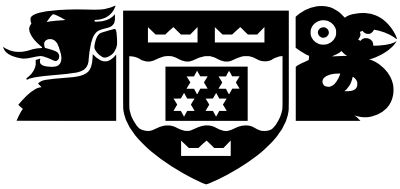
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui



School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Client-Server Model for Preserving Old Computer Games

Tania Jacob

Supervisors: Dr Stuart Marshall and Dr Ian Welch

Submitted in partial fulfilment of the requirements for
ENGR489 - Bachelor of Engineering in Software
Engineering with Honours.

Abstract

Computer games are a vibrant and important part of our culture and economy. Some old computer games are no longer able to be played. This is because the hardware necessary to run old computer games are becoming obsolete. Another barrier to playing old computer games is the fact that they are not yet in the public domain. This project explores a method to solve the issue of obsoletion and intellectual property through client-server emulation of old systems. A real world scenario where this game will be used is an exhibition setting where users can play these games via a web browser on desktops. Built in browsers can act as clients to servers where existing emulators are hosted. A benefit of taking this approach is simplified deployment of emulators and more control over game distribution. This latter aspect of tightened distribution control may act as an incentive for copyright owners to donate their game software to these archives. We contribute THPGame: a prototype of our distribution mechanism that specifically targets performance and usability.

Contents

Figures

Acknowledgments

Firstly I would like to thank my supervisors Dr Stuart Marshall and Dr Ian Welch for their continuous support and guidance throughout the project in their expert domain areas. Without their technical guidance I would not have been able to got to this point with my project. Also thanks goes to my family and friends for their ongoing support.

Chapter 1

Introduction

Digital cultural preservation of old computer games is a growing area of interest. Obsoletion of the original hardware and operating systems used to run these games has caused this issue. In addition, the access to these Microbee games is not available on the public domain due to copyrights. THPGame, a casual and competitive gaming system with synchronised video display on a web browser has been developed. This allows old computer games code to be executed on a server and the display is exported to a client running in a web browser. At the same time users are able to experience 'historic' user interfaces [31]. Easy deployment of games to a server is possible through the delivery mechanism that was developed. Deployment for this project is on the desktop with access via a web browser. The long term aim is to produce a usable user interface and delivery mechanism where performance is indistinguishable to human perception whilst multi player games are played either locally or via a wireless internet connection. In addition, producing a usable user interface is targeted. Performance testing and a heuristic evaluation will be conducted to test the system. Results of this project contribute to the larger Play It Again (PIA) project which aims to deploy these games in an exhibition setting to make them available to be used on modern platforms. This will also allow for geographical relevance to occur. [40].

The context for this work is in an exhibition setting, where visitors can play games on the available desktop machines. It would be expensive to develop or port a range of existing emulators to these platforms or others as they are developed. Utilising a client server approach means only the client needs to be ported and existing browsers can act as the client. Over the years, preference has grown towards using HTML5/ JavaScript to deliver web applications via a web browser. Consequently, a front end web interface is built that allows for remote access into the emulators which run on a central server. Permission is not granted for the client to download games meaning the source code is protected.

1.1 Motivation

The ability to play old computer games is an eye-opening experience for the new generation. Unfortunately, there were two major barriers that prevented access to these games which were obsolete hardware and copyright law. Obsoletion of hardware was an issue as future hardware and software are not generally backwards compatible. Also, copyright laws are a barrier as it is illegal to make a copy of media. Consequently, such games tend to be never played again. Using an emulator provided a solution to solve having obsolete hardware by running these old computer games that were working on outdated platforms. However, the longevity of copyrights for 50 years remains an issue [27]. Copyrights pre-

sented a problem as this meant games could not be copied or reused without the copyright owner's permission. A need therefore existed to develop a system for easier game deployment to a common location where copyright owners did not feel they lost control over the games they owned.

1.2 Key Issues

The project targets to solve the following issues:

- I1: No access to play old computer games on the desktop due to intellecutal property rights
- I2: Games not having synchronised audio and visual components
- I3: Users needing a collective place to share gaming experiences and allow for playing as a social construct
- I4: All emulators are not cross platform compatible (run on all operating systems)

1.3 Contributions

Contributions that the system envisages to make are:

- Architecture for transmitting synchronised audio/video over Wi-Fi in a shared environment such as a museum
- Proof of concept prototype that demonstrates modifying an existing emulator to support video transfer in a client/server model.
- Evaluation as to whether performance of the model is noticeably detected by human perception and if a usable user interface has been developed.

Chapter 2

Background

This section will give insight about games and their preservation. A strategy of digital preservation known as emulation will then be discussed followed by related work. Lastly I will give a background about the computer platform utilised for this project and the project methodology adhered to.

2.1 Importance of Videogames

Games are a vibrant and important part of our culture and economy, despite its history in the youth culture. [31]. They are not only growing in significance in a cultural, political and economical sense, but also are becoming legitimate objects of academic enquiry [25]. Growing confidence in using a computer is possible through playing games [34]. Massive multiplayer online role playing games (MMPORG) are a new generation of games developed that have dramatically increased the concept of virtual worlds [35]. An increasing number of people in society are playing games: around 65 percent of American households own console or computer games. Evidently, the gaming target market is not primarily teenage boys since the average game player is 35 years old. It has also been found that a larger portion of women aged 18 or over play more games (33 percent) than boys aged 17 or under (17 percent) [43]. Preserving games is therefore a good idea as it is both popular and playing games is pleasurable to humans [34]. Measures are being taken to preserve such digital media.

2.2 Digital Preservation

Digital Preservation is the process of keeping electronic material accessible and usable for a certain period of time [33]. Inaccessibility or obsoletion of digital objects due to anticipated changes in technology drives such an issue [43]. Digital games are effectively a living mirror of any given society and should be preserved[2]. New Zealand's digital gaming history includes a significant quantity of locally written software from the 1980s and not many people know about the existence of such software. [34] Therefore, in order to keep the content of such games alive, accessible and functionally as authenticated as possible, digital preservation must be used.

Three digital preservation strategies that exist are:

- Refreshing: Process that updates storage mechanisms
- Migration: Process which converts data to a newer not obsolete format

- Emulation: Keeps data in the original format and recreates the rendering environment for the digital object in a different form

Migration and emulation were listed as the main strategies for digital preservation according to the United Nations Educational, Scientific and Cultural Organisation (UNESCO) guidelines. Over time, the embedding of games presents the following difficulties in game preservation:

- Availability of game source code
- Changes in display technology
- Game versioning
- Keeping the original look and feel of a game
- Legal Issues
- Missing Documentation of console system hardware
- Preserving the multiplayer experience and behaviour of players

Emulation is the most commonly accepted digital preservation method for many kinds of digital artifacts including variable media art and videogames [43].

2.3 Emulation Process

Emulation is known as the 'heart of software preservation' aiming to recreate the functionality of original hardware in an accessible form [2]. Keeping data in its original, unaltered form whilst using its associated software occurs in the process. The operating system must run on the hardware for which it was developed for and at the same time the software has to run on the same operating system. Evidently, to keep this chain alive, emulators exist [11]. In other words it provides a duplication of functionality of one system to another to run a file of an outdated version or format.

Emulation results in a a form of loss since it focuses on surface reproduction but changes the artifact's underlying computing environment. Immersion and interaction are key features of digital games which need to be maintained [2]. Elimination of the need to install additional software is removed through using emulation services [31]. Gaming consoles such as Sony PlayStation3 and Xbox360 have their own emulators.

Emulation as a service (EaaS) can help simplify the digital preservation process and access strategies through end users interacting with emulators remotely through standardised (web) clients on their various devices. However the correct channels and bandwidth must be provided through a network link to allow remote emulation to properly function [30]. For this project, Microbee will be the platform upon which emulation services will be performed.

2.4 Microbee Platform

Microbee was the first commercially marketed Australian PC manufactured by Applied Technology in June 1982. It ran off MicroWorld BASIC DGOS (David Griffiths Operating

System) and its introductory price was AUD399.00 for the original kit in 1982. Later it was sold assembled in Australia and Sweden. Use of the Microbee computer started in Swedish schools. Specifications for the Microbee was a memory of either 16 or 32 kilobytes and a Zilog Z80 CPU (with two MegaHertz clock speed). Originally its architecture consisted of a two board unit. One board had a keyboard, Zilog Z80 microprocessor, Synertek 6545 CRT controller, two kilobytes of screen RAM, two kilobytes of character ROM (128 characters) and two Kilobytes of Programmable Character Graphics (PCG) RAM (128 characters). A byte within the screen RAM addressed a character in either the character ROM or PCG RAM. Whilst, the other board (core board) had the memory and eventually had a floppy disk controller. The Microbee had graphic characters, programmable on the same 8 x 16 pixel pattern as screen characters (ASCII)[12]. Implementation of programs enabling a user to draw fine lines at any orientation and curve was made possible due to the existence of the graphic characters. A ROM formed the architecture of the Microbee in addition.

The Microbee was successful in its initial years. Competing with Apple which provided superior functionality and range of software was infeasible. Within a year of the Microbee's first release, the IBM PC (the first Intel based PC) was released. The manufacturer of Microbee was slow in upgrading the early Microbee to have full graphic capabilities making the Apple computer more useful for school purposes. Officially the use of Microbee discontinued in 1990 [36].

2.5 Related Work

2.5.1 Past Projects

In 2007, a pilot project partly funded by Victoria University was carried out involving my project supervisors which focused on preserving early examples of Sega game software. The aim of this particular project was to emulate and port one software title to a contemporary mobile device with the permission of rights holders [34].

In 2010, Jay Shepherd carried out a similar project of developing a web interface to a game emulation server. Jay's project allowed the use of a software without distribution of the executable . The project that I am carrying out builds upon this foundation by the provision of video streaming.

2.5.2 Guacamole

Guacamole is an open source well established remote access method for audio and video streaming and input handling [30]. It is an HTML5 remote desktop gateway that translates remote desktop protocols such as Remote Desktop Protocol (RDP) or Virtual Network Computing (VNC) for direct browser access without the need of a special browser plugin [2]. Using Guacamole allows emulation as a service to be independent of the actual hardware platform used on a wide range of end user devices without particular software requirements. Performance of Guacamole is not optimal and the need for an application specific protocol exists. A facility therefore to run old computer games where network lags remain undetectable to the user has been developed. [10].

2.5.3 Sony Gaikai and PlayStation4

Sony Gaikai is a high quality, fast and interactive cloud streaming platform in the world where games can be streamed to internet connected devices. Games are uploaded to datacentres around the world and streamed using high-end servers to internet connected devices. This is similar to the method used to stream videos to computers except interactively [32]. Sony's PlayStation 4, due to be released, will take advantage of Gaikai's streaming service. Gaikai will give the capability to instantly plan, share and purchase games when necessary. However the closed nature and narrow range of supported emulation by Sony Gaikai and PS4 makes it unsuitable for this project [6].

2.6 Methodology

A Gantt Chart was used to track the progress of work. Weekly meetings were held with my supervisors roughly lasting one hour. In each meeting, work completed was initially discussed. Targets of work to be finished by the next meeting was also established. Effectively this was an iterative development cycle with weekly sprints of work. A focused approach to help track progress was provided through using an iterative approach. Difficulties discussed later affected my progress. Despite setting target dates that lightly followed a waterfall approach, alternative methodologies may have not be suitable [20].

User centered design was used to design my system. [1]. With such an approach, goals and typical usage of the system were considered whilst developing the system. To support my work, fictional representations of potential users called personas were created. These outlined the possible interactions and actions performed by a user to help gain better focus and understanding of a user [29]. Clear communication of potential users motivations, behaviours, environment and goals were covered in these personas which are explained in Chapter 3. Usage centered design was not selected because a focus of the project involved providing a good gameplay environment for users whilst being to play old Microbee games.

Chapter 3

Requirements Analysis

A requirements analysis phase was conducted using the user centered design approach. It is the closest de-facto standard in industry for modelling user interfaces. Before this, background research was carried out through a pre-study questionnaire. From the data gathered and project issues, functional requirements and non functional requirements were set.

3.1 Background Research

To set the foundation for future decisions to be made, a pre-study questionnaire was carried out. Data gathered from the questionnaire helped determine which browser to run the system on and helped gather potential participants for the user testing phase of the project.

Content of the Pre-Study Questionnaire

The objective of the questionnaire was to gather data about potential system users mobile phones, phone activities/experiences and gaming interests/experiences. Participants were recruited via a Facebook event and were aged between 18 - 25 years old. Consent had to be received from the head of school to run the questionnaire. An information sheet (Appendix B), alongside a consent form (Appendix C) was submitted with the pre-study questionnaire (Appendix D).

Results obtained from Pre-Study Questionnaire

From the sample of 94 participants taking part in the survey, 40'%' were male and 60'%' were female. Most of the participants either studied: Engineering/Computer Science, Commerce or Law. Out of this group, 42'%' owned Android devices (majority on Android 4.2 - Jellybean operating system), 36'%' owned iPhones (primarily iPhone4), and the remaining 22'%' did not own a smartphone. A minute 0.40'%' difference existed between the user rating levels of iPhones and Android devices. Google Chrome proved to be the most popular web browser.

The second half of the the questionnaire was focused on gaming and 76'%' of users played games. Such result meant most people filled out answers to the second half of the survey. At least 65'%' of the participants played either competitive or social games (or both in some cases). Most people gamed on PC (77'%)', closely followed by portable devices such as cellphones and ipods (75'%'). To my advantage, 55'%' of the participants had taken part in server based games in the past such as multiplayer pacman with 49'%' having experience

in detecting network performance lags. This was pleasing as it showed hope for having potential participants to take part in the user testing.

3.2 Users Models

A game player is the primary user for the system. This user will have the capability to do tasks such as register user, play game, view player statistics and suggest a new game to add. Two types of game players exist. A competitive game player uses THPGame for its gaming facilities (including leaderboard). Whilst a casual game player is interested in playing historic games with no real intentions of getting top of the score table.

A game portal administrator is the secondary user for the system. Adding games to the system is their job which needs minimum knowledge of Javascript. Personas describing potential users and their associated user scenarios will now be outlined.

3.2.1 Personas and Scenarios

To follow the standard practices of user centered design, it is important to describe personas about the potential users. Forming context scenarios based on these personas developed is another aspect of user-centered design. These scenarios describe the user's environment, the system's interaction with the persona's life and help ascertain the goals of such user. Key-path scenarios are the most relevant scenario describing the persona's interaction with the system and focus more on system detail than context scenario [23].

Casual Game Player

Matthew is a 48 year old casual game player who is interested in old computer games. He has an active interest in the history of computers and is working as a developer at BNZ. Matthew read about the Microbee Preservation Projects and came to know about the development of THPGame. He wants to gain knowledge on how old Microbee computer games run with no real aim of winning or being good at games. Viewing the high scores is less of a concern for him.

Scenario 1: Register User
Matthew has an interest in playing old historic games and wants to run THPGame. But first needs to register as a user on the system.

Goal: Matthew needs to register as a new user to access the functionality of THPGame.

Keypath Scenario:

1. Matthew is presented with the Login page
2. He opts to create a new user
3. Matthew enters his details to complete registration and is presented with a success message
4. Matthew is prompted to enter his new details to login to the system

5. Providing his credentials validate, he is redirected to the Home page as a registered user

Scenario 3: Play Game
Matthew wants to experience playing old Microbee games. Games requiring less skills are the games he prefers playing unlike the case for TJ.

Goal: Matthew is interested in playing Microbee games on THPGame.

Keypath Scenario:

1. Matthew logins into THPGame with his username and password
2. He navigates to the Games page via the navigation bar
3. Reading the game descriptions he takes note of the particular games that currently exist
4. Matthew selects the game he wants to play and follows the instructions to start playing

Competitive Game Player

TJ is a 22 year old competitive game player who loves gaming. She is looking to build her gaming skills and score high. As a commerce student she does not have technical knowledge beyond HTML coding. Despite this, she is competent with computers so will be able to register herself and play games. Regularly, she will check the high scores table to see how she fares against her friends. Responsibilities she has includes playing the games. Encouragement is made to TJ to suggest new games to add to the system and email requests to the administrator at thpgameadmin@gmail.com.

Scenario 1: Register User
Recently, TJ discovered THPGame, a new system where she could play games. However, she needs to register herself on the system.

Goal: TJ needs to register as a new user to play the Microbee games on THPGame

Keypath Scenario:

1. TJ is presented with the Login page
2. She opts to create a new user
3. TJ enters her details to complete registration and is presented with a success message
4. She is prompted to enter her new details to login to the system
5. Providing her credentials validate, she is redirected to the Home page as a registered user

Scenario 2: View Games List
TJ wants to know the games available on THPGame. Game descriptions on the Games page help her decide which games she wishes to play.

Goal: TJ wants to see the games available in THPGame.

Keypath Scenario:

1. TJ logins into THPGame with her username and password
2. She navigates to the Games page via the navigation bar
3. TJ reads the game descriptions on the left hand panel of the page.

Scenario 3: Play Game
Upon deciding a game to play (scenario 2), TJ wishes to play a game on THPGame.

Goal: TJ wants to play inaccessible Microbee platform games to achieve high scores.

Keypath Scenario:

1. TJ logins into THPGame with her username and password
2. She navigates to the Games page via the navigation bar
3. TJ selects the game she wants to play and follows the game play instructions

Scenario 4: View Player Statistics
TJ wants to view her friends high scores on THPGame. She wants to know whether she needs to improve her strategy or if she has a significant lead. Her ultimate aim is to remain at the top of the table.

Goal: TJ wishes to see her friends high scores

Keypath Scenario:

1. TJ logins into THPGame with her username and password
2. She navigates to the Player Statistics page and views the score table

Scenario 5: Suggest Game
TJ has discovered a new game that she wants added to THPGame. She believes a competitive challenge can be setup through adding a game that interests both her friends and herself.

Goal: TJ wishes to add a new game to the system

Keypath Scenario:

1. TJ sends an email to the THPGame administrator with the software attached and associated parameters for tape loading.
2. The administrator adds the game to the system

The Administrator

Kelsey is a 27 year old Computer Science graduate. She has a full time job at Datacom as a Ruby on Rails developer. Her work responsibilities do not allow her to fulfill her interest in monitoring and administering others actions. Therefore, she accepted the job to be the administrator for THPGame. Every six months she has the duty to email the current users of THPGame to ask if they want any new games added to the system (unless someone sends suggestions on a regular basis). Software for such games need to accompany such email.

Scenario 6: Add Game
Kelsey needs to fulfill her duties as an administrator on THPGame upon receiving an email request.

Goal: Kelsey has to add a new game to THPGame

Keypath Scenario:

1. Kelsey gets an email request to add a game to the system
2. She adds the game details to the software file in the source code
3. The new game software is added with the existing games
4. On the Games page source code, Kelsey adds a game description and allows the game name to display on the page
5. Logging in as an administrator, Kelsey checks the game successfully runs

The use case diagram for THPGame is as follows:

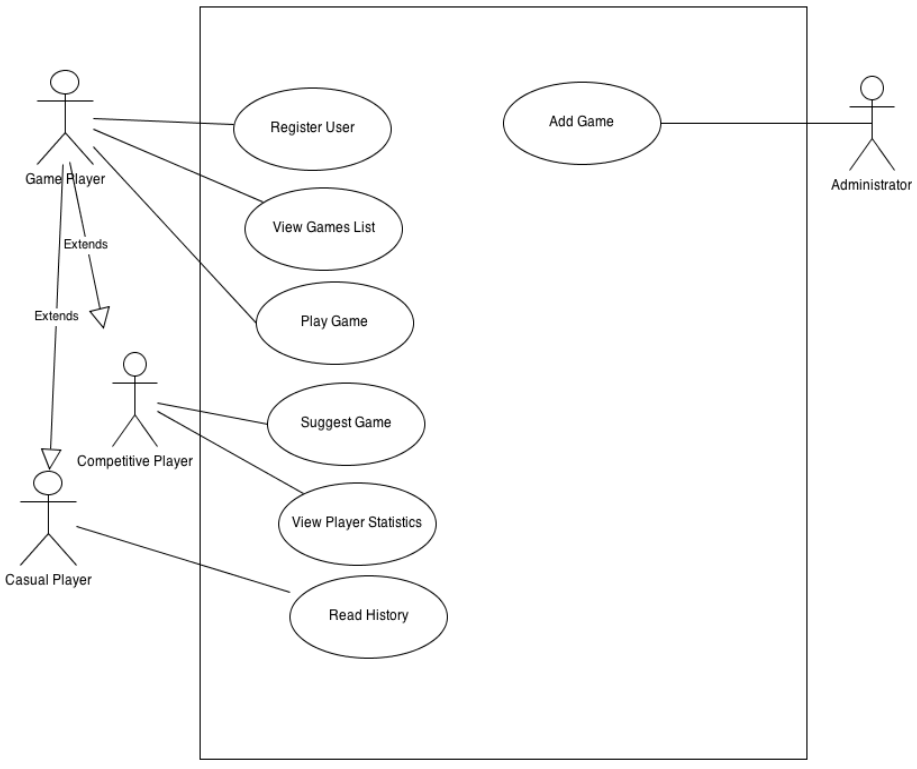


Figure 3.1: Use Case Diagram for THPGame

3.3 System Requirements

3.3.1 Functional Requirements

The following five functional requirements will be met by the system:

F1: Run Microbee Platform Games

Microbee games were only accessible via a website: www.nanowasp.org. Obsoletion of hardware and legal constraints caused such inaccessibility. The solution should run Microbee games and meet Matthew's desire to experience ancient games.

F2: Portability to the Desktop

THPGame's deployment to the desktop will be possible through a web browser. All the personas outlined will be able to access the system on their PC. This functionality is important to Kelsey as she might prefer to do her work on a PC as opposed to a mobile device.

F3: Cross-Platform Solution to run Games

A necessity is for the games to run off a cross platform emulator. In effect this would support the museum environment where there could be a range of computers available. This means the solution would work on any operating system (Windows, Mac OS X) causing no compatability issues. Restrictions are therefore not imposed on potential users such as TJ to use the system.

F4: Multiplayer Game Handling

The system will allow for multiple people to connect to a server and run their own instance of the game. Naturally this seems suitable for a game playing system. Competitive players such as TJ would be interested in having such a feature to serve her needs.

F5: Store Persistent Game Session Data

Storing game session data is necessary in order to provide an additional competitive aspect to the system. Users can target to become the highest scorer in the system. Competitive game players such as TJ would desire such requirement to be fulfilled in order to meet her expectations of a gaming system.

3.3.2 Non-Functional Requirements

Similarly, there are three non functional requirements that are targeted to be achieved by the system:

NF1: Highly Intuitive User Interface

Making an intuitive user interface will allow for easy navigation in the user interface. Games will be played off the Games page which is accessed via the navigation bar. Simplicity has therefore been considered to make the user interface highly intuitive. Having a highly intuitive user interface will aid the user experience of non technical users such as TJ.

NF2: Game Performance indistinguishable to human perception

Users should not be able to detect if they are playing THPGame with the emulator running locally or via the server. Satisfying this requirement helps provide the original game feel. Making the performance indistinguishable serves to impress technical users such as Matthew who have the ability to detect network lags.

NF3: Protection of Intellectual Property Rights of Game Software

Inaccessibility to Microbee platform games was caused by the unwillingness of copyright owners to make their games available. Microbee game software is still governed by the Copyright Act despite the age of the software so a local copy is not able to be downloaded. A built-in mechanism is necessary for protecting the system. As mentioned in Chapter 2, copyrights last for 50 years. The system would encourage current copyright owners to donate their Microbee games to THPGame. Potential users such as TJ would have confidence in telling copyright owners she has contact with that THPGame will not make the local software available to clients.

A relationship matrix is illustrated below to depict how the system requirements address the project issues:

	Issue 1	Issue 2	Issue 3	Issue 4
F1	x	x	x	x
F2	x		x	x
F3	x		x	x
F4	x	x	x	x
F5			x	
NF1			x	
NF2		x	x	
NF3 x		x		

Table 3.1: Relationship matrix showing how project issues correlate to system requirements

Chapter 4

Design of THPGame

4.1 Design Decisions

In the design phase of the project, the choice of computer platform, system architecture, web browser, deployment device and survey tool took place which will now be discussed.

4.1.1 Computer Platform

To kickstart the project, the consideration of computer platform games to run was necessary. Prior work by my supervisors, enabled us to have contact with a leading researcher in Australia: Melanie Swalwell. She recommended the scope of the computer platforms to be limited to the Microbee and Commodore 64. Melanie was a partner in the Play It Again project and had links with the developers of one Microbee emulator. Microbee was built in Australia and had local software written for it. Cultural value would have been provided through utilising Microbee. Melanie offered access to copyright free Microbee games to run that were not available online. Three emulators existed that ran Microbee games [28]. Preservation initiatives are currently in place to bring back the capability to run Microbee games.

Melanie's influence's forced us to consider the Commodore 64 as the alternative platform. It was an 8-bit home computer which was introduced in January 1982 by Commodore International. There were six emulators that ran such platform games: three Windows emulators, two Mac (OS X) emulators and one for modified Xbox [5].

Microbee was selected due to its provision of cultural value and because of the access to copyright free Microbee games. Notably, the cross compatability factor of Microbee was the driving factor to finalise Microbee as the chosen platform and this decision helps meets requirement F3.

4.1.2 System Architecture Model

The architecture for the system was client-server based. Client-server architecture was considered as this was the original specification for the project. Using a client-server architecture meant users got the information they wanted when desired. A disadvantage was that sophisticated security, system management and application development was necessary using such an architecture but this was less of a concern for this project.

Peer to Peer was an alternative architecture to use where each component acts as its own process and as both a client and a server to other peer components. In such architecture any component could initiate a request to any other peer component. Advantages of using such approach was that it scales up well, it is highly tolerant of failures and has increased system capabilities. Other architectures exist such as repository, publish-subscribe and layering but were not considered [22]. Using a client-server model helped provide the capability to meet requirement F4.

4.1.3 Web Browser

A narrow range of browsers could only be used for the project, due to the need for modern browsers to run Microbee games. The default Android browser was out of consideration as the system is only deployed on the desktop. Two methods were found to decide which web browsers to utilise for development purposes. Statistics were found online and showed the following results:

Web Browser	Usage
Google Chrome	52.70
Internet Explorer	12.70
Mozilla Firefox	27.90
Opera	1.70
Safari	4.0

Table 4.1: Web Browser Usage [42]

Results from the pre-study questionnaire also aided in deciding which web browser to use:

Web Browser	Usage
Default Android Browser	15.90
Google Chrome	53.62
Mozilla Firefox	1.45
Other	2.90
Safari	42.03

Table 4.2: Web Browser Usage

Combining the results from both sources showed Google Chrome was the most popular web browser. Hence it was decided to run the application on the Google Chrome web browser on the desktop and allowed requirement F2 to be met.

4.1.4 Deployment Device

Deployment was selected to be on the desktop as the performance of the server running on the phone is questionable. Regardless, this design decision helped meet requirement F2.

4.1.5 Survey Tool

SurveyMonkey [41] and Qualtrics were two tools investigated to run the pre-study questionnaire, with Qualtrics being heavily used by Victoria University. Stemming from this, it

seemed appropriate to use Qualtrics as it was a tool used within the university [39].

4.2 Software Engineering Considerations

Geographical relevance, reliability and reusability were considered in the design phase of the project.

4.2.1 Geographical Relevance

Geographical relevance is provided through THPGame as Microbee was developed in Australia (providing local cultural value) and so is relevant to the Australia and New Zealand region [40].

4.2.2 Reliability

Monitoring that the system will not cause identifiable network lags was necessary. Performance testing results was carefully analysed to ensure reliability. In addition, reliance was placed on getting support from contacts in Australia if we were placed in a difficult situation. Also it was expected that the system will be reliable in giving the original game experience despite the games running on a newer platform.

4.2.3 Reusability

A simple design has been considered for the project in order to make components of the system reusable in the future.

4.3 System Architecture

In order to run the Microbee games, an emulator is utilised to run on a backend server. The role of the emulator is to execute game logic and it sends video and could send audio. The server has the role to run the Microbee games and sends character images to the user interface. Users need to be authenticated in order to play the games. Verification of users is provided through checking a database if the given login details are valid. A user is similarly able to register on THPGame and their details are stored in the database. If this succeeds, users are able to play the games. Storage of these games is in a separate database so that the software is inaccessible to unauthorised users in the museum setting (meeting requirement F1). When users request a particular game to be played, a request is made to the backend server to fetch the games from the games database.

4.3.1 Component Diagram

The components of my system include a user interface which connects to a user database. This user interfaces communicates with a back-end server. The emulator is run off the back-end server and this server also communicates with the Games database.

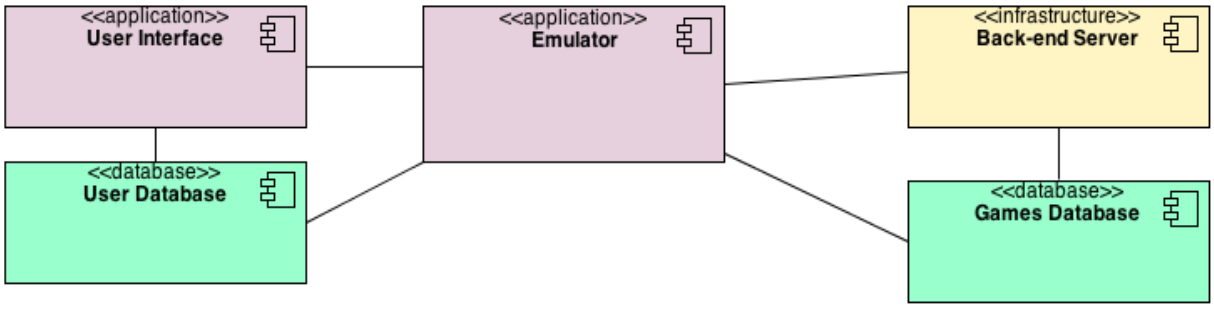


Figure 4.1: Component Diagram

4.3.2 Deployment Diagram

The deployment of my system consists of a back-end server that runs both the emulator and user interface. In addition, both the games and users database are accessed by the back-end server. This server will be deployed on many web browsers on the desktop (with the focus on serving content on Google Chrome).

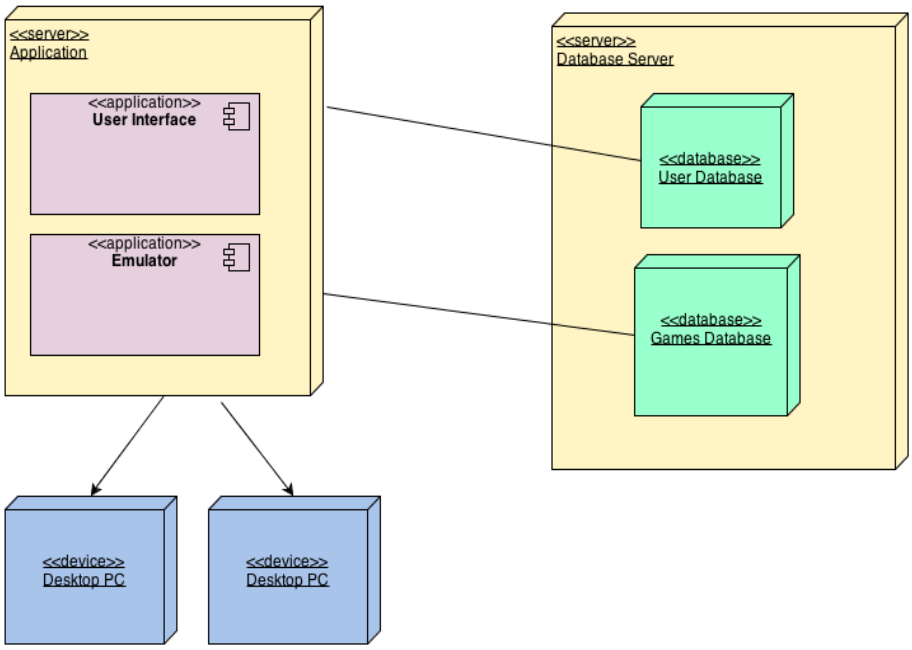


Figure 4.2: Deployment Diagram

Chapter 5

Prototype of THPGame

5.1 Scope

A proof of concept has been developed to demonstrate an implementation for the described system. The time frame of 300 hours limited the scope of implementing all components outlined in the system architecture. As a result, certain parts of the system have been developed in this project, the first of which was the emulator.

5.2 Emulator

5.2.1 Choosing an Emulator

A crucial implementation decision for the project involved selecting an emulator to run the Microbee games. Three emulators existed that ran Microbee platform games: Nanowasp, uBee512 and MESS.

Nanowasp is a Javascript based emulator that was written by Dave Churchill. It was ported from running in C to allow for increased usage of the emulator. Running Nanowasp requires a modern web browser. Source code for the emulator is available on Github at [1] and there is a website that currently runs Microbee games via the Nanowasp emulator [2].

uBee512 is an older emulator that was developed by Stewart Kay. This emulator was written in C and used as a foundation to develop the Nanowasp emulator. A vast amount of support was available using such an emulator [13]. MESS (Multi Emulator Super System) was the other option to use that was written in C++. It was based on the MAME (Multiple Arcade Machine Emulator) core which was an emulator application used to recreate the hardware of arcade game systems in software on modern personal computers and other platforms [24].

The driving factor to use Nanowasp was the greater experience of both my supervisors and myself in Javascript compared to C and C++. Although C++ is known to have similarities with Java, it was a new language to me. Having a limited knowledge of C restricted me. Evaluating my decision, a lack of language familiarisation should not have been the prime reason to discard MESS and uBee 512. Viewing the code before making the decision would have been better as the code for the other two emulators may have not been as difficult to

[1] https://github.com/dgchurchill/nanowaspjs
[2] http://www.nanowasp.org/

understand as expected. Regardless, using an emulator allowed Microbee games to be run without breaking intellectual property rights (meeting requirement F1).

5.2.2 Nanowasp Architecture

Many components make up Nanowasp as illustrated in Figure 5.1. Nanowasp.js is the primary class of the emulator connecting to the user interface and tying the components of the emulator. The emulation process occurs in the microbee.js class with the creation and connection of the emulation devices and registering of ports occurring before this process.

Another component of the emulator is the Cathode Ray Tube controller (crtc). The Cathode Ray Tube is used to view images off a fluorescent screen. Setting up the graphics context and rendering the game data to the canvas are the vital functions of the class. An ImageData object is used to represent the output consisting of a red, green, blue and alpha component. The crtc also has a memory associated with it known as the CrtcMemory which is where all the rams and roms are built with the characters and graphics context to be utilised. Characters are built, stored and retrieved from this class. The roms and rams are registered into the memory within the memory class. Memmapper is the last memory class which registers z80 memory devices and maps memory addresses to values.

In regards to the games and software utilised, the software class registers the tapes that will be run off the system. The tape injector class adds tapes to the z80 cpu. The virtual tape class registers new tapes that are added to the system so that they can be used in the software class.

Key mapping is necessary via a keyboard class as punctuation characters on a standard keyboard are laid out differently than Microbee keyboards. As a result a special keyboard is needed. Other additional classes exist including utils, latchrom, z80cpu, tapeview and debugger.

Compilation of the javascript files, roms, rams and software is completed by the Makefile to build the emulator as shown in Figure 5.1. The Makefile takes the source files and generates files in the debug directory (which acts as the output directory). A range of files are produced in this process. To note: three javascript files are produced: nanowasp.js, z80.js and data.js(containing the roms and rams). As well as this an index.html page is produced off which games can be played provided there is a back-end server.

Rendering character images is worth outlining in detail. It occurs within the render method in the crtc class. Data currently stored in the CrtcMemory class is rendered through calling the getCharacterData() method to retrieve the data. This particular method returns a bitmap for a single character on the screen. A calculation is made of what the bitmap looks like from the data presently stored in the PCG-RAM (Programmable Character Graphics) or the character ROM. The built-in font is stored in the character ROM. Data in the PCG-RAM can be modified by the emulated program. A single call of putImageData() is used to render pixels to the canvas [4].

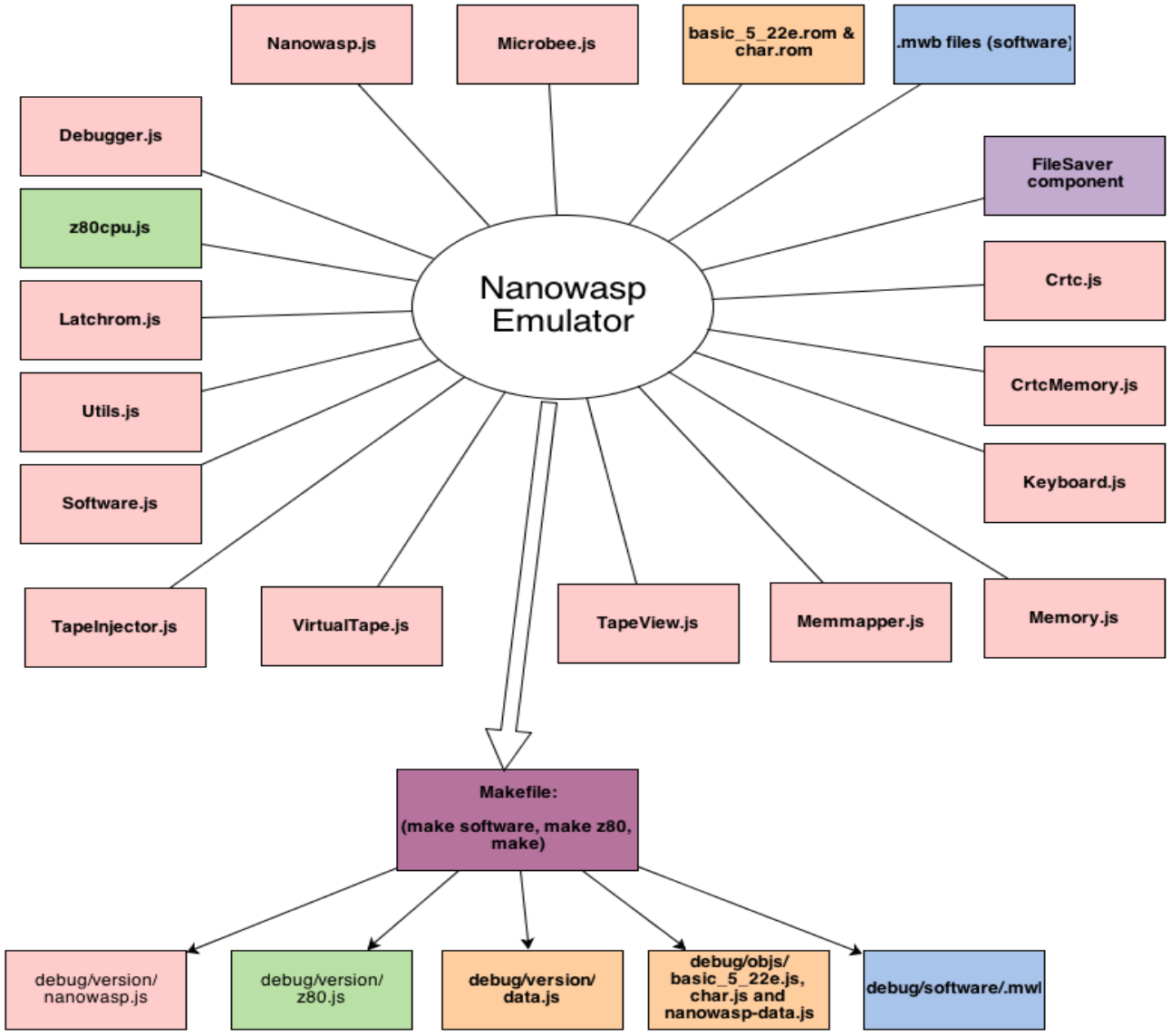


Figure 5.1: An illustration of the building of the Nanowasp emulator. The colours used in this picture are significant. For example the pink components all feed into the Makefile to produce a single nanowasp.js file (pink component). A similar idea happens with the groups of coloured components of the emulator illustrated.

5.3 Back-end web server

5.3.1 Selecting a back-end server

To allow for the games to run, a back-end server was required. Considering the Nanowasp emulator was written in Javascript, a back-end server that ran Javascript files on the server was necessary.

NodeJS was utilised to run the emulator as it is a lightweight and efficient technology [18]. It is built on Chrome's Javascript runtime for easily building fast, scalable network applications. The speed for a server to be setup and running with NodeJS was impressive. In addition, there was a vast amount of documentation outlining the use of NodeJS. Many NodeJS libraries existed that supported a range of functionality in addition. Scalability of NodeJS however was the key reason for using NodeJS as the back-end server because this is a non functional requirement systems attempt to achieve. [37].

Alternative back-end server technologies considered were a Ruby server (alongside having a Ruby on Rails front end user interface). Interest in learning Ruby stemmed after a group project earlier this year forced me to consider the possibility of using such a server. With my supervisors guidance, this idea was discarded because of my higher confidence in developing a Javascript/HTML based web application. Also considering the initial specification for the project at initiation was about building a HTML5/Javascript application, it was best to keep to the supervisors preferences.

The final alternative was to use an Apache web server (XAMPP-HTTP) [7]. This was used for a short time in running the emulator. A recommendation made by the Nanowasp developer drove us to use the technology. However, work with this server did not continue for long as it did not have the capability to run Javascript files on the server.

Therefore utilising NodeJS as the back-end server technology allowed for Microbee games to be run off the Nanowasp emulator without a breach of legal restrictions which satisfies requirement F1. In addition, due to its scalability and high performance it supports requirement NF2.

5.3.2 Using NodeJS

Before integrating NodeJS with the emulator components, experiments of various modules of the project occurred independent of the project code. Though this was a slow process initially, latter integration was easy in most cases.

Installing libraries was possible through simply executing the npm install 'library' command in the terminal. The libraries used in the system were as follows:

Library	Description
chalk	Adds style to the output from the terminal
connect-flash	Flash middleware for the Connect framework
consolidate	Template engine consolidation library
express	Sinatra inspired web development framework
fabric	Object model for HTML5 canvas
fs	File system
http	HTTP server
mustache	Templates to display dynamic content
passport	User authentication
passport-local	Local username password authentication
pg	PostgreSQL client for database connection
socket.io	Websocket capability for client-server

Table 5.1: Utilised NodeJS libraries [17]

Other frameworks were considered such as connect and hogan. But frameworks such as connect were not selected as they provided basic features. For these libraries, their base features were built upon in the libraries I used in THPGame so were not explicitly used.

5.4 Databases

5.4.1 Selecting a Database

Once the server and emulator were decided, it was vital to choose a database to store both the user details and games. Two options were considered: a PostgreSQL database or a NoSQL database(such as MongoDB) [9]. Consideration of a NoSQL database primarily occurred due to the use of such database in online NodeJS user authentication tutorials. The existing setup of the PostgreSQL database environment at university drove me to choose using this type of database. By using PostgreSQL, time did not need to be spent on familarisation with database query languages such as SQL.

5.4.2 Using the PostgreSQL Database

The Nanowasp database was setup in the university environment containing a users table. The fields within the table were: userid, firstname, lastname, username, password, email and score. When users need to be authenticated , the database is queried, On the other hand, a new record is added to the database when a new user registers. Beyond this functionality, users scores are extracted and displayed on the Player Statistics page from the database. Storing such persistent records of user data fulfills requirement F5.

5.5 Combining Nanowasp and NodeJS

A successful compilation of the Nanowasp emulator presented it in a state where it could be integrated with the NodeJS server. The first task to combine both the emulator and the server was to get the emulator running locally. Less time was required than anticipated in getting the emulator running locally. Games were able to be played within the user interface

developed which will be later explained. The next stage was to get the emulator running on the server.

NodeJS's incapability to run browser-oriented code, meant references to the browser within the emulator code had to be removed. Time had to be spent on performing such activity. Focus was then on rendering game frames to the client on the user interface.

The pixel rendering had to be modified in order to handle the client-server architecture. Instead of rendering images to the canvas from the server, the data to be displayed had to be sent to the client first. Consideration had to be made on the most effective method to serve data from the server to the client. Using AJAX or WebSockets were the two options available. WebSockets provided a low-latency, bi-directional option to enable a long running connection between a browser and server [14]. They worked cross platform (meeting requirement F3) and provided a good wrapping mechanism. Effectively it provided an extension to AJAX capabilities. For these reasons WebSockets were chosen to send the data.

Utilisation of the NodeJS library, socket.io, helped send the character image to be displayed on the client's screen between the server and client using JSON. As the graphics context had to be removed earlier (since it was browser-oriented code), the NodeJS canvas library fabric had to be used to cover missing functionality. Handling both sending and receiving data from the server and rendering the display through a socket existed in the html pages which will now be discussed.

5.6 User Interface Design

Developing a user interface formed the second half of the project and earlier a decision had to be made on the technology to use for development purposes.

5.6.1 Selecting a Framework

Twitter Bootstrap was utilised to develop the front end web interface for the project using HTML [38]. The standard colour theme of black and white was used in order to not make it intrusive to the eyes. Previous experience leaned me towards using Twitter Bootstrap. Bootstrap has become the modern choice of technology to use to develop a front end web interface. In addition, alot of built-in code exists meaning that a basic user interface could be up and running within a short span of time.

An alternative technology that could have been used was Skeleton. This was a small collection of CSS files with a lightweight grid as its base. It elegantly scales down to downsized browser windows, tablets, mobile phones (both in landscape and portrait). Providing the flexibility and basic styles as a foundation, it is a beneficial development kit [8]. But my familiarity with Twitter Bootstrap and the fact that the primary concern of the project was about getting the games running, forced me to keep to using Twitter Bootstrap. Easy navigation through a basic user interface helped provide a highly intuitive user interface to utilise, which meets requirement NF1.

5.6.2 Developing the User Interface with NodeJS

Four pages primarily made up the user interface: Games, Home, Login and Player Statistics. Using the NodeJS Express framework provided the backbone to handle the requests

sent from the user interface to the server with users authenticated using the passport and passport-local libraries. Socket connections and image renders were setup on the Games page. s Once the static components of the user interface pages were developed (using Twitter Bootstrap components), serving dynamic content was the next task. Using the mustache NodeJS library, it was possible to display the currently logged on user. A single file retrieval call had to be modified on the server side in order to use mustache. In such a scenario, the current user's username was sent. But in order to get the username, the Nanowasp database on the server was queried using the id returned in the deserialise method.

5.7 Prototype Integration

When the server starts up users are presented with the Login page. Here users are requested to enter their login details (Figure 5.2). To support persistent login sessions, the NodeJS passport library needs users to be serialised into and deserialised out of a session. Serialising involves storing a user's username, whilst deserialising involves users being searched by userid. A post request is sent upon a user logging in to the system to the NodeJS server and passport.authenticate is called. A query is made to validate the login details in the users table in the Nanowasp database. If the user details fail to authenticate, the user is informed and is able to re-enter their details (Appendix A Figure A.1). Users are able to create new accounts in the system if they are not already registered and their details are posted to the server and added to the Users table (Figure 5.3).

Upon successful creation of a user account, users are notified and asked to login with their new details (Appendix A Figure A.2). Successful login redirects a user to the Home page which gives information about Microbee (Figure 5.4). From this page users are able to go to the Games page to play games (Figure 5.5) which also has a list of available games (with hover descriptions implemented using JQuery). Alternatively they can visit the Player Statistics page to view high scores (Figure 5.6) for social reasons such as competition between friends. Finally when users choose to logout out of the system a request is made and the user is successfully redirected to the Login page.

Whilst the prototype works as outlined above, difficulties were encountered in implementing the system and these difficulties are discussed in the next section.

5.8 Project Difficulties

Understanding the concept of a client-server architecture was difficult to grasp at project initiation. Time had to be spent on researching and understanding tutorials demonstrating such architecture with the necessary technology. Using an Apache server as suggested by the Nanowasp developer was a challenge, however when it came to using NodeJS, things became easier. Reasoning for this is because of a better grasp of client-server architecture. Therefore my knowledge increased during the project in this domain. However, it would have been beneficial if I understood client-server architectures earlier.

Another problem faced in the project was understanding the emulator's code. Using Javascript beyond Document Object Model (DOM) code to help render HTML pages was a new experience. The Nanowasp code had a small number of comments and poor use of variable names. A lesson has been learnt in this regard to using valid comments when writing code

in the future as a similar situation may be presented to someone unfamiliar with a project.

Rendering character images to the client side from the server was the biggest difficulty encountered. Vast amounts of time had to be spent understanding the building of images to be displayed on the client. Repeatedly, the same character was rendered to the canvas. It could render individual characters correctly but there was a problem with the video RAM that prevented a program being shown correctly. Eventually, this problem was resolved to an extent.

Working on my project on the university environment presented limitations particularly due to a lack of administrative rights. Stemming from this, there was reliance on the Engineering and Computer Science technical staff. At times, these rights prevented the use of certain libraries for the project. An example of note is the inability to utilise the express framework (well established for NodeJS applications) meaning a less powerful library had to be used (connect). Since connect was less frequently used this meant that there were less libraries available to use to meet my needs. Leading from this, it meant that more time had to be spent on finding libraries suitable for Connect based NodeJS applications (less existed compared to Express libraries). Another example was for setting up a database. It was easy to request a database to be setup through an email. However, roughly one day was spent trying to connect to the PostgreSQL database that was setup and the issue was not clear. A conversation with the technical staff revealed that a special password had to be setup for my user to access the database. These issues would have been more easily tackled if I was working on my project outside the university environment.

5.9 Factors affecting Implementation

Despite facing these difficulties, whilst implementing the system legality and maintainability were two software engineering considerations that had to be pondered about.

5.9.1 Legality

Nanowasp was a free emulator that was used to run the Microbee games and was downloadable from GitHub [16]. The Microbee software had been sourced from the Nanowasp developer [28]. So no legal constraints were breached in the project.

5.9.2 Maintainability

To allow for maintainability to be achieved, Nanowasp was used so that the front end web interface and emulator were all written in one language. This means the solution is easily maintainable in future.

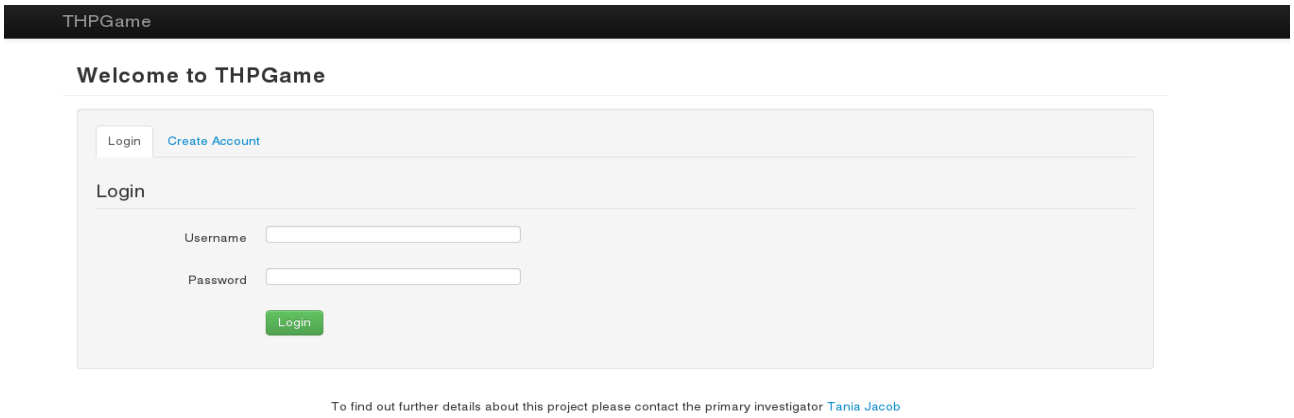


Figure 5.2: Login to THPGame

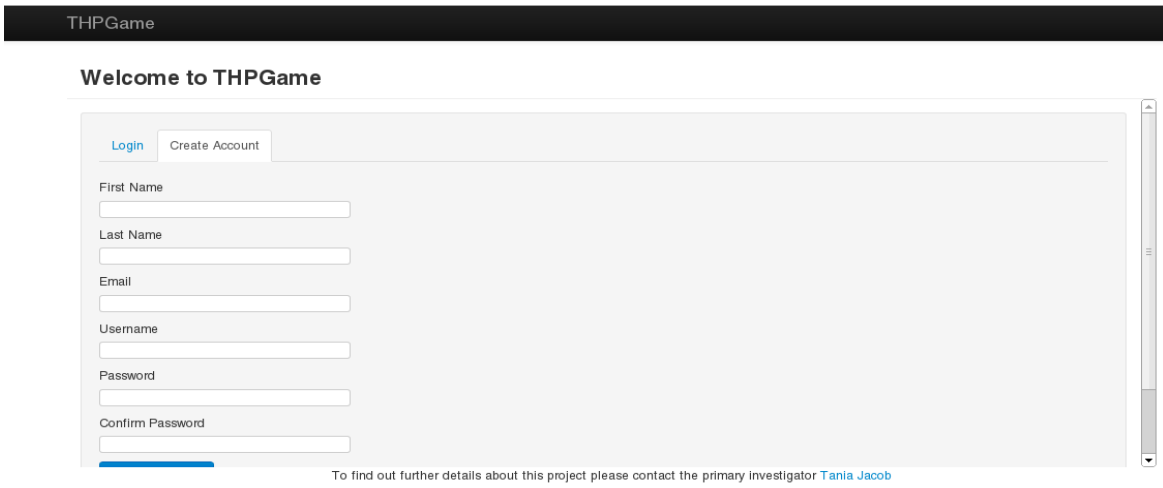


Figure 5.3: New user registration on THPGame

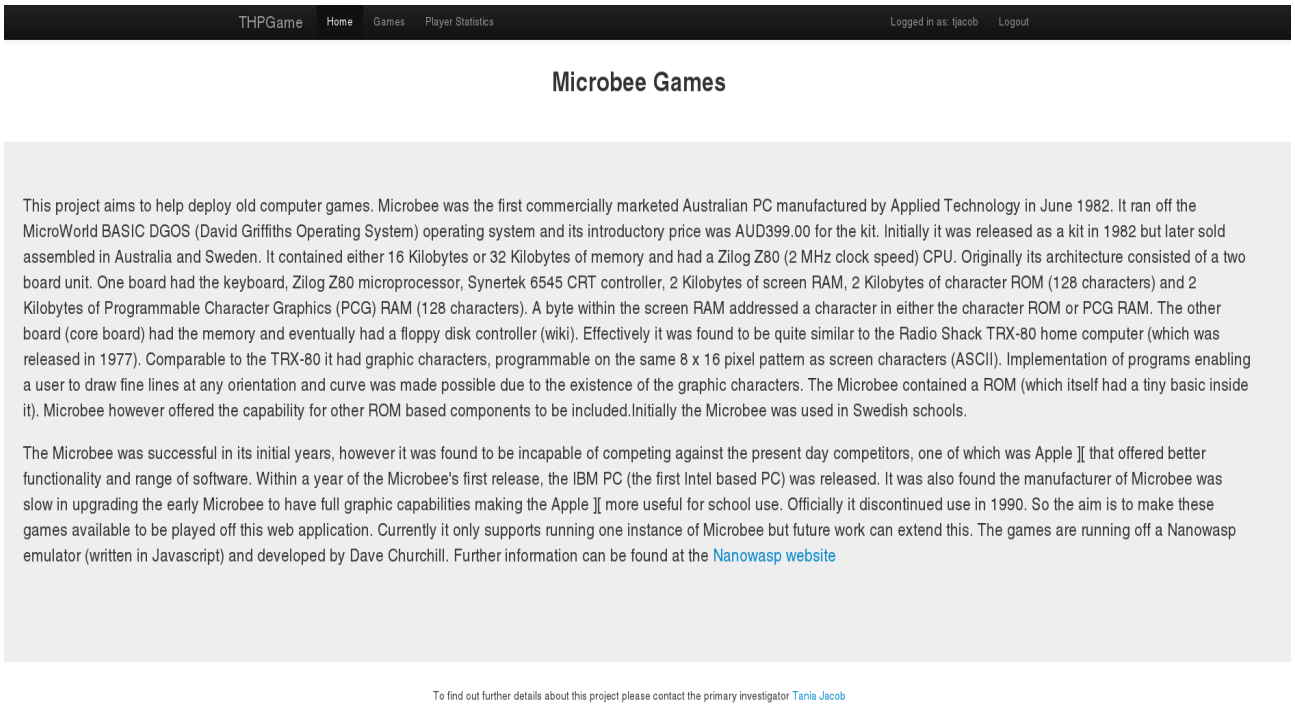


Figure 5.4: Homepage of THPGame

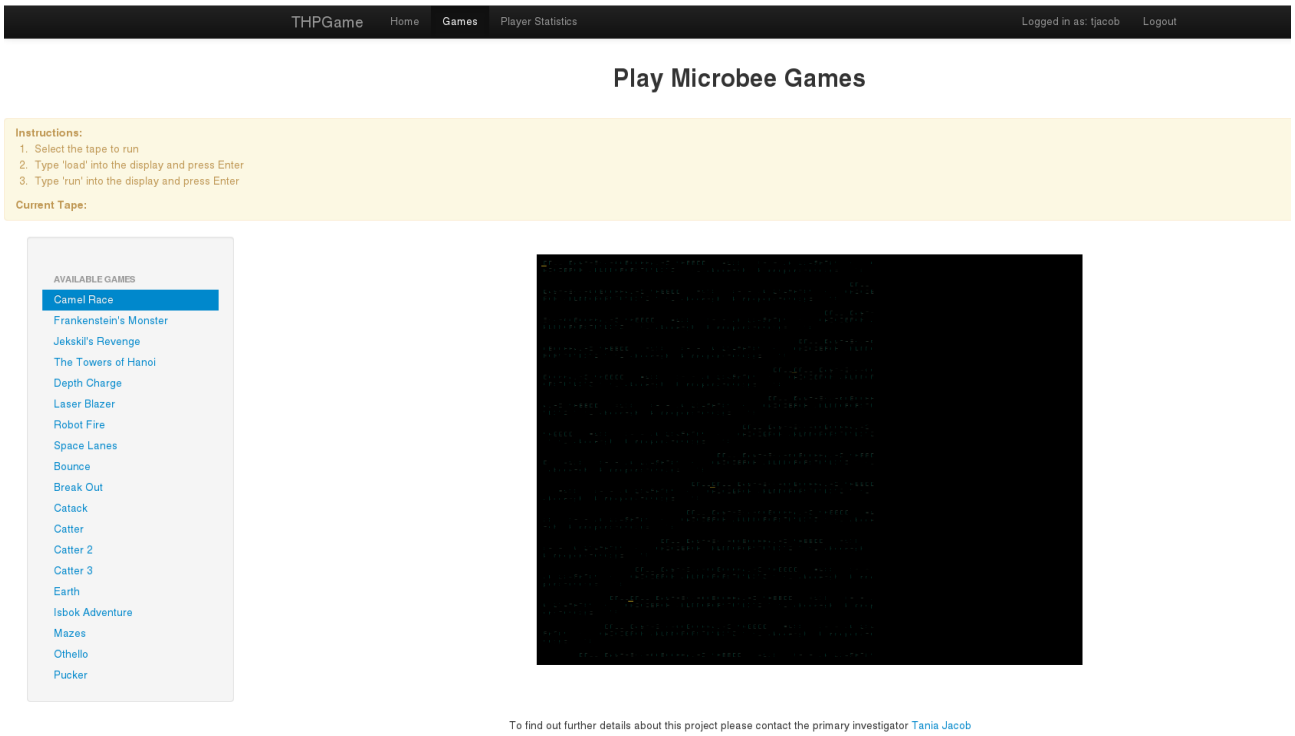


Figure 5.5: Play Games on THPGame

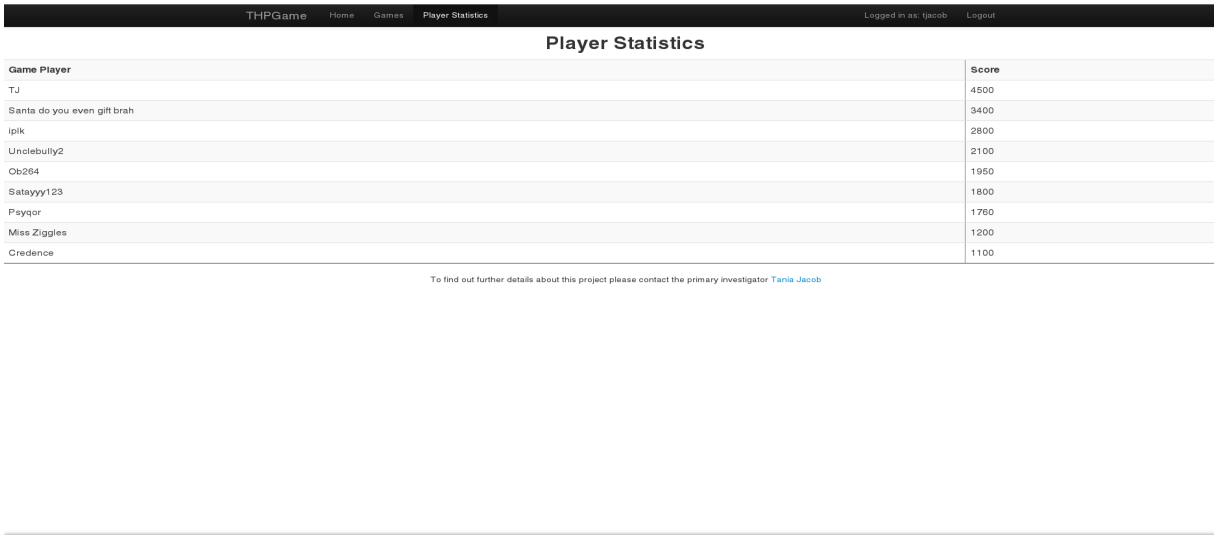


Figure 5.6: Player Statistics on THPGame

Chapter 6

Evaluation

This chapter discusses the heuristic evaluation, performance testing and future experiments. To conclude,the use of supported practices will be covered.

A standard approach to validating a user interface design is through user testing. Potential users casually playing the games and reporting that the games were boring and hard to play because of the unfamiliar game control keys caused us to turn to using an expert-evaluation technique. A heuristic evaluation is one of these techniques. We performed a full heuristic evaluation and carried out performance tests that targeted the scalability and performance of THPGame.

6.1 Heuristic Evaluation

A heuristic evaluation was conducted to evaluate THPGame. Usability problems of the user interface were detected through evaluation against Jakob Nielsen's heuristics. Carrying out a heuristic evaluation is an extremