A New Gaming Device and Interaction Method for a First-Person-Shooter

Steffi Beckhaus, Kristopher J. Blom, Matthias Haringer
interactive media/virtual environments, University of Hamburg
imve@informatik.uni-hamburg.de

Abstract

This paper presents a new gaming device and interaction method for First-Person-Shooters (FPS) based on ChairIO, a novel intuitive hands-free interface. ChairIO is based on a stool and is similar to a joystick, but controlled by the user's body motion. The ChairIO interaction is augmented by a game console gun to form a new interaction method for FPSs. An initial evaluation compares several ChairIO and gun based interaction methods with traditional keyboard and joystick controls. Results suggest that the combined interface creates a method which helps beginners to enjoy playing a FPS immediately and gives experienced players a new gaming experience.

1 Introduction

In "real world" games, most gaming metaphors involve physical movement and engagement. Hide and seek, "ring around the rosey", playing with dolls or Lego™, and even card and board games have a physical component. In computer games game control is mostly reduced to operating the standard input devices: mouse, keyboard, and sometimes a joystick. This interaction limits the user's physical involvement to typing and moving the mouse. New game interfaces like the Sony EyeToy™ show the gaming potential of full body interaction.

In this paper, we present and evaluate a gaming interface based on a novel chair-based interface, the ChairIO, and an additional prop, a game console gun with additional gamepad functionality (see Figure 1 and 2). The ChairIO is based on an ergonomic stool with several unique properties that afford its usage as a gaming device. A user study already showed its potential as a highly intuitive and fun to use navigation device in a Virtual Reality (VR) application [3]. We believe that the stool device is not only successful for navigation, but that it also has a large realm of possibilities in gaming. As the stool device is mapped to a joystick input, it is usable in a variety of gaming applications with minimal effort. Here, we explore part of its potential as a gaming device in a very popular gaming application, a First-Person-Shooter (FPS). The low cognitive load in operating the ChairIO plus its hands-free navigation method allow the intuitive use of a second gaming input device, here a gun. We believe that the combined interface creates a method which helps beginners to enjoy a FPS immediately and gives experienced players a new level of gaming experience.

Figure 1. The ChairIO and a Game Gun Used as a First-Person-Shooter Interface.

In Section 2 we give a short review of research involving FPSs and related input devices. Section 3 presents the ChairIO and details of the current implementation, providing joystick-mapped input. Section 4 discusses our chosen experimental platform, interaction aspects of action games, and our various implemented interaction methods. We then describe our informal user study and present our experiences with the interface in Section 5. Finally, we conclude the paper and give directions for future work.

2 Related Work

This section briefly describes the work related to our research. We will first outline the research done on chair-based interfaces to control computers. Then, we will present other research on using FPSs together with alternative displays and devices.

To our knowledge, 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 [10]. He used a chair to control the direction of movement by tilting the chair and the rotation of view by rotating the chair. In [3] we presented a chair-based interface to provide intuitive and hands-free navigational control of a virtual environment. A cushion based interface with custom electronics built into the seat was published in [5].

Much research involving FPSs has gone into reusing them and extending their interaction methods. A number of FPSs have been brought into other display systems and combined with other interfaces. Projects have been created in both Virtual Reality (VR) and Augmented Reality. For an overview of these fields please see [4] and [2] respectively. Popular versions of such environments are *CAVE Quake* [7], *CaveUT* [6], and *ARQuake* [8]. In the case of *CAVE Quake* the models were used from the software and a new engine was written to allow use of VR standard interaction metaphors and viewing inside of VR. *CaveUT* uses the *Unreal Tournament* engine for all aspects. The project centers on using the engine for other purposes than the original game play. Additional works have been largely demonstrations, often having no interaction possibilities outside of navigating the environment.

ARQuake is an application of the *Quake* software from *id Software* in the field of Augmented Reality. In this application, the user navigates the physical environment by traditional means, namely by foot. The opponents are augmented into the surroundings. In an earlier version of the research, the user interacted with a two button device [12]. In a later version, the user interacts using a prop pistol developed by the researchers. The developed device included force feedback, activated when firing the weapon or when the user was hit by an opponent [8].

Action games, particularly FPSs, typically have a fairly large set of interactions possible, some of which must be simultaneously used, for instance navigating the environment plus the various input needs for firing weapons, selecting weapons, and interacting with objects such as doors. Providing appropriate interaction with the game is therefore quite challenging. The standard solution to this is to use dedicated keyboard keys in conjunction with the mouse. These mappings have been highly refined over the course of the genre's development.

3 Interface Design and Implementation

This section presents the design and implementation of our interface. The interface designed is more physically engaging than traditional interfaces. In the first subsection we describe the ChairIO interface and implementation. We then briefly present the gun interface and the implementation we used. Finally, we wrap up this section with a description of the integration of the devices into the computer and software architecture created.

3.1 ChairIO, the Chair-Based Interface

The interface is based on a commercially available seat, the Swopper™ [1] (See Figure 1). The stool is an ergonomic seat for use in an office environment. It has the following properties: rotatable seat, 360° pivot point, height and damping adjustment, and a linkage arm consisting of a spring/shock combination. The seat can tilt in any direction. The spring/damper system potentially allows the user to bounce. To adjust to different users, the seat height and the spring strength of the stool can be altered. The seat itself is on a rotational system on top of the linkage arm, allowing it to independently rotate.

3.1.1 ChairIO User Interaction Metaphor

To operate the ChairIO the user sits on the device and, by shifting their body weight, tilts it in any direction or rotates the seat. This physical movement of the seat is mapped to viewpoint/direction movement in the game environment. For example, to move forward, the user simply moves their body forward, tilting the seat forward. Rotating the view requires slightly rotating the seat, thereby triggering slow or faster rotation of the view in that direction. [3] shows that, for a 3D ground following movement, this method is easy and highly intuitive to use and, furthermore, is fun.

The movement is computationally divided into the component translation and rotation. Translation of the current viewpoint is performed by tilting of the seat in any direction and translation speed is non-linearly mapped in relationship to how far in the direction of the desired travel the user tilts the stool. In an area surrounding the center the mapping is linear; Thereafter, we map the distance as linear plus a cubic factor. This allows the user to travel at higher speeds by tilting the seat further in the direction of travel. In contrast to that presented in [3], we have removed the zero zone centering, assuming it's use was limited in this context due to the fast paced nature of FPS game play. The rotation functions regardless of the tilt of the seat for the translational component. Figure 1 shows the ChairIO in use.

3.1.2 Tracking Method

The current method of determining the position and orientation of the seat uses two points on the seat determined by a magnetic tracker. This method was chosen primarily for its robustness in initializing the interface, as it is not position dependant and allows re-adjustment of the seat's height. An initialization procedure sets a few initial values used in the calculation, such as the rotation of the seat and the position of the Swopper. From the two positions we are able to obtain the translational component from the initial position. We are also able to calculate the rotation of the seat by applying the inverse tilt transform to the seat and comparing with the initial rotation. In the future we plan to integrate low-price standard sensors into the chair.

3.2 Gun Interface and Implementation

The ChairIO interface described above enables the user to navigate the game world. The remainder of the user's interactions have to be mapped onto another input device. For a FPS, the obvious choice for an interface is a gun. The device we have used is a light gun intended for use with a gaming console. Due to limitations of the light gun technology and the time needed to implement this into the game, we have chosen not to pursue that aspect of the gun. Instead, we make use of the gamepad components built into the gun, namely the trigger, the grip button, the "B" button(roughly were a typical trigger lock is), and the d-pad (see Figure 2). In addition, for two of our test setups, we required knowing the orientation of the gun in three degrees of freedom. For our testing we have used an Inertia Cube 2 from InterSense. This tracking device provides an absolute rotation in three axes with real-time update frequency and minimal drift.

Figure 2. Gun Interface with d-Pad, Buttons, Trigger, and Orientation Tracker.

3.3 Interface Architecture and Software Interface

Both the ChairIO and the gun (buttons, d-pad and orientation input) are connected to the game through a single joystick metaphor. The tilting of the stool maps naturally onto the joystick movement. Modern joysticks often incorporate twist measurements on the handle which correlates directly to the rotation of the seating platform (z axis). In addition, joystick drivers often a throttle and a dial input.

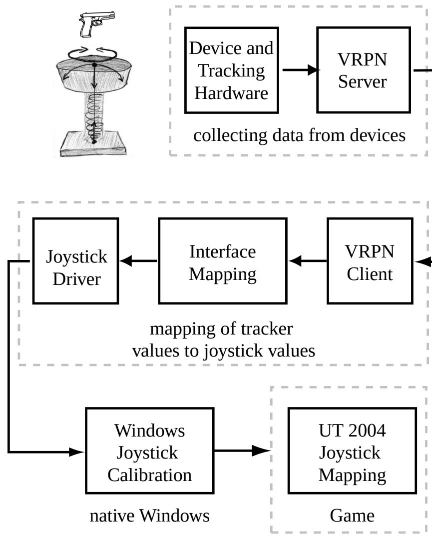

Figure 3. The ChairIO Interface Architecture.

