

DECLARATION BY MARK CLAYPOOL, PH.D.

I, Mark Claypool, hereby declare the following to be true to the best of my knowledge, information, and belief.

1. I am over twenty-one (21) years of age, and I am fully competent to make this declaration.
2. I am a Professor of Computer Science and Interactive Media & Game Development at Worcester Polytechnic Institute (WPI). I have been a professor at WPI since 1997. I have published over 100 peer-reviewed conference and journal papers and written two books on computer games. I have also chaired several ACM conferences on networks, games and multimedia and served on the technical program committee for over 40 of them.
3. I served on the steering committee of NetGames Conferences from 2008 until the final conference in 2017. I attended the conferences in 2003, 2004, 2005, 2008, 2011, and 2013.
4. Due to my involvement with the NetGames Conference as a steering committee member and an attendee at the 2003 NetGames Conference and several after, I am knowledgeable of the publication of articles at the NetGames Conferences. Each conference made available and disseminated to all attendees of the NetGames Conferences several peer-reviewed research papers ("NetGames Conference Proceedings") that had been accepted for publication at the NetGames Conference. I have published several articles in the NetGames Conference Proceedings over the years, with my publications appearing in the NetGames Conference Proceedings in 2003, 2004, 2005, 2011, 2012 and 2014.
5. To my knowledge, each NetGames Conference Proceedings was published online in the ACM Digital Library.
6. I have been asked to certify the public availability date of "*Human Pacman: A Sensing-Based Mobile Entertainment System with Ubiquitous Computing and Tangible Interaction*," authored by Adrian David Cheok, Siew Wan Fong, Kok Hwee Goh, Xubo Yang, Wei Liu, and Farzam Farzbiz ("Human Pacman"), which is attached to this Declaration as Exhibit A. To my knowledge, Exhibit A is a true and correct copy of Human Pacman downloaded from the ACM Digital Library.
7. Human Pacman was published in the conference proceedings "NetGames '03: Proceedings of the 2nd Workshop on Network and System Support for Games" ("NetGames '03 Conference Proceedings").

8. I am an author on an article that was also selected for publication in the NetGames '03 Conference Proceedings entitled "*The Effect of Latency on User Performance in Warcraft III*," authored by Nathan Sheldon, Eric Girard, Seth Borg, Emmanuel Agu, and myself, which is attached to this Declaration as Exhibit B.
9. I attended the NetGames 2003 Conference which took place on May 22-23, 2003, in Redwood City, California at the Electronic Arts (EA) Headquarters.
10. The NetGames '03 Conference Proceedings, including Human Pacman, was disseminated to conference attendees on the first day of the NetGames '03 Conference (May 22, 2003). I received a printed copy of the NetGames '03 Conference Proceedings, including Human Pacman, when I attended the conference on May 22, 2003. I also received a flash drive containing an electronic version of the NetGames '03 Conference Proceedings, including Human Pacman, at that time.

If requested, I will appear for a deposition on this declaration. I declare under penalty of perjury under the laws of the United States of America that the foregoing statements, made on my knowledge, information and belief, are true and correct.

Executed on: March 20, 2025



Mark Claypool, Ph.D.

Exhibit A



Human Pacman: A Sensing-based Mobile Entertainment System with Ubiquitous Computing and Tangible Interaction

Adrian David Cheok, Siew Wan Fong, Kok Hwee Goh, Xubo Yang, Wei Liu, Farzam Farzbiz

National University of Singapore

Department of Electrical and Computer Engineering

adriancheok@nus.edu.sg / <http://mixedreality.nus.edu.sg>

ABSTRACT

Human Pacman is an interactive ubiquitous and mobile entertainment system that is built upon position and perspective sensing via Global Positioning System and inertia sensors; and tangible human-computer interfacing with the use of Bluetooth and capacitive sensors. Although these sensing-based subsystems are weaved into the fabric of the game and are therefore translucent to players, they are nevertheless the technical enabling forces behind Human Pacman. The game strives to bring the computer gaming experience to a new level of emotional and sensory gratification by embedding the natural physical world ubiquitously and seamlessly with a fantasy virtual playground. We have progressed from the old days of 2D arcade Pacman on screens, with incremental development, to the popular 3D game console Pacman, and the recent mobile online Pacman. With our novel Human Pacman, we have a physical role-playing computer fantasy together with real human-social and mobile-gaming that emphasizes on collaboration and competition between players in a wide outdoor physical area that allows natural wide-area human-physical movements. Pacmen and Ghosts are now real human players in the real world experiencing mixed computer graphics fantasy-reality provided by using wearable computers that are equipped with GPS and inertia sensors for players' position and perspective tracking. Virtual cookies and actual tangible physical objects with Bluetooth devices and capacitive sensors are incorporated into the game play to provide novel experiences of seamless transitions between real and virtual worlds. In short, we believe Human Pacman is pioneering a new form of gaming that is based on sensing technology and anchored on physicality, mobility, social interaction, and ubiquitous computing.

Keywords

Mobile entertainment, ubiquitous computing, tangible interaction, computer entertainment.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NetGame '03, May 22-23, 2003, Redwood City, California.
Copyright 2000 ACM 1-58113-734-6/03/0005 \$5.00.

1. INTRODUCTION

In recent years the world has seen the proliferation of highly portable devices, such as personal digital assistants (PDAs), laptops, and cellular telephones. At the same time, trends in computing environment development suggests that users are gradually freed from the constraints of stationary desktop computing with the explosive expansion in mobile computing and an increasing availability of cheap sensors. With this technological progress in mind, we have developed Human Pacman, which serves as a pioneer in the new genre of computer game that is based on real-world-physical, social, and wide area mobile-interactive entertainment. Concepts of context-awareness through location and body tracking, tangible interaction with objectification of virtual objects in game play, ubiquitous computing in the form of mobile and social gaming, and lastly physical gaming supported by the use of wearable computers are explored.

1.1 Context-Awareness in Outdoor Environment

Human Pacman allows players to be physically and immersively role-play the characters of the Pacmen and Ghosts as they move about freely in the real world over wide area outdoor spaces. There is a seamless merging of the fantasy computer digital world with the real physical world realized using location and body tracking technologies, namely the Global Positioning System (GPS), Dead-Reckoning module (DRM), and inertia sensors, to provide the necessary context-awareness of the system for the game play. The players who are role-playing Pacmen have to physically roam around the outdoor area and collect virtual cookies in the process. In addition, they have to find and collect 'Ingredients' that are actually Bluetooth embedded objects for the making of 'Super Cookies' which give them special Ghost-devouring ability. These physical objects are linked in real time to their virtual Pac-world representation, thus providing the player with a sense of presence and immersion with the virtual Pac-world. Another role in Human Pacman is known as the Helper. She is participating in the game remotely via a computer terminal over Wireless LAN. In other words, the Helper role can be played by anyone connected to the game's Wireless LAN network, or an online player who is physically anywhere on earth and is joining the game via Internet.

Researchers at Olivetti Research Ltd. (ORL) and Xerox PARC Laboratory pioneered the context-aware computing area [Want92, Schilit93] with the introduction of Active Badge System and PARCTab. However, these systems were expensive with the extensive use of infrared transceivers, and were limited in scope as their applications were confined to an indoor room. Therefore with the introduction of GPS and emergence of cheap but accurate sensors, more context-aware systems for outdoor applications were built. One notable system was the Georgia Tech Cyberguide project [Long96] where mobile context-aware tour guide prototypes were made to provide information to a tourist based on her position and orientation. Similarly, at the University of Canterbury, some context-aware fieldwork tools have been developed: an archeological assistant tool [Ryan97], a giraffe observation tool [Pascoe98], and a rhino identification tool [Pascoe99] to enable the users to make location dependent notes using a PalmPilot as terminal and GPS for positioning. Unlike Human Pacman that uses Augmented reality techniques as its main computer human interface, these systems have only primitive 2D maps and text presented on palmtops.

Another tourist assistant called Smart Sight was developed at the Carnegie Mellon University [Yang99], which was able to translate from and to local language, handle queries posed and answer in spoken language, and aid navigational around the campus with the use of wearable computers. Nevertheless since laptops were used as part of the mobile computer system, their sheer weight and bulkiness have greatly reduced user's mobility and comfort of use. In Human Pacman, players are provided with custom-built wearable computers that are designed, built, and developed in our lab especially for this application.

The use of GPS and outdoor physical area for computer gaming is pioneered by ARQuake [Thomas02]. This game is an AR extension of the original desktop Quake game of player shooting virtual monster (in this case the monsters are presented in physical world using AR techniques) in first person perspective. However it is a single player game with practically no social interaction.

1.2 Tangible Interaction at Game Play

Even though Graphical User Interface (GUI) has been and still is the dominant paradigm for interactions with computers, we are increasingly encountering computation that moves beyond the traditional confines of the desk and attempts to incorporate itself more richly into our daily experience of the physical and social world. Work on physical interaction started to appear in literatures in the early 90s with the introduction of Computer-Augmented Environments [Acm93] that have visioned the merging of electronic systems into the physical world instead of attempting to replace them as in virtual reality environments.

Over the years, a number of projects have explored this new paradigm of interaction termed tangible computing. Early attempts include Bishop's Marble Answering Machine [Smith95] that has made a compelling demonstration of passive marbles as "containers" for voice messages; "Bricks" by Fitzmaurice [Fitzmaurice95] that are essentially new input devices that can be tightly coupled to virtual objects for manipulation or for expressing action (e.g., to set parameters or for initiating processes); "Tangible Bits" and "mediaBlocks" from MIT media lab [Ishii97] that allows users to "grasp & manipulate" bits in the

center of their attention by coupling the bits with everyday physical objects and architectural surfaces, and "contain, transport, & manipulate" online media using small, electronically tagged wooden blocks that serve as physical icons ("phicons") respectively. Nevertheless in all of these implementations of tangible computing, computer interaction remains passive with human initiates communication with the tangible object, and confined only between virtual objects and humans. However, in Human Pacman, with the use of embedded Bluetooth devices for tangible interactions, we explore the active communication between computers and human players instantiated by Bluetooth devices as the players walk within the range of objects which Bluetooth devices are attached. Also, the use capacitive sensors on objects as well as wearable computers allow the game play to include graspable interaction between humans and computers, and between human players themselves. Thus there is tangible touch interaction between humans which has real time affects on the virtual computer world.

1.3 Ubicomp elements in connected mobility for social and interactive gaming

Björk suggested ubiquitous games [Björk02] as a new form of gaming. Coupling that with the development in Wireless LAN technology and mobile computing using wearable computers, we have combined connected mobility with elements in ubicomp (i.e. natural interfaces, context-awareness, and automated capture and access) for realizing social gaming components in Human Pacman.

Several pioneering works have been developed on ubiquitous and social gaming. Multi-players mobile gaming is demonstrated in 'Pirates!' [Björk01]. 'Pirates!' implements the game on PDAs with proximity sensing technology to incorporate a player's contextual information (such as physical co-location of players and objects in the world) into the game context as important elements of the game mechanics. However, visual and sound effects of game play are limited by relatively low computing power of PDAs. Augmented Reality and Virtual Reality cannot be implemented; therefore immersive experience is rather limited due to the flat 2-D display used on PDAs. The E3 project [Mandryk01] examines the essential elements of free play, and multi-user social interaction. It focuses on human-to-physical interaction and human-to-human interaction. However it does not explore large-scale configuration where users walk around.

Today's mainstream entertainment revolves around interactivity. Gone are the days when people were satisfied with passive form of entertainment as provided by television and cinema. People today enjoy entertainment they can control, and experience in which they are full involved [IDSA01]. In fact not only do they want such entertainment; people want to enjoy it together with family and friends. As shown in a certain survey [IDSA02] one of the top reasons why game players like to play games is that game playing is a social activity people can enjoy with family and friends. With advancement in networking technology, social gaming has gained popularity since the introduction of networked games [IDSA02]. Networked games overcame the barrier of distance, enabling real people to play against each other over large areas. After all there is no opponent like a live opponent since no computer model will rival the richness of human interaction [Dodsworth98a]. According to a recent study by Nezlek [Nezlek02], enjoyable and responsive

interactions increase life satisfaction scores among people. Nevertheless, even in networked computer games, social interaction between players is limited since natural interactions such as gestures, body language and movement, gaze, and physical awareness are lost. Thus, by bringing players in physical proximity for interaction, Human Pacman brings networked social computer gaming to a novel higher level because humans enjoy being physically together, and socially interacting with each other (as mentioned in the following section with the work from [Bowlby83]). Essentially, Human Pacman brings the exciting interactive aspects of networked gaming, and merges it with the real physical world, allowing social translucence between humans players in both the real and virtual worlds simultaneously.

1.4 Physical Gaming with Wearable Computers

Human Pacman has its roots in serious research about human's interaction with their world. Human, as social creatures find physical interaction, touch, and human-to-human presence essential for the enjoyment of life [Bowlby83]. In pre-computer age, games were designed and played out in the physical world with the use of real world properties, such as physical objects, our sense of space, and spatial relations. Nowadays computer games focus the user's attention mainly on the computer screen or 2D/3D virtual environment, therefore constraining physical interactions. However, there seems to be a growing interest in physical gaming and entertainment. Commercial arcade games have recently seen a growing trend of games that require human physical movement as part of interaction. For example, dancing games such as Dance Dance Revolution and ParaParaParadise [Konami01] are based on players dancing in time with a musical dance tune and moving graphical objects, as seen in Figure 1.



Figure 1. A player dancing to the music in ParaParaParadise arcade game

However these systems still force the person to stand in more or less the same spot, and focus on a computer screen in front of them. Nevertheless, our philosophy is similar as one of the goals for Human Pacman to bring physical gaming into computer

entertainment. The game of Pacman has come a long way from the 2D arcade version to the current popular game console and computer version of 3D Pacman. Now with Human Pacman, we have a physical role-playing computer fantasy that takes place in a wide outdoor physical area, and promotes natural wide-area human-physical movements.

1.5 Overview

In this article, we present a novel game system, Human Pacman, which anchors upon sensing-based subsystems that are the GPS merged with Dead Reckoning module and inertia sensor; and Bluetooth device and capacitive sensor. Firstly we describe the roles of the various sensors and tracking devices that are made used of in Human Pacman. Then we proceed to detail gaming experiences involved by clarifying the actual game play designed. More importantly in the section that follows, the details of the sensing-based subsystems are analyzed in the context of addressing various Human Computer Interaction issues. Lastly we conclude with our reflections on the future impacts of the system on everyday life.

2. ROLE OF SENSORS AND TRACKING DEVICES

Numerous sensors and tracking devices that are used in the system – dead reckoning module (DRM) with GPS and digital compass, inertia sensor, Bluetooth device, and capacitive sensor – are vital to the manifestation of Human Pacman. First of all, the DRM provides the all-important mobile and physical gaming infrastructure for indoor and outdoor location tracking of players. It is an electronic module comprising of a 12-channel Global Positioning System (GPS) receiver, digital compass, pedometer, and altimeter from Point Research Corporation [Pointgrey02]. The DRM measures the displacement of the user from an initialization point by measuring the direction (with data obtained from the compass), and distance traveled (using accelerometer data) with each footstep taken. Although the DRM is a self-contained navigation unit, when GPS position data is available, it can be used to correct both the distance and direction calculations with the help of a Kalman filter algorithm. Besides, in conjunction with the step detection logic (pedometer), the module can detect running, sideways, and backwards walking which is necessary for our application.

The real-time position of each mobile user is sent periodically to the server through wireless LAN. Upon receiving the position data, the server sends an update to each wearable computer the position of other mobile players, as well as the positions of all “non-eaten” plain cookies.

The position of a player with respect to another player or location in the physical world is calculated as follows. Taking two physical locations as *point1* and *point2*, the distance *d* between *user1* and *user2* in radian can be calculated by:

$$d = \cos^{-1} \{ \sin(lat1) \sin(lat2) + \cos(lat1) \cos(lat2) \cos(lon1 - lon2) \}$$

Where *lon1*, *lat1*, *lon2* and *lat2* are the longitude and latitude of *user1* and *user2*, respectively which all are in radian. To compute the distance in kilometers instead of radians, *d* is multiplied by the radius of earth, which is estimated at 6371.0km.

The course *c12* from *user1* to *user2* will be:

IF $\sin(\text{lon } 2 - \text{lon } 1) < 0$

$$c12 = \cos^{-1} \left\{ \frac{\sin(\text{lat } 2) - \sin(\text{lat } 1) \cos(d)}{\sin(d) \cos(\text{lat } 1)} \right\}$$

ELSE

$$c12 = 2\pi - \cos^{-1} \left\{ \frac{\sin(\text{lat } 2) - \sin(\text{lat } 1) \cos(d)}{\sin(d) \cos(\text{lat } 1)} \right\}$$

The above $c12$ is valid when the *user1* is facing to the North Pole. Otherwise the yaw of the North Pole for the *user1* must be subtracted from $c12$.

Since augmented reality is implemented as an interface feature between the players and their wearable computers, head tracking for the calculation of view perspective of each player is important. We use Inertiacube2 from InterSense [InterSense01] for this purpose. It is an inertial three degree-of-freedom orientation tracking system that gives yaw, pitch and roll of the object that it is attached on. By placing the sensor on the cap of each player, her head movement and orientation is tracked to high accuracy.

Consequently using the player's position and head movement, the wearable computer calculates the relative position of each plain cookie within the view of the camera, and superimposes a 3-D virtual cookie image of a proportionate size on corresponding position on the video stream. The position of other mobile players obtained from the server is used to update the 2-D virtual map in the HMD view of the player. Similarly the point-of-view of each mobile player is sent to the computer of a helper through the server, and the corresponding view in the virtual Pac-World can be displayed.

As for the Bluetooth device, it is embedded in a physical object (as the sugar jar in Figure 2) which seamlessly correlates to a particular virtual ingredient in the game play. When a player is within range of the Bluetooth object (about a distance of 10 meters), communication takes place between the wearable computer and the Bluetooth device. The wearable computer sends the unique address of the Bluetooth device to the server, upon receiving it, the server will then decide if the player is eligible to collect the virtual ingredient that is associated with the physical Bluetooth object. If the player is not eligible (for example she has already collected the ingredient), she will not be alerted to the object. Otherwise, an alert message will be shown on the player's HMD display.

Most importantly, when the player collects the object in the real world, her action can be seen in real time in the virtual Pac-world. The Bluetooth-embedded object is a direct link between the real world and fantasy world. When Pacman collects the physical sugar object in the real world, it is shown as a sugar collection seamlessly in the corresponding position in the virtual Pac-world. This highlights how the object is directly embedded in both the physical and virtual world. The object (physical sugar can) has a direct link and representation with the virtual Pac-World (virtual sugar can). Thus, this further increases the sense of presence and embodiment of the virtual world in the real world for the players.

The player has to hunt for the Bluetooth embedded object upon receiving the alert message in the surrounding physical area and thus adding elements of fun and adventure to the game play.

Once the object is found, she has to physically hold the object in her hands in order to collect the virtual ingredient associated with it. This is achieved by the use of charge transfer sensing on the object that detects the player's touch. We have designed this capacitive sensor using QT161 IC chip from Quantum Research Group [Quantum02]. Once haptic data is collected by the sensor, the Bluetooth device embedded on the object will send an alert message to the wearable computer, which will in turn be relayed to the server. The server performs legitimacy check on the player's action, and then proceeds to updating its database as well as informing the wearable computer.



Figure 2. Bluetooth-Embedded object

Capacitive sensing is also important in another part of the game play where a player devours her enemy by tapping on the shoulder of the latter. When player A taps on the shoulder on player B where the sensor is attached, the sensor is activated. Then the wearable computer on player B sends this information to the server via Wireless LAN communication. The server, after considering the positional and status information of player A and B, will decide if it is legal for player A to devour player B. If the devouring action is legitimate, both wearable computers on player A and B are notified and the databases on the server are updated.

Devouring involves tangible physical touching contact between two players. As close proximity is involved, other forms of human interaction come into play. The act of devouring makes the game more tangible and fun by involving more types of natural physical movement. As in when a Pacman player is the prey, her agility determines the "life-and-death" of her virtual Pacman role. Not only tangibility is brought to play in this fantasy world, but also other human perceptions and instincts. Thus this computer game provides the benefits of natural wide area free bodily movements as part of humanistic interaction between each person.

3. BRIEF SYSTEM DESCRIPTIONS AND GAME PLAY

Human Pacman features a centralized architecture that is made up of four main entities, namely the central server, wearable computers, Sony VAIO laptops, and Bluetooth embedded objects. An overview of the system is shown in Figure 3.

The Wireless LAN serves as a communication highway between the wearable computers, the helper computers (Sony VAIOs laptops), and the server desktop computer. The underlying

program is built on a client/server model with wearable computers and helper laptops as clients, and the desktop computer as a central server. Physical location and players' status updates are done between the client wearable computers and the server on a frequent and regular basis. The server maintains up-to-the-minute players' information (location, status etc.), and presides over any communication between Bluetooth objects and the wearable computers.

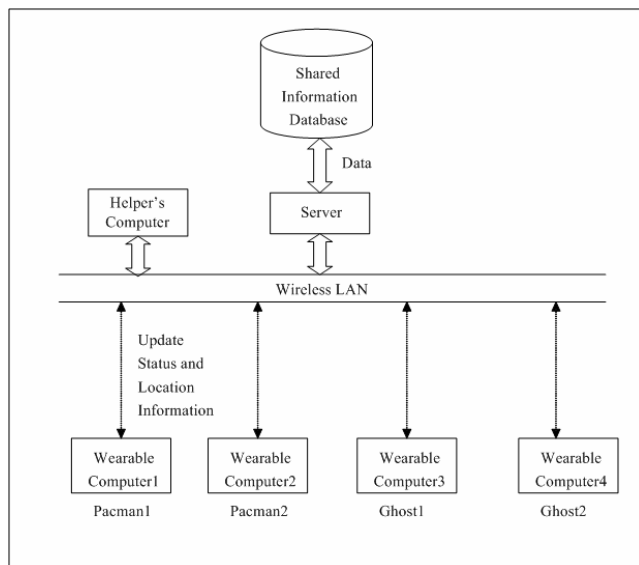


Figure 3. Complete system overview of Human Pacman

Since the wearable computers were designed and developed in the lab, we proceed to discuss its detail configuration as seen in Figure 4. The main components are Transmeta Single Board Computer (PC104 plus compliant motherboard running on Crusoe processor), Twiddler2 (handheld keyboard and mouse), Cy-Visor Head Mounted Display (video see-through HMD) with FireWire camera attached, two Sony F960 InfoLithium batteries, InertiaCube2 (inertia sensor), DRMI module (GPS and Dead-Reckoning device from Point Research Corporation), and JVC Bluetooth device.

With the software architecture mentioned as the backbone of the game engine and the hardware as enabling tools, we proceed to describe the game play of Human Pacman. The main concepts of the game are first given in terms of team collaboration, ultimate game objectives, and essential nature of the game's playground named Pac-World. Then, we move on to present the details on the players' roles as Pacman, Ghost, and Helper respectively. We end this section by giving examples on several actual game play situations.

3.1 Main concepts: team collaboration, ultimate game objectives and the nature of Pac-World

The players are assigned to two opposing teams, namely the Pacman team and the Ghost team. The former consists of two Pacmen and two Helpers; correspondingly, the latter consists of two Ghosts and two Helpers. Each Pacman/Ghost is in coalition with one Helper, promoting collaboration and interaction between the users. Since a Helper player is essentially participating in the game remotely using a computer terminal over Wireless LAN,

Human Pacman can effectively be expanded to include online players anywhere on earth who can view and collaborate via the Internet with real human Pacmen and Ghosts who are immersed in the physical playground.



Figure 4. Detail configuration of wearable computer

Ever since its introduction by Namco to Japanese arcade fans in 1979, Pacman has gone through numerous stages of development yet the ultimate goal of the game remains fundamentally unchanged. We have designed Human Pacman to be in close resemblance to the original Pacman in terms of game objectives so that the players' learning curves are very much leveled to the point that they can pick up the game in no time and enjoy the associated familiarity. Basically the goal of the Pacman team is to collect all virtual plain cookies and hidden ingredients in Pac-World while avoiding the Ghosts. On the other hand, the aim of the Ghost team is to devour all Pacmen in the Pac-World. To add to the excitement of game play, after 'eating' certain special ingredients, a Pacman gains Ghost-devouring capability and henceforth can attack her enemy head on for a limited period of time.

Pac-World is a fantasy world existing dualistically in both Augmented Reality (AR) and Virtual Reality (VR) mode. Pacmen and Ghosts, who are walking around in the real world with their networked wearable computers and head mounted displays (HMD), are allowed to switch between the two viewing mode. Helpers, on the other hand, can only view in VR mode since they are stationed in front of networked computers. Most importantly there is a direct and real time link between the wide-area physical world and the virtual Pac-World at all times, thus providing the users with a ubiquitous and seamless merging of the fantasy digital world and the realistic physical world. As seen in Figure 5 where the 2D map of the selected game play area in our university campus and the 3D map of Pac-World are shown side-by-side, we have converted the real world to a fantasy virtual world while ingrained the latter with direct physical correspondences.

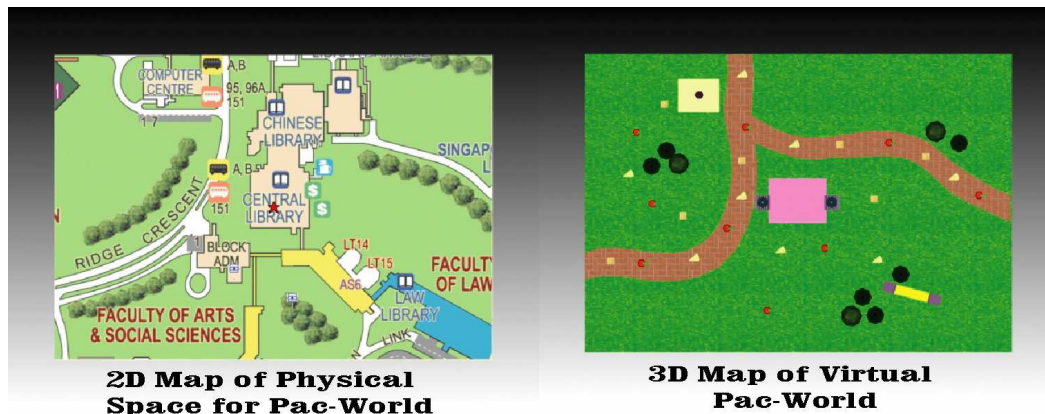


Figure 5. 2D map of game play area and its corresponding 3D map of Pac-world



Figure 6. Correspondence between the physical world and the virtual Pac-world

3.2 Pacman, Ghost, and Helper

Pacman has to physically move around the game area attempting to collect all virtual cookies. When she is in AR mode, through the HMD, she sees the real world being overlaid with virtual plain cookies as shown in Figure 7. In addition to the virtual plain cookies, she has to find and collect physical ingredients that are actually Bluetooth embedded objects (Figure 2). These objects have direct links and representations in the virtual Pac-World. This provides a sense of presence and immersion with the virtual Pac-World, as well as a feeling of actively participating in the game in the real world.

On the other hand, the Pacman should avoid being devoured by Ghost, i.e. not letting Ghost tapping on her shoulder where the capacitive sensor is attached on. This physical touch interaction between the players exemplifies tangible physical interaction between humans, which is commonly found in traditional games such as hide-and-seek and the classic “catching” game, but is now being revived in computer gaming arena.

The role of a Ghost is rather straightforward; she has to track down all Pacmen and devour them. Nevertheless, she has to

beware of Pacmen with Ghost-devouring power and avoid being devoured by them.

Helper is a new character in Human Pacman who does not exist in the original Pacman game. Each Pacman and Ghost will be assigned a partner Helper who acts as an intelligence, advisor, and coordinator for her in her quest for achieving her goal. To enhance gaming experience for Pacman and Ghost, the players in the physical world with wearable computer are not able to see enemy mobile units and hidden ingredients when they are in AR mode. Therefore the Helper, who is always in VR mode and sees all, guides her partner by messaging her with important information as shown in Figure 8 and thus this promotes collaboration and interaction between human.

It must be noted that when Pacman or Ghost switches to VR mode, she is completely immersed in virtual Pac-world. However virtual paths are drawn in close correspondent to roads in real world as seen in Figure 6. Despite being physically separated from one another and highly mobile, each player is constantly tracked and her identity as unique entity in Pac-world is maintained in real time.

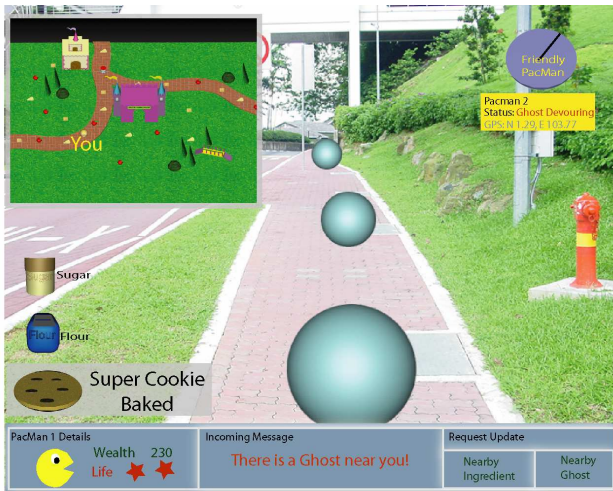


Figure 7. First person view as presented in HMD display of Pacman



Figure 8. Close collaboration between the outdoor Pacman and her Helper

3.3 Actual game play

Starting the game: Pacmen and Ghosts start from the Pac-castle and Ghost-house in Pac-world respectively. These two places correspond to two different physical locations in the game area.

Collection of plain cookies: Pacman collects a cookie by walking through it. Such physical action is reflected visually in Pac-World through the disappearing of the cookie in both the AR and VR mode. The collection of cookies by a Pacman will be reflected in the Pac-World map seen by all players in real time, be it on the player's HMD or the helper's Sony VAIO. In Figure 8,

the top images show the HMD view of the Pacman player as she collects a cookie. When she walks through the cookie, the cookie disappears. Note that this is also reflected in the virtual Pac-World in real time by the disappearing of the cookie in the corresponding location. This is shown in the images of the figure on the next page.

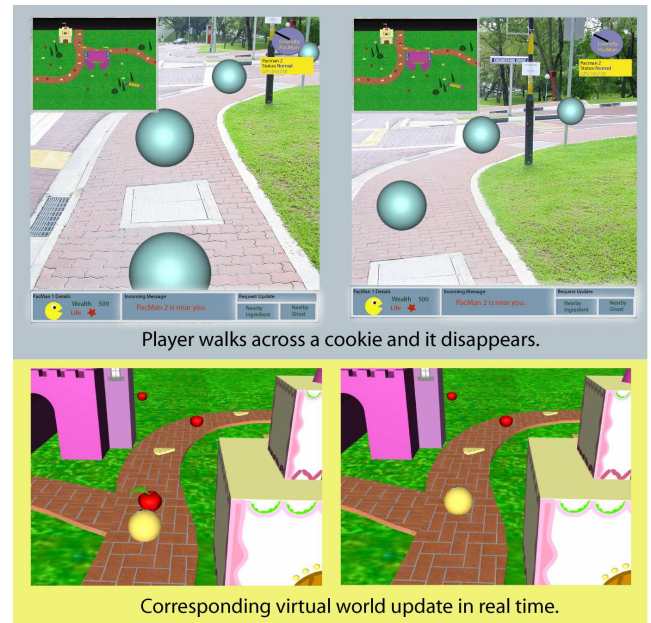


Figure 9: Pacman collecting cookies

Ghosts are not allowed to collect cookies. Although a Ghost is not able to see enemy Pacman on the map, the disappearing of cookies in her map can give her a hint to where to find a Pacman. Therefore a Pacman has to be careful as her physical interaction with the real world (i.e. movement) can be digitally reflected in the virtual world, and be made used of by a Ghost. Novelty is again seen in such intimate relationship between interaction in the physical world and its effect in the fantasy virtual world. Neither physical distance nor mobility could restrict each player from seeing this effect real-time as all players, including the Ghosts can see an update of the virtual map in real-time.

Collection of ingredients: In the game, Pacmen collects ingredients to make special cookies. Ingredients include flour, butter, sugar, and special ingredients (e.g. Chocolate Chip, Almond). There are two types of special cookies; a butter cookie is made up of flour, butter, and sugar; a super cookie is made up of butter cookie and a special ingredient.

When Pacman eats a butter cookie, she achieves 10 seconds immunity from being consumed by a Ghost. When Pacman eats a super cookie, it takes a time lag of 5 seconds before she achieves 10 seconds of ghost-devouring power. (5 seconds is for the Ghost to run or devour the Pacman).



Figure 10: Sequence of pictures showing the collection of an ingredient



Figure 11: HMD display and the corresponding label under VR mode when Pacman gains power

In the game, real Bluetooth-embedded objects are placed in different parts of the game area. In Figure 10 on the next page, a sequence of pictures shows a Pacman collecting an ingredient. When Pacman is within ten meters range from the object, she will be alerted to its presence visually with an alert message as seen in the figure. However she still has to search for the Bluetooth-embedded object, as it might not be in her line of sight. Having found the object, collection is done simply by physically holding the object in her hands. This exemplifies a natural tangible interaction is involved through physically interacting with this

object by human touch. Pacman is able to hold a real object naturally in hand as should be in real-life treasure finding. Such tangible action is provides the player a sense of touch to the fantasy world. The collection of the ingredient will be kept in a virtual inventory list and be immediately reflected in the display as well as having the action occurs in real-time in the virtual world. As seen in the figure, collection is shown as an addition of an icon of a sugar jar to the inventory list after the ingredient has been collected. Pacman need not lug the physical object with her as she has collected the ingredient virtually.

Collaboration between players: There is an essential element of collaboration in the game play between a Pacman/Ghost with her Helper, and between any allied Pacmen.

(i) *Pacman/Ghost and Helper Collaboration* - The Helper is in a good position to assist her partner as she has a complete view of Pac-world all the time, including the positions of all players and ingredients. Mobile players can only see the complete Pac-World under the VR mode. However AR mode is more advantageous for mobility because under VR mode, movement is restricted since she does not have a view of the real world. Furthermore as Helpers within the same team are physically close, they are able to collaborate between themselves and work out a strategy to achieve the team's goal. The advantage of this setup is that social interaction and collaboration is significant between Helpers, as well as between Helpers and her partner.

(ii) *Pacman and Pacman Collaboration* -- Pacman players can collaborate through transferring of ingredient between them, even if they are physically far from each other with the support of Wireless LAN. For example Pacman A can initiate request for the list of unused ingredients Pacman B has. Upon approval, A can request for transfer of ingredient from B, subjected to approval by B. Transfer of ingredient is important as Pacman may not be able to comb the whole game area for ingredients. She may lack some ingredients, which may have been collected by her ally. Strategy could be implemented, with the coordination from the Helpers, to distribute ingredients between the Pacmen. However, Pacman are not allowed to transfer special cookies so as not to disadvantage the Ghosts.

Using special cookies: All special cookies can only be used once. When a Pacman consumes a special cookie, she will see an alert message in her HMD, informing her of the power she acquired. Furthermore, in real-time a label describing her acquired-power will be placed on top of her Pacman avatar in the VR mode. This serves to inform all Helpers, including those from the Ghost-team, of her ability. These are illustrated in Figure 11.

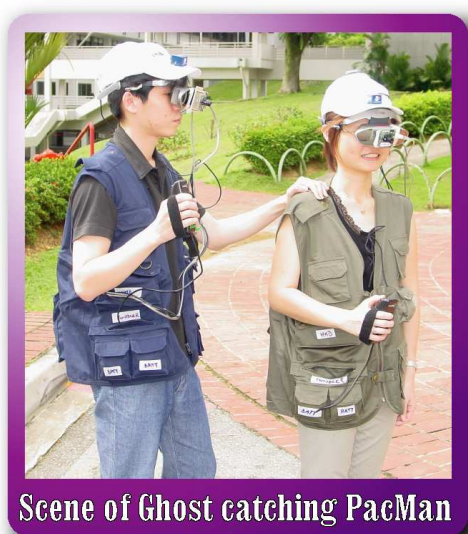


Figure 12: Ghost catching a Pacman

Devouring enemy player: To devour a Pacman, a Ghost must physically touch the Pacman's capacitive sensor pads on her shoulders as shown in Figure 12. The same applies when a Pacman with Ghost-devouring capability devours a Ghost. When a Pacman is devoured, she loses one life point. Each Pacman is given two life points. The same goes for the Ghost.

Ending the game: The game ends when either team meets their goal or when a time limit of ten minutes has been reached.

4. UBICOMP AND SENSOR TRACKING ISSUES

Ubicomp, also known as "Ubiquitous Computing", is a phrase which late Mark Weiser (1952-1999) described in 1988 as "calm technology that recedes into the background of our lives". Though not strictly making computers available throughout the physical environment but invisible to the user as described by Weiser [Weiser93], Human Pacman envisions applying the same concept of calm technology into computer gaming by experimenting with tangible interfaces and context-awareness entertainment and communication, which are two of the interaction themes in UbiComp.

4.1 Tangible Interfaces

There are two distinctive manifestations of tangible interfaces in Human Pacman; the first is being implemented in 'Special Ingredient', which is actually a Bluetooth embedded object with capacitive sensor, and the second is the capacitive sensor shoulder pads on the wearable computers for Pacmen and Ghosts. Bluetooth is incorporated into the system where there is already Wireless LAN support for communication because firstly it provides paired communication with security which is essential for one-to-one communication between the 'Ingredient' and the player; secondly Bluetooth devices support automatic device discovery and connection setup when they are within range therefore provide the backbone for reasoning by close physical proximity in the game play. This allows Pacman to search nearby area for 'Ingredient' once being alerted of the presence of Bluetooth embedded device.

Tangible interaction between the Bluetooth embedded object and the player is made possible by using capacitive sensor for detecting the action of touch by the player. In this way, we harness the physical and tactile abilities of Pacmen to support the computational task of registering the discovery and collection of virtual ingredient. Another important aspect of this design is the clever exploitation of the affordances of the object's physical properties whereby without prior training, players can intuitively associate the action of picking up the 'Ingredient' object with the collection of it in their virtual inventory as well as having the action simultaneously occur in the virtual world.

The use of capacitive sensor shoulder pads of wearable computer for the detection of 'Devouring' action in game play, serves the purpose of demonstrating how computation can be used in concert with naturalistic activities, in this case, the action of physically catching the enemy on the shoulder. Also, by making the distinction between "interface" and "action" very much reduced, i.e. physical action of tapping versus a mouse-click for interaction, Human Pacman allows the players to experience transparent interchange between human and computer as never before in computer gaming.

Table I. Five questions and answers posing human-computer communication challenges for interaction design in the case of Human Pacman.

Basic Question	Human Pacman Interface Answers
Address: How do I address one (or more) of many possible devices?	With the implementation of Ubiquitous Computing, the system constitutes a more amorphous concept with automated interactions between sensors and computer. The existence of unique address for each Bluetooth device disambiguates the Bluetooth embedded objects. Furthermore, centralized control of the server prevents ambiguity of intended target system even when there are more than one players are near the Bluetooth device. Keyboard and mouse are used for messaging and selection of the 'Ingredients' to be exchanged between Pacmen.
Attention: How do I know the system is ready and attending to my actions?	Graphical feedback is used extensively from providing alert message in popped up window, to refreshing virtual inventory after Pacman picked up Bluetooth embedded object. Also, since this graphical information is provided in the HMD directly in the zone of the user's attention, they are highly effective.
Action: How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?	The Pacman/Ghost click on preset messages to be sent to Helpers. Pacmen click on graphical representation of 'Ingredient' to be exchanged. Clearly labeled Bluetooth embedded objects are to be found in physical space where interaction is intuitive. According to Norman's Theory of Action [Norman90], this form of tangible interface bridges the 'Gulf of Execution'.
Alignment: how do I know the system is doing (has done) the right thing?	Real time graphical feedback presents distinctive and timely graphical elements establishing the context of the system.
Accident: How do I avoid mistakes?	Pacman right-click on virtual ingredient in order to dump the ingredient.

4.2 Context Awareness in Entertainment and Communication

There are three different ways in which the idea of context awareness is being applied to in Human Pacman. Firstly, with the use of GPS and DRM to provide various data required for tracking the players in wide area outdoor environment, location awareness of the system is made possible. Despite being the most widely publicized and applied location-sensing system, GPS suffers from accuracy and selective availability. The problems are compensated through sensorfusion with DRM. In Human Pacman, we are taking advantage of user's mobility in the wide outdoor

area to adapt the system's behavior based on her current location. This location context is being made use of throughout the game play for augmented reality (AR) placing of virtual cookies, as well as calculating the relative positions of allied players.

Another important component in realizing AR elements in Human Pacman is the inertia sensor. Through data collected from it, the system is aware of current perspective of the player and thereby displays virtual objects accordingly. Besides, Human Pacman also experiment with information context in Human Computer Interface with the Helper player having information access to other players via Wireless LAN and providing them with necessary and timely information.

4.3 Addressing sensor-tracking issues

In Human Pacman, we tried to combine materials from cognitive psychology and sociology with that from computer science. However the vast amount of issues encountered have exceeded the scope of this paper. Therefore we will concentrate on discussing sensor-tracking issues with respect to Human Computer Interface design. According to Bellotti [Bellotti02], there are five questions posing human-computer communication challenges for interaction design. In Table I, we summarize the sensing approaches to interaction in Human Pacman with respect to the five questions raised.

5. CONCLUSION

In this paper we presented a sensing-based mobile entertainment system that brings computer gaming experience to permeate natural physical world ubiquitously and seamlessly. Context awareness computing is being applied to wide outdoor physical area for realizing mobile gaming and social collaboration elements in Human Pacman. At the same time, tangible interface is being explored with tangible physical objects incorporated into the game play to provide the novel experience of seamless transitions between real and virtual worlds. These objects in the real world are embedded with a real-time link and meaning with objects in the virtual world. Table II summarizes the various novelties about Human Pacman.

The continual propagation of digital communication and entertainment in recent years forces many changes in societal psyche and lifestyle, i.e. how we think, work and play. With physical and mobile gaming gaining popularity, traditional paradigms of entertainment will irrevocably shake from the stale television-set inertia. We believe that Human Pacman heralds the conjuration and growth of a new genre of computer game that is built on mobility, physical actions and the real world as a playground. Reality, in this case, is becoming more exotic than fantasy because of the Mixed Reality element in the game play. On the other hand, emphasis on physical actions might even bring forth the evolvement of professional physical gaming as competitive sport of the future, for example 'PacMan International League'. Element of social gaming in Human PacMan symbolizes the nascence of humanity in future digital entertainment. People are looking forward to widening their circle of friends and colleagues through social collaboration in game play. A new form of interactive entertainment is evolved. Another important area of impact is the field of education. The technology presented in Human PacMan can be exported to applications in educational training that stresses on "learn by experience".

Students are immersed in real site of action, and are given instructions visually through head mounted display or verbally through speaker/earphone. This technology serves as a powerful instrument of cognition since it can enhance both experimenting and reflective thoughts through mixed reality and interactive experience.

In conclusion, we believe Human PacMan is a pioneer in the new hybrid of physical, social, and mobile gaming that is built on ubiquitous computing and networking technology. The players are able to experience seamless transition between real and virtual world and therefore a higher than ever level of sensory gratification are obtained.

Table II. Novel features of Human Pacman

Novelty	Improvement
Ubiquitous Computing	Everyday objects throughout the environment seamlessly have a real-time fantasy digital world link and meaning. There is automatic communication between Wearable Computers and Bluetooth devices embedded in certain physical objects used in game play.
Tangible Interaction	Throughout the game people interact in a touch and tangible manner. For example, Players need to physically pick up objects and tap on the shoulder of other players to devour them.
Physical Gaming	Players are physically role-playing the characters of Pacmen and Ghost; with Wearable Computers donned, they use wide area free bodily movements as part of interaction between each person, the real and virtual world, and objects in the real and virtual environments.
Social Gaming	Players interact both directly with other players when they are in physical proximity or indirectly via the Wireless LAN network by real time messaging. There is a seamless networked social contact with players in both the real-world and virtual-worlds, as well as throughout their boundaries. People from all around the world can also participate in the Human Pacman experience by viewing and collaborating in real time over the internet with the physical Human Pacmen and Ghosts who are immersed in the physical real world game.
Mobile Gaming	Players are free to move about in the indoor/outdoor space without being constrained to the 2D/3D screen of desktop computers.
Outdoor Wide-Area Gaming Arena	Large outdoor areas can be set up for the game whereby players carry out their respective missions for the role they play. This could even be linked throughout cities.
Seamless Transition between real and virtual worlds	Players swap freely between immersive first person augmented reality view (with virtual cookies and instructions overlay the real world) and full virtual reality view of the Pac-world in the game.

6. REFERENCES

- [Acm93] Computer Augmented Environments: Back to the Real World. Communications of the ACM, July 1993, 36, 7.
- [Bellotti02] BELLOTTI, V.M.E., BACK, M.J., EDWARDS, W.K., GRINTER, R.E., LOPES, C.V. AND HENDERSON, A., 2002. Making sense of sensing systems: Five questions for designers and researchers. Proc. CHI 2002, 415-422. ACM.
- [Björk01] BJÖRK, S., FALK, J., HANSSON, R., AND LJUNGSTRAND, P., 2001. Pirates! – Using the physical world as a game board, Interact 2001, IFIP TC. 13 Conference on Human-Computer Interaction, Tokyo, Japan 2001.
- [Björk02] BJÖRK, S., HOLOPAINEN, J., LJUNGSTRAND, P., AND MANDRYK, R., 2002. Special Issue on Ubiquitous Games, Personal and Ubiquitous Computing, 6, 5-6, 358-361.
- [Dodsworth98a] CRAWFORD, C., 1998. Live: What a Concept!...Networked Games. Digital Illusion, DODSWORTH, C. JR, Ed. ACM Press, 241-248.
- [Bowlby83] BOWLBY, J., 1983. Attachment and Loss, Volume I: Attachment. Basic Books, New York, 1983.
- [Fitzmaurice95] FITZMAURICE, G., et al., 1995. Bricks: Laying the Foundations for Graspable User Interfaces, Proceedings of the Conference on Human Factors in Computing Systems (CHI '95), ACM, Denver, May 1995, 442-449.
- [IDSA01] Interactive Digital Software Association, State of the Industry – Report 2000-2001, [online document] 2001, Available at <http://www.idsa.com>
- [IDSA02] Interactive Digital Software Association, Essential Facts About the Computer and Video Game Industry, [online document] 2002, Available at <http://www.idsa.com>
- [InterSense01] InterSense Inc, Inertia Cube2 Bringing 3D To Life – Manual for Serial Port Model, 2001.
- [Ishii97] ISHII, H., AND ULLMER, B., 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. Proceedings of the Conference on Human Factors in Computing Systems (CHI '97), March 22-27, 1997.
- [Konami01] Konami Corporation, ParaParaParadise, [online document] 2001. Available at <http://www.konami.com.hk/am/ppp.html>
- [Long96] LONG, S., et al. 1996. Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study. 2nd ACM International Conference on Mobile Computing and Networking (MobiCom'96) 1996 November 10-12, 1996.
- [Mandryk 01] MANDRYK, R.L., INKPEN, K.M., 2001. Supporting free play in ubiquitous computer games, Workshop on Designing Ubiquitous Computer Games, UbiComp 2001, Atlanta, GA 2001.
- [Nezlek02] NEZLEK, J.B., RICHARDSON, D.R., GREEN, L.R., AND SCHATTEN-JONES, E., 2002. Psychological well-being and day-to-day social interaction among older adults, Personal Relationships, 9, 57-71.
- [Norman90] NORMAN, D.A., 1990. The Design of Everyday Things. Doubleday: New York, New York.
- [Pascoe98] PASCOE, J., RYAN, N.S., MORSE, D.R. 1998. Human Computer Giraffe Interaction - HCI in the Field. Workshop on Human Computer Interaction with Mobile Devices.

- <<http://www.dcs.gla.ac.uk/~johnson/papers/mobile/HCIMD1.html>>
- [Pascoe99] PASCOE, J., RYAN, N.S., AND MORSE, D.R. 1999. Issues in Developing Context-Aware Computing. Proceedings of the International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, September 1999, Springer-Verlag, 208-221.
- [Pointgrey02] Point Grey Research Corporation, <http://www.ptgrey.com>
- [Quantum02] Quantum Research Group Ltd, [online document] 2002, Available at <http://www.qprox.com>.
- [Ryan97] RYAN, N., PASCOE, J., MORSE, D. 1997. Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant. V., GAFFNEY, M., VAN LEUSEN, S., EXXON, Eds. Computer Applications in Archaeology. <<http://www.cs.ukc.ac.uk/pubs/1998/616/content.html>>
- [Schilit93] SCHILIT, B.N., THEIMER, M.M., AND WELCH, B.B. 1993. Customizing mobile applications. In Proceedings of USENIX Mobile & Location-Independent Computing Symposium, Cambridge, Massachusetts, August 1993. USENIX Association, 129-138.
- [Smith95] SMITH, C.G., 1995. The Hand That Rocks the Cradle. I.D., May/June 1995, 60-65.
- [Thomas02] THOMAS, B., CLOSE, B., DONOGHUE, J., SQUIRES, J., BONDI, P.D., AND PIEKARSKI, W., 2002. First Person Indoor/Outdoor Augmented Reality Application: ARQuake, Personal and Ubiquitous Computing, 6, 1, 75-86.
- [Want92] WANT, R., HOPPER, A., FALCÃO, V., AND GIBBONS, J. 1992. The Active Badge location system. ACM Transactions on Information Systems, 10(1):91-102, January 1992.
- [Weiser93] WEISER, M., 1993. Some computer science issues in ubiquitous computing, Communications of ACM, 36, 7: Special issue on computer augmented environments: back to the real world.
- [Yang99] YANG, J., YANG, W., DENECKE, M., AND WAIBEL, A. 1999. Smart sight: a tourist assistant system. 3rd International Symposium on Wearable Computers, San Francisco, California, 18-19 October, 1999, 73-78.

Exhibit B



The Effect of Latency on User Performance in Warcraft III^{®*}

Nathan Sheldon, Eric Girard, Seth Borg, Mark Claypool, Emmanuel Agu
`{claypool,emmanuel}@cs.wpi.edu`
 CS Dept., Worcester Polytechnic Institute
 100 Institute Road,
 Worcester, MA 01609, USA

Abstract

Latency on the Internet is a well-known problem for interactive applications. With the increase in interactive network games comes the increased importance of understanding the effects of latency on user performance. Classes of network games such as First Person Shooters (FPS) and Real Time Strategy (RTS) differ in their user interaction model and hence susceptibility to latency. While previous work has measured the effects of latency on FPS games, there has been no systematic investigation of the effects of latency on RTS games. In this work, we design and conduct user studies that measure the impact of latency on user performance in Warcraft III, a popular RTS game. As a foundation for the research, we separated typical Warcraft III user interactions into the basic components of explore, build and combat, and analyzed each individually. We find modest statistical correlations between user performance and latency for exploration, but very weak correlations for building and combat. Overall, the effect of even very high latency, while noticeable to users, has a negligible effect on the outcome of the game. We attribute this somewhat surprising result to the nature of RTS game-play that clearly favors strategy over the real-time aspects.

*Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. NetGames'03, May 22-23, 2003, Redwood City, CA, USA. Copyright 2003 ACM 1-58113-734-6/03/0005...\$5.00

1 Introduction

Over the past decade, the Internet has grown in popularity and capability at exceptional rates. In 1997, there were 36.6 million homes with computers and only 18 million of them had Internet access [6]. By the year 2000, the number of homes with computers had grown to 51 million, 41.5 million of which had Internet access, and many with broadband Internet connections such as cable modems and DSL lines.

This growth in popularity and capability of the Internet has led to an increasingly diverse set of applications with varying network behaviors and requirements. Key metrics of latency and throughput are useful in characterizing application behavior. Traditional applications such as file transfer, Usenet news and email are primarily concerned with throughput and can tolerate delays on the order of minutes. Web browsers are also concerned about throughput, but the interactive nature of browsing requires latencies on the order of seconds or at most tens of seconds [4]. Emerging real-time applications such as IP telephony and networked games typically have the lowest throughput requirements but are even less tolerant of latency than other applications. Knowing how these real-time applications react to latency and loss is the crucial first step in designing the next generation network hardware and software that will support their requirements. In addition, classifications of real-time applications according to latency tolerance will enable designers, developers and engineers to make informed decisions on appropriate quality for classes under such architectures as Diff-Serv [5].

The most popular real-time applications are

multi-player network computer games that can make up around half of the top 25 types of non-traditional traffic for some Internet links [13] and are predicted to make up over 25 percent of Local Area Network (LAN) traffic by the year 2010. In 2000, the U.S. economy only grew 7.4% while the computer and video game industry grew by 14.9%, out-pacing growth in other high-tech industries and even Hollywood over the previous five years [10]. In 2002, over 221 million computer and video games were sold, or almost two games for every household in America.¹ Knowledge of how network related issues, such as latency and packet loss, affect the usability of games can be of great use to the companies that make these games, network software and equipment manufacturers, Internet Service Providers (ISPs), and the research community at large. In particular, if established latency requirements and any associated trade-offs were known, ISPs could establish tariffs based on customers' indicated maximum delays, requested Quality of service (QoS) and the ISP's ability to meet these demands.

Two of the most popular categories of real-time network games are *First Person Shooter* (FPS) games and *Real Time Strategy* (RTS) games. FPS games, first made popular by Doom,² have the player view the world through the eyes of a character (the first person). They then move around slaying monsters and other players, with an amalgamation of ranged weaponry (the shooter). RTS games, first made popular by Dune 2,³ are generally characterized by resource collection, unit construction, and battles that consist of large numbers of animated soldiers going through a repetitive, animated attack.

While there has been research qualitatively characterizing the effects of latency for some popular FPS games [2, 9] as well as a general awareness of latency issues [3, 11, 14], quantitative studies of the effects of latency on network games have been lacking. Moreover, it is unlikely that FPS games have the same network requirements as do RTS games. In FPS games, exact positioning and timing is re-

quired, because, for example, a target must still be at the location where the player aimed in order for the shot to hit. In RTS games, the positioning and timing is more forgiving because, for instance, a command can be issued to attack a unit, regardless of its current location or its direction and time of movement.

This work studies the effect of latency on user performance and network traffic for the best selling [15] RTS game, Warcraft III.⁴ We quantify the effect of latency on user performance in Warcraft III by analyzing the results of controlled research experiments designed to measure quality of service over a range of induced latencies. As a foundation for RTS research, we divide Warcraft III up into fundamental game components of building, exploring and combat. We then develop multiple criteria for measuring user performance in Warcraft III and use these criteria in very carefully designed experiments to determine user performance over a range of latency conditions.

We find that latencies up to several seconds have little effect on the final outcomes of building, exploring, and most combat. Although, the effectiveness of certain strategies that involve precise timing of events are influenced by the amount of latency, very few such strategies prevail in typical Warcraft III games. Overall, strategy plays a much larger role in determining the outcome of the game than does latency. We conclude that Warcraft III should be placed in a different QoS class than applications with stringent latency constraints, such as FPS games or audio-conferences, since Warcraft III has latency requirements more similar to those of Web browsing.

The rest of this paper is organized as follows: Section 2 presents background information on Warcraft III; Section 3 describes our approach to measure the effects of latency on Warcraft III; Section 4 analyzes the application, network and user results from our experiments; Section 5 summarizes our conclusions; and Section 6 presents possible future work.

¹Top Ten Industry Facts, IDSA, <http://www.idsa.com/-pressroom.html>

²<http://www.idsoftware.com/games/doom/>

³<http://www.dune2k.com/duniverse/dune2/>

⁴<http://www.blizzard.com/war3/>

2 Background

Warcraft III is an RTS game in which players construct buildings and fighting units, and issue commands that cause the units to move, engage enemy units in battle, and build structures. Games are played on one of many possible maps, which are either provided with the game or custom built by players. During the game, Warcraft III uses a centralized server in a client-server architecture, either over the Internet with possibly thousands of participants or in a LAN scenario with a few players. For most Internet games, the server is via Battle.net,⁵ a free service that allows Blizzard's Starcraft, Diablo and Warcraft players to initiate multi-player games over the Internet. For a LAN game, users can have a client's machine act as a server, too, by choosing a map and then letting other clients join the game.

At the beginning of a game, players can choose to be a member of one of four "races" (Humans, Orcs, Undead and Night Elves). Our research focuses on the Humans, but since Blizzard put great effort into making the races equivalent in power, our results should generalize to the other races. There are a number of ways in which players can be competitively grouped. In a free-for-all game, all players vie to have the last remaining army on the map. Players can also team up against each other and/or against artificially intelligent computer-controlled players in myriad ways.

Figure 1 shows a screenshot of a Human town under attack from an Undead army. The Undead are in the upper left area of the screen. Human peasants can also be seen carrying lumber to the town hall and other activities.

Town control and unit control are two major aspects of the game. Town control consists of selecting what buildings are to be built or upgraded, what units are to be produced and what technologies are to be researched. In order to accomplish these tasks gold and lumber are required which are gathered by peasants. The Human buildings range from farms, which provide food, to town halls, which are the central building in any Human town. Buildings such as barracks produce standard army units such as footmen and knights while buildings



Figure 1: Screenshot of Undead Army Attacking a Town.

such as the arcane sanctum produce units such as priests and sorceresses. Town control involves a lot of strategy in knowing when and where to build, upgrade, and research.

Unit control can be broken up into three sub-categories: building, exploration and combat. Building overlaps with town control as it is the management of peasants in harvesting gold and lumber as well as building and repairing buildings. Exploration allows players to determine geography, find creeps (neutral enemies on which to gain experience for their heroes), enemy towns or units, and neutral shops at which their heroes can buy various items. Combat allows units to kill other units, to defend towns, secure territory or gain experience points for Heroes. There are various battle strategies that can be deployed, but at a minimum the player can let the computer's artificial intelligence handle the units. Warcraft III has special units called heroes which are more powerful than standard units, and have additional abilities.

3 Approach

In order to empirically measure the effects of latency on Warcraft III, we employed the following methodology:

- Categorize user interactions in typical RTS

⁵<http://www.battle.net/>

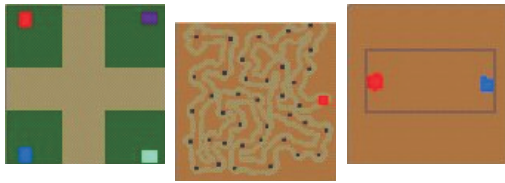


Figure 2: RTS Component Maps: Build (left), Explore (middle), Combat (right).

games and construct Warcraft III campaigns that exercise each category (see Section 3.1).

- Determine criteria to quantitatively measure Warcraft III performance (see Section 3.2).
- Construct an environment for measuring the effects of latency on Warcraft III (see Section 3.3).
- Conduct pilot studies (see Section 3.4) and then numerous user studies for each RTS category over a range of latencies, recording the performance measurements.
- Analyze the results (see Section 4).

3.1 Categories of RTS Interaction

Through pilot studies, we determined there are three main user interaction components of an RTS game: *building* when players gather resources, construct defenses and recruit units; *exploring* when players send units out to determine geographic layout and location of other players' units; and *combat* when players engage their units with other units in battle. Since all components require user interaction, we hypothesized that under each component, user performance would degrade as latency increased. We built multi-player maps⁶ that isolated each component so that we could use experiments to measure the effects of latency on that component.

For the building map (Figure 2 (left)), we divided the map into four quarters using mountain ranges that units could not cross. Each player started with a town hall and four peasants, had unlimited gold and lumber available, and had to research, build, and upgrade the complete Human technology tree as fast as possible. We added triggers to the map

⁶The maps can be downloaded at <http://perform.wpi.edu/downloads/#war3>

that disabled players' ability to build more than one building in order to provide consistency and reduce confusion, as well as a trigger to display the total time since the beginning of the game.

For the exploration map (Figure 2 (middle)), we designed a raised path that kept units on a general exploration course. The player had to guide a unit along the winding path and step on numerous waypoints. Map triggers kept track of the player's time to complete the map.

For the combat map (Figure 2 (right)), we designed a small player versus player arena in which each player controlled a small army consisting of a level 6 Hero (a Mountain King), two Knights, four Footmen, two Riflemen, a Sorceress, and two Priests.

3.2 Warcraft III Performance Criteria

For both the building and exploration maps we recorded the game length as a measure of performance. For the combat maps, in addition to the game length, we recorded each player's unit score and which player won. The number of units a player starts with plus the number of units killed determines the unit score. The breakdown of points for the individual Human units used in our combat map are listed in Table 1.

Unit	Score
Footman	160
Priest	170
Sorceress	200
Rifleman	270
Knight	350
Level 6 Hero	600

Table 1: Unit Score Values

3.3 Experimental Setup

Figure 3 depicts our experimental testbed setup, which consisted of PCs connected on a private network subnet. Computer A was a dual-processor Pentium-2 300 MHz machine running Mandrake Linux that routed packets with 100 Mbps connections to the Warcraft computers B, C and D. Computers B and C were Pentium-2 350 MHz machines with 256 MB of RAM, and a 64 MB Geforce 2 Ti

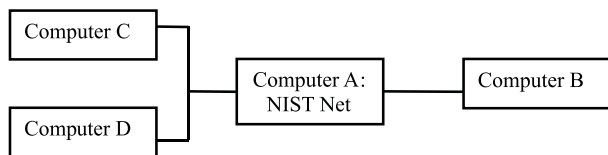


Figure 3: Experimental Testbed Setup.

graphics card running Windows 98. Computer D was a Pentium-4 1.3 GHz with 256 MB of memory and a 64 MB Geforce2 graphics card running Windows XP.

The recommended specifications for Warcraft III are a 400 MHz Pentium-2 or equivalent, 128 MB of RAM, and an 8 MB 3D video card (TNT, i810, Voodoo 3, Rage 128 equivalent or better) with DirectX® 8.1 support. Although machines C and D were only 350 MHz machines, the graphics cards and extra memory that they contained made up for this slight deficiency, and all machines were capable of rendering 30 frames per second⁷ even during battle. We used Warcraft III version 1.04 for all user tests and version 1.05 for the network traces due to the Battle.net requirements.

We installed NIST Net⁸ on computer A. NIST Net allows emulation of a wide variety of network conditions by giving control at the IP level, including fine tuning of latency and variation in latency (jitter). We used NIST Net to induce latency (and jitter) for one of the machines in a game, while the others played with no induced latency. Also, in order to analyze the network footprint of Warcraft III, we ran Ethereal⁹ to capture packet traces for network analysis.

3.4 Pilot Studies

First, we conducted pilot studies to help determine the range of viable latencies on which to focus. Our first pilot studies consisted of two-player games in which one player was subjected an increasing amount of latency and the other player experienced none. Initially, each player had a town hall and a gold mine placed a fixed distance away from the town hall; and second, each player had

two identical units that did one point of damage per hit. We setup triggers in the maps so games could be run automatically and ran repeated tests with one player (the lagged player) having increasingly greater latency. We found both players did equally well, gaining gold and inflicting damage at exactly the same rate. In addition, both players saw exactly the same events on each screen, except the player with added latency saw events later than the player without added latency.

From these pilot studies, we made two important observations about latency compensation in Warcraft III:

First, the game does not use handicapping in the game to equalize latencies across all players. Both lagged and non-lagged players see events happen at the real-time rate, regardless of the latency of the other player. The lagged player has events executed later by an amount equal to the induced latency.

Second, the game does not have inconsistent game states, which implies no dead reckoning [8]. The actions that occur on each machine are identical; there is no prediction of user actions and then correction upon some later time if the predictions are inconsistent with the actual game state.

Thus, clients must communicate any user actions to the server before executing them. After that, the commands themselves are executed identically on all machines in the game.

For the real experimental runs, the maps were not automated and we pitted one player against another player. The first player was the server with no induced latency. The second player was the client that was subjected to induced latencies ranging from 0 to 3500 ms, a range even broader than typically found in dialup modems [12]. Hence, we concentrated our data points on ranges of more typical latencies [1] which are less than 1000 ms.

From traces collected during our pilot studies, we determined that clients communicate only with the server but not directly with other clients. Servers combine data from multiple clients before distributing data. Each machine maintains a complete copy of the game state, and to an extent, all outcomes are predetermined upon initiation of the action. Command data is only transferred upon the issuance of a command, and never again during

⁷Tested with *fraps*, <http://www.fraps.com/>

⁸<http://snad.ncsl.nist.gov/itg/nistnet/>

⁹<http://www.ethereal.com/>

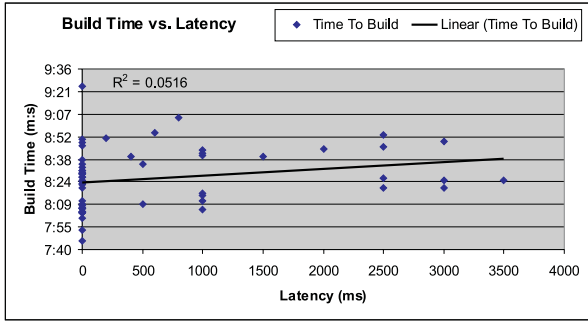


Figure 4: Build Time versus Latency.

the life of the event. For instance, the commands to initiate a large-scale battle are propagated to all clients once, resulting in an increase in the packet payload size, but the battle itself has no effect on traffic unless further commands are issued as the battle is carried out.

4 Analysis

We analyzed our experimental data at three levels: Section 4.1 contains our analysis of the application level data we collected from our Warcraft III user studies; Section 4.2 analyzes network level traffic for a LAN game and two Battle.net Internet games as well as network level traffic for combat games with three levels of induced latency; and Section 4.3 summarizes the observation data we collected during the user studies.

4.1 Application Level Analysis

This section analyzes the results from each of our test maps, starting with building (Section 4.1.1), then exploring (Section 4.1.2) and lastly combat (Section 4.1.3).

4.1.1 Building

Figure 4 illustrates the effect of latency on the total time required to construct every building and research every upgrade (the technology tree) for the Human race from our test map. The graph shows the build time versus latency for all runs, as well as a best-fit line for the data. Under conditions with no induced latency, building the technology tree

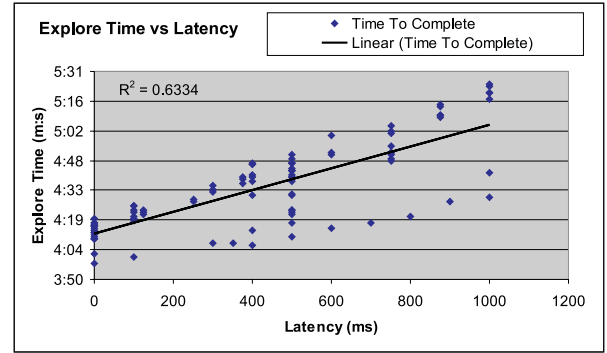


Figure 5: Explore Time versus Latency.

takes about 8 minutes. Latency values of up to 3.5 seconds increase total build time by at most 14 seconds, which is less than 1% of the total time for this short game. The coefficient of determination¹⁰ is 0.05, indicating there is very little statistical correlation between latency and building. In addition, the statistical correlation observed in a real game environment is likely to be even lower. This is due in part to a longer game time and that different numbers of buildings would be produced (such as more than one farm). In addition, in a real game, players may build their towns in a strategic layout instead of in a random pattern. Finally, time is often spent in a real game attending to other matters so that the speed of building the base is not of utmost importance. Our conclusion is that any effect latency may have on building would have no significant impact on the outcome of typical Warcraft III games.

4.1.2 Exploring

Figure 5 illustrates the effect of latency on the exploration of our test map. The graph shows the exploration time versus latency for all runs, as well as a best-fit line for the data. The overall correlation between explore time and latency is modest (0.63), but can be high (0.95) for individual users. The first 8-10 games of a test typically showed a vertical

¹⁰The coefficient of determination (R^2) represents the fraction of variability in y that can be explained by the variability in x . In the simple linear regression case, R^2 is simply the square of the correlation coefficient. An R^2 of 1 represents perfect correlation.

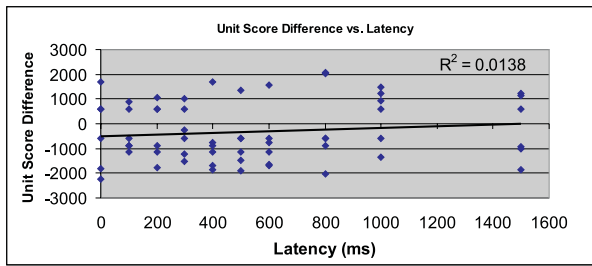


Figure 6: Unit Score Difference versus Latency.

component where exploration times decreased. We attribute this to the player learning the map, gaining from the knowledge in subsequent games. Once the map is known, all data show a nearly perfect linear relationships between latency and time to explore. Overall, while there is a statistical correlation for explore time versus build time, the effect of an additional 6 seconds of exploration time for every 100 ms of latency would be insignificant in a real game. In addition, it is likely that high latency players in a real game may try to adapt in various ways during exploration. For instance, high latency players may discover that they achieve better results by spending less time actively controlling their units during exploration and thus decide to send them for long distances with each move command instead of micro-managing them for shorter distances.

4.1.3 Combat

Figure 6 shows the unit score difference versus latency for all runs, as well as a best-fit line for the data. The unit score difference is the non-lagged player's unit score minus the lagged player's unit score. The maximum difference (if one player loses all units and the other loses none) is ± 3020 . From Figure 6, there is a slight upward trend in that the score difference increases as latency increases, but the coefficient of determination is an extremely low 0.01. Moreover, the difference in points from no induced latency to one second of induced latency is only about one unit, an insignificant amount in the large battles that are typical in Warcraft III. Thus, we conclude that latency has little effect on the individual units in combat.

Figure 7 illustrates the effect of latency on com-

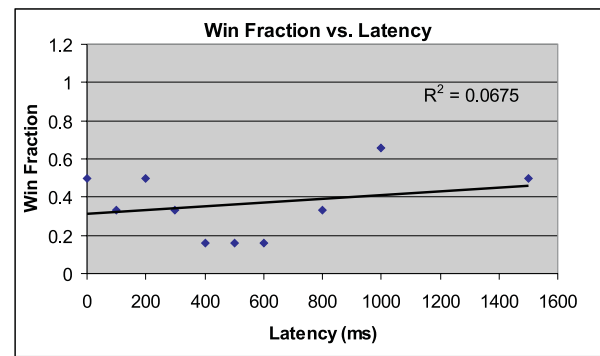


Figure 7: Combat Games Won by Host (non-lagged) versus Latency (on the Client).

bat from our test map. The graph shows the percentage of games won by the non-lagged host versus the latency of the lagged client. Even though there is a slight upward trend in the data, the coefficient of determination is an extremely low 0.07, indicating there is little statistical significance. Thus, we conclude that latency has little effect on the overall outcome of combat.

While the previous studies measured the effects of fixed latency on user performance, we also examined the effects of variable latency. For these tests, we set NIST Net to induce an average latency of zero¹¹ with a range of standard deviations.

Figure 8 shows games for 2 pairs of users with a standard deviation of latencies from a normal distribution with mean zero. Figure 8 (top) shows player 1 winning two games, one at 100 ms, and then again at 750 ms, while losing the games in between, all by similar margins of 1 or 2 units. Figure 8 (bottom) shows Player 3 consistently beating his opponent in every game, but by varying margins. Neither graph shows a significant statistical relationship between the variable latency and success in battle, similar to the results with constant latency.

Overall, both from a direct conclusion from our data and with extrapolation into a complete game, we found that the effect of latency on the outcome of a Warcraft III game is negligible over a range of practical latencies.

¹¹Our testbed had 1 ms of base latency from client to server.

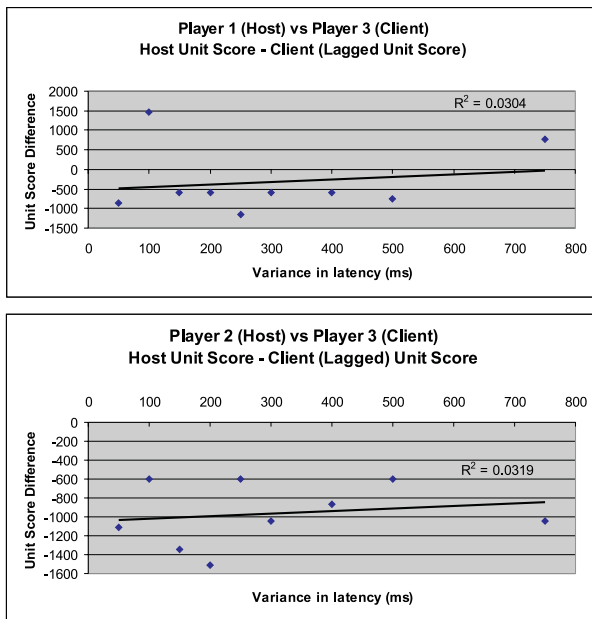


Figure 8: Unit Score Difference versus Variable Latency: Player 1 versus Player 3 (top), Player 2 versus Player 3 (bottom).

4.2 Network Level Analysis

Among other things, a better understanding of network game traffic can help design networks and architectures that more effectively accommodate their traffic footprints. Furthermore, careful empirical measurements of network games can provide the data required for accurate simulations, a typical tool for evaluating network research.

4.2.1 Traffic for Full Games

We packet traced three full (20-30 minute) games, two played over Battle.net and one played over a LAN.¹² The LAN game was 1 player versus 1 player (1v1), and the Battle.net games had a 1 player versus 1 player game and a 2 player team versus another 2 player team (2v2) game. Unlike other popular networked games [7], Warcraft III always uses TCP as the transport protocol with port 6112 for the server. All IP traces were performed on the client machines. For reference, the average round-trip times for the Battle.net games was about 100

¹²All network traces can be downloaded at <http://perform.wpi.edu/downloads/#war3>

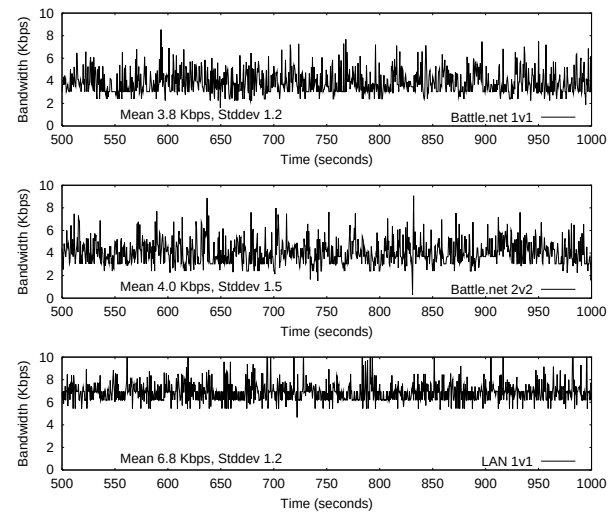


Figure 9: Bandwidth versus Time.

ms and there was less than 0.1% data loss.

Figure 9 depicts the bandwidth (including IP headers) taken in 500 ms intervals for the three packet traces. Only the intervals 500-1000 seconds are shown to illustrate more detail, but the bandwidth pattern throughout each game is similar to the shown interval. Overall, the variance in network bandwidth for all three is similar, with the average data rate for the LAN being slightly higher (6.8 Kbps) than the Battle.net traces (3.8 Kbps and 4.0 Kbps). All three traces have very low data rates that can easily be achieved with a modem. In comparison, Starcraft,¹³ the previous generation RTS game from Blizzard, has a data rate of about 5 Kbps for a 2 player game [7], similar to that of Warcraft III.

Figure 10 depicts the cumulative density functions (CDFs) of the payload sizes for all packet traces (incoming and outgoing). The median payload sizes are all very small, only 9 bytes. The two most common payload sizes are 6 and 9 bytes. Less than 1% of the payloads for any game are over 40 bytes with the Battle.net games having slightly more larger packets. The 2v2 player Battle.net game has a distribution with slightly larger payloads, most likely because of command aggregation across users at the Battle.net server.

Overall, Warcraft III sends considerably smaller

¹³<http://www.blizzard.com/worlds-starcraft.shtml>

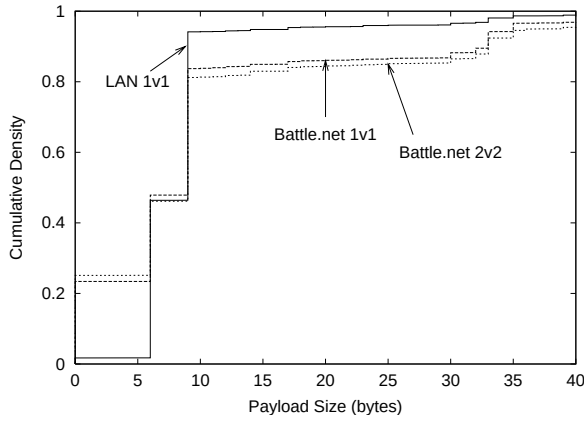


Figure 10: Payload Distributions.

packets than the typical Internet traffic packet size of over 400 bytes [13]. The number of players does not have a significant effect on the packet sizes, either. Warcraft III packet sizes are consistent throughout the game and are not significantly influenced by the action in the game. Since current Internet routers are designed for large transfers with large packets, there may be opportunities to improve network architectures to better manage and support game traffic. Starcraft has typical packet sizes of 122 and 132 bytes [7], while Warcraft III packets are most commonly 46 or 49 bytes in size (including headers).

Warcraft III sends out packets at regular intervals. Table 2 shows the inter-packet times that we observed for incoming and outgoing packets during the games we traced. In our local area network game, Warcraft maintained a very steady inter-packet rate of approximately one packet every 1/10th of a second both incoming and outgoing. With our Battle.net games the timing interval was lower, down to one packet every 200 ms incoming and every 160 ms outgoing.

	1v1 LAN	1v1 B.net	2v2 B.net
In Mean	104	201	201
In Std Dev	18.6	79.1	78.1
Out Mean	104	165	159
Out Std Dev	19.4	87.4	88.2

Table 2: Inter-packet Summary Statistics (ms).

Figure 11 depicts the CDF for inter-packet times

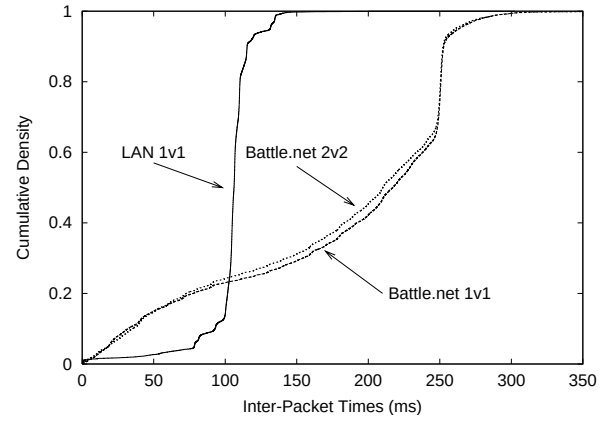


Figure 11: Inter-packet Distributions.

(incoming and outgoing). The LAN game has a much more consistent packet rate while the Battle.net Internet game varies considerably more. The median times for the Battle.net games are around 225 ms compared with around 100 ms for the LAN game. The 1v1 player Battle.net game exhibits about the same inter-packet times as does the 2v2 player Battle.net game.

4.2.2 Combat Traffic and Latency

From Section 4.2.1, the differences between the Battle.net game traces which had latencies around 100 ms and the LAN game traces which had latencies around 1 ms, suggest Warcraft III network traffic patterns change at least slightly with changes in latency. In this section, we analyze traces over a range of latencies in an attempt to quantitatively determine how Warcraft III network traffic differs with different latencies.

We packet traced games with our combat map at latencies of 0 ms, 500 ms, and 1000 ms with three games at each latency. All games took similar amounts of time (around 2 minutes each). The first phase (about 30 seconds long) of the combat games mostly involved the two armies moving towards each other, so there were few user commands and little network traffic. Thus, we removed the first 30 seconds of data from each trace for all subsequent analysis.

Table 3 shows the mean number of packets sent and the standard deviation across the three runs for each latency. Also shown is the mean bandwidth

Latency	Packet Count		Bandwidth (Kbps)	
	Mean	Std Dev	Mean	Std Dev
0 ms	1886	230	7.2	0.6
500 ms	550	292	2.3	0.6
1000 ms	255	123	2.1	0.5

Table 3: Packets and Bandwidth.

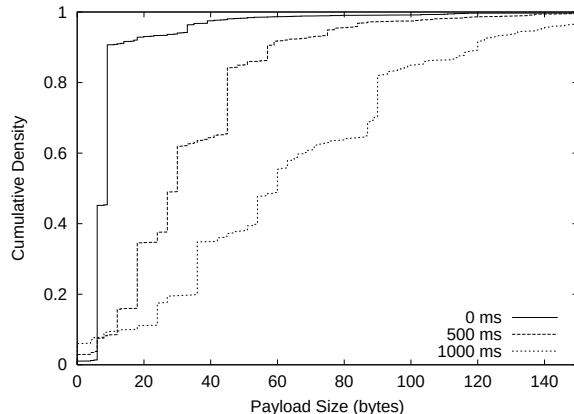


Figure 12: Combat Payload Distributions.

(including IP headers) over 500 ms intervals as well as the standard deviation. The number of packets (incoming and outgoing) decreases as the latency increases, with the combat games with 500 ms and 1000 ms of latency sending only about 1/3rd and 1/8th as many packets, respectively, as the game with no added latency.

The 0 ms latency combat game produces about the same bandwidth as does the full LAN game, shown in Figure 9. The 500 ms latency and the 1000 ms latency combat games have about 1/4th the bandwidth as the 0 ms latency game and both the 500 ms latency and the 1000 ms latency games produce less bandwidth than do the Battle.net games. This data suggests that Warcraft III bandwidth decreases with an increase in latency up to 500 ms, but remains constant for latencies beyond 500 ms.

Figure 12 depicts the CDFs of the payload sizes for all packet traces (incoming and outgoing), grouped into the three latencies. The median payload sizes increase from 9 bytes at 0 ms, to 30 bytes at 500 ms and to 60 bytes at 1000 ms. Less than 10% of the packets for any game are empty ac-

Latency	Commands Payload
0 ms	45.2 Kbytes
500 ms	46.3 Kbytes
1000 ms	45.0 Kbytes

Table 4: Commands Payloads.

knowledgments (payload size of 0). Overall, the distributions vary considerably with latency with higher latencies having larger packets. This suggests that at higher latencies, there is command aggregation at either the TCP or application level, meaning more Warcraft III commands are placed into each IP packet.

Based on Warcraft III traffic analysis during our pilot studies, we assume that there is an application overhead of 6 bytes for each packet issued, possibly used by Warcraft to indicate command sequence numbers or timing information. If we remove this overhead from the traces by subtracting 6 bytes from each packet, we can assume the “left-over” payloads are the result of user commands. Table 4 shows the sum of the command payloads over all the traces for each latency. The sum of the command payloads is very similar for each latency, which suggests that the commands issued by users are very similar, regardless of the network latency.

4.3 User Level Analysis

While we did not provide a way to quantify player perceptions, we did note game player comments and observed trends during and after our user studies.

Players observed that it was relatively easy to adjust their strategy to compensate for latencies between 0 ms and 500 ms. The game still appeared to run smoothly, and although the delays in executing commands were perceptible, it was relatively easy to estimate this delay and react accordingly. For latencies above 800 ms, the game appeared erratic which made for a degraded game experience. Without a short response time when executing commands, gamers found it difficult to implement particular strategies.

The exact point at which a player perceived a degraded game experience was between 500 ms and 800 ms but varied from person to person based

on strategy and skill level. A strategy that relied heavily on micro-management of units was more sensitive to latency than a strategy that was less focused on individual unit control. What game aspects that a player chooses to micro-manage also had an effect on how perceived latency affected the gaming experience. A player that micro-managed the town rather than the combat units was much less likely to be aggravated by latency than a player that micro-managed units in combat. Also, a mistake during combat that appeared to be the result of high latency was viewed, rightly or wrongly, as more serious than a mistake during building.

Thus, while latency does not necessarily affect the outcome of a Warcraft III game, if high enough, perceived latency does affect a user's gaming experience.

5 Conclusions

Understanding the effects of latency on application performance is important in order to design networks that meet application requirements. The growth in interactive network games demands better understanding the effects of latency on user performance in network games.

In this work, we investigated the effects of latency on user performance in Warcraft III, a popular Real Time Strategy (RTS) game. We divided RTS games into their fundamental components of build, explore and combat and designed experiments to isolate and measure the effects of latency on each component.

We find that overall user performance is not significantly affected by Internet latencies ranging from hundreds of milliseconds to several seconds. There is some statistical correlation between latency and the exploration game component, but the overall impact is minimal and there is even less correlation between latency and building and between latency and combat.

While these results are, at first glance, somewhat surprising they can be explained by the fact that the nature of RTS game play emphasizes strategy more than the interactive aspects. While Warcraft III is played in real-time, reaction time plays a small role compared to understanding the game, knowing a campaign map, and having a good strat-

egy. Since RTS user strategies take seconds or even minutes to carry out, the effects of typical network latencies (less than a second) do not impact the overall outcome. This relative insensitivity to latency is further illustrated by Warcraft III's use of TCP as the underlying transport protocol. TCP retransmits lost packets, with the retransmissions increasing latency on the order of a round-trip time, at best, and several seconds (upon timeout) at worst. Overloading at the game server is another factor which potentially adds to game latency. The fact that Warcraft III's game play over the Internet via Battle.net is very effective further underscores the lack of significant impact of latency on game outcome.

Overall, in terms of general classification of traffic, RTS games do not have the very strict latency requirements (on the order of hundreds of milliseconds) of audio-conferencing or First Person Shooter network games, but instead have latency requirements most similar to that of Web browsing (on the order of seconds).

At the network level, Warcraft III basically produces small, regularly-spaced packets and modest aggregate bandwidths which make it suitable for play over a low speed modem. At higher latencies, Warcraft III aggregates multiple commands in each packet, resulting in fewer, but larger packets. By placing multiple commands in each packet, Warcraft III amortizes the overhead of each IP header cost, thus reducing network bandwidth slightly. Our network analysis suggests that the aggregate of user commands sent are comparable over a range of latencies.

6 Future Work

The component-based studies presented here do not allow users to choose long-term strategies. Evaluating the effects of latency on how users choose what components to micro-manage, how they select and form long-range, even full-game strategies may provide insights beyond the results presented here.

While the results in this project have focused on Warcraft III, we assume that they generalize

to other RTS games (such as Age of Empires¹⁴ or Command and Conquer¹⁵) as well, since most RTS games have the same fundamental components (build, explore, combat) and similar user-interaction models. Studies to confirm this through select user-studies, possibly less extensive than those presented here, would be useful to verify our assumption and help generalize our results.

The effects of latency on user performance in other game genres, such as First Person Shooters or Massively Multi-player Online Role Playing Games, is also still an open issue. However, it is clear that several network games consist of distinct phases which vary greatly in their interaction model and hence network behavior. The component-centric methodology presented here, which entails categorization of the game play and running of controlled users studies in each category, can perhaps be applied to these games as well, in order to increase overall understanding of network games.

References

- [1] Mark Allman. A Web Server's View of the Transport Layer. *ACM Computer Communication Review*, 30(4), October 2000.
- [2] Grenville Armitage. Lag Over 150 Milliseconds is Unacceptable, May 2001. [Online] <http://gja.space4me.com/things/quake3-latency-051701.html>.
- [3] Paul Bettner and Mark Terrano. 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond. *Gamasutra*, March 2001. [Online] <http://www.gamasutra.com/features/20010322/terrano02.htm>.
- [4] Nina Bhatti, Anna Bouch, and Allan Kuchinsky. Integrating User-Perceived Quality into Web Server Design. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *IETF Request for Comments (RFC) 2475*, December 1998.
- [6] U.S. Census Bureau. Home computers and internet use in the united states, September 2001. [Online] <http://www.census.gov/prod/2001pubs/p23-207.pdf>.
- [7] Mark Claypool, David LaPoint, and Josh Winslow. Network Analysis of Counter-strike and Starcraft. In *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC)*, April 2003.
- [8] Chris Haag. Targeting, a Variation of Dead Veckoning, May 2001. [Online] <http://www.gamedev.net/reference/articles/article1370.asp>.
- [9] Tristan Henderson. Latency and User Behaviour on a Multiplayer Game Server. In Jon Crowcroft and Marcus Hofmann, editors, *Proceedings of the Third International COST Workshop (NGC 2001)*, number 2233 in LNCS, pages 1–13, London, UK, November 2001. Springer-Verlag.
- [10] International Digital Software Association. Economic Impacts of the Demand for Playing Interactive Entertainment Software, 2001. [Online] <http://www.idsa.com/releases/EIS2001.pdf>.
- [11] Peter Lincroft. The Internet Sucks: Or, What I Learned Coding X-Wing vs. Tie Fighter. *Gamasutra*, September 1999. [Online] <http://www.gamasutra.com/features/19990903/lincroft01.htm>.
- [12] Dmitri Loguinov and Hayder Radha. Measurement Study of Low-bitrate Internet Video Streaming. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.
- [13] S. McCreary and k claffy. Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. In *Proceedings of ITC Specialist Seminar on Measurement and Modeling of IP Traffic*, pages 1 – 11, September 2000. [Online] <http://www.caida.org/outreach/papers/2000/AIX0005/AIX0005.html>.
- [14] Yu-Shen Ng. Designing Fast-Action Games for the Internet. *Gamasutra*, September 1997. [Online] <http://www.gamasutra.com/features/19970905/ng01.htm>.
- [15] Blizzard Entertainment Press Release. Warcraft III - Shatters Sales Records Worldwide with over 1 Million Copies Sold, October 2002. [Online] <http://www.blizzard.com/press/020722war3.shtml>.

¹⁴<http://www.microsoft.com/games/empires/>

¹⁵<http://westwood.ea.com/>