Games Technology: Console Architectures, Game Engines and Invisible Interaction

Edmond Prakash and Jim Wood
University of Bedfordshire, United Kingdom
{Edmond.Prakash | Jim.Wood} @beds.ac.uk

Baihua Li, Michael Clarke, George Smith and Keith Yates
Manchester Metropolitan University, United Kingdom
{b.li | k.yates} @mmu.ac.uk; mclarke01@live.co.uk
georgesmith25@googlemail.com

***Abstract*— This presentation will look at three core developments in games technology. First we will look at the architectural foundations on which the consoles are built to deliver games performance. Millions of consoles are sold and the console performance is improving in parallel. Next we look at the cutting edge features available in game engines. Middleware software, namely game engines, help developers build games with rich features and also simultaneously harness the power of the game consoles to satisfy gamers. The third part focuses on Invisible Game Interaction. The Nintendo Wii games console was an instant success because of the Wiimote. Old and young alike embraced it. The Microsoft Kinect pushed the boundary even further, where the interaction device is slowly becoming invisible and the human body becomes the interface. Finally, we look at novel research developments that go beyond current game interaction devices.**

***Keywords*— Games Technology, Console Architectures, Game Engines and Invisible Interaction.**

I. INTRODUCTION

Games technology is going to be extremely important for the coming next five to ten years. Why? When we look at the future, we have to look a little at the past on how games technology has emerged. This will help us to figure out how certain trends are moving forward. We quickly find that there has been considerable development in game hardware architectures, game software architectures as well as in game interaction. The Graphical Processing Unit (GPU) has revolutionized a wide spectrum of computing activities from game consoles to the modern supercomputing systems. We explore the various advances in games technology because of the underlying GPUs.

II. GAMES TECHNOLOGY

For developing high performance games, developers rely on the GPUs that have become integral in the computer game consoles.

A. *Console Architectures*

First we look at the various game consoles and how the GPU helps in achieving the performance. Figure 1 shows the organization of the execution units (ALUs) in a CPU compared with the GPU. The GPU has tiny computing elements, but has several of them and facilitates single instruction that operates on multiple data in parallel. The GPU also is designed for high computation per memory access. The CPU on the other hand is flexible and can perform logical operations, branch-intensive operations and also random access, memory intensive operations.

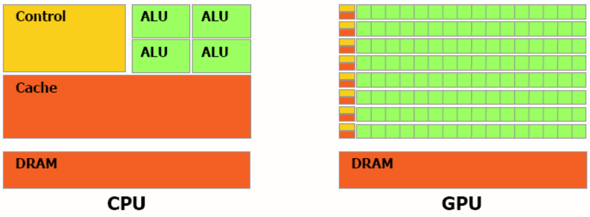

Fig 1. GPU devotes more registers for data processing.

There has been an exponential growth in the demand for games consoles. The development of consumer electronics has accelerated the adoption rate of games consoles. Figure 2 shows the total number of the current generation of consoles that have been sold until April 2011. The sheer number of consoles sold exceeds 50 million for every vendor has been the driving force for the success of the console and its underlying hardware. Next we explore the console hardware architecture.

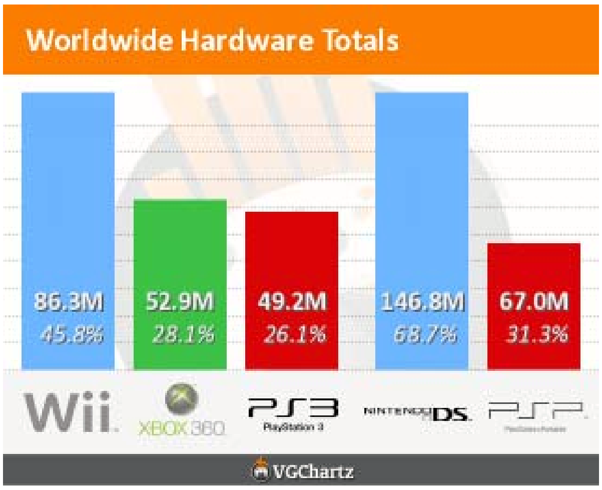

Fig 2. Game consoles as of April 2011 [22]

B. *The PS3 Console Architecture*

Figure 3 highlights the top-level block diagram of the Cell processor in the Playstation 3 (PS3). The Reality Synthesizer

Proc. of the 4th Annual International Conference on Computer Games, Multimedia and Allied Technology (CGAT 2011)
Editor: Dr. Edmond Prakash
Co-Editor (CCV 2011): Prof. San Murugesan

ISBN: 978-981-08-8644-8

(RSX) is the raw graphics power that boosts the performance of the PS3 console. The Cell processor has been a wide success too that accelerates the performance of the other aspects of game software (such as physics and AI).

