Univ Access Inf Soc
DOI 10.1007/s10209-009-0146-3

LONG PAPER

Gaze controlled games

Poika Isokoski · Markus Joos · Oleg Spakov · Benoît Martin

Abstract The quality and availability of eye tracking equipment has been increasing while costs have been decreasing. These trends increase the possibility of using eye trackers for entertainment purposes. Games that can be controlled solely through movement of the eyes would be accessible to persons with decreased limb mobility or control. On the other hand, use of eye tracking can change the gaming experience for all players, by offering richer input and enabling attention-aware games. Eye tracking is not currently widely supported in gaming, and games specifically developed for use with an eye tracker are rare. This paper reviews past work on eye tracker gaming and charts future development possibilities in different sub-domains within. It argues that based on the user input requirements and gaming contexts, conventional computer games can be classified into groups that offer fundamentally different opportunities for eye tracker input. In addition to the inherent design issues, there are challenges and varying levels of support for eye tracker use in the technical implementations of the games.

P. Isokoski (✉) · O. Spakov
Department of Computer Sciences/TAUCHI,
University of Tampere, 33014 Tampere, Finland
e-mail: poika@cs.uta.fi

O. Spakov
e-mail: oleg@cs.uta.fi

M. Joos
Department of Psychology,
Dresden University of Technology, Dresden, Germany
e-mail: joos@applied-cognition.org

B. Martin
LITA, University of Paul Verlaine, Metz,
57045 Metz Cedex 1, France
e-mail: benoit.martin@univ-metz.fr

Keywords Computer games · Eye tracking · Taxonomy

1 Introduction

Computer games are a significant part of popular culture. Eye trackers offer interesting human-computer interaction opportunities for people with and without disabilities. This paper discusses these opportunities from two points of view in the context of computer games. Firstly, eye trackers are widely used to assist people with disabilities that prevent manual operation of computers. For such users, many of whom have limited access to collaborative gaming, the possibility of playing computer games using eye trackers would be highly desirable. Secondly, eye trackers may enhance the playing experience for all players, with or without the usual manual input devices. Mainstream gaming industries are unlikely to adapt their products significantly for the benefit of the minority of users who currently rely on eye trackers. However, if benefits for all players were found, mass-market products may be more likely. People with disabilities would benefit in the form of increasing availability and decreasing prices of eye trackers, while gaming in general would benefit from incorporating a new and powerful interactive and attention-aware tool.

At first glance it may seem as though little is needed in order to make standard computers eye tracking capable. The most popular operating principle is the video-based corneal reflection eye tracking [4]. A typical corneal reflection eye tracker follows the way that light reflects from the cornea and the retina. To avoid complications due to light sources outside the tracker, infrared light is usually used. Not having other intense infrared radiation sources near the computer is a reasonable assumption in indoors environments where light bulbs are usually in the ceiling. In addition to the infrared illumination, the trackers consist of a video camera that is sensitive to infrared light, a CPU that is powerful enough to analyze the video stream in real-time, and software that calculates the point where the user's gaze intersects the display.[1] All these components, except the infrared illumination and the eye tracking software, are present in many computers sold today. Most video cameras are sensitive to infrared light. In fact they are so sensitive that they have a filter which blocks infrared light. Making this filter adjustable so that it could be made to block visible light for eye tracker use is probably not expensive. The cost of adding a few infrared LEDs to the sides of displays in order to illuminate the user's eyes is also rather small. Thus, there appear to be no significant technical hindrances.

Unfortunately, this is not quite the case. Typical cameras do not have sufficient resolution or the kind of optics necessary. Although the addition of viable optics, increased resolution, and infrared illumination does not necessarily imply a huge rise in cost, computer manufacturers are unlikely to add hardware to their products without a significant market demand. Therefore, it seems likely that unless new applications are found, eye trackers will remain specialized devices and will not become a part of the standard computer setup. However, if a successful mass-market product is found, it is likely that unit prices of eye tracking equipment will decrease, even dramatically, as the demand for eye trackers brings about larger manufacturing volumes.

In search for applications that could make large-scale eye tracker manufacturing a realistic development, this paper reviews past work in eye controlled games and discuss possible future developments, concentrating on the use of an eye tracker as the only input device. It should be kept in mind that this is an unlikely scenario for the majority of computer users. Most users today use several input devices including mice, keyboards and dedicated game controllers according to the demands of the game and the context of use. This practice is likely to continue with eye tracking as a possible addition to the arsenal. Thus, the point of tight focus on eye tracking is only to highlight the strengths and weaknesses of eye trackers as gaming devices, not to propose abandoning other devices or to exclude multi-modal use of all available input and output devices. Under a pragmatic point of view, the main interest here are the technical and practical issues involved in eye tracker gaming.

Eye tracker gaming is in its infancy. Most researchers have approached the area from the point of view of existing games and asked questions like how could eye trackers be integrated in a particular game. This is the approach adopted in this paper as well. In the following, two ways of classifying computer games are presented. Firstly, according to the type of input that a player must accomplish in order to play successfully. This classification allows identifying games which could be controlled solely by eye movements without hindering successful gaming. Secondly, games can be classified according to the means that must be used to make them work with eye trackers. This is a pragmatic classification which allows estimating the skills and effort necessary in order to produce an eye tracker compatible version of a particular game. Finally, examples of eye controlled game projects are discussed, followed by conclusions on the current state and likely future developments of eye controlled games.

1.1 Eye movements and games

Eyes are primarily sensory organs. While the field of view of human eyes is wide, both eyes have only a small area (fovea) that can see accurately. The fovea is circular and covers about 1–5 degrees of the visual field [4]. The range from 1 to 5 degrees is not intended to represent variation in the actual size of the fovea, but rather the varying use of the term. The accuracy of vision does not change abruptly from sharp to fuzzy at a certain point. Instead, there is a gradual degradation when moving further from the center of the fovea, thus the ambiguity on where to draw the border.

When perceiving a scene, humans move their eyes in rapid jerky movements known as saccades to direct the light from different parts of the scene onto the fovea. Between saccades the eyes remain relatively fixed. These stops are known as fixations. In addition to saccades and fixations, eyes can follow moving objects in a slow smooth movement known as smooth pursuit. Because receptor cells in the eyes only react to changes in the amount of light they receive, and in lack of changes quickly adapt and stop responding, the eyes need to move during long fixations. These movements are small and usually not noticed without special equipment such as eye trackers.

The eyes can be controlled with remarkable speed and accuracy even in difficult situations, such as under simultaneous body and head movements. However, because of the size of the fovea, the accuracy needed for large vision-related saccades is not very high. Abrams et al. [1] report standard deviation of saccade endpoints between 0.7 and 0.9 degrees depending on saccade length. This is understandable, since there is a need for rapid eye movements to be accurate enough to quickly project any given point in

[1] Calling such devices gaze trackers might be more accurate, since their most important function is that they track the eyes with such precision that they can calculate where the eye is pointing. We call these devices eye trackers to follow the convention within the field.

visible space on the fovea, but accuracy beyond that is superfluous. Small microsaccades can be used to further adjust eye position more accurately.

A good eye controlled game might utilize the knowledge of these naturally occurring eye behaviors—either to by supporting their use or by challenging the player to learn new behaviors.

In a game the use of the eyes for observation of the scene must also be possible at any time. If the natural use of eyes for information gathering causes undesired actions in the game, the player may feel distracted and trapped. Thus, switching between observing and controlling must be effortless. Preferably, the game should be modeless, so that observing and controlling happen simultaneously without conscious effort.

For good results in fast-paced games rapid and fluid use of eye control is essential. Otherwise, an eye control bound player will be at a disadvantage against other players with other input devices. In turn-based games the requirements for the efficiency of eye control are lower, and inefficient control methods are less likely to interfere with a satisfying gaming experience.

These considerations allow to characterizing the degree to which a game is compatible with eye control. Turn-based and other games which do not require continuous on-line control can be easily adapted for use with an eye tracker. The same is true for many single player games which require real-time control, because the difficulty of the game can be adjusted to fit the player's performance. Games that require continuous control and are played competitively against people with other input devices constitute a significant challenge for eye tracker input design, because other players and other input devices determine the difficulty level. Table 1 summarizes the situation. It divides gaming situations into those which require turn-based and continuous control on the one hand, and to solitary and group situations on the other.

The following sections discuss the relationship of different in-game interaction techniques and the framework in Table 1. Interaction techniques are intended as the different ways that the players use the input devices to control the game. Pointing and clicking is perhaps the most common interaction technique. It is used in different contexts in different games. For example, in chess it could be the only interaction technique needed to choose the piece to move and then the square to move it to. In other games pointing and clicking may appear as a part of more complex interaction techniques. For example, in a first person shooter game, the player may be pointing and clicking on a target to shoot at it, while simultaneously moving to avoid enemy projectiles. Situations like this where there are multiple degrees of freedom that must be controlled simultaneously pose significant challenges to eye tracker input as the only input technique. In multimodal use of several input devices these situations offer opportunities for designs that offer efficient combinations of the input modalities.

1.2 Methods of implementing eye control in games

On the technical side, there are four different ways of implementing eye control in games. The simplest of these requires no game specific modifications. Most eye trackers offer an eye-mouse mode of operation where the tracker software commands the operating system to place the mouse cursor at the point where the user is looking at. Most trackers also emulate the mouse button press by sending a button press to the operating system whenever the user's gaze remains within a small area for a pre-determined amount of time. This selection technique is known as dwell-time based selection. It is a trade-off between accidental selections due to too short dwell-time and slow operation due to too long dwell-time. The optimal settings depend both on the context of use and on the user's preferences. However, together these mouse emulation techniques are sufficient to play some games.

The second simplest solution is to use additional software outside both the eye tracker and the game itself. In a sense this is what the eye-mouse software bundled with most trackers does. Sometimes, however, no such generic solution exists and the external software needs to be tailored for each game separately. For example, a program which repeatedly sends events corresponding to pressing the trigger button may be sufficient to make some games playable with the eye-mouse functionality of an eye tracker.

Table 1 The challenges in adapting computer games for eye tracker use as a function of control mode and number of players

	Control	
	Turn-based	Continuous
One player	Easy (chess against a computer)	Moderately difficult (FPS against a computer)
Many players	Easy(chess against a human)	Difficult (FPS against humans)

FPS refers to first person shooter games where the player moves in a three dimensional virtual world constantly controlling the direction and rate of movement

The third possibility is that the source code of the game is available and can be modified to allow eye control. This approach tends to be more labor intensive than the two previous ones. It is also rarely applicable, since the source code of most commercial games is not available.

The most labor intensive method of all is building a new game from scratch. Although expensive, this method also allows the game design to maximize the potential of eye tracker input. Eye controlled games which represent a genuinely new gaming experience for all players, and differ significantly from the dominant game genres, can be achieved only by creating completely new games.

The level of effort is also a reasonable predictor of the number of games that are available in each category. There are a large number of games that can be played with the basic eye-mouse and text entry functionality available in most eye tracker packages. There are some projects that have created add-on software that makes playing of some games possible[2] and some projects that have modified the internals of existing games. On the other hand, no commercial projects are known to develop a completely new game for eye tracker use. Examples of student projects that have implemented eye tracker controlled game prototypes can be found on the COGAIN website at http://www.cogain.org.

2 The state of the art in eye controlled games

Long before the integration of eye trackers in interactive computer systems, special eye tracking technologies such as cameras were developed in order to study human perception and cognition through experimentation [2]. Eye tracking experiments that presented stimuli and recorded the viewer's reactions were developed in order to understand various aspects of attention and cognition. Wade and Tatler [16] provide an impressive overview of the historical development of eye tracking research and methodology. Although computerized eye tracking systems were not involved until much later, the experience of a participant in these early experiments was sometimes close to a computer gaming experience. Just like many computer games, these experiments required fast reactions and sometimes a degree of problem solving. However, usually there was no storyline and no intention of immersing the participants in a game world. These missing aspects are often considered distinctive characteristics of computer games.

One of the first documented prototypes with computerized real-time eye tracking and intentionally constructed storytelling was the gaze responsive self disclosing display by Starker and Bolt [15]. The display showed a planet that rotated slowly. On the planet an observer could see objects such as mountains and staircases. When the observer's gaze dwelled on these objects long enough, the system gave more information about the object of interest using synthesized speech. While such dynamic story telling systems may not be considered games by today's standards, many games have similar features. Exploring one's surroundings and interacting with objects and characters that are found during these explorations is a frequently used way of revealing the plot of the game to the player.

Eye trackers have traditionally been expensive and difficult to use in comparison to other input devices. For example, traditional eye tracker issues such as restrictions to head movements and need for calibration are not a part of the use of most mainstream input devices. Consequently, the potential market for eye controlled games has been small. Eye controlled games have so far been research prototypes or training tools to be used when introducing an eye tracker to a new user, although therapeutic applications have also been proposed [11]. In recent years, eye trackers have developed into a more user-friendly direction and both the number of games bundled with eye trackers and the number of research prototype games has increased. In the following, some of these games are described in order to give a flavor of the current state of the art.

The Eyegaze communication system by LC Technologies includes paddle games and a Score Four game in its software package. The paddle games involve an on-screen paddle which is used to stop an on-screen ball from hitting one of the sides of the playing area. This is very similar to 'PONG', one of the earliest video games, released 1972. Score Four is a game where the players alternately place marks on a grid and the first to have four marks in a straight line wins. In addition, LC Technologies has developed a gaze-operated version of Mahjong, an old Chinese board game.

Eye tracker input is especially suitable for paddle games. The player's task is to place the paddle so that it stops a ball from escaping the playing area. This task becomes trivial if the paddle follows the player's gaze. All the players need to do is to follow the ball with their eyes. Dorr et al. [3] conducted an experiment to verify that using eye tracker input instead of a mouse indeed improves player performance in this kind of a situation. They set up an experiment with 20 students playing against each other in pairs. One player played with a mouse defending one edge of the screen and another with an eye tracker defending the other edge. Each player in a pair played with both input devices. The eye tracker had a statistically significant advantage demonstrating its suitability to fast-paced paddle games.

[2] For an example see the Chicken shooting game example in this paper.

At the time of writing this article, the MyTobii system by Tobii Technology included a Minesweeper game and a Gobblet game. The Minesweeper is an eye controlled version of the well known strategy game made famous by its inclusion in the Microsoft Windows operating system. Gobblet is a four-in-a-row variant played on a 4×4 board. It deviates from the traditional tic-tac-toe mode of play by allowing the moving of pieces and having playing pieces of different sizes. A big piece can be placed on a smaller piece thus stealing the opponent's position.

Oleg Spakov has implemented Chess, Tic-tac-toe, and Lines in the MyTobii environment [13, 14]. Chess and tic-tac-toe are well known games. Lines is a puzzle game invented by Oleg Demin [17]. The goal is to arrange similar items into line or cross formations that, when completed, disappear. The challenge in the game arises from the fact that new pieces appear on the display after every move. One must make moves which are likely to help the formation of multiple shapes to avoid having the game board getting completely filled with pieces.

Jönsson [8] experimented with Half Life (First Person Shooter) and Sacrifice (Shoot-em-up). Her participants claimed that eye controlled gaming was more fun than using traditional input devices. Additionally, players achieved higher scores in Sacrifice with eye control than without it. Player performance in Half Life was not reported. Before deciding to use the Sacrifice game for her experiment, Jönsson had tried another Shoot-em-up game where targets moved across the screen. She found that players tend to track the target which leads to shots landing slightly behind the target.

The same finding was reported by Smith and Graham [12], who experimented with three games: Quake 2 (FPS), Neverwinter Nights (Role playing game), and Lunar command (Shoot-em-up). Lunar Command is a two-dimensional shooting game where the player defends bases at the bottom of the screen from missile attacks by shooting at the missiles by pointing at the desired interception point and pressing a button. Smith and Graham reported lower scores with eye control than with mouse control. They explained that this is mostly because it is very difficult to aim at the empty space in front of the target using eye pointing. Aiming at an intersection of the missiles extrapolated trajectory was necessary in Lunar command in order to compensate for the time of flight of the projectiles.

Quake 2 is a First Person Shooter game where the player moves in a three dimensional world and shoots at the creatures found there while avoiding being shot. Neverwinter Nights is a computer-based role playing game where the player's avatar is shown from a third person perspective and moved by pointing and clicking. For both these games Smith and Graham report lower performance with eye control than with mouse control. The players received little training. With training in eye control comparable to their experience with the mouse the situation might be different. However, the questionnaire data reported by Smith and Graham shows that the players felt more immersed in the game when using eye control. An overall verdict on the value of eye trackers as gaming devices is difficult to make based on the evidence gathered by Smith and Graham. This is because it is difficult to weigh performance against immersion. If the game does not require competition against people using other input devices, the value of immersion may be greater than the value of a high score. If, however the other players have more efficient input devices and can effortlessly beat the eye tracker based player, performance may be felt as more important, no matter how immersed the losing player was.

A third report on eye trackers in FPS games is from Isokoski and Martin [6] and Isokoski et al. [5], who implemented an eye tracker input mode in their input device testing software. This software looks like a FPS game to the player, but was written for the purpose of testing input devices in pointing and navigation tasks in three dimensional virtual worlds. Isokoski and Martin were interested in comparing the efficiency of different input devices. Their comparison included a gamepad with thumb-operated joysticks, keyboard and mouse, and a combination of mouse and eye tracker. In the eye tracker mode, aiming could be done with two crosshairs. One stayed in the middle of the display and was used for aiming by rotating the point of view using the mouse. The other crosshair followed the players gaze. During the experiment only the eye controlled crosshair was used in the eye tracker mode. The results suggest that the eye tracker mode was competitive with the gamepad, but the mouse and keyboard mode was more efficient than the two other modes. Isokoski and Martin report only preliminary results for one player, thus it is difficult to generalize the results.

Isokoski et al. [5] reported a further study where six participants played for 50 min in 10 five-minute blocks using three different input device configurations. This study used the same environment as the 2006 study testing three different input device configurations. The configurations were: (1) full gamepad control, (2) moving with a gamepad and aiming with eyes, and (3) steering and aiming with eyes (only velocity control and trigger for shooting were operated manually with the gamepad). The results showed that the increasing eye control did not affect the players' performance in terms of targets hit, but did increase the number of shots fired.

Generally speaking, all eye tracker gaming experiments have been short. It is hard to predict what users will think of eye trackers in the context of computer games in the long run. Additionally, not much is known about the potential dangers of eye tracker gaming in the form of repetitive strain injury, for example.

3 Game genres and their challenging features

This section discusses traditional computer games and the suitability of eye tracker input for controlling them. The notion of game genre is used in a colloquial way to mean the often poorly defined and constantly changing loose grouping of games into categories based on similarity. Often a genre in this meaning is born when a game is particularly successful and several similar products appear as a result. What makes these groups of games poorly defined is that some contain features that were not present in the original one. Over time, successful genres tend to split into several sub-genres that may merge with other sub-genres creating a mesh that is very difficult to describe comprehensively. Rather than comprehensive, the current discussion is intended to be of practical value by highlighting properties of games that make them particularly suitable or unsuitable for the use of eye trackers as input devices.

Table 2 presents a brief overview of the positive and negative indicators for eye tracker control for each game genre. The issues identified in Table 1 are included (one vs. many players and real-time vs. turn-based game). In addition, multiplayer situations present two levels of difficulties. The first is the need to communicate and perform well enough to manage in a multiplayer situation (Online multiplayer), and the second is to do this in a game that is not turn-based, but requires instant reactions (Online real-time multiplayer). Games requiring continuous position or velocity control or activation of controls while visually attending to something else pose special challenges in implementing eye tracker input. These, as well as other aspects of these game genres, are discussed in more detail in the following sections.

3.1 Computerized board games and puzzles

Board games, as well as their computerized versions, are usually one player or multiplayer games with turn-based control. Therefore, they do not place excessive demands on the efficiency of eye tracker input implementation. For example, chess, go, monopoly, checkers, and many other traditional games do not have any inherent hindrances against successful eye controlled versions. Eye controlled versions can be implemented by merely making sure that all user interface elements are big enough for easy selection with a pointer controlled with an eye tracker. The same holds for many puzzles. For example, implementing eye controlled versions of crosswords, Sudoku, and missing square puzzles does not pose significant challenges for the user interface design.

Unfortunately, theoretical simplicity does not necessarily mean that all these games are easy to play with eye trackers in practice. Currently available versions may have targets which are too small to select comfortably with eye pointing. Games may also be programmed using platforms which read pointer events directly from the device, making it difficult for eye tracking software to feed mouse events into the game. An example of such problems are games that are based on the DirectX library and programming interface by Microsoft; using direct input from keyboard or mouse through DirectX makes it difficult to inject events generated by an eye tracker.

