the camera at a distance from the chair increases the possiblity of occlusion of all the markers. Conversely, mounting the camera on the chair itself requires the use of expensive optics that are not available on webcams. The last major issue is that the method is highly sensitive to the lighting conditions in the environment. Even small changes to the lighting can cause the markers not be recognized.

Laser Mouse Solution: Current laser mice have a very high resolution, up to 2000 dpi. When moved along a surface, even a shiny one, these mice recognize a precise relative 2D motion. Exploiting this property is intriguing for detecting the movements of the ChairIO. As there are 4 DOFs of interest on the Swopper, tilt (2D), rotation (1D), and height (1D), at least two laser mice are required to track the chair. Here, our experimental laboratory setup is presented. This setup uses two mice: one is used to measure rotation and the seat height and the other for the movement of the chair's linkage.

The mouse, which is used to measure rotation and the seat height, is mounted below the seat on the upper end of the spring, providing a mostly stable position. The sensor is oriented towards the shaft holding the seat. The shaft rotates under the sensor, causing left/right translations of the "mouse," as the seat rotates. The up and down motion of the seat registers as up/down movements of the mouse. The second mouse is firmly mounted upside down to the Swopper base to measure the chair tilt. The sensor is pointing up, and a semi-flexible sheet is fixed to the suspension system slightly above the base and placed upon the mouse sensor, as seen in Figure 12(c). A slot cut into the sheet and a fixed bolt in the chair base restricts the movement of the paper to a foreward/backward motion, if the chair is tilted forward or backward. If the chair is tilted left or right, the bolt in the slot forms a pivot point for the sheet, causing it to rotate over the mouse. Adding a correction for the rotational motion, the original 2D tilt can be computed to achieve a pure translational component in the x axis.

7.2 Additional Controls and Feedback:

The described tracking methods capture the motion of the chair itself, which are the four degrees of freedom of the chair. However, many meaningful computer applications require more input than a 2D cursor position or a 3D position and orientation in a virtual world. The desktop solution, discussed in Section 5, needs mouse clicks to initiate actions. VR and gaming solutions may need to activate objects or selections or may need different additional motion control or mode changes, for example crouch, jump, climb, or hide. Even the control of the ChairIO interface itself requires further inputs, e.g. to switch modes, start and stop, recalibrate, or adjust values like sensitivity of chair or speed.

A simple way to add input is fixing a mouse onto the side of the chair. This was done in an exhibition at ars electronica 2006. Other possible solutions are: buttons placed on the side of the chair seat, buttons on a fixed platform beneath the chair, a control panel fixed to the the chair and following the tilt and rotation movements, foot switches, and handheld button devices. Foot switches for enabling full mouse supplement in desktop environments were already successfully tested and described in a previous section. The usage of additional devices, such as the gun presented in Section 4.2, has also been successfully explored.

The ChairIO can also be equipped to provide feedback to the user. Various kinds of pneumatics or robotic device could be attached to actuate the chair. A simple solution to provide feedback to the user was implemented for the ChairIO. The Buttkicker Gamer (from The Guitammer Company Inc.) is a commercially available device, which delivers a vibration to a chair, similar to what vibration elements in handheld gaming devices do (but much stronger). The device attaches to the seats supporting rod, directly above the suspension system. The generated vibrations are transmitted to the chair's seat. If the application supports this, low frequency sound, music, or special sound effects can be used for feedback of events or the ground's

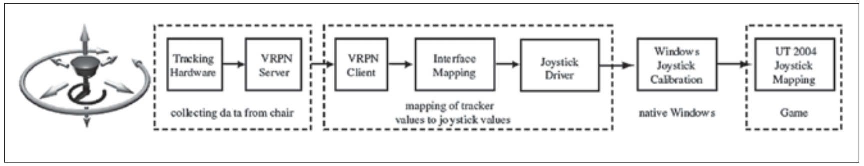


Figure 13. Windows ChairIO to joystick interface architecture.

texture. Moving over gravel, for example, can be supported by a low vibration of the seat. Alternatively, a collision could be signalled, by playing a short, loud sound.

