Eye tracking is a process that allows us to determine where an observer is focusing at a given time. The gaze direction indicates where people focus their attention. Recent innovations in the video game industry include alternative input modalities for games to provide an enhanced, more immersive user experience. Gaze control has recently been explored as an input modality in games [Isokoski et al. 2009]. As eye trackers become more accurate, cheaper, and less intrusive to the user, the technology could well be integrated into the next generation of games. It is important therefore to ascertain its viability as an input modality and explore how it can be used to enhance the gamer experience.

However, gaze based interaction is not without its issues. It tends to suffer from the “Midas touch” problem. This is where everywhere one looks, another command is activated; the viewer cannot look anywhere without issuing a command [Jacob 1990]. To combat this problem gaze is often used in conjunction with another input mechanism such as a mouse click. This course presents two case studies that have explored if the “Midas touch” problem can be overcome by combining voice recognition with gaze to achieve a completely hands-free method of game interaction.

The first case study “The Revenge of the Killer Penguins” is a third person adventure puzzle game using a combination of non intrusive eye tracking technology and voice recognition for novel game features [Wilcox et al. 2008]. The second case study “Rabbit Run” is a first person maze game which was created to compare gaze and voice input with traditional techniques, such as mouse and keyboard [O’Donovan et al. 2009].

This course is for people who are interested in incorporating eye tracking in games and virtual environments. The attendees will be given an introduction to attention, eye movements, and different eye tracking technologies. Previous work in the field of gaze in gaming will be summarized, as well as future ideas to create richer interaction and attention-aware behavior algorithms for characters and crowds in virtual environments. The lessons learned in the case studies will be presented and issues relating to incorporating eye tracking in games will be discussed.

We believe that alternative input modalities can be used in novel ways to enhance gameplay, for example by using them to allow virtual humans to interact with objects, navigate through environments, and interact with other characters and crowds [Sundstedt 2009]. Alternative means of interaction in games are especially important for disabled users for whom traditional techniques, such as mouse and keyboard, may not be feasible. Given that gaze and voice are entirely hands-free it also presents a real opportunity for disabled users to interact fully with computer games.

Acknowledgments

The author would like to thank Jonathan O’Donovan, Jon Ward, Scott Hodgins, Tom Wilcox, Mike Evans, Chris Pearce, Nick Pollard, and Carol O’Sullivan for their collaboration on the research and ideas presented in this course.

References


ACM SIGGRAPH 2010 Course – Gazing at Games: Using Eye Tracking to Control Virtual Characters.

Instructor: Veronica Sundstedt, Blekinge Institute of Technology, Sweden & Trinity College Dublin, Ireland.

veronica.sundstedt@bth.se

Wednesday, 28 July, 3:45 PM – 5:15 PM, Room 406 AB, Los Angeles Convention Center.
Welcome to the course: Gazing at Games: Using Eye Tracking to Control Virtual Characters. I will start with a brief introduction to the course which will give you an idea of its aims and structure. I will also talk a bit about my background and research interests and motivate why I think this work is interesting. There are no particular prerequisites for this course.
This course is for everyone who is interested in learning more about eye tracking and how it can be incorporated in games and virtual environments. You will be given an introduction to visual attention, eye movements, and different types of eye tracking technologies. I will also summarise some previous work in the field of gaze-controlled games and present two case studies which I have been involved in. In these case studies gaze and voice have been used in combination to control virtual characters and to interact with computer games. The lessons learned in the case studies will be presented and some issues relating to incorporating eye tracking in games will be discussed. Towards the end I would also like to encourage audience discussion about using novel interaction techniques for computer games. I will also present some ideas for future research using eye tracking to create novel algorithms for virtual characters. So if you are interested in these topics please do come and talk to me after the course as well.
Here you can see a brief overview of the course structure. After this introduction I will talk about attention, eye movements, and different eye tracking technologies. This will be followed by a presentation of some of the key related work in the field of using gaze as an input device in computer games. After this I will present two of our case studies which use gaze and voice recognition in combination to control virtual characters and to interact with computer games. The first case study “The Revenge of the Killer Penguins” is a small third person adventure puzzle game using a combination of non intrusive eye tracking technology and voice recognition for novel game features [Wilcox et al. 2008]. The second case study “Rabbit Run” is a small first person maze game which was created to compare gaze and voice input with traditional techniques, such as mouse and keyboard [O'Donovan et al. 2009]. The lessons learned in these case studies will be presented and issues relating to incorporating eye tracking in games will be discussed. After this I will present some ideas for how the research can be developed further to create richer interaction for characters and crowds in virtual environments. This will be followed by a 10 min Q&A session in which I encourage audience discussion around the topic. This is a short course which will end at 5:15 PM.


Veronica Sundstedt was recently appointed as a Lecturer in Computer Graphics at Blekinge Institute of Technology, Karlskrona, Sweden. Prior to this she was working as a Lecturer in the Graphics, Vision, and Visualisation Group in the School of Computer Science and Statistics at Trinity College Dublin, Ireland. Previously, she was a Postdoctoral Research Associate in the Department of Computer Science at the University of Bristol and the University of Bath, UK. She holds a PhD in Computer Graphics from the University of Bristol and an M.Sc. in Media Technology from the University of Linköping, Sweden. Part of the research presented in this course was undertaken at the University of Bristol and at Trinity College Dublin. Her research interests are in graphics and perception, in particular perceptually-based rendering techniques, multi-modal interaction, experimental validation, and eye tracking techniques.
Traditional input devices include mice, keyboards, and specific game controllers. Recent innovations in the video game industry include alternative input modalities for games, such as motion sensing, gesture recognition, pointing, and sound. We believe that alternative input modalities can be used in novel ways to enhance gameplay, for example by using them to allow virtual humans to interact with objects, navigate through environments, and interact with other characters and crowds [Sundstedt 2009]. Our aim is to create an enhanced, more immersive user experience. Alternative means of interaction in games are especially important for disabled users for whom traditional techniques, such as mouse and keyboard, may not be feasible.

The EyeToy for PlayStation 2 for example is a colour digital camera (like a web camera). The camera allows players to interact with the game using motion, colour detection, and sound via its microphone. To process the images (edge detection, colour tracking, and face mapping) taken by the camera the technology uses vision and gesture recognition. The idea with the EyeToy was to have a more natural user interface. The camera needs to be used in a room which is well-lit in order to process the input from the player. The PlayStation Eye for PlayStation 3 is a successor of the EyeToy camera. The PlayStation Eye make use of a microphone array which allows for tracking multi-dimensional voice locations, echo cancellation, and to suppress background noise.
The Wiimote has motion sensing capability allowing the user to interact on screen via gesture recognition and pointing through the use of accelerometer and optical sensor technology. Project Natal for Xbox360 by Microsoft allow the player to have a completely controller free gaming experience. The add-on equipment allows the user to interact with the game using gestures, spoken commands, and by presenting objects or images.
Recent innovations in the video game industry include alternative input modalities for games to provide an enhanced, more immersive user experience. Eye tracking input has recently been explored as an input modality in games [Isokoski et al. 2009]. Using eye tracking in computer graphics and interactive techniques user studies is a relatively new phenomenon. Nowadays eye tracking technology has advanced and it is possible to obtain both cheaper, easier to use, faster, and more accurate eye tracking systems [Duchowski 2003]. As eye trackers become more accurate, cheaper, and less intrusive to the user, the technology could well be integrated into the next generation of games. It is important therefore to ascertain its viability as an input modality and explore how it can be used to enhance the gamer experience. The images here shows a few different types of eye trackers... for example a T-series eye tracker from Tobii which has the eye tracker built into the screen, the older portable x50 from Tobii and a the head mounted EyeLink II eye tracker from SR Research. Next I will be talking a little bit more about what eye tracking is and how we can use it as input to games...