Figure 3 gives an overview of the architecture. In the previous section we detailed the tracking method used for both the ChairIO and gun interfaces. This information is then delivered over a network interface via an external project, VRPN [11]. This information is interfaced into the OS using a modified generic joystick driver. We have combined both the ChairIO and gun interface information into one "joystick" device due to limitations of the game to deal with more than one. The primary axes of the chair interface are mapped on the axes of the joystick. The rotation is mapped naturally onto the appropriate z rotation. The seat height also maps naturally onto the z axis. The orientation of the gun is mapped onto rotational components of the x and y axes of the joystick respectively. The final component is that of bounce on the seat, which we map as button. From the seat height we perform a simple threshold algorithm to determine when the seat is below a certain height.

With this architecture we have a generic interface between our devices and any joystick supporting game.

4 First-Person-Shooter and Game Control

In the area of classic 2D arcade games, the ChairIO is easily applicable using the joystick metaphor that many such games were outfitted with. For 2D and 3D ground following movement, this method is highly intuitive. Incorporating the additional property of the spring/damper of the stool, "jump and run" style games take on a new aspect as the user performs physical movements which are metaphorically akin to the motion of the user's avatar.

For several reasons, explained more in depth in the next subsection, we chose to use a FPS as our first exploratory gaming environment.

4.1 FPS

First-Person-Shooters or Ego Shooters are typically either classified as their own category of computer game or as part of action games.

> ”The primary model in action games is based purely on fast interactions – hand-eye coordination and reaction speed.” [9]

FPSs follow several principles, some implied by its name. First, they are almost always played from a first person perspective. Second, the main (non-navigational) interaction with the world is shooting. The user navigates the world and shoots objects/opponents. Typically one sees their virtual weapon in front of them and usually including some sighting (targeting) mechanism, commonly crosshairs centered on the screen.

There are three further characteristics of FPS game play which contribute to our work: the speed of navigating, the targeting aspect, and the number of interaction possibilities. Running, typically set extremely fast for experienced users, is usually in the direction of the viewpoint. The viewpoint is rotatable, providing a method of both navigating non-straight paths and also used heavily in target acquisition. Rotation is often set on the mouse movement, again set at a very high rate for experienced users. The final point is the number of interaction possibilities. While FPSs typically have many possible (configurable) interactions, there is a relatively small set of interactions typically used by players. While small in number, the users must perform several simultaneous interactions. In Section 4.2 we highlight a standard set of interactions in FPSs.

Our choice of the FPS as a testbed application for the ChairIO stems from these characteristics. The high number of simultaneous inputs creates a high cognitive load on the user. A large portion of that task is that of navigation, which we have found to be intuitively performed with the ChairIO and with low cognitive load. We conjectured that the ChairIO would help users, particularly those who are not experienced players of FPS or action games. Conversely, the exaggerated navigational speeds of the FPS creates a situation which is potentially a problem for the ChairIO. A chair-based interaction is physically bound in a way that mouse control isn't, as it requires movement of the torso including twisting for the rotational component. In choosing the FPS, we have chosen an extreme case.

For this study we use Unreal Tournament 2004(UT). The primary reason is that UT supports a joystick interface, which we desired for the ability to input pseudo analog values for the motion inputs. UT has another advantage, namely its openness to modifications at the user level. This support opens many possibilities for future work, including incorporation of various mods already available within the large modding community.

4.2 Standard Control of a FPS

As with most computer games, the FPS genre uses mainly the keyboard and mouse as interface. Over the course of the development of the genre, the interaction method has been refined and generally is fairly standardized. Figure 4 lists in column 1 a set of common interactions and in column 2 the commonly implemented interaction methods/devices. A noted interaction missing from this list is that of talking, either through typing text or through actual speech. Both of these tasks remain the same, although more difficult when the keyboard is the input method for text input.

Interaction	Standard Method	Joystick	V1	V2	V3	V4
Fire	L mouse button	trigger	P.Trigger	P.Trigger	P.Trigger	P.Trigger
Fire (alt)	key or R mouse button	top button	P.Grip	P.Grip	P.Grip	P.Grip
Cycle Weapons	key or scroll wheel	top button2	P.BButton	P.BButton	P.BButton	P.BButton
Crouch	key or M mouse button	base button	-	-	-	-
Jump	key	base button2	C.B	C.B	C.B	C.B
Move Forward	key or M mouse button	tilt	C.tilt	C.tilt	C.tilt	C.tilt
Move Backward	key	tilt	C.tilt	C.tilt	C.tilt	C.tilt
Sidestep Left	key	tilt	C.tilt	C.tilt	C.tilt	C.tilt
Sidestep Right	key	tilt	C.tilt	C.tilt	C.tilt	C.tilt
Turn Left	mouse movement	handle twist	C.LRot	P.Dpad	P.Orient	C.LRot
Turn Right	mouse movement	handle twist	C.RRot	P.Dpad	P.Orient	C.LRot
Look up	mouse movement	throttle	P.Dpad	P.Dpad	P.Orient	P.Orient
Look down	mouse movement	throttle	P.Dpad	P.Dpad	P.Orient	P.Orient