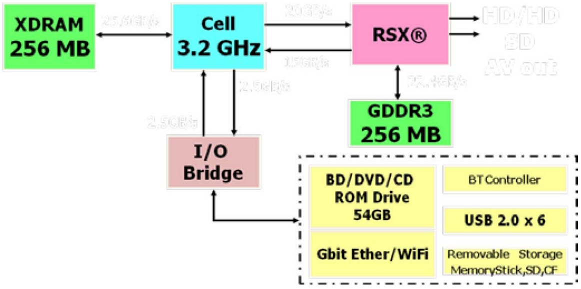

Fig 3. The PS3 architecture

The Power processing element (PPE) and the Synergistic Processing Elements (SPE) form the work horse of the Cell processor. The PPE is PowerPC based and has 2 hardware threads. The SPEs are dedicated vector processing units. The PPE and SPEs communicate over the Element Interface Bus (EIB). The block diagram of the Cell processor is shown in Figure 4.

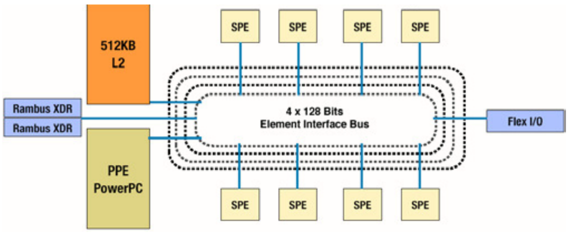

Fig 4. The Cell Processor in PS3

C. *The XBOX 360 S Console Architecture*

Over the past several years there has been a debate on unifying both the CPU and GPU in a single architecture.

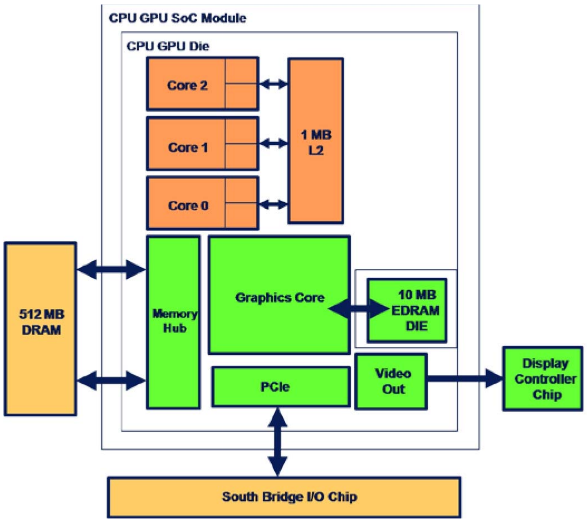

Fig 5. Xbox 360 S 250GB System

The unified processor can deliver better performance for the same number of transistors. The XBOX 360 S as shown in Figure 5 attempts to unify 3 CPU cores with the GPU in a single processor. Another benefit of a unified processor architecture is reduced power consumption.

III. Game Engines

A. *Game Engines: Commercial vs Custom Development*

There has been a great push to use commercial engines for rapid game development. The advantage of such commercial game engines is their cross platform nature. The game is developed on a single game engine and the engine is ported on to a variety of game platforms/consoles. Examples of such game engines are the id Tech5, Unreal UE3 and Cryengine3 engines. The components of a typical game engine software are as shown in Figure 6.