I will now talk a little bit about visual attention, eye movements, and different eye tracking technologies.
The information in the environment that reaches our eyes is much greater than our brain can process. Humans use selective visual attention to extract relevant information. Visual acuity relates to the resolution limit of the eye and our ability to resolve fine details [Snowden et al. 2006]. Due to the uneven distribution of photoreceptor cells in the retina humans have higher visual acuity in the fovea. The figure shows a schematic representation of the human visual field. The foveal vision has an area of approximately 2°. To reposition the image onto this area, the human visual system uses five basic types of eye movements.


Eye movements

- Five basic types
  - Saccades
  - Smooth pursuits
  - Vergence
  - Vestibular ocular reflex
  - Optokinetic reflex
- Fixations
  - 200-300 ms, 90% of viewing time
- Scan paths
  - Trajectory between fixation points

As described in the previous slide, the highest visual acuity is obtained in the foveal region. To reposition the image onto this area, the human visual system uses five basic types of eye movements: saccades, smooth pursuits, vergence, vestibular ocular reflex, and optokinetic reflex [Duchowski 2003]. The eye movements are controlled by six muscles, which allow the eye to move within six degrees of freedom. The five types of eye movements are summarised below based on descriptions by Vilis [2006] and Duchowski [2003]:

Saccades are fast and ballistic eye movements used to reposition the fovea. Ballistic means that when the saccade has begun the final destination cannot be changed. These movements are both voluntary and reflexive and last from 10-100 ms. There is virtually no visual information obtained during a saccade.

Smooth pursuits are movements used when tracking a visually moving target. Although it is dependent on the range and the speed of the target, the eyes are normally capable of matching its velocity.

Vergence movements are used for depth perception to focus the pair of eyes over a distant target.

Vestibular ocular reflex (VOR) movements are used to fixate the eyes on an object even if the head rotates. It works even if the eyes are closed.

Optokinetic reflex (OKR) movements are used to account for the motion of the visual field. They produce a sense of self-motion which can be experienced when sitting in a stationary train and the opposite train starts to move.

Between eye movements fixations occur, which often last for about 200-300 ms and are rarely shorter than 100 ms [Snowden et al. 2006]. Approximately 90% of viewing time is spent on fixations [Duchowski 2003]. During a fixation the image is held approximately still on the retina; the eyes are never completely still, but they always jitter using small movements called tremors or drifts [Snowden et al. 2006]. The trajectory between fixation points is usually called a scan path [Noton and Stark 1971]. The image shows a scan path from an observer looking at the Lucy statue. The circles indicate fixation points where a larger radius represents longer fixation duration. Rendering of
the Lucy scene (Image courtesy of Diego Gutierrez and Oscar Anson). The model of Lucy was created by the Stanford University Computer Graphics Laboratory. The eye tracking was done using the Tobii x50 eye tracker.


Vilis T., The Physiology of the Senses Transformations for Perception and Action. Course Notes, University of Western Ontario, Canada, 2006.
The information in the environment that reaches our eyes is much greater than our brain can process. Selective visual attention is a complex action consisting of conscious and subconscious processes in the brain, which are used to extract relevant information in a quick and efficient manner [Rensink 2003]. There are two general visual attention processes, termed **bottom-up and top-down**, which determine where humans locate their attention. A lot of recent work has focused on trying to model these processes separate or in combination. This includes models (or maps) that tries to predict what will attract attention, including task maps aiming to model what the user will look at while performing a task or saliency maps, which tries to model what will automatically attract attention in a scene in the case where no task is present. Models of human visual attention have been used in computer graphics to guide rendering resources for example in ray tracing. For example brighter regions in an importance map will receive more samples. Eye tracking has also been used in the past to evaluate if these maps accurately predict where people tend to focus.

In bottom-up processing the visual stimulus capture attention automatically without volitional control [Itti et al. 1998]. Low-level, bottom-up features which influence visual attention include contrast, size, shape, colour, brightness, orientation, edges, and motion. Motion is a very strong cue for attracting attention. Examples of maps from the corridor scene. Frame 1 (left) and saliency map (right). The saliency map is generated using the algorithm by Itti et al. [1998]. So objects that are designed to be salient in real life such as the fire extinguishers appear as salient in the image.

In contrast, top-down processes focus on the observer’s goal; they depend on the task. The top-down approach was studied by Yarbus who asked an observer to look at a picture while their eye movements were recorded. Yarbus [1967] study showed that the scanpath was influenced by the question being asked of the observer while studying the picture. The figure shows seven scanpaths from one observer: (1) freeviewing, (2) estimate the material, circumstances of the family, (3) estimate their age, (4) guess what the family did before the unexpected visitor arrived, (5) remember their clothes, (6) remember the position of the people and the objects, and (7) estimate how long the unexpected visitor had been away. So here you can clearly see that the scan path is altered based on the task.

What is eye tracking?

- Eye tracking allows us to determine where an observer’s gaze is fixed at a given time.

Eye-tracking is a process that records eye movements allowing us to determine where an observer’s gaze is fixed at a given time. The point being focused upon on a screen is called a gaze point. Eye tracking allows us to capture the gaze of an observer. The direction of gaze indicates where humans focus their attention. Eye-tracking techniques make it possible to capture the scan path of an observer. In this way we can gain insight into what the observer looked at, what they might have perceived, and what drew their attention [Duchowski 2003]. We can also study how tasks affect gaze behaviour as seen in the figure above. The left image shows an observer free-viewing the Kalabsha temple scene and the image to the right shows the eye movements of the same observer performing the task of counting the stones in the courtyard.

There are several application areas for using eye tracking [Duchowski 2002]. Eye tracking has previously extensively been used in psychology, neuroscience, human factors, human computer interaction, in particular in the evaluation of web design and advertising to find out what people look at and how sites can be made more efficient for specific tasks etc.