Figure 4. Table of common tasks in FPSs (legend: Pistol=P, ChairIO=C, left=L, right=R, Rotation=Rot, Bouncing=B)

4.3 The ChairIO and Gun interfaces in a FPS

Using the ChairIO plus the gun provides for various possibilities to map device features to the standard FPS interaction tasks outlined in Section 4.2. As we have mentioned before, the mapping of the ChairIO onto navigation is already set. However, the mapping of the other interaction task is less straightforward. We have developed and tested four different mapping versions (see overview in Figure 4, columns 4-7 or the description below).

In all cases the ChairIO is used for the translational component of the movement and the bouncing, mapped to a button, is used for jump. Likewise, in all cases the gun is used for the weapon functions: fire, alternative fire, and weapon selection cycling.

- V1: The horizontal rotation component is mapped onto the ChairIO (view: left/right). The gun's D-pad up/down is mapped to control the vertical component of the viewing rotation (view: up/down).
- V2: The rotational component of the viewpoint manipulation (view: up/down, left/right) is mapped solely onto the d-pad. This method was created in order to keep a consistency in the method for rotation.
- V3: The orientation of the gun is used to control the viewpoint (view: up/down, left/right), using a incremental rotation scheme as explained in the ChairIO's rotation aspects.
- V4: The horizontal rotation component is mapped onto the ChairIO (view: left/right). The gun's orientation controls the vertical component (view: up/down) of the viewing rotation. Since FPSs are typically ground based travel, this maps all travel components onto the ChairIO. The targeting components are then fully mapped on the gun.

5 Evaluation

In order to evaluate the usability of our interface we have performed an informal user study. In this section we discuss our experience with the ChairIO and gun interfaces. We first explain the study's design and how it was performed. Then we present the results of the study and discuss our experiences with the interface.

5.1 Informal Study

Eight users took part in the study, ranging in age from 21 to 39, two of which were female and six male. The prior experience of the users ranged from non-gamers and those with no FPS experience to experienced FPS gamers. With a pilot-study prior to the user study, we narrowed the number of interfaces to try to V1, V3, mouse/keyboard, and joystick. V2 was found to be unusable for the required reaction times needed in a FPS. In the course of testing, several users on their own suggested V4, leading us to test on that setup additionally. The users with more experience did not test the mouse case.

For each test setup the user was given two tasks. The first task was following a winding path leading to a "generator," a non-moving target. At the generator, the users were advised to aim and shoot at a white square (about double crosshair size) from a defined distance. The exact UT setting was an "Onslaught" scenario without bots on the map called "Frost-bite." This task was relatively short, taking at most a few minutes. The purpose of the task was two-fold: to assess how intuitive the interface was to use and to give the subjects a chance to learn the interface without the stress of the next task. The second task was to play the game. The user was placed in a "Death Match" against novel level bots in the map called "Rankin." We assessed their ability to play the game with the given interface. After a few minutes of play time, we ended the session for that test setup.

The technical setup was as follows: The keyboard/mouse setup used the standard setup from the game. The joystick used was a Logitech Wingman Force 3D and configured as stated in Figure 4. All setups were on standard 17" LCDs at a resolution of 1280x1024. Other settings were kept to the defaults that UT set.

Prior to their first task, subjects were requested to fill out a short questionnaire over their data and experience with gaming. After completion of the test cases, the users were given questionnaires for each method to fill out regarding their experiences. The questions asked concerned their assessment, on a scale of 1-5, of how intuitive the method was, how precise in navigation, how good it was possible to see around and above/below, how precise it was in targeting, how easy it was to operate the controls, and, if the method was fun to use. Finally, overall ratings of the best method for targeting, navigation, and fun were asked.

While the user was performing the task, the observer was recording his impressions on how well the interface and the method worked for the subject. Additionally, all user comments were noted.

5.2 Discussion of Results

Observing the navigation task, we found that novice users had particular problems with the standard and v3 method (30s-2min chaotic view, followed by a slow learning curve), while with the ChairIO navigation(V1 and V4) they were able to navigate well within moments. For novices users targeting/aiming of moving targets in the second task was difficult with all the methods and in some cases even for the static target in the first task.

The questionnaire responses provide an informative user view on the experience. The selection for the best navigation method was distributed among the methods. The preferred method for aiming was the standard method (6 users) followed by V3 (2). Asked for which method was the most fun, V3 (5) led, followed by standard method (2) and V4 (1). Those that selected the standard method, said the reason was a better capability to get more "Frags." For all users that tested the joystick method (3) it was regarded worst, both for precision and fun. These overall judgements were mostly supported by the data collected for each method.

For general assessment we can say an number of things from the study and our experience. While experienced users found that the Swopper navigation was less precise, they had more fun playing with it. Several felt that the same efficiency as with the standard method could be achieved with practice and tweaking. They commented that the flow of game was best in V3 and V4. Lastly, we found that the sensitivity values set for the method is important for successful use, particularly in the case of beginners and in learning.

6 Conclusion

In this paper we have presented a new gaming interface, the ChairIO, and explored one of its many potential interaction methods. We coupled the ChairIO with a common hand-held light gun to create a complete interface for playing a First-Person-Shooter. Various interface combination and interaction methods were explored. The ChairIO was used primarily for the navigational aspects (running, sidestepping, jumping) of the gaming interaction and the gun for the weapon functions. Differing methods combining the interfaces were explored for the more difficult task of targeting.

An informal study of the developed interaction concepts was performed. The study showed that the interface has several benefits. As expected, the ChairIO succeeds well in it navigational portion of the task; However, because it is physically based, it does not allow the extreme rotational speed and accuracy that experienced users achieve with the traditional mouse. The gun's natural affordance to the weapon functions also performed as expected. Test subjects with little to no experience with the traditional mouse interface were able to begin successfully playing the FPS within a few minutes and reported enjoying the experience. We feel quite encouraged by these results, particularly in light of the comment from Rollings and Adams:

> "*Quake*-style games, even though they have standardized on a fairly logically consistent control system, are by no means simple for a complete beginner. Learning to use the *Quake*-style interface is the biggest barrier to mass-market success for first-person 3D games." [9]

Besides the aspects intuitivity and low learn curve, the ChairIO allows combined interaction with the gun creating an fun and natural interface. Beginners enjoyed the game almost immediately and even though the expert users were not able to perform at their usual level, they enjoyed playing the game with our interface as well. All subjects selected one of the new interfaces as the most fun to play.

7 Future Work

Encouraged by our results, we have several areas in which we plan to improve the interface and research further. Broadly, these include: building a simpler interface for detecting the stool input, further refining the FPS interface, and exploring ChairIO potential in other gaming genres and in a more general context.

We are currently working on interfacing the stool device to the computer with a new method of capturing of the input data: tilt of the seat, rotation of the seat, etc. Circuitry integrated into the stool will interface as a HID joystick. This seems to be the most flexible, easily integratable, and configuration friendly method available.

We have presented here the most requested use of the ChairIO, for FPS style games. However, the interface's use is interesting within a broader context of gaming than FPSs. Application of the travel metaphor can be imagined for various "jump and run" and side scrolling style games. We believe the immersive quality and enjoyment of the games will be improved by including things such as jumping and physical action. The interface as a general input device within the operating system is another avenue of interest, mainly concerned with applying and evaluating the ChairIO in standard desktop applications.

References

[1] aeris Impulsmöbel GmbH. The swopper. http://www.aeris.de.

[2] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355 – 385, 1997.

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

[4] G. C. Burdea and P. Coiffet. *Virtual Reality Technology, Second Edition with CD-ROM*. Wiley-IEEE Press, June 2003.

[5] P. Holleis, M. Kranz, A. Winter, and A. Schmidt. Playing with the real world. Workshop on Pervasive Games (PerGames 2005) on Pervasive 2005, May 2005.

[6] J. Jacobson and Z. Hwang. Unreal tournament for immersive interactive theater. *Commun. ACM*, 45(1):39–42, 2002.

[7] R. Paul. CAVE QUAKE. http://brighton.ncsa.uiuc.edu/ prajlich/cave.html).

[8] W. Piekarski and B. Thomas. ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM*, 45(1):36–38, 2002.

[9] A. Rollings and E. Adams. *Andrew Rollings and Ernest Adams on Game Design*. New Riders Publishing, May 2003.

[10] C. Sommerer and L. Mignonneau. *Art and Science*. Springer-Verlag Telos, March 1998.

[11] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN: a device-independent, network-transparent VR peripheral system. In *VRST*, pages 55–61, 2001.

[12] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi, M. Morris, and W. Piekarski. ARQuake: An outdoor/indoor augmented reality first person application. In *ISWC*, pages 139–146, 2000.