3.2 Computerized card games

Card games that are mostly about strategy and do not require quick reactions are also based on turn-based control and, therefore, well suited for eye control. Computerized poker over the Internet, for example, has recently become very popular. Again, in theory these games should be easy to operate with eye trackers, but there are practical technical hindrances. Typical online casinos use software based on bitmap graphics. Consequently, the user interface is not resizable. While many in-game controls, such as buttons to select the next action, are large, the "lobby" interface used for selecting the game to play and managing the system settings tends to be full of very small targets making eye controlled use difficult.

Given suitable client software, eye controlled play would be possible in such games. Unfortunately, online casinos typically do not allow third party clients to be used on their servers. In addition to card games, many other online casino games are turn-based, with generous time limits that make them, in principle, suitable for eye controlled play.

3.3 Shoot-em-up (scrolling shooters)

Shoot-em-up games are games with usually two degrees of freedom position control in addition to a trigger and a possible weapon/shield switching control. An archetypical shoot-em-up game has a downward scrolling background that brings with it targets that appear at the top of the screen. The task of the player is to maneuver his or her game character (often a spaceship) around the screen and shoot as many of the targets as possible while avoiding the projectiles launched by the targets and other hostile objects.

Eye control in these games case is challenging, since shoot-em-up games require constant or at least frequent control of position. They often also require a dissociation

Table 2 Positive and negative indicators for eye tracker compatibility of a game genre

Game genre	Indicators for eye tracker use						
	Positive		Negative				
	One player mode	Turn-based gameplay	Online multiplayer	Online real-time multiplayer	Continuous position control	Dissociation of focus of attention and control	Large number of commands
Board games and puzzles	x	x	x				
Card games	x	x	x				
Shoot-em-up	x				x	x	
Beat-em-up	x				x	x	x
First person shooters	x			x	x		
Flight simulators	x			x	x	x	x
3rd person action and adventure	x						x
Level jumping (platform)	x				x	x	
Turn-based Strategy	x	x	x				x
Real-time strategy	x			x			x
Turn-based role playing games	x	x					x
Real-time role playing games	x			x			x
Racing	x			x	x		

of attention to the target, and position control of the shooting device. The player needs to look at a target, but simultaneously stay out of the line of fire if the target defends itself. Also, when avoiding enemy projectiles the player needs to steer the avatar through empty areas. The natural reaction is to look at the approaching projectiles. Therefore, direct mapping of gaze position to avatar position would probably lead to a disaster.

Since shoot-em-up games are usually one player games, the input efficiency requirements are not necessarily high. Furthermore, with slight changes to the control requirements, eye controlled play can be made quite rewarding. For example, the need for the trigger can be eliminated by implementing an auto-fire function. This modification removes the tactical aspect of conserving ammunition in the game, but in many games, the maneuvering task alone can be rewarding.

3.4 Beat-em-up (fighting games)

Typical gameplay in a beat-em-up game consists of well timed operation of mini joysticks and buttons on a gamepad. The player's character in the game uses its limbs and the weapons it holds to attack its opponents. Such games usually involve a wide variety of possible attack moves which must be executed in sequences that penetrate the defenses of the opponent. The player must also move the game character around the scene to avoid being hit by the opponents.

Implementing eye controlled mode is difficult. The large number of controls to be operated is difficult to map to eye movements. It may be possible to build a game where one aims the attacks by looking at the desired part of the opponent's body and the correct move sequence for the attack is then automatically selected. However, there are no games reported in the literature that implement this kind of control method. Consequently, it is difficult to tell h a good gaming experience is achievable. However, it can be anticipated that this kind of eye control mode changes the nature of the game considerably. The manual skill involved in operating the controls is one of the features which attract players to beat-em-up games. The player learns fighting skills and enjoys the power that applying those skills gives in the game world. Automating the fighting may make the game less attractive. However, it may also enable the player to be more immersed in the game, and promote the importance of tactics over manual dexterity.

3.5 First person shooters

First person shooter (FPS) games are a game genre where the game world is shown from the point of view of the player's avatar and where the interaction with the objects and creatures of the game world happens mainly by shooting. Traditionally, moving in a three dimensional world requires at least four degrees of freedom (2DOF for both translation and rotation) and several discrete controls such as buttons for shooting, jumping, crouching etc. In addition, FPS games are often played competitively over the Internet. Therefore, a competitive eye control implementation would need to be comparable to skilled two handed play.

A special difficulty is that the ability to aim accurately is very important in FPS games. If a player can hit far-away targets accurately, he or she has a significant advantage, because the distance gives enough time to evade enemy fire and the likelihood of chaotic close combat situation is smaller. The preferred aiming device of many FPS players is the mouse. Eye trackers are not competitive with the sub-pixel positioning accuracy of modern mice. Aiming is also problematic when using the thumb-operated joysticks of game controllers (such as those delivered with Microsoft Xbox 360, Sony Playstation 2 and others). Some games have implemented systems which assist with aiming. For example, Klochek and MacKenzie [9] describe a game (Halo 2 by Bungie Studios and Metroid Prime by Retro Studios and Nintendo) where the game helps the player in maintaining a good aim even if the target moves. The aiming help needed in eye-based aiming is different. Acquiring and following targets with eye-based aiming is easy. Due to the inaccuracy in fast eye movements, tracker inaccuracies, and tracker noise, hitting small targets efficiently is difficult. It may be possible to add some aiming assistance which could be used when eye tracker FPS performance needs to be competitive with players using a mouse. In one player situations such systems may not be necessary.

Navigation in three dimensional spaces shown from the first person perspective is a special situation of gaze-based interaction, because players have extensive experience in real world navigation with the same point of view. Eyes can be used for controlling the orientation of the game character in a fairly natural way, by turning the field of view whenever the player looks away from the center of the display. When an object of interest becomes visible, the player naturally tracks it with his or her eyes and the turning of the field of view could automatically stop when the object of interest reaches the center of the display. Overall, however, FPS games cannot currently be played efficiently using eye tracker input alone. Good solutions for eye-based velocity control and trigger operation are needed.

3.6 Flight simulators

There are different kinds of flight simulators. Some are based on scenarios of realistic navigation of passenger and cargo planes on rather peaceful routes. The pace of gameplay is often slow enough to be categorized into the turn-based category in Table 1. However, simulators that require continuous control are almost as difficult to use by eye control as FPS games. There is a whole class of flight simulators where the difficult flying conditions are an important part of the game. For example, damaged fighter planes may need to fly under enemy attacks in a difficult weather. All flight simulators require at least periodically observing the environment while manipulating the controls to make the simulated airplane turn. This is difficult to achieve with eye tracker input.