8 Technical Implementation – Software

The methods described in the previous section focus on the acquisition of data from the chair. In order to create meaningful behavior in applications, processing of this data is required, embedded in suitable device drivers for VEs, games, and desktop applications. Additionally, achieving meaningful results in the applications requires calibration of the current environment, chair, and user configuration.

8.1 Device Software Design

This sub-section presents a number of software designs that were developed for using the ChairIO. Each of the designs has its strengths, largely dependent on the purpose the ChairIO is fulfilling. In the first sub-section, the software design for using the ChairIO as a standard VR device is described. In the second, a solution for using the ChairIO as a Windows joystick is presented. In the third, a mouse interface of the ChairIO is introduced.

A commonality to all version is the choice of mapping method for the chair's sensor data to the interface data, which can be performed in either a linear or nonlinear manner. With a linear mapping, tilting the ChairIO to half of its possible range, would generate half the maximal effect in the final application. However, non-linear behaviors can be useful, if value changes of very fine granularity on one side and very rapid value changes on the other side are required. Standard desktop environments use such a non-linear mapping for mouse speed mappings commonly.

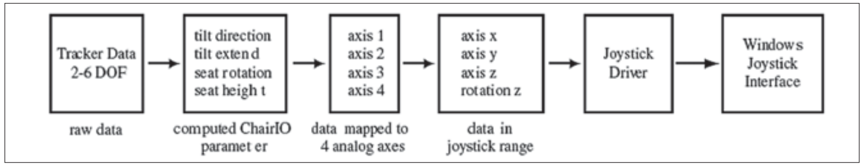


Figure 14. Data flow in the Windows ChairIO interface.

In the navigation application presented in Sections 3.2 and 6.1, a non-linear mapping enables moving with low speeds, useful for fine movement, was well as high speeds for large physical movements. In this way difficult small movements as well as fast travel are possible, without the typical explicit speed or mode changes.

ChairIO as VR Device: In our initial study in Section 6.1, the ChairIO was used to provide navigational steering in the open source VR software framework AVANGO [11]. The AVANGO framework simplified our software system design, as it already provides for connecting very different and experimental devices, as well as providing basic 3D navigation methods. The tracking device presented in Section 7.1 is a common VR device and, consequently, easily integrateable, using standard pathways. A software system, called Virtual Reality Peripheral Network (VRPN [12]), was used to poll the device and deliver a matrix specifying the 6-DOF position of both receivers in an abstracted way and over a network connection.

The computations described in Section 7 are performed in a newly developed AVANGO object, one of the basic building blocks in the framework. The resultant values: the tilt vector, the seat rotation, and the height, are made available to other AVANGO objects. This is performed through AVANGO's field and field connection principle. AVANGO also has several built in navigation metaphors. The "mover” built for the ChairIO was based on the Walker, a mover that performs ground-following, described in Section 3.2.

ChairIO as Joystick: The previously described method is adaptable for usage within any standard VR system. VR hardware is, however, unsupported by standard operating systems. More problematic is that programs designed solely for standard desktop usage can neither be modified nor provide provisions for the VR system solution. Desktop environments typically use one of two standard input solutions, the keyboard and mouse or a joystick. Due to the functional similarity of modern joysticks and the ChairIO, incorporating the ChairIO as a joystick is relatively straightforward.

Rather than writing a joystick device driver from scratch, a device driver framework was used. The PPJoy (Parallel Port Joystick) device driver, developed by Deon van der Westhuysen, was originally designed to input a parallel port joystick into the system as a generic joystick, transparent to the OS. In current versions, PPJoy also allows the creation of virtual joysticks. The values of these virtual joysticks, i.e. the analog axes and buttons, can be set via a C++ API.

The raw data of the ChairIO device is processed in a custom VRPN Client. The two 6-DOF inputs of the Patriot sensors are transformed into the tilt, rotation, and height motions of the ChairIO, like in the AVANGO implementation. The values are then further transformed into the analog joystick axis ranges 1 to 32767 for the x, y (movement), z (height), and z-rotation axis (rotation). The x-axis values from 1 to 16383, for example, represent a left motion, the values from 16385 to 32767 a right motion, and the value 16384 represents the zero point. If digital values are needed, for example, if the height change is needed as a button press, a threshold range for on and off is defined.