There are different types of eye tracking applications shown in the hierarchy. Broadly eye tracking systems can be divided into two categories: interactive and diagnostic systems. The figure is adapted from Duchowski [2001]. In interactive systems the users gaze is used to interact with the application. The users gaze can be used as an alternative input device. Early work in perceptually adaptive graphics mainly falls into gaze-contingent where parts of the virtual environment are modified based on the gaze of the observer [Luebke et al. 2000]. Gaze-contingent techniques are also divided into screen-based and model-based techniques. Screen-based manipulate frame buffer before display. Model-based techniques reduce resolution by modifying geometry prior to rendering [Duchowski 2001]. Today I will be talking about selective applications where the users gaze is used to interact with and control virtual characters.


Many different types of eye tracking systems have been developed since it was first used in reading research about 100 years ago [Rayner and Pollatsek 1989, Nilsson 2009]. There are two general techniques for studying eye movements: (1) by measuring the position of the eye relative to the head, or (2) by measuring the orientation of the eye in space [Duchowski 2003]. The most common system for the second technique is the video-based corneal reflection eye-tracker [Duchowski 2003]. Video-based eye tracking systems can be head-mounted, portable, or wearable. There are also electronic methods in which skin electrodes are used around the eyes to measure the potential differences in the eye or mechanical methods which uses contact lenses with a metal coil around the edge [Aaltonen 2005]. These are very intrusive techniques. Most eye trackers today use video images of the eye to determine where a person is looking, i.e. their point of regard [Poole and Ball 2004].


Nilsson T., A Tobii Technology Introduction and Presentation, And How Tobii Eye Tracking Could be used in advertising, at Beyond AdAsia2007, Jeju Island, Korea, 2009.


There are several important terms used in eye tracking. The accuracy of the eye tracker shows the expected difference in degrees of visual angle between true eye position and mean computed eye position during a fixation [Aaltonen 2005]. The accuracy is usually 0.5-1°. My experience is that it can be quite hard to obtain this kind of accuracy in many applications. The spatial resolution of the eye tracker shows the smallest change in eye position that can be measured and the temporal resolution or sampling rate is the number of recorded eye positions per second [Aaltonen 2005]. Many eye trackers has a sampling rate of about 60-120Hz. Isokoski et al. [2009] highlights that it is much more important to have rapid and fluid use of eye control in fast-paced games and that it can be lower in turn-based games. In general you need a higher temporal resolution if you want study different kinds of eye movements.


In video-based eye-trackers, the light source reflection on the cornea (caused by infra-red light) is measured relative to the location of the pupil’s centre [Duchowski 2003]. These two points are used as reference to compensate for head movements [Duchowski 2003]. The corneal reflections are also known as Purkinje reflections or Purkinje images [Duchowski 2003]. Due to the properties of the eye four Purkinje reflections normally appear [Räihä 2005]. The first reflection is at the front of the cornea, the second at the rear of the cornea, the third at the front of the lens, and the fourth at the rear of the lens. Video-based eye-trackers normally locate the first Purkinje image [Duchowski 2003]. For example, the Tobii x50 eye-tracker uses near infra-red light-emitting diodes (NIR-LEDs) and a high-resolution camera with a large field of view [Tobii 2006]. The NIR-LEDs and the camera are used to generate the Purkinje images of the eyes and to capture images of the observer. Eye tracking software consists of image processing algorithms that extract important features, such as the eyes and the Purkinje images generated by the NIR-LEDs. It can also calculate a three-dimensional position in space of where the eyes were located to determine the gaze point at a given time. The image shows Purkinje images (left) and (right) relative positions of pupil and first Purkinje images as seen by the eye-tracker’s camera during calibration.


Before starting a recording the video-based eye trackers need to be fine-tuned to the participant in a calibration process [Poole and Ball 2004]. Calibration of the eye-tracker is achieved by measuring the gaze of the observer at specific grid points (usually 5, 9, or 16) [Duchowski 2003, Goldberg and Wichansky 2003]. The Purkinje images then appear as a small dot close to the pupil. The calibration process can be incorporated within the game so that the player is being calibrated before the game starts. It can be hard to keep a good calibration so sometimes drift correction techniques are applied. There is still future research needed to explore how drift correction techniques can be done within an interactive context.


Eye trackers

- COGAIN (Communication by Gaze Interaction)
- http://www.cogain.org/wiki/Eye_Trackers
  - Eye trackers for interactive applications
  - Eye trackers for research on eye movements
  - Open source, low cost, and freeware solutions
- Gaming
  - Non-intrusive that can deal with head movements

I will not go into more details on different types of eye trackers, because different systems are useful for different purposes. For further information about different kinds of eye trackers the COGAIN (Communication by Gaze Interaction) has a very good summarised list of different trackers and extensive information about their target users, technical specifications, and features. For gaming it is important to have non-intrusive eye trackers that can deal with head movements. In general more accurate results can be obtained if the users head is fixed. Another limitation with eye trackers is that they can only track over a specific area.

http://www.cogain.org/wiki/Eye_Trackers
In eye tracking there are several issues that can affect the gaze measurements [Schnipke and Todd 2000]. These can broadly be categorised into two areas: (1) physiological reasons, and (2) external reasons. First, eye trackers are known to have some issues calibrating various lenses (bi-focal/tri-focal, super-dense, hard), lazy eye, large and small pupils, if there is a low contrast between pupil and eye white, and if the eyelid covers part of the pupil [Pool 2005]. Another reason is that the pupil does not reflect enough light. Participants wearing eyeglasses or hard contacts can also be two external reasons why recording can be difficult [Schnipke and Todd 2000]. There are also other issues that can affect recording. For example if the participant is moving their head significantly this could cause a delay until the eye tracker is capturing the eye again and it can also be a reason for calibration loss [Schnipke and Todd 2000].


Schnipke S. K. and Todd M. W., Trials and tribulations of using an eye-tracking system, CHI '00: CHI '00 extended abstracts on Human factors in computing systems, pp 273-274, 2000.
Eye-trackers normally produce a large amount of raw data since humans perform several saccades per second. This raw data contains many parameters (for the left and right eye), including gaze point \((x,y)\), pupil location in the camera image \((x,y)\), distance from camera to eye, pupil size, time stamp in ms, and a frame number [Tobii 2006]. The raw data needs to be filtered and reduced before it is analysed. In this process it is common to identify fixations and saccades [Rothkopf and Pelz 2004]. Blinks can also be identified. This process allows us to get a better idea of the users intentions [Jacob 1990]. Eye tracking can be used for both offline analysis or real time interaction.


However, gaze based interaction is not without its issues. It tends to suffer from the “Midas touch” problem. This is where everywhere one looks, another command is activated; the viewer cannot look anywhere without issuing another command [Jacob 1990]. Midas is popularly remembered for his ability to turn everything he touched into gold: the *Midas touch*. It is not ideal to have the player look at something and then issue a command all the time in the game. Ideally the gaming interface should allow the user to look around the game and interact with objects and characters they choose to interact with. Jacob [1990] was one of the first to look into feasibility of gaze based selective interaction. He first identified “Midas Touch” and suggested dwell time to overcome it. To combat this problem gaze is often used in conjunction with another input mechanism such as a mouse click or dwell time; the object will be marked when the user looks at it and after a certain delay it will be chosen. Instead of using a mouse click or dwell time in combination with gaze input our work has explored if voice recognition can be used instead, which I will talk about more later in the case studies.

I will now talk a little bit about some of the previous work done in this field.
Starker and Bolt [1990] introduced one of the first systems with real-time eye tracking and intentionally constructed storytelling. A rotating planet was displayed with various features including volcanoes, staircases, and flowers. When the gaze of the observer was focused on these objects for a certain duration, the system provided the user with more information regarding the object using synthesized speech. Although this was not a game as such, games of today allow the player to investigate the surroundings and interact with the environment and other characters.

A few games, such as Tom Clancy’s EndWar™ Ubisoft, use voice recognition allowing the player to input voice commands to the game. In this game, voice control is used to issue orders to troops for example. The following table shows some different games that make use of voice recognition in gaming. There are two different categories of speech recognition technologies, speaker dependent and speaker independent [Mehdi et al. 2004]. Speaker-dependent requires each user to go through a process of training the engine to recognise his/her voice. Speech independent recognition avoids this by training with a collection of speakers in the development phase [O’Donovan 2009]. Our case study Rabbit Run presented later is using a speaker independent technology. Instead of using a mouse click or dwell time in combination with gaze input our work has explored if voice recognition can be used instead. Our work has investigated if the combination of gaze and voice as an input modality offers a hands-free solution to the “Midas Touch” problem. I will talk more about this in a bit.


Several computer games have exploited the concept of visual attention in gaming. For example, in *The Legend of Zelda: The Wind Waker™*, Nintendo, the avatar of the player can direct attention to important objects or approaching enemies by looking at them. This can aid the player in puzzle solving.
Attention in gaming

- Sennersten [2004] – action game tutorial
- Kenny et al. [2005] – fps games, focus in center
- El-Nasr and Yan [2006] – top-down & bottom-up
- Sennersten et al. [2007] – eye tracker + HiFi engine
- Sundstedt et al. [2008] – active vs. passive
- Stellmach [2007], Sasse [2008] – psychophysiological logging, data acquisition

I will describe some other work in more detail that uses gaze in gaming but first I just wanted to mention some references that has looked at attention in gaming. Sennersten [2004] studied eye movements in an action game tutorial (Counterstrike v. 1.0). The SMI iView© eye-tracker was used to gather information about the players eye movements. Recent studies suggest that in adventure games, fixation behavior can follow both bottom-up and top-down processes [El-Nasr and Yan 2006]. Visual stimuli are reported to be more relevant when located near objects that fit players’ top-down visual search goals. In first-person shooter games, gaze tends to be more focused on the center of the screen than in adventure games [Kenny et al. 2005; El-Nasr and Yan 2006]. Sennersten et al. [2007] have also performed a verification of an experimental platform integrating a Tobii eye tracking system with the HiFi game engine. In an experiment involving active video game play, nine low-level heuristics were compared to gaze behavior collected using eye tracking [Peters and Itti 2008]. This study showed that these heuristics performed above chance, and that motion alone was the best predictor. This was followed by flicker and full saliency (color, intensity, orientation, flicker, and motion). Nonetheless, these results can be improved further by incorporating a measure of task relevance, which could be obtained by training a neural network on eye tracking data matched to specific image features [Peters and Itti 2007]. Stellmach [2007] developed a psychophysiological logging framework, which allows psychophysiological data can be correlated with in-game data in real time. The framework is also capable of logging viewed game objects via an eye tracker integration, which can inform us of how game elements affect our attention. Sundstedt et al. [2008] performed a psychophysical study of fixation behaviour in a computer game. Their study particularly looked at the differences in eye movements for active and passive game play.


Gaze in gaming resources

- Survey paper
- COGAIN – Communication by Gaze Interaction
  - http://www.cogain.org/links/gaze-controlled-games

In the last few years there have been an increasing amount of work done in the field of gaze in gaming. Although this work is still in its early stages. It is still not common to use eye tracking in gaming and there are not many games that support eye tracking technology [Isokoski et al. 2009]. For an extensive survey on gaze controlled games please see the survey paper by Isokoski et al. [2009]. Some previous work is also summarised in the M.Sc. Thesis of O’Donovan [2009]. When looking at eye tracking as a means of computer game interaction it is important to look at different genres and the challenges they present. Isokoski et al. [2009] discuss this in some detail. In this course I will mentioned a few projects that have used gaze in games. I will particularly focus on 3D games. For more information on previous projects using gaze in virtual environments please see:

http://www.cogain.org/wiki/Gaze-Controlled_Games


Leyba and Malcolm [2004] used eye tracking as an aiming device in a computer game. They compared mouse and gaze for aiming. They thought that although the accuracy would be less with gaze the speed to remove 25 balls with random velocity vectors would be faster. The aim of the game was to remove 25 balls by clicking on them using mouse. To overcome the “Midas touch” problem a conservative form “Manual and Gaze Input Cascaded (MAGIC) [Zhai et al. 1999] pointing was used. This kind of pointing warps the cursor to the general point of regard. The user can then make small adjustments to be directly on target with a mouse. Leyba and Malcolm [2004] adapted method used the gaze point as the cursor position on screen when the mouse was clicked. They found that as expected mouse input was more accurate. Gaze also proved to be slower but the authors say this is most likely due to problems with calibration and the players tending to click the mouse repeatedly rather than aim accurately when using the mouse as input.


Jönsson [2005] used gaze input in two open source games, the FPS game Half Life and the Shoot-em-up Sacrifice. Open source versions of these games were adapted to accept gaze input from a Tobii eye tracker. Two demos were created using Sacrifice, one were aim was controlled by mouse and one by gaze. In Half Life three demos were created, one were the weapon sight and field of view were controlled by mouse, one were weapon sight and field of view were controlled by gaze and one where weapon sight was controlled by eyes and field of view with mouse. Participants achieved higher scores with eye control when playing Sacrifice than without it. Jönsson also reported that playing with gaze was more perceived as more fun. Jönsson also experimented with another shoot-em-up game with moving targets. It was reported that players tend to track the target which resulted in that their shots were missing the target and landing behind it.

Kenny et al. [2005] conducted a study in which they looked at players fixations after playing a first person shooter (FPS) game. Their results showed that players spend 82% of the game time looking in the near centre of the screen, the inner 400 x 300 rectangle of a 800 x 600 screen. They FPS game was created using the Torque Game Engine and they used the SR Research EyeLink®2 eye tracker to record the eye movements. Our work presented later in one of the case studies “Rabbit Run” is inspired by their result.

Smith and Graham [2006] used the Tobii 1750 eye tracker as a control device for video games. They examined how it could be used within different game genres, such as in Quake 2 (FPS), Neverwinter Nights (roleplay), and in Lunar Command (action/arcade). In Quake 2 eye movements were used to control the orientation. In Neverwinter Nights they study eye tracking as a means of interaction with characters. Finally in Lunar Command eye tracking was used to target moving objects through pointing. The participants played all three games using both means of control. Users performed significantly better with the mouse in Lunar Command but no significant performance difference was found for Quake 2 or Neverwinter Nights. One of their main results indicate that using gaze control can increase the immersion of a video game. Smith and Graham also found that players tend to fire behind missiles using eye control which agrees with the result by Jönsson [2005].


Isokoski and Martin [2006] developed a FPS style game and compared the score obtained using different input techniques. They compared the score obtained when using keyboard/mouse/gaze, only keyboard/mouse, and with an Xbox 360 controller. The game used gaze for aim, mouse to control the camera and the keyboard to move the player around the scene. They report that using eye tracking does not improve performance in comparison to the mouse/keyboard condition, but it is competitive with the Xbox 360 controller. They only evaluated the system with one player (one of the authors).

Isokoski et al. [2007] presents some additional results using the same FPS game as presented in their previous work. They used the same game as in 2006 with three different input techniques. The first one was to have full gamepad control. In the second one they moved with the gamepad and aimed with the eyes. Finally in the third condition they were both steering and aiming with their eyes. Their results showed that increasing eye control did not affect the performance in terms of target hits but they did fire more shots.

Isokoski et al. [2009] reports that eye tracking is particularly useful in paddle games such as PONG style games. Dorr et al. [2007] investigated if the performance by participants controlling a paddle using gaze or mouse in a modified version of LBreakout2, published under the GNU General Public License (GPL), would differ. The results showed that the eye tracker had a statistically significant advantage over mouse control. When playing with gaze the ball was released after 5 seconds to avoid a needed mouse click. They used the SensoMotoric Instruments iViewX Hi-Speed tracker running at 240 Hz [Dorr et al. 2007]. They also report that they have played it successfully with a 50 Hz SMI RED-X remote tracker, in which they did not need to fix the head of the player.


Istance et al. [2008] have developed a software tool called Snap Clutch. This is an application which uses gaze data to generate normal key and mouse events. These events can then be used to use applications such as World of Warcraft and Second Life. The application makes different gaze interaction techniques available to the user and allow them to switch between these in an efficient manner [COGAIN wiki]. Eye trackers that support Snap Clutch include Tobii and the ITU GazeTracker [COGAIN wiki]. This tool has also been used in Vickers et al. [2008].


http://www.cogain.org/wiki/Snap_Clutch

The Second Life 3D virtual environment (Linden Labs, www.secondlife.com)
Istance et al. [2009] reports on the development of a software device using gaze input in different modes for emulating mouse and keyboard events when interacting in on-line games. They explored how gaze could be used to control the game World of Warcraft using this device. They report that it is possible to perform tasks at a beginners level using gaze alone. Dwell time can be used to drop a magnifying glass to inspect things further for example [Istance et al. 2009].

Castellina and Corno [2008] created some simple 3D game scenarios to test multimodal input. They used semitransparent buttons (or “Virtual Keys”) to rotate the camera and move the avatar. As can be seen in the figure these were used to move forward/backwards and to look left/right. These were activated by dwell time. In their studies gaze input was found to be as accurate but since dwell time was used they mention that it was not as fast as mouse/keyboard interaction.

Hillaire et al. [2008] developed an algorithm to simulate depth-of-field blur for first-person navigation in virtual environments. Using an eye-tracking system, they analysed users’ focus point during navigation in order to set the parameters of the algorithm. Using this focus point they propose rendering techniques which aim to improve the users’ sensations. The results achieved suggest that the blur effects could improve the sense of realism experienced by players. Hillaire et al. [2010] has also used a model of visual attention to improve gaze tracking systems in interactive 3D applications.


Sébastien Hillaire, Gaspard Breton, Nizar Ouarti, Rémi Cozot, Anatole Lécuyer, Using a Visual Attention Model to Improve Gaze Tracking Systems in Interactive 3D Applications Accepted for publication in Computer Graphics Forum, 2010
Ekman et al. [2008] developed a game which only use the eyes as input. They make use of gaze, blinking, and pupil size to affect game state. They report that the pupil size is affected by physical activation, strong emotional experiences, and cognitive effort. They discuss limitations of using pupil size as an input modality. Pupil size is increased when the player is engaged in the game interaction. They use this to model magic powers. Gaze blinking was also used to disappear into a puff of smoke.


Previous work summary

- Studies have looked at various aspects
  - Different input devices, game genres, duration of play, eye trackers, game engines
- Difficult to assess the significance of the results
  - Performance vs. immersion
- More research is needed…

So to summarise some of the previous work...

The studies have looked at various aspects, for example...

  Different input devices, game genres, duration of play, eye tracking systems, game engines

Difficult to assess the significance of the results using so many different setups...

  Sometimes performance was better with one technique and sometimes not, also the level of immersion was changed based on the type input device

I believe more research is needed in this area...
I will now present two case studies in which gaze and voice were used in combination to control virtual characters and their behaviour. The first game is described further in Wilcox et al. [2008] and the second in O’Donovan [2009] and O’Donovan et al. [2009]. The first case study “Revenge of the Killer Penguins” is the result of a group project developed at the University of Bristol by Tom Wilcox, Mike Evans, Chris Pearce, and Mike Pollard. The second case study “Rabbit Run” is the result of my M.Sc. Student Jonathan O’Donovan from the Interactive Entertainment Technology programme at Trinity College Dublin.


Gaze and voice in gaming

- Can the Midas touch problem be overcome by combining voice recognition with gaze?
  - Hands-free method of game interaction
  - Positive implications for disabled users
  - Novel game features

Our two case studies use gaze and voice to interact with games. We have explored if the “Midas touch” problem can be overcome by combining voice recognition with gaze instead of using a mouse click or dwell time. This gives us a novel and hands-free method of game interaction. Given that gaze and voice are entirely hands-free it can also have positive implications for disabled users for whom traditional techniques might not be possible. I will also talk a little bit about some novel game features which we have developed.
Our first case study, *The Revenge of the Killer Penguins*, is a small third person adventure puzzle game using a combination of non intrusive eye tracking technology and voice recognition for novel game features [Wilcox et al. 2008]. The first game consists of one main 3rd person perspective adventure puzzle game and two first person sub-games, a catapult challenge and a staring competition, which use the eye tracker functionality in contrasting ways. I will talk a little bit more about the different features available in these games. It also contains a novel game feature for displaying in game text called SmartText which I will describe later.

The game framework was written in C++ and implemented using Ogre3D. This framework included driver objects for the eye tracker and speech input, which involved implementing the TET Components API provided by the Tobii Eye Tracker SDK and the Microsoft Speech SDK respectively. This project used the portable Tobii x50 system which was positioned in front of a monitor.
Inspiration

- Monkey Island 1, The Secret of Monkey Island, Lucasarts
  - Point and click style games suitable
  - Mouse click for object interaction and walking longer distances in the environment

As inspiration for the game we chose the bright and colourful style of the Monkey Island series by Lucasarts. These games are point and click style type games which we thought would suit eye tracking and voice commands well. For example in these games the player uses the mouse to point at actions to do and also the objects to interact with. Mouse clicks are also used to walk longer distances in the game environment.
To play the catapult game, the user can simply look at the target and voluntary blink to fire a projectile towards the object under the crosshair. Voice commands could be used to initiate the power bar increments and launch the projectile as an alternative option. In the staring competition on the right the player was challenged to stare at another character without blinking for the longest duration possible. This sub game also had different difficult levels.
Another feature is SmartText in which dialogue and text are displayed on the screen until it has been detected that the player has read it. Other features include detecting voluntary blinks and winks to be used as controls. The idea behind SmartText is that people read at a massive range of different speeds which makes it a problem attaching a time limit to how long a text should be displayed for. The benefits of SmartText are that a player can look away from the screen, be a slow reader or fast and impatient and the text will display long enough for them to read it and no longer, guaranteeing they get the information.
There are two different modes of control in the main game. The user can select objects by looking at them and perform **look**, **pickup**, **walk**, **speak**, **use** and other commands by vocalizing the respective words. Look would give you more detail about the object. Pickup would put the object in the inventory. Walk would allow you to go to a new location easily. Speak allow you to talk to other characters. And the use command could be used to use one object with another, for example catching a fish with the pot. For example, the figure shows how the behaviour of the virtual character can be controlled by looking at a pot (highlighted in yellow) while issuing a voice command. Alternatively, they player can also perform each command by blinking and winking at objects. The ‘main menu’ or ‘inventory’ overlays will also be displayed upon commands. Other features include detecting voluntary blinks and winks to be used as controls.
A problem identified in testing was that users would look towards a new target whilst attempting to apply a vocal command on a previously selected object. By implementing a lag in selecting items, we were able to solve this issue without impeding the user’s natural motions by providing the user with enough time to voice any command. Our experience with the system was that sometimes the calibration could become poor resulting in users not being able to select objects easily. It was possible to snap the cursor to objects nearby to help mitigate this problem. By designing levels with careful attention to camera positions, size and the locations of selectable objects we were also able to mitigate this problem. Building scenes for the purposes of gaze selection requires careful positioning and making selectable objects larger and more spaced apart. A smoothing filter was also used to compensate for the fact that the eye is never static resulting in an unstable cursor. Overall it was found that when designing a point-and-click style game for use with the eye tracker, scenes and selectable elements must be laid out and scaled for optimal usage; buttons must be bigger and targets must be carefully positioned in order to produce an effective system.
This case study is the result of the M.Sc. (Interactive Entertainment Technology) undertaken by my student Jonathan O'Donovan. The purpose of the project was to create a novel game evaluation framework that could be controlled by input from gaze and voice as well as mouse and keyboard. This framework was evaluated both using quantitative measures and subjective responses from participant user trials. This is the first evaluation study performed comparing gaze and voice against mouse and keyboard input [O'Donovan et al. 2009]. As far as we know this is the first study performed comparing gaze and voice against mouse and keyboard input. The main result indicates that, although game performance was significantly worse, participants reported a higher level of immersion when playing using gaze and voice.

The game needed to be relatively simple so users could understand it quickly and finish within a reasonable time frame. We also wanted it to be challenging enough to bare relevance to a real game. It needed to include common gaming tasks, such as navigation and object selection. The premise decided upon was Rabbit Run. The player is trapped in a rabbit warren, inhabited by evil rabbits, from which you must escape. The main objective is to navigate through the warren maze and find the exit in the shortest time possible. To earn extra points coins distributed throughout the maze could be collected. In order to pass by the evil rabbits they needed to be shot. Once the exit was reached the game ended.
A map would also be provided (upon a key press or voice command depending on the input) in order to assist players finding their way through the maze. The game was also purpose built for user evaluation tests. The game also stored relevant game data, such as distance travelled (in squares), speed, coins collected, rabbits shot, etc. It was also decided to put a time limit on game so that each user trial would not go on for too long in case the user found the game too difficult.
The Tobii T60 eye tracker was used in this project. The eye tracker is integrated in the monitor making the setup easier than when using a portable system. The game evaluation framework was developed using Microsoft XNA. We think this is the first use of XNA with gaze based interaction. All relevant game data such as shots fired and coins collected were stored in XML format once a game trial was completed. Calibration and processing of gaze data from the Tobii T60 eye tracker was accomplished using the Tobii SDK. Upon start of the application all previous calibration data was cleared and a new calibration was initiated. Once calibrated real-time gaze data was relayed to the application. The gaze data includes the area being looked at on screen by both the left and right eyes and the distance of both eyes from the screen. This information was averaged to give a gaze point. The Microsoft Speech SDK was used to implement voice recognition. A few problems were encountered with the voice recognition in an early user test. More intuitive voice commands were not always recognised so instead more distinct voice commands were chosen. For example, instead of saying “Map” to bring up the game map, “Maze” had to be used instead. When selecting menu items “Select” also proved to be inconsistent so the command “Option” was used instead.
Game controls

- Mouse/Keyboard
  - Arrow keys used to move player
  - Change game view using mouse
  - Walking and run pace (shift)
- Gaze/Voice
  - Voice commands to move player (walk, run, stop)
  - “Gaze Camera” used to change game view

The framework allowed the game to be controlled by mouse/keyboard, gaze/voice, mouse/voice, and gaze/keyboard. Ultimately only mouse/keyboard and gaze/voice were tested in the user evaluation. A menu system was also provided controllable by mouse/keyboard and gaze/voice. Navigation for mouse/keyboard in the game environment was implemented using the arrow keys. Players could move forwards, backwards, left, or right relative to the direction the camera was pointing. This updated the position at a “walking” pace. If players wanted to increase their speed to a “run” pace they needed to hold down the shift key. Navigation with gaze/voice was achieved using three commands “Walk”, “Run” and “Stop”. When the “Walk” command was issued the camera proceeded to move, at a walking pace, in the direction the camera was facing until it encountered an obstacle, such as a wall, a rabbit, or the “Stop” command was issued. A continuous movement was required so that users would not have to keep repeating voice commands. The “Run” command worked in a similar way except at a faster pace. The Gaze Camera was used to change the game view – which I now will talk a little bit more about...
The cameras in the game played a dual role of showing the game play and acting as the player’s avatar. The position of the camera was checked for collisions with the surrounding game objects. The camera implementation differed between the gaze/voice and mouse/keyboard input types. The mouse/keyboard camera works like most FPS cameras where the target is at the centre of the screen and the mouse movement is used to shift the camera in a given direction. For targeting rabbits to shoot a cross hairs was rendered at the current position of the mouse, which is almost always at the centre of the screen. The gaze/voice camera builds upon the idea of Castellina and Corno [2008]. As described earlier they used semitransparent buttons to rotate the camera and move the avatar. As shown by Kenny et al. [2005] the vast majority of game time in a FPS games is spent looking in the near centre of the screen. Our implementation is based on this idea by using the outer rectangle of the screen to place the semitransparent gaze activated buttons. The inner rectangle could be left alone to allow normal game interaction. By placing the buttons in this outer area it was hoped that they would not interfere with game play. The buttons would also not be displayed unless activated, by looking in that area of the screen, to avoid distracting the player. The purpose of the buttons was to rotate the camera in a given direction. By looking at the left and right part of the screen the camera would shift left and right respectively. The original idea had been to place the buttons in such a way as to form an eight sided star.


Images from O’Donovan et al. (2009).
An early user test showed that the star gaze camera buttons were difficult to use. The camera rotated when the user did not want it to, causing it to spin in a disorientating manner. After feedback this was simplified to only use the left and right arrows. Original eight-sided star gaze camera is shown on the left. This was abandoned in favour of a simpler gaze camera with only left and right arrows shown on the right. The buttons act as a visual aid to the player indicating the direction the camera was shifting. For targeting rabbits a cross hairs (in green) was displayed using the current gaze point as screen coordinates. This separated the targeting from the camera view much in the same way as Jönsson [2005] did in her Half Life demo.

A map was provided in order to assist the players navigate the warren. The map was generated on the fly showing the places where the player had been in the warren. There was a subtle difference between its implementation using mouse/keyboard versus gaze/voice. For mouse/keyboard the map was only updated with the coordinates of where the player currently was. So if the player travelled to a new location that location would be revealed on the map. A novel game feature was used for gaze/voice input. In this case locations were revealed based on where the player had looked. So if the player looked at a particular location that area would be shown on the map without requiring the player to physically move to those positions. This novel game feature could be useful in puzzle based games where the player needs to memorise parts of the virtual environment or objects seen. Think of a game like Zelda in which the areas you have been to are displayed on a map, instead the areas you have “looked at” could be displayed.
A user evaluation was performed to evaluate how suitable gaze/voice is as a means of video game control compared to mouse/keyboard. Because the game could be controllable by mouse/keyboard as well as gaze/voice it facilitated direct comparisons between the two methods of interaction. This meant that each participant would need to play the game using both modes of interaction. Ten participants (8 men and 2 women, age range 22-33) with normal or corrected to normal vision were recruited for the user evaluation. Participants were sought by recruiting volunteering postgraduates and staff in the college. The participants had a variety of experience with computer games. One of the main objectives of the user study was to gather both quantitative measures and subjective comments. In addition to saved game data questionnaires were given to the participants to ascertain subjective data, such as how effective gaze/voice was perceived to be and how immersive or entertaining the experience was. As the eye tracker occasionally loses calibration during a trial, especially for participants wearing glasses, trials which clearly produced unreliable data were removed. In total eight users successfully completed their trials. Half the participants played the gaze/voice game first, while the other half played the mouse/keyboard first.
Equipment and setup

- Tobii binocular T60 eye tracker
  - 60cm away from the user
  - 17 inch TFT, 1280 x 1024 pixels
  - 0.5 degrees accuracy, 60Hz
- Stand-alone unit
  - No restraints on the user
- Sound proof video conference room
  - To negate any possible ill effects of background noise

The Tobii binocular T60 eye tracker was placed at a distance of 60 cm away from the user. The T60 is a portable stand-alone unit, which puts no restraints on the user. The freedom of head movement is 44 x 22 x 30 cm. The eye-tracker is built into a 17 inch TFT, 1280 x 1024 pixels monitor. The eye-tracker has an accuracy of 0.5 degrees and has a data rate of 60 Hz. This degree of accuracy corresponds to an error of about 0.5 cm between the measured and actual gaze point (at a 60 cm distance between the user and the screen). An adjustable chair was provided to allow participants to make themselves comfortable and place themselves at the correct distance from the monitor. The user trial took place in a sound proof video conferencing room. This was to avoid any interference background noise might have on voice recognition. The hardware setup consisted of a laptop running the application while connected to the T60 eye tracker via an Ethernet connection. A keyboard, a mouse, and a microphone headset were also connected to the host laptop to allow for keyboard, mouse and voice input. The lighting was dimmed throughout the experiment. A calibration was carried out for each participant, prior to each trial, to ensure that the collected data would be reliable.
Procedure

- Consent form and background questionnaire
  - Age, profession, gaming habits
- Instruction pamphlet
- Demo version
- Questionnaires

Upon arrival each participant were asked to fill in a consent form and answer a short background questionnaire about age, profession, and their gaming habits. After this the participants read a sheet of instructions on the procedure of the particular game they would play. It was decided that an instruction pamphlet would be used to inform all users in a consistent way. The participants were first asked to play a demo version of the game using the selected interaction method. They were allowed to play the demo version of the game for as long as they wanted. They were also encouraged to ask any questions they might have at this stage. Once satisfied with the controls participants were asked to complete the full user trial version of the game. Immediately after playing each trial, participants were asked to answer a second questionnaire. This questionnaire aimed to get the participants subjective opinions on that particular input. After the second and final game was played (and the post-trial questionnaire was completed) a third and final questionnaire was given to the participants. This final questionnaire aimed to compare the two different games played by the participant to gauge which one they preferred.
The game was designed to bear relevance to a real game, while being controlled enough to allow for analysis. Each game needed to be played twice in order to yield comparable results. It was decided that the exact same layout would be used in each trial by only swapping the start and exit points. The different start and end points would eliminate the possibility of skewed results from learning. The exact same number of coins and rabbits were used, distributed in the same positions in each setup.
The table shows the quantitative measures recorded while participants played both versions of the game. Values in bold indicate no significant difference. Mouse/Keyboard performs better on most although there was no significant difference for the shooting accuracy and the number of times the map reference was used...

<table>
<thead>
<tr>
<th>Measure</th>
<th>ME (mean)</th>
<th>GV (mean)</th>
<th>Related t-test (two tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance travelled (squares)</td>
<td>122.5</td>
<td>75.25</td>
<td>(T(df = 7) = 2.556, p &lt; 0.05)</td>
</tr>
<tr>
<td>Time (seconds)</td>
<td>228.47</td>
<td>418.86</td>
<td>(T(df = 7) = 4.683, p &lt; 0.01)</td>
</tr>
<tr>
<td>Speed (distance/time)</td>
<td>0.583</td>
<td>0.176</td>
<td>(T(df = 7) = 7.490, p &lt; 0.001)</td>
</tr>
<tr>
<td>Coins collected ($)</td>
<td>9.125</td>
<td>5.625</td>
<td>(T(df = 7) = 3.004, p &lt; 0.02)</td>
</tr>
<tr>
<td>Rabbits shot ($)</td>
<td>10</td>
<td>6.625</td>
<td>(T(df = 7) = 2.791, p &lt; 0.05)</td>
</tr>
<tr>
<td>Shooting accuracy (%)</td>
<td>72.6</td>
<td>64</td>
<td>(T(df = 7) = 1.034, p &gt; 0.2)</td>
</tr>
<tr>
<td>Map references ($)</td>
<td>10.1</td>
<td>9.8</td>
<td>(T(df = 7) = 0.08, p &gt; 0.1)</td>
</tr>
</tbody>
</table>
Here you can see the statistics for each subjective measure gathered using the questionnaires. Values in bold again indicate no significant differences. Again mouse/keyboard was perceived easier than gaze/voice on most measures. Although — no significant difference was given between the following measures: How easy it was to shoot rabbits, game enjoyment, and the usefulness of the normal map vs. the novel gaze map... Note the mean score is higher for gaze/voice than mouse/keyboard in the immersive ratings. The participants did report that gaze/voice was more immersive to use than mouse and keyboard.
Results

• The main result indicates that, although game performance was significantly worse, participants reported a higher level of immersion when playing using gaze and voice
  – Menu and game navigation, control and precision, coin collection, effort, difficulty
  – Shooting rabbits, map usefulness

The main results indicate that although game performance was significantly worse, participants reported a higher level of immersion when playing using gaze and voice... There was no statistically significant difference between the rankings in game enjoyment but maybe this was more due to the actual game than the interaction device... 75% of the participants selected gaze/voice as more enjoyable to play.
One of the biggest issues in the game was the collision response. It was decided to stop the motion of the player should they collide with a wall. This was worked ok for mouse keyboard since participants simply adjusted with the arrow keys. However for voice control this was awkward since participants had to: Adjust the gaze camera, then reissue a voice commands. So in future work we would like to improve the collision response to allow for smoother voice/gaze control... Although adapting open source games reduces the implementation time required, these games were originally created with mouse and keyboard in mind. So the adapted game is restricted to using gaze as a replacement for the mouse rather than an input device in its own right, with the gaze input acting only as a mouse emulator. When a game is developed from scratch it should be free of this restriction. However, it was possible to play the game hands-free, which is obviously significant for disabled users for which regular means of input (such as mouse/keyboards) may not be feasible.
Future directions

Gaze voice games:

- Our study suffered from *mouse emulation*
- Future work could exploit the use of gaze and voice in the real world to create novel and exciting ways to interact with video games
- The game experience itself could be measured using a game experience questionnaire

In terms of future research into gaze voice games one of the lessons from this study is that of *mouse emulation*. By this I mean that the gaze acts as a mouse/mouse emulator. When a game is developed from scratch it should be free of this restriction. However perhaps the goal of comparing gaze and voice directly against mouse and keyboard unwittingly manifested this idea on the project. Future work in this area could look at how we use gaze and voice in the real world – to create some novel/exciting ways to interact with video games. To help avoid this we would suggest that measurement of such a game would be conducted in terms of a *game experience questionnaire* rather than attempting to directly comparing mouse/keyboard VERSUS gaze/voice.
Future directions

Head tracking and gaze input:

- Investigate how Natural Point’s TrackIR system could be used in conjunction with gaze data to reduce camera motion issues

Some users experienced issues when rotating the camera using Virtual Keys. Ways around this could include: Blinking etc could be used to help the interface OR using Head Tracking device such as Natural Point’s TrackIR – head direction to control the look direction in-game. Head movements seemed to be a suitable interaction method for controlling the field of view but poor for aiming a weapon.
Future work

• Discussion of future research and directions

Future interaction with computer games is likely to be made using novel multimodal interaction techniques. We believe that alternative input modalities can be used in novel ways to enhance gameplay. I will now discuss some ideas for interesting future research.
Computer games have also started to provide more advanced crowd interaction, such as in Assassins Creed in which the crowd interacts with the player (based on its position for example). I believe this is an interesting area of research, combining gaze input to interact with crowds. For example you could allow virtual humans to interact with objects, navigate through environments, and interact with other characters (enemies) and crowds. Another area worth investigating is how the animation and AI of game characters could be adapted to react to the player’s gaze and voice. More socially realistic scenarios, based on social rules, could be created if game characters responded to the voice and gaze of the player. Say if you stare at someone for long enough time this would be deemed inappropriate.
Memory algorithms

- This idea could be extended further to equip virtual characters with novel memory algorithms constructed based on eye movement information.
  - Hints based on what has been seen / not seen
  - Abilities to remember which objects to use (inventory), where to go (map)

This idea could be extended further to equip virtual characters with novel memory algorithms constructed based on eye movement information. By gathering information about player eye movements we can store information on what the player has seen in a game. Maybe this idea could be used to equip virtual characters with novel memory algorithms.
Another interesting area is to figure out the state and interest of the player... What are they looking at? How can this information be used to create novel game features? Pupil size (Engaged, scared, stressed), Skin sensors
Explore possible effects of background noise on voice recognition. User evaluation held in sound proof room. No game sounds were used. There is still a great need for cheap, accurate, and novel solutions which could be used for mass-market products. These low cost solutions needs to be accurate, non-intrusive, and deal with different gaming setups and game genres. Alternative means of interaction in games are especially important for disabled users for whom traditional techniques, such as mouse and keyboard, may not be feasible. More work is needed in how to make computer games content more accessible to all users. There could also be situations in which voice recognition might not be an option.
Further resources

- COGAIN: Communication by Gaze Interaction
  - http://www.cogain.org/
- ETRA 2010: Eye Tracking Research and Applications
  - http://etra.cs.uta.fi/
- APGV 2010: Symposium on Applied Perception in Graphics and Visualisation
  - http://www.apgv.org/
- TAP: ACM Transactions on Applied Perception
  - http://tap.acm.org/

Here are some relevant conferences and journals for more related work.
Acknowledgements

- Jonathan O’Donovan, Tom Wilcox, Mike Evans, Chris Pearce, Nick Pollard, Mashhuda Glencross, Marina Bloj, Mary Whitton, Diego Gutierrez, Oscar Anson, Erik Reinhard, Francisco Lopez Luro, Carol O’Sullivan, Andrew Moss, and Trinity College (GV2)
- Jon Ward and Scott Hodgins, Acuity ETS Ltd. for the loan of the Tobii T60 eye tracker
- University of Bristol, Blekinge Institute of Technology
- Stephen Spencer and the support from Vetenskapsrådet
Thank you for attending the course: Gazing at Games: Using Eye Tracking to Control Virtual Characters. I would now be happy to answer any questions you might have. I would also like to encourage any discussion around novel multi-modal interaction techniques for games, in particular of using gaze as input to control games. If you would like any further information please do not hesitate to contact me on: Veronica Sundstedt, Blekinge Institute of Technology, Karlskrona, Sweden (email: veronica.sundstedt@bth.se). From next week the updated course material can also be found online at: http://www.scss.tcd.ie/Veronica.Sundstedt/gazingatgames2010.html

The full course notes also contain all the references that have been used throughout this talk and some additional ones.
ACM SIGGRAPH 2010 Course – Gazing at Games: Using Eye Tracking to Control Virtual Characters.

Instructor: Veronica Sundstedt, Blekinge Institute of Technology, Sweden & Trinity College Dublin, Ireland.

veronica.sundstedt@bth.se

Wednesday, 28 July, 3:45 PM – 5:15 PM, Room 406 AB, Los Angeles Convention Center.