In addition, implementing control modes where a lot of the manual control is automated changes the game. The whole point of a simulator is to learn and use the skills needed in the simulated environment. Removing the need to do so removes something essential from the game. Thus, it would appear that flight simulators are not a fruitful avenue to pursue in eye tracker gaming at least if traditional airplane controls are being simulated and only eye tracker input is available.

3.7 3rd person action and adventure games

Instead of the first person perspective, many action and adventure games use a third person perspective. These games typically relieve the player from the need to control the camera angle. In first person games the scene is rendered as if it was seen through the game character's eyes. Third person games are shown from different angles, often as if there was a cameraman following the game character. The camera can be locked into a specific position in relation to the game character or the game may position the camera automatically. Depending on the camera angle, the player can see more of the events around the character than in games shown from the first person perspective. However, controlling the game character remains complicated and it is difficult to map all the controlled degrees of freedom to eye movements.

Sometimes detailed control over the character's movements is not necessary. For example, moving can be completely automated so that the player only needs to point to where he or she wishes the character to move, and the game controls the character's movement to the desired location, even dealing with the obstacles on the way. The character navigation in the role playing game example given by Smith and Graham [12] was of this sort. In the gaze controlled version the player pointed with eyes and clicked the mouse button to make the game character move to the pointed coordinates.

Overall, using eye trackers to control these games is possible in principle, but most games do not work without modifications. The issues may include incompatibility with the eye trackers mouse emulation, too small targets, need for text input, numerous keyboard controls, and action sequences that require fast and precise pointing with a mouse.

3.8 Level jumping (platform games)

The platform game genre involves horizontal movement along platforms shown on the display and jumping from one platform to another while avoiding obstacles and collecting points or other items for later use. Originally, level jumping games used a simple two-dimensional bitmap as the background for the game animation. Later when three dimensional graphics have become commercially viable, platform games in three dimensions have been produced as well. Three dimensional platform games are sometimes difficult to distinguish from 3rd person action and adventure genre. Consequently, the possibilities for eye control are similar.

No eye controlled platform games are known to have been developed. The basic controls are simple. One degree of freedom plus a button for jumping is enough for basic operation. However, eye control may not be simple to implement. The players may need to look around the display to be aware of dangers and to plan a safe and effective route. Therefore, direct mapping of gaze direction to movement may not be possible. Also, careful timing of jumps may be challenging with eye control. Controlling the game character in a three dimensional world is more complicated, and many platform games offer automatic help to the player. Still, it is difficult to imagine how an efficient gaze controlled version could be built. However, platform games are usually single player games played without a network connection. In the long run the player's goal is to explore new strategies and to beat his or her own record. Therefore, even inefficient controls may afford rewarding playing experience.

3.9 Turn-based strategy

Turn-based strategy games (e.g., Civilization) give the player as much time as needed to complete one round of information gathering and commanding of units in the field. When all players have completed their turn, the game simulates and displays the results and waits for the players to react during their next turn. Assuming that the user interface elements are big enough, turn-based strategy games should be playable with eye controlled mouse emulation. Unfortunately, this is not always the case because of the same reasons mentioned for other turn-based genres. The games often contain user interface elements that are too small to operate comfortably with an eye tracker. Strategy and role playing games tend to have a huge number of commands that the user can give at any time. A field commander may have dozens of units to command with many possible orders to give. When played manually, the usual strategy is to select units with the mouse, and give the orders through the keyboard. An obvious alternative for eye controlled use is a menu. Extended eye controlled use of menus can, however, be distracting and tiresome. If the menus are too small for eye-based selections, playing with the eye tracker is difficult or impossible.

There is a special genre of simulators for simulating sports teams, such as football (mostly the European variant), or ice hockey teams. Some of these consist mainly of large menu structures that display various parameters of the team and the players. The games that the simulated team plays are sometimes not simulated in detail—only enough to present strategic options to the player in order to make decisions on pulling away a player, or pushing for victory even with the risk of injuries. In other games the strategy game is combined with a low level game simulator where the player controls individual players during a game. In relation to eye controlled play the strategy component is just like any turn-based strategy game. The low level game simulation component, on the other hand, is a mix of a real-time strategy game and a 3rd person adventure where the action occasionally escalates to situations similar to beat-em-up games. Accordingly, the challenges for eye control are considerable as detailed in the sections on these genres.

3.10 Turn-based role playing games

The eye control features of turn-based role playing games are similar to other turn-based games. If sufficient time for operating the user interface is available, the technology used in the game does not interfere with the eye-mouse, and the user interface elements are of sufficient size for eye-mouse operation, the game should be playable out of the box.

3.11 Real-time strategy

Real-time strategy (e.g., Command and Conquer by Electronic Arts) is a game genre where the game does not wait for the players to complete their actions. Instead, events in the game world go on as the player is sending commands to his units. Because of this the game can overrun a player if he or she is slow because of the use of eye tracker input. When playing against computer opponents, the pace of the game can usually be adjusted. Unfortunately, these games require very intensive mouse usage. Therefore, players using eye control may have a significant disadvantage when playing against human opponents over the Internet.

3.12 Real-time role playing games

Real-time role playing games have become very popular recently. Most notably, the World of Warcraft servers (by Blizzard Entertainment) have attracted millions of players.

Eye control problems are similar to those encountered in turn-based role playing games and three dimensional adventure games. When playing against computer opponents the game may be interesting, but against human opponents over the Internet, it may be impossible to design eye control methods that would be competitive. However, because of the cooperative nature of the game, it may be possible to navigate the player's character through the game so that it can avoid fast-paced combat and still offer enough support for its allies to earn its place in the social hierarchy of the game world.

3.13 Racing

Racing games come in many forms. They range from very simple games where the players task is to maintain full speed and steer while hoping to avoid obstacles, to high fidelity simulations. The point of view can be from within the cockpit of whatever craft is used for the race, from the outside, or even from a distance showing the whole track. Some games allow the players to choose from several camera positions. The control requirements are similar to FPS games. Especially when the game world is viewed from the driver's seat, there is practically no difference in the control requirements between a FPS game and a racing game. The authors of this paper are not aware of examples of successful eye controlled racing games.

Racing games also feature some dissociation between the focus of attention and movement control. For example, they often include gauges for speed, fuel, time, etc. Direct mapping between gaze position and steering, for example, is not easy because glancing at the speedometer would cause steering actions. Overall, racing games range from simple games where gaze-based control may be possible given sufficient freedom to modify the game to games where efficient gaze-based control is impossible.

4 Examples

The previous section has discussed the possibilities of eye controlled games in general terms. Different game genres have been described, outlining their position in the continuum of the level of effort that is needed to adapt the games to eye tracker use. This section describes concrete examples. It starts with a game that can be played using the mouse emulation mode offered by most eye trackers that are being offered for augmentative and alternative communication (AAC) use, and it ends with a project which involved a complete rewrite of the user interface. Apart from the chicken shooting game (external modifications) and the Eye Chess (complete rewrite) these examples have not been tested or developed with players with disabilities.