The result of this processing is passed to the axes and buttons of the virtual joystick. The virtual joystick is registered as a Windows joystick and can be used with any joystick capable application. How the data is processed through these stages is shown in Figures 13 and 14.

The mapping from the joystick data to application functionality is performed by the application software itself and may not match well with the ChairIO mapping. For instance, an axis with a range of 1 to 32767 might be mapped to three movement modes: slow, medium, and fast. The precision of the analog ChairIO is reduced to something that makes controlling the movement difficult on the ChairIO. Fortunately, some of today's games support modification of the input mapping to different functionalities. For instance, games using the Unreal™ Engine have the ability to specify the use of different joystick axis mappings, scales, and zero zones via a configuration file. This permits the user to configure the system as appropriate. This ability was used in our study using Unreal Tournament™ and also used with the Harry Potter™ games.

ChairIO as Mouse: Unfortunately, today there are relatively few applications that support joysticks, even in the (PC) gaming community. For the ChairIO's use in games without joystick support and other standard applications, a mouse based interface is required. Having such an interface allows the ChairIO to be used in almost every modern desktop application.

A proof of concept implementation of this uses an external "joystick to mouse" converter program on top of the joystick driver described above. This has established a baseline for functionality as a mouse substitute. Of course, going through too many layers adds unnecessary complexities to building a stable and accurate system. A more recent development addresses this problem. It is a combined work with the "two mouse" hardware system described in Section 7.1. The developed driver combines the two mice's inputs in a meaningful way, including creating a button press out of the jump motion. This remains an active area of research in our group, particularly on what mappings are effective.

8.2 ChairIO Calibration

Due to the differences in weight, size, and sitting-posture of different ChairIO users and the various possible adjustments built into the Swopper stool, a calibration has to be performed to make the ChairIO drivers work correctly. The calibration process can be seen as a multi-step process. Central to the process is determining the "zero position," which defines the point, where no movement takes place. The user's sitting position and posture highly influence this position. For this reason, the zeroing step is typically performed with the user on the ChairIO. The second step is determining the values produced by the user's movements. The simplest method assumes a maximal distance based on the physical properties of the chair and interprets values based on this. However, with the individual user's sitting positions, this is not always accurate or desirable.

To combat this and to compensate for tracking inaccuracies, more advanced software approaches for the calibration can be taken. Determining the maximal values for each of the axes independently allows the software to creating a mapping from tracking values to software interface values, such that the maximal possible values for each direction are used. This can be done in an interactive calibration step, directly after the initial zero point is established. By adjusting the maximal values for each of the axes as new values come, the software can allow the user establish the maximal values themselves, for example by moving in each direction to the desired maximum position. When the desired calibration is ready, the values can be set for permanent calibration.

A ChairIO setup that has many users changing often, such as a public display, could potentially benefit from a continuous calibration. In this case, the system would continue to calibrate itself throughout the user's entire session. This has the advantage of the user being able to use the system immediately, with only zero point determination done in the "calibration phase." Unfortunately, it also has the disadvantage that any large movements could potentially alter the mapping in undesirable ways. For example, if the user has only gently moved the ChairIO in the vertical axis and then performs a large bounce, all bouncing after that point must also be so large in order to trigger the bounce event. Such unwanted changes to the calibration can severly impact the usability of the ChairIO.In the setting of International Automobile Salon, a fixed, one-time calibration for all users, based on an average user, was sucessfully deployed, avoiding those calibration issues an creating a less support intensive system.

The calibration method to this point has only be concerned with mapping the tracking values to use the complete range of values for the output of the software interface. There are, however, two additional points to address. These are somewhat specific to the combination of tracking in 3D space and the Swopper based ChairIO. They can, however, be addressed through the introduction of compensation methods built in software.

As most people using the Swopper place their feet on the floor or on the outer chair base ring, more of the user's mass is positioned on the chair when leaning back. While leaning forward, the user's mass on the chair lessens, as most of their weight is supported by their legs. As a result, there is a correlation between the y and z axis movement. If the user moves forward, the software produces values that correlate to a movement forward and upward, due to the weight shift. To compensate for this, a scaling factor can be added to the computations.

The height differences between users requires further consideration and a possible second compensation. A tall user typically has the seat positioned higher, causing the seat to travel a larger distance from the zero position to the outermost position. In a system based solely on the tilt distance, the speed of the induced motion is dependent on the user's height. Adjustments similar to the above mappings can be made, such that all users can move with the same speed range. Conversely, this acts to increase the tall user's input granularity, creating a situation where fine grain movements are less possible for a smaller user.

9 Conclusion and Future Work

The ChairIO has proven to be a highly intuitive and joyful interface to use to control movement in Virtual Environments and games. In this chapter, we have presented the ChairIO on hand of a number of application areas and presented different solutions, both hardware and software, to interface the chair based device to those applications. Additionally, through the results of user studies, tests, and observations made in our lab and at several exhibitions, the ChairIO's high level of usability is demonstrated.

The ChairIO's development is, however, still in progress. For example, developments worth looking into include: novel ways to support impaired users, applications for computer supported muscle training, and more joyful, computer motivated preventive health care. Many more application and research areas can be explored, including the formal evaluation of chair based interfaces for Virtual Reality interaction and the development of new metaphors for desktop applications that use all of the chair's features. The development of a commercial chair with integrated and cost efficient sensors and basic actuators would be most desirable to support all this and to provide a general availability of the ChairIO.

Regardless of the future direction taken, enabling pervasive tools such as the chair in our environment will again revolutionize the way we work.

10 Acknowledgements

Without the contribution of several people, some of the developments related to the ChairIO would not have been possible. The authors would like to thank Jan-Tajo Fittkau, Jan Brauer, and Roland Schröder-Kroll, who inspired or conducted the development of the laser mouse based tracking system, and Martin Berghoff, who assisted in producing some of the images for this bookchapter. The authors would also like to thank Josef Glöck from aeris-Impulsmöbel for generously supplying us with chairs to further develop the ChairIO.

References

1. Beckhaus, S., Blom, K.J., Haringer, M.: Intuitive, hands-free travel interfaces for virtual environments. In: New Directions in 3D User Interfaces Workshop of IEEE VR 2005. (March 2005)

2. Beckhaus, S., Blom, K.J., Haringer, M.: A new gaming device and interaction method for a First-Person Shooter. In: Computer Science and Magic 2005, GC Developer Science Track, Leipzig, Germany (2005)

3. aeris-Impulsmöbel GmbH: The Swopper^{TM} . http://www.aeris.de (last accessed: March, 2007)

4. Sommerer, C., Mignonneau, L.: Art and Science. Springer-Verlag (1998)

5. Paradiso, J., Gershenfeld, N., Hammond, E., Rice, P., Egozy, E.: Sensor Chair. http://brainop.media.mit.edu/Archive/Hyperinstruments/chair.html (last accessed: March, 2007)

6. Cohen, M.: The Internet Chair. Int. Journal of Human Computer Interaction 15(2) (2003)

7. Tan, H.Z., Lu, I., Pentland, A.: The chair as a novel haptic user interface. In: Proceedings of the Workshop on Perceptual User Interfaces, Banff, Alberta, Canada (1997)

8. Holleis, P., Kranz, M., Winter, A., Schmidt, A.: Playing with the real world. Workshop on Pervasive Games (PerGames 2005) on Pervasive 2005 (May 2005)

9. interactive media / virtual environments (im/ve): ChairIO website, featuring videos. http://imve.informatik.uni-hamburg.de/projects/chairio/index.htm (last accessed: March, 2007)

10. Laurel, B.: Computers as Theatre. Addison-Wesley, Reading, MA (1993)

11. Tramberend, H.: Avango: A distributed virtual reality framework. In: Proceedings of IEEE Virtual Reality, IEEE Society Press (1999)

12. Taylor, R.M., Hudson, T.C., Seeger, A., Weber, H., Juliano, J., Helser, A.T.: VRPN: A Device-Independent, Network-Transparent VR Peripheral System. In: Proceedings of ACM Symposium on Virtual Reality Software & Technology (VRST), Banff Centre, Canada (November 2001)