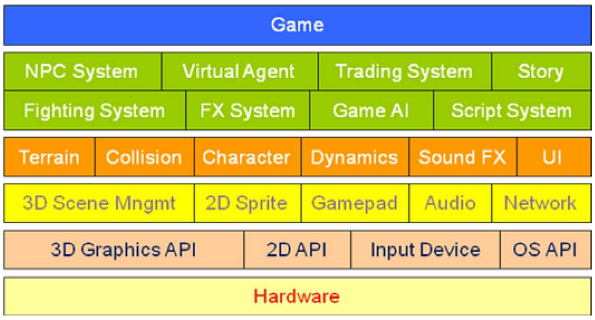

Fig 6. Game software system

B. *Game Engines: Environment Modelling*

To get best performance, it is necessary to develop a standalone custom engine that is free from other overheads. One such project is to create a library capable of displaying large, outdoor environments. To achieve this, C++, OpenGL, SDL and wxWidgets are used. One of the primary areas of focus has been the need to optimise terrain rendering so that a high framerate may be maintained. After researching numerous techniques, a highly customized geo-mipmapping algorithm is implemented to enable the user to explore large, outdoor areas in real-time. For videos of the current working prototype of Scenergy please see [21].

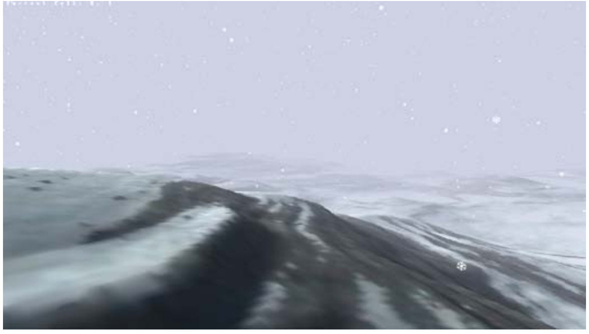

Fig 7. A mountainous scene using a particle weather system

The Scenergy system has

- three separate components which includes
  - A library to facilitate rendering of outdoor scenes.
  - An editor application allowing the user to import assets, customise an outdoor scene, and export the result to a bespoke file format.

  - A demo showing how the scene would look in a game or other application.
- the ability to display both high and low detail terrain patches (utilizing OpenGL's Vertex Buffer Objects) to ensure that environments can be rendered efficiently, even with large amounts of terrain.
- particle effects such as rain and snow with properties that can be edited in real-time.
- selective rendering of terrain patches and particle emitters based on proximity to the viewpoint.

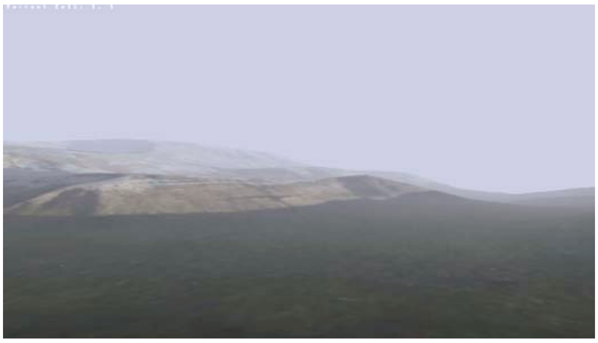

Fig 8. Scenergy displaying a mist-enshrouded highland setting

To further develop and enhance the Scenergy library, the plan is to implement dynamic skydomes that reflect a realistic day/night cycle. Other extensions planned are to implement foliage and to explore the various methods used to render clouds, including shaders and Perlin noise.

A really interesting future direction for Scenergy would be combining it with a rendering framework, giving us the opportunity to craft a game engine that could be used to render large-scale indoor and outdoor environments utilising the latest versions of OpenGL, including OpenGL ES 2.0 for mobile platforms.

C. *Game Engines: Realistic Rendering*

This section describes the implementation of Grendl a game rendering framework in C++, OpenGL 3.2 and GLSL. This has been motivated by the difficulties encountered when rendering large 3D scenes, and provides developers with access to state-of-the-art visual effects such as normal mapping and depth of field blur.