4.1 Go over the internet using Cgoban3

Go is an ancient Chinese game played on a 19 × 19 grid, that is usually drawn on an approximately half meter square wooden board. Smaller 9 × 9 and 13 × 13 grids are often used for quick games and for training beginners. The gameplay consists of two players (black and white) placing stones on the intersections of the grid. The objective of the game is to surround territories on the board. This can be achieved by surrounding empty territory or by capturing the opponent's stones. Stones are captured when one of the players manages to populate all intersections surrounding an opponent's group of stones with his or her own stones.

Go, like Chess, is a game of complex strategy and tactics. Placing of stones is separated by contemplation that in some cases can take hours or even days. In addition to being turn-based, the pace of game is usually relaxed and during a game the player needs to give only one kind of command to place the stones. These properties make Go, in principle, well suited for eye controlled playing. Therefore, it was decided to try if it is possible to use Go software with eye pointing. The test was performed using Cgoban3,[3] the most popular western Go software. Some of the discussion below is not Cgoban3 specific, but applies to Go software in general.

The game window of Cgoban3 is shown in Fig. 1. As we can see, most of the window is taken by the Go board. In addition the right hand side of the window has a bar of additional controls. At the top there is a number of buttons for checking the game rules, resigning the game, asking for the right to undo a move, etc. Next to these buttons is a list of players and observers that are present in the game. Below these are the player information and a chat window.

We used CGoban3 with a Tobii 1750 eye tracker that is integrated into a 17 in. LCD display. The Windows control functionality of the MyTobii software package was used for mouse emulation. To facilitate eye tracker pointing, we set the display resolution to 800 × 600 pixels and increased CGoban font size to 20. Cgoban3 scaled the user interface elements according to the font size. The only element that was difficult to use even with large font sizes was the scroll bar that remained too narrow. We constrained ourselves to the 9 × 9 board. According to our experience 13 × 13 boards might be playable, but 19 × 19 is beyond the upper limit of eye tracker accuracy, at least if mouse emulation is used for pointing.

Gaze aware Go software could alleviate the problems with the pointing accuracy. For example, we found the dwell-time based mouse click emulation difficult to use because the timer animation that MyTobii mouse

[3] Cgoban3 is available for free at http://www.gokgs.com.

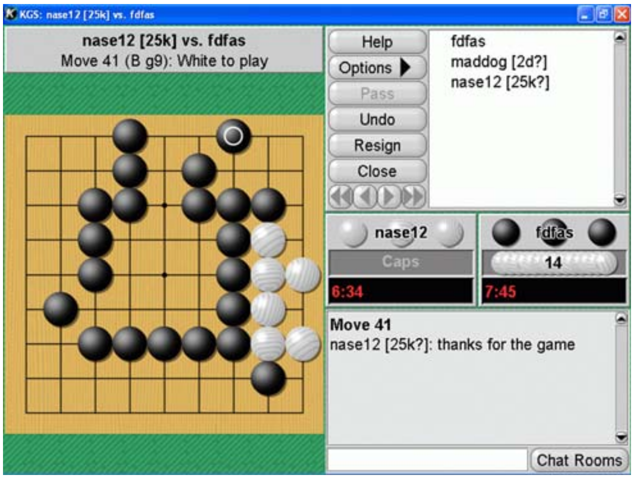

Fig. 1 The game window of CGoban 3: White is taking a beating from black who is using an eye tracker to play

emulation displays leads to a drifting phenomenon that complicates placing the stones. Namely, as the transparent animation is displayed, the player's eyes tend to follow it rather than remain fixed on the position of the desired stone placement. The tracker's gaze point estimation is usually somewhat off the mark. As the gaze shifts on the animation, it shifts further away from the desired stone position, the eyes again follow it, etc. By the time the dwell-time timer fires, the gaze may have drifted away from the desired stone position, and the stone is placed on a wrong intersection. A mistake like this can be fatal in a Go game. If the pointing system could be made aware of the grid on the Go board, the animation could be shown only on the intersection that is nearest to the initial gaze position. This way the iterative drift loop would not occur.

It turned out to be necessary to turn off the mouse click emulation in between moves. Planning a move involves lengthy periods of staring at various positions on the board to mentally visualize the effects of moves and the following move sequences. A dwell time based click emulation would produce unwanted moves. The MyTobii Windows control system has a fairly easy way of turning the click emulation on and off, so this was not a problem.

Despite the difficulties, it was possible to play 9x9 go with eyes only using Cgoban3. Unfortunately, in addition to MyTobii mouse emulation, an on-screen keyboard for text entry is needed. Text entry is necessary when logging onto the server to enter user name and password. Text entry is also desirable for chatting with the opponents and other players. Players who do not respond to greetings and other messages are often considered rude.

Apart from playing, online Go offers other activities. The teaching functionality in Cgoban3 requires a lot of menu usage and scrolling. Both of these activities are troublesome with eye pointing. Observing teaching games and following lessons given by others, on the other hand, requires few interactions and is, therefore, fairly easy to do. Observing the games of strong players and having access to their games for learning purposes is one of the greatest advantages that online Go offers to amateur players. Both of these activities can be achieved with eye pointing on CGoban3, although going through the games offline requires a fair amount of clicking the buttons that control the replay.

Overall, it appears that Cgoban3 could be used with an eye tracker and mouse emulation. Playing on small boards and observing games was easy. Using some of the more advanced functionality and text based chat was more difficult.

4.1.1 Chicken shoot (external modifications)

As an example of a game where external modification enabled use with eye control, Joos et. al. (Dresden University of Technology) have augmented a two-dimensional FPS game called 'Chicken Shoot'.[4] This game is a classic *Fun Shooter* with adventure elements. The main task in Chicken Shoot is simply to blow the chickens away. In each mission the player can collect a number of items and receive extra points for various actions. The initial levels are fairly easy, but the difficulty increases from mission to mission. In normal playing mode the player can move the cross-hair used for aiming with the mouse and shoot using a mouse click. In addition, the 2-D scene can be navigated by scrolling with the cursor keys on the keyboard. After 10 shots the player must reload the gun by pressing the right mouse button. A screenshot of the Chicken Shoot game is shown in Fig. 2.

In order to make this game playable solely through eye control, an external eye gaze to mouse and-keyboard wrapper was developed. The goal was to translate the eye tracker data into mouse and keyboard events and then inject these into the game. The most obvious and simplest task was the positioning of the cross-hair, which was achieved by directly driving the mouse cursor with gaze position. The second task was to emulate the scrolling functionality. This was achieved by using off screen gaze aware regions on both sides of the screen. These regions generated scroll events when the player's gaze dwelled on them longer than 500 ms.

Since the aim was to render the game controllable through gaze alone, the trickiest task was to emulate the shooting command. Since the content of the game is highly dynamic, with both chickens moving in the scene, and the scene itself also being navigable, a dwell time approach was not feasible for two reasons;

[4] http://www.chickenshoot.com.

Fig. 2 The Chicken Shoot game; the cross-hair slightly above the center of the screen is controlled by gaze

(a) The natural types of eye movements in this dynamic situation are smooth pursuit and saccades rather than fixations and saccades. This means that one would have to dynamically move the dwell time sensitive regions according to the objects under scrutiny. Since the 2 D object location is not accessible from outside the game this approach doesn't work.

(b) Even if one could compute dynamic dwell times for smooth pursuit eye movements one would become caught in the trap of the so called Midas Touch problem [7], which means that either a lot of false alarms would result in unintended "friendly fire like" shoots, or dwell time selection thresholds would need to be set so high that the likelihood of hitting a chicken would approach zero.

For these reasons, a switchable automatic machine gun mode was implemented, using an off screen toggle region above the screen. Once this machine gun mode was switched on, left mouse clicks were injected into the game at a rate of 2 Hz until the user toggled the machine gun mode off. Reloading of the gun was performed automatically by counting left mouse clicks and issuing a right mouse click after every 10th left click.

A restriction of the approach became obvious during testing. Although testing was completed with the LC Technologies EyeFollower system, sampling at 120 Hz, the 20 sample delay introduced by the adopted gaze smoothing algorithm complicated the cross- hair positioning control due to the temporal delay and associated spatial offset. Reducing to 10 the number of gaze samples used for smoothing solved this problem. Therefore, in order to control this game, eye tracking devices must have a higher sample rate and sufficiently low noise levels in their data.

The final system was tested by eight subjects at Dresden University of Technology, as well as by participants of the 2006 COGAIN Camp in Turin, Italy. Results showed that subjects' performance with the eye controlled system was initially worse than when using mouse and keyboard for playing. However, after four to five trials, most subjects outperformed the mouse and keyboard control condition in the final score. This result can be explained by the faster positioning of the cross-hair by gaze compared to the manual positioning which involves both a relocation of attention to the target and a subsequent manual relocation of the positioning device.

The final judgment of the successful adaptation to eye control is the subjective experience of the player. In this case, players reported a strong feeling of immersion in the game, and lower feeling of impediment in the eye controlled mode than in the mouse and keyboard mode. For those players that do not have the option of playing with a mouse due to physical disabilities (all disabled participants in the above-mentioned test in Dresden had a late stage ALS diagnosis) a direct comparison cannot be made. However, the physically disabled showed strong feelings of joy being able to play a game after many years when this had been impossible for them.

4.2 First person shooter (internal modifications)

Three projects have investigated the use of eye trackers in first person shooter games. Jönsson finished her Master's thesis in 2004, Smith and Graham reported their results in 2006 and so did Isokoski and Martin. Smith and Graham do not provide details on their implementation, but Jönsson as well as Isokoski and Martin [6] and Isokoski et al. [7] found that they needed to modify the game in order to make eye tracker control possible. Jönsson modified a game called Half Life, Smith and Graham worked with a Java-based implementation of Quake 2 known as Jake2, and Isokoski and Martin used a game-like FPS input device testing framework originally implemented by Laurent Gomilla and Maurice Svay. A screenshot of the display of this testing platform in eye tracker mode is shown in Fig. 3.

In an FPS game the direction of the weapon and the field of view are usually connected. There is a crosshair in the middle of the display and aiming happens by rotating the field of view, so that the target is under the crosshair. In other words, in order to aim, the player must manipulate the field of view so that the target is at the middle of the display. Usually the field of view can be rotated by moving the mouse or displacing a stick on a gamepad.

There are different ways that the eye tracker input could be used in a FPS game. Jönsson implemented a mode where one could aim the weapon with the gaze and move the field of view with a mouse and a mode where the weapon and the field of view followed the gaze. Isokoski and Martin reported only the former, Isokoski et al.

Fig. 3 The display of the FPS input device testing system used by Isokoski and Martin. Notice the two crosshairs. The big white one (on the left edge of the leftmost penguin tablet) was the standard FPS crosshair that stays in the middle of the screen. The small red crosshair (on the neck of the penguin tablet on the right) followed the gaze

reported both, and Smith and Graham reported only the latter.

Jönsson did not report the performance of the participants in her experiment. Smith and Graham and Isokoski and Martin found that their participants performed better with manually operated pointing devices. Isokoski et al. concentrated on various levels of distributing work between eye tracker input and gamepad input, finding similar performance regardless of the distribution.

Jönsson did not have enough participants to perform statistical tests on the data that she collected on the participants' opinions and impressions regarding eye control in FPS games. However, she reported that the participants experienced the use of eye trackers in this context positively. Smith and Graham did not report statistically significant differences in player performance between mouse operated aiming and eye operated aiming. Their participants reported higher level of immersion with eye controlled aiming. In both cases, it is difficult to assess the significance of these results. The participants had a very short exposure to the eye tracker. With longer playing sessions their skills and opinions might change. Isokoski and Martin report longitudinal data, but only for one participant (Isokoski himself). Isokoski et al. found differences in the consumption of ammunition and shooting distance when more work was done with the eye tracker input. However, the longest experiment [5] lasted only for ten 5-min sessions per input device configuration. Therefore, the long-term effects of learning to the relative efficiency of manual input devices and eye trackers remain unknown.

Efficiency is a central issue in FPS games, because they are very popular in competitive gaming over the Internet. Eye trackers are unlikely to be popular if using them leads to losing the game.

Both Jönsson and Isokoski and Martin report the need to alter the internals of the game software in order to make the eye tracker work. The most important reason for this was that they wanted to make the aiming crosshair follow the gaze rather than staying fixed in the middle of the display. Smith and Graham do not give details on their technical implementation. They did not need to make the crosshair move, so it is possible that they did not need to modify the behavior of the game.

Jönsson used a fairly early tracker model from Tobii technology, and possibly because of that decided to implement a two-part software where a server process outside the game communicated with the tracker and then transmitted the eye gaze data to a module within the game that controlled the game according to the gaze input. Isokoski and Martin were using a later model of the same tracker. They used a game-like input device testing environment where a dedicated input module was to be written for all different input devices. Thus it was natural that one was implemented for receiving the input from the Tobii tracker as well.

Overall, it appears that some ways of using eye trackers in FPS games require internal modifications of the game. Note, however, that none of these examples suggests that FPS games could be played efficiently using only eye tracker input. In all of the known examples moving in the game world and shooting still happened with manual input devices. If these functions need to be gaze-operated as well, it is even more likely that modifying the source code of the game is necessary.

4.3 Eye chess (a complete UI rewrite)

Eye tracking seems to offer many opportunities for interesting entirely new kinds of games that utilize the gaze position information. However, no such games are known to be available. So far, most games that have been written with eye trackers in mind are adaptations of old games. In some cases a game can be used directly, like Cgoban3. In other cases small internal modifications are needed, as it was the case with the FPS games. Sometimes, however, it is better to re-write the whole user interface to better support eye tracker usage. This is what Oleg Spakov did in EyeChess [13]. A screenshot of the Eye Chess playing window is shown in Fig. 4.

As was mentioned in the context of Go, dwell time based mouse click emulation is troublesome. If no feedback on the dwell time timer is shown, accidental clicks can lead to a lost game. If feedback is shown in a generic

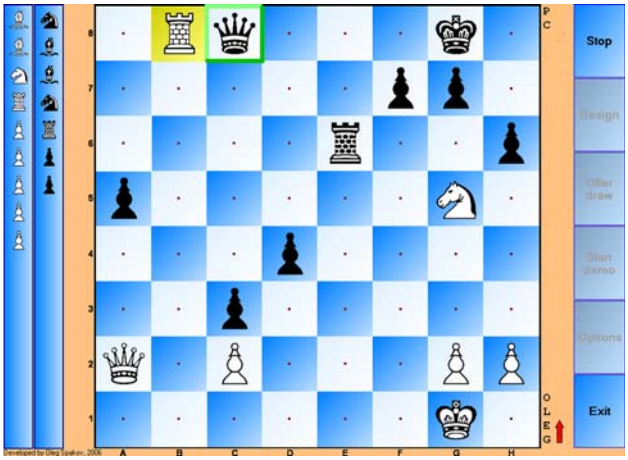

Fig. 4 EyeChess playing window. The square of the selected piece is highlighted, and the square that the user is looking at is shown with the *green border*

pixel pointing mode where the pointer is not aware of the squares of the game board, the feedback can lead to drifting to neighboring squares. In EyeChess the squares of the chessboard show the feedback on the selections. In addition, the squares have a center marker, so that it is easy to focus the gaze on the center of a square in order to maximize eye pointing accuracy.

Interaction with the chess pieces happens by first staring a piece to select it for moving and then staring at the square where the piece is to be moved. In addition to the interaction with the board, games like Chess have a large number of options to set. The user interface elements in EyeChess have been made large enough to use easily with eye pointing. The changes in comparison to mouse operated chess programs may not appear large, but they are necessary. One critical control that is too difficult to operate with eye control can render the whole game unusable.

EyeChess is compatible with the MyTobii environment and has been demonstrated for example at the COGAIN demonstration day in Turin Italy. The reaction from chess players, especially those with disabilities that hinder manual play, has been very positive.

5 Conclusions

Interest in eye controlled gaming has increased. Most of the research projects mentioned in this paper have taken place during the last few years. It is expected that this wave of interest will continue until academic researchers have satisfied their initial curiosity. Further progress will depend on whether commercial endeavors in the area are successful or not. The effect of some critical factors is difficult to anticipate. For example, a lot of computer gaming happens in a living room setting. In addition to online collaborators and opponents, friends that are playing or just hanging around are often present. Remote eye trackers are well suited to situations where the computer is used by only one player. In multi-player situations eye control may be more difficult to use, or require several head mounted eye tracking devices.

For people with disabilities which interfere with manual control in computer gaming, eye control offers an opportunity to play many games. Turn-based games and puzzle games can usually be adapted for eye tracker use. Adapting games that require constant and fine control is more difficult, and even when possible the adaptations tend to require access to the source code of the game. However, these same games offer the greatest opportunities to improve the gaming experience of people without disabilities. Many games require so much complex fine motor control that potential players are overwhelmed with the arrays of sticks and buttons they must operate. If some of the input can be achieved effortlessly with an eye tracker, the gaming experience may be improved.

While participants in the early studies on eye controlled gaming have been very positive in their assessments of the eye controlled modes in comparison to the traditional manually controlled modes, there is still room for skepticism. Eye tracking technology tends to evoke positive reactions. When a person is using an eye tracker for the first time with a well designed interface, the experience may feel as if things happen merely by thinking about them. This experience may distract the participants in short studies from critically assessing their performance with the eye tracker. In the long run, the initial enthusiasm may wear out along with novelty, and without smooth game play attitude may turn against eye control.

Eye controlled gaming is in its infancy. It is too early to draw final conclusions on the value of eye trackers as gaming devices. In the near future interesting demonstrations, prototypes, and experimental results are expected. It is time to assess the situation after seeing this evidence. So far, eye controlled gaming seems promising for people with and without disabilities.

References

1. Abrams, R.A., Meyer, D.E., Kornblum S.: Speed and accuracy of saccadic eye movements: characteristics of impulse variability in the oculomotor system. J. Exp. Psychol. Hum. Percept. Perform. **15**(3), 529–543 (1989)
2. Dodge, R., Cline, T.S.: The angle velocity of eye movements. Psychol. Rev. **8**, 145–157 (1901)
3. Dorr, M., Böhme, M., Martinetz, T., Brath, E.: Gaze beats mouse: a case study. In: Proceedings of COGAIN, pp. 16–19 (2007)

4. Duchowski, A.T.: Eye Tracking Methodology: Theory and Practice. Springer, London (2003)
5. Isokoski, P., Hyrskykari, A., Kotkaluoto, S., Martin, B.: Gamepad and eye tracker input in FPS games: data for the first 50 min. In: Proceedings of COGAIN, pp. 10–15, (2007) available at http://www.gogain.org
6. Isokoski, P., Martin, B.: Eye tracker input in first person shooter games. In: Proceedings of COGAIN 2006: Gazing into the Future, pp.78–81, available at http://www.cogain.org
7. Jacob, R.J.K.: The use of eye movements in human-computer interaction techniques: what you look at is what you get. ACM. Trans. Inf. Syst. **9**, 152–169 (1991)
8. Jönsson, E., If looks could kill—an evaluation of eye tracking in computer games. Master's Thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden (2005)
9. Klochek, C., MacKenzie, I.S.: Performance measures of game controllers in a three-dimensional environment. Proceedings of Graphics Interface 2006, pp. 73–79. CIPS , Toronto (2006)
10. LC Technologies, Eyegaze User Manual, September 2006
11. Lin, C.-S., Huan, C.-C., Chan, C.-N., Yeh, M.-S., Chiu, C.-C.: Design of a computer game using an eye-tracking device for eye's activity rehabilitation, optics and lasers in engineering, 42(1), 91–108, Elsevier (2004)
12. Smith, J.D., Graham, T.C.N.: Use of eye movements for video game control.. In: ACM Advancements in Computer Entertainment Technology (Hollywood, CA, USA, June 14–16, 2006). ACE 2006. ACM Press, New York (2006)
13. Spakov, O.: MyTobii Applications available at: http://www.cs.uta.fi/~oleg/mytobii.html. Accessed 28 Jan 2007
14. Špakov, O., Miniotas, D.: EyeChess: a tutorial for endgames with gaze-controlled pieces. (abstract) In: Abstracts of ECEM13, Bern, p. 32 (2005)
15. Starker, I., Bolt R.A.: A gaze-responsive self-disclosing display. In: Proceedings of CHI 90, pp. 3–9. ACM Press (1990)
16. Wade, N.J., Tatler, B.W.: The Moving Tablet of the Eye: The Origins of Modern Eye Movement Research. Oxford University Press, NY (2005)
17. Wikipedia, Lines (video game) available at. http://en.wikipedia.org/wiki/Lines_(video_game). Accessed 28 Jan 2007