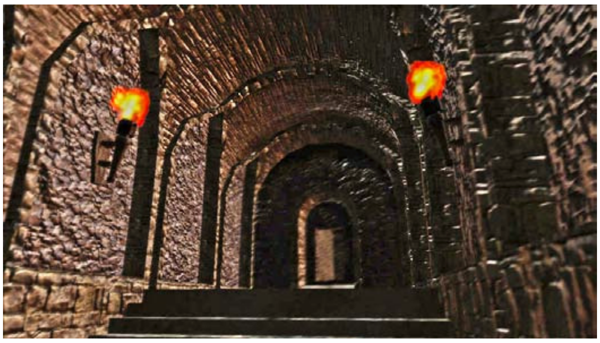

Fig 9. Grendl: Particle effects and depth of field blur

Performance and memory management have been key to the design of the framework, with rendering optimized through the use of a custom octree data structure. Resources such as models and textures are dynamically loaded/unloaded based on usage patterns in order to ensure that the overall memory footprint remains low.

The features implemented are:

- State-of-the-art shader effects (such as normal mapping and depth of field blur).
- Forward-compatible OpenGL 3.2 / GLSL 1.50 code utilizing Vertex Array Objects (VAOs) and Frame Buffer Object post-processing.
- A custom, frustum-culled octree data structure which vastly reduces the number of rendering calls per frame.
- A particle system capable of simulating dynamic effects such as fire and smoke.
- Singleton resource management and runtime profiling classes.
- Vector maths operations converted to Assembly for maximum performance.

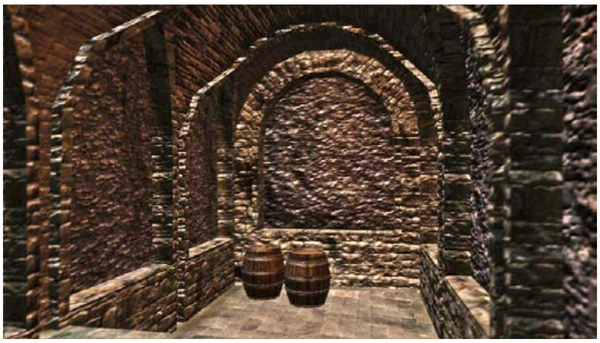

Fig 10. Grendl: GLSL normal mapped surfaces

There are a number of ways in which this project could be expanded upon - these include the implementation of additional shader effects such as deferred rendering and high dynamic range lighting. Benefit may also be gleaned from offloading particle calculations to the graphics card in order to further improve runtime efficiency.

As Grendl has been designed with indoor environments in mind, one particularly exciting possibility would be combining it the Scenergy library in order to allow for the seamless transition between indoor and outdoor scenes.

For a short demonstration video, please see Grendl [23].

IV. Invisible Interaction

A. *Kinect Interface*

Heralded as one of the consumer electronics revolution is the Kinect interface that has sold more than 10 million units in the first 4 months of its launch. It has also led to the sale of 10 million Kinect enabled game titles during the period. motion-controlled gaming has thus entered the main stage. Such mass production could effectively lower the price to $150 per unit as opposed to the first prototype cost of $30,000.

Developers on one hand build tools to support off-the couch motion capture to enable players to use their body to interact with the game. However, there are still traditional gamers, who would like to sit on the couch, hold their coffee mug on one hand and interact with the kinect using another hand. Games developers thus need to abstract the motion to higher level interpretation, what would otherwise could be captured using large body motion. Now due to this abstraction, simple hand movements could be effectively used to play a kinect enabled game.

Fig 11. Kinect interface (above) and body interaction (below)

B. *Attention-aware gameplay with eye tracking*

Eye tracking devices allow real-time tracking of eye movements and gaze. In the past decade, eye tracking tended to be a specialized technology used only for research in areas such as psychological and cognitive studies of human brain and visual system, attention patterns, usability evaluation, HCI, problem solving strategies and improving the learning experience in education.

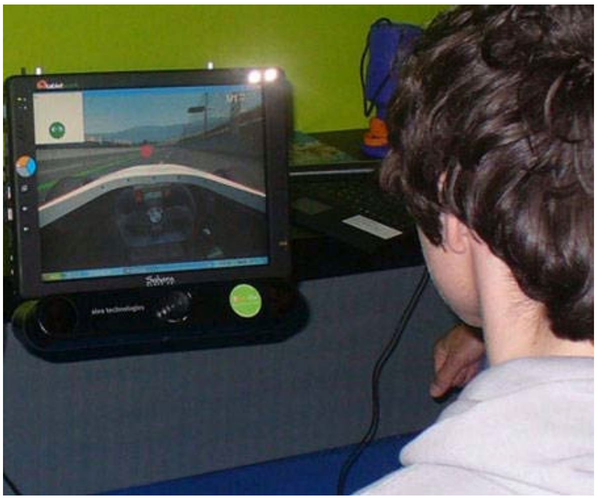

Fig 12. Gameplay using eye control [3]

Recently eye tracking products are becoming more user-friendly and are available in a variety of ranges. The most common systems are desktop "corneal reflection" trackers implemented using computer vision technology (e.g ToBii [1]). Higher sampling rate trackers for fast eye movement recording, wide-screen eye trackers for large stimulus display and head movement, and wearable lightweight systems for real-world activities are available from manufactures across the world. Eye tracking devices are becoming easy to use, less or non-intrusive, more accurate and cost affordable for ubiquitous computing scenarios. These have made it possible to use human eyes as a natural input modality to interact with virtual game worlds. Eye tracking technology provides a significant opportunity and value for people with disabilities who cannot manually operate screen-based computer applications. However, it is still an open problem as to how ordinary users could benefit and enhance their gameplay experience by using eye tracking [2].

Preliminary research has been undertaken in recent years to assess the usability of incorporating eye trackers in games of different genres and with different purposes of using eye trackers. It was found eye tracking input was particularly beneficial in paddle games [4]. In the game [4], the player's gaze was used to control the paddle to prevent the ball from escaping the playing area. It was reported that eye control has significantly improved their gameplay performance. The players felt more relaxed and subjectively immersive when using eye input than using mouse. This positive result can be explained as being due to observation and control being achieved simultaneously without much conscious effort in such attention-aware gameplay.

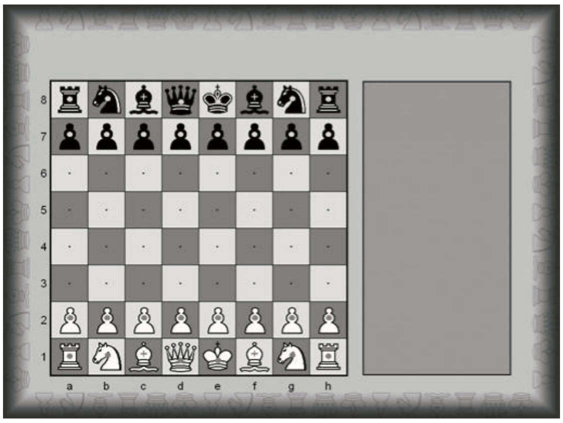

Fig 13. EyeChess: Virtual chessboard [5]

Board games, such as chess, card and puzzle games, in principle are other well suitable game genres for eye trackers. They are usually turn based and do not require intensive use of eyes. There could be immersive benefit for players if these games can be modified to ensure that all interactive game elements are big enough for eye selection and control. In the chess game EyeChess [5], the user interface has been completely rewritten in order to adapt it for the use of eye trackers. As shown in Fig.13, the centre of each square is indicated to ease focusing. In addition, the eye selected pieces for moving and the eye selected square where the piece is to move are highlighted on the chessboard. With these modifications, players could receive a considerable increase of the feedback from the game. This has helped to reduce the problem caused by dwell-time based mouse click emulation and has prevented eye drifting to neighbouring squares that are often caused due to saccade behaviour of the eyes.

The possibility of using eye trackers in FPS was discussed in several studies. In [6, 7], gaze was used for aiming the weapon at a moving target, as shown in Fig 14. Gaze has also been used to control orientation of an avatar [8] or field of view [7, 9]. It has been found that aiming with the gaze was faster and easier than aiming with conventional devices such as mice, keyboards or Xbox controllers. Results based on subjective evaluation in these studies indicate that eye control was perceived to be more fun and natural. It has remarkably enriched gameplay experience and the amount of immersion.

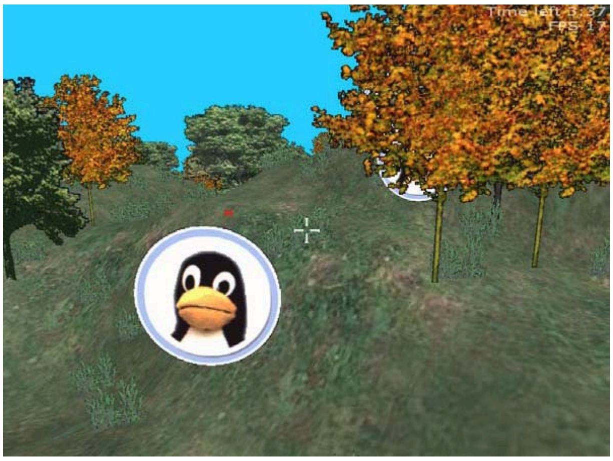

Fig 14. FPS testing game used in [6]: the big white crosshair in the middle of FOV is the standard FPS crosshair; the short red line above the penguin is controlled by player's gaze for aiming at a target penguin. The penguin has to be big enough for aiming by eye trackers.

A strong preference towards gaze control was also demonstrated in role playing games. Gaze has been used for pointing the desired destination where the character would walk to [8], or to select an object in the interactive environment. The benefit discovered in gameplay is not only because using gaze can largely reduce the amount of effort required to direct an avatar around a screen by pointing, but also because natural eye movement is a common means when communicating with other objects in real or virtual worlds.

It was also found that eye trackers did not always improve players' performance. For example in FPS games, eye trackers were competitive with gamepads [6], but mouse and keyboard control was more preferable and achieved higher scores than eye control [6, 8]. Although the performance improved after a significant amount of training on eye tracking devices were provided, the players felt that aiming and pointing with their eyes was difficult and less competitive than using manual control. This low performance is mainly because of "Midas Touch" effect as discussed below. Additionally, there might be potential health issues that could be caused by intensive use of eyes in the long term.

"Midas Touch" is an important problem affecting gameplay performance if only eye tracking is used as control input [10, 11]. When a player uses eye gaze to control the orientation of an avatar or perform pointing tasks, all the objects or directions the user is looking at is considered to be gaze selected, even when the user would just want to navigate the world or simply look at an object and not to consciously perform a task. Direct mapping of gaze for in-game control could lead to undesired results, such as constantly changing traveling direction of the avatar or FOV, unnecessary game events and interaction, and distraction of user's attention. Combining manual control with gaze mode has been reported as a possible solution [2, 6, 7, 8] and snap clutch methods were proposed in [11, 12] to eliminate Midas Touch problems.

Aiming, pointing and selecting with gaze at small objects or over long distances is also difficult due to the accuracy of eye trackers. In the literature, gaze is often used as a coarse control to position the cursor at a long distance roughly close to the object, and then a fine mouse control is involved to refine the location of selection. This method is effective as eye muscles are the fastest in the body.

Other studies on eye-tracking in gaming relevant areas have also been reported. EyeDraw is a system for drawing pictures with eye movements [13]. A computer game was designed to use an eye-tracking device for rehabilitation of eye activity [14]. Eye gaze that informs the user's psycho-perceptual attention has been used to improve rendering efficiency, where the player looks, where correspondingly a higher level of detail will be rendered [15, 16]. In the cognitive research on gameplay experience, a psycho-physiological logging system was implemented to demonstrate how certain game elements affect the player's attention [17]. A multi-mode gaze and sound control interface was proposed in [18]. In computer-based educational games, it was found that although mouse interaction was the easiest to interact during problem-solving, gaze-based interaction brought more subjective immersion [19]. Bulling et al. demonstrated how eye movement obtained from wearable eye trackers could be recognised and used for context-aware gaming [20].

Eye control seems promising for attention-aware gamming, however incorporating this technology in games is still in its infancy [2]. Eye trackers are not currently widely supported by the gaming industry. There is only a limited volume of research and experiments on eye-controlled gaming and most of them have just taken place in recent years. Eye-controlled games have been so far research prototypes. The methods used to incorporate eye tracking in games were usually based on replacing mouse/keyboard control by eye-mouse mode or simply modifying user interfaces. Games specifically developed with eye tracking function are rare. It is too early to assess the full features and benefits that we may gain from this new gaming input modality. To maximize the potential of eye control and obtain a truly novel gaming experience, it appears that completely new games and game genres incorporating such an attention-aware tool need to be developed. Mainstream gaming industries could be only likely to develop or adapt their game products to eye-control, if benefits for all users and mass-market were found.

C. *Examples of Invisible Interaction in Media Arts*

In general peripherals of all shapes and sizes and specifications seem to be commonplace, as enhancements to game play and experience [24]. Taking as an example pop and rock music games, this is no longer the preserve of 'toy' makers. In the near future the instrument makers may no longer make guitars solely for classic or punk rockers, heavy metal thunderers, but guitars for game supremos and their fans. [25]

The use of camera technology for invisible interaction or natural user interface (NUI) has been under focus recently, but this is not a new area. In fact as pointed out in a recent article for the HCI community [26], there is a long history of interchange and interface between these technology developments and experimental Media Arts.

We present some examples from Media Arts history and contemporary to give a situated critical perspective on how interfaces like the Kinect may be assimilated.

Known as one of the first proponents of 'artificial reality' Myron Kruger [27], developed 'Videoplace' a series of work starting in the 1970s that used first analogue video processing, and then computing and video recognition techniques to respond to participants movement or shadows in a projected virtual environment. This work is a clear forerunner of many current video based interaction techniques. Looking at the few sample images available [27] there is a clear similarity between this and some of the recent open Kinect demos [30]

David Rokeby's Very Nervous System (VNS) (1986-1990) is a hallmark for interactive sound installations that use whole body movement and interaction to control music and audio processing. Video cameras, image processors, computers, synthesizers and a sound system were used to create a space in which the movements of one's body create sound and/or music. [28] Rokeby continues to be an influential artist in computer based arts and has also developed VNS as a software, *softVNS*.

Text Rain (1999) by Camille Utterback and Romy Achituv, [29] is an interactive installation in which participants use their bodies, to lift and play with falling letters. *"Like rain or snow, the letters appears to land on participants' heads and arms. The letters respond to the participants' motions and can be caught, lifted, and then let fall again [29]."* It is another much cited example of video based interaction, and has a poetic quality about it.

These three historic examples have been hugely influential, with an ongoing interest in invisible interaction developing throughout the 90s and beyond. Now the Kinect sensor, and its relatively low cost has broadened this category, developed a whole genre [30] which has already shown examples both fun [31], and practical use [32].

In the past technical R & D had led from the labs to experimental artists and media labs, before emerging into the mainstream. Now mainstream devices are re-appropriated back into the hands of artists and then modified by researchers and developers in HCI labs to feedback to the community. This leads to a healthy state of cross-fertilization amongst platforms and a lot of opportunities for development of engaging new experiences.

V. Conclusion and Future Work

Development of games technology is taking place at a rapid pace. Areas of development have been in the hardware and software for games. Games hardware development in the form of consoles has been successful and we have reviewed the current generation console architectures in this paper. We have also presented the development in game engines. Custom game engine developments are crucial for achieving best performance.

A new challenge in game development is the interaction. Game players interact with newer ways than it was ever possible. The Wii revolutionized game interaction. The Kinect interface also has gained wider attention with interaction extending beyond games. We have also presented how media arts push the boundary of invisible interaction.

A comprehensive games technology review should also explore further to include the developments in networked and multiplayer games. We will explore this in the near future.

References

[1] ToBii Technology, http://www.tobii.com

[2] Isokoski P., Joos, M., Spakov, O., & Martin, B. (2009). Gaze Controlled Games. Universal Access in the Information Society 8(4). Springer.

[3] GameBase, http://www.gamebase.info/forum/topic/74

[4] M. Dorr, M. Böhme, T. Martinetz and E. Barth, Gaze beats mouse: A case study. In proceeding of COGAIN, pp16-19, 2007.

[5] O. Spakov, D. Miniotas. EyeChess: a tutorial for endgames with gaze-controlled pieces. In: Abstracts of ECEM13, p. 32, 2005.

[6] Isokoski, P., & Martin, B.. Eye Tracker Input in First Person Shooter Games. In Proc. of COGAIN 2006: Gazing into the Future, 78-81.

[7] E. Jönsson. If Looks Could Kill - An Evaluation of Eye Tracking in Computer Games. Master's Thesis, Royal Insittute of Technology, Stockholm, Sweden, 2005.

[8] Smith, J. D., & Graham, T. C. N. (2006). Use of Eye Movements for Video Game Control. In ACM Advancements in Computer Entertainment Technology, ACE 2006. ACM Press, New York, NY.

[9] Lee, Eui-Chul and Park, Kang-Ryoung. 3D First Person Shooting Game by Using Eye Gaze Tracking . The KIPS transactions. Part B, Volume b12, Issue 4, Korea Information Processing Society, ISSN 1598-284x, pp. 465-472, 2005.

[10] R. Jacob, 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.

[11] H. Istance, R Bates, A. Hyrskykari and S. Vickers. Snap Clutch, a Moded Approach to Solving the Midas Touch Problem. *Proceedings of the 2008 symposium on Eye tracking research & applications ETRA '08*, ACM Press, 2008.

[12] S. Vickers, H. Istance, A. Hyrskykari, N. Ali, and R. Bates, R. Keeping an Eye on the Game: Eye Gaze Interaction with Massively Multiplayer Online Games and Virtual Communities for Motor Impaired Users. Proceedings of International Conference on Disability, Virtual Reality and Associated Technologies; ICDVRAT 2008

[13] A. Hornof, A. Cavender, R. Hoselton. EyeDraw: A System for Drawing Pictures with Eye Movements. Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility, 2004.

[14] C. Lin, C. Huan, C. Chan, M. Yeh, C. Chiu. 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.

[15] B. Bazyluk, Eye Tracking in Virtual Environments: The Study of Possibilities and the Implementation of Gaze-point Dependent Depth of Field, Proc. Of Central European Seminar on Computer Graphics, CESCG, 2010. http://www.youtube.com/watch?v=lVaPzUGHhxY

[16] A. Kenny, H. Koesling, D. Delaney, S. McLoone and T. Ward. A preliminary investigation into eye gaze data in a first person shooter game, in Proceedings of European Conference on Modelling and Simulation, ECMS 2005.

[17] S. Stellmach. A Psychophysiological Logging System for a Digital Game Modification. Technical Bachelor's Report, 2007.

[18] T. Wilcox, M. Evans, C. Pearce, N. Pollard and V. Sundstedt. Gaze and voice based game interaction: the revenge of the killer penguins. In ACM SIGGRAPH, 2008.

[19] T. Gowases, R. Bednarik and M. Tukiainen. Gaze vs. Mouse in Games: The Effects on User Experience. In Proceedings of the International Conference on Computers in Education, ICCE 2008, pp. 773-777.

[20] A. Bulling, D. Roggen and G. Tröster. EyeMote - Towards Context-Aware Gaming Using Eye Movements Recorded from Wearable Electrooculography. In International Conference on Fun and Games, LNCS 5294. pp 33-45, 2008.

[21] Scenergy demo http://vimeo.com/20434383 http://vimeo.com/20435013

[22] Game Console Hardware total sales, Source: http://www.vgchartz.com/

[23] Grendl http://vimeo.com/20430199.

[24] Rock Band , 'Rock Band 3 Gear' http://www.rockband.com/gear

[25] Fender http://www.fender.com/promos/2010/rockband3/index.php

[26] Eva Hornecker. Let's Get Physical: The Role of Physicality in Tangible and Embodied Interactions. ACM interactions magazine vol 18, iss.2 (March/April), pp. 19-23 , 2011.

[27] M. Kruger, Videoplace, http://www.medienkunstnetz.de/works/videoplace/

[28] 'Very Nervous System' http://homepage.mac.com/davidrokeby/vns.html

[29] 'Text Rain'http://camilleutterback.com/projects/text-rain/

[30] Kinect Hacks website http://kinecthacks.net/

[31] Kinect Ultra http://code.google.com/p/kinect-ultra/

[32] FAAST project, http://projects.ict.usc.edu/mxr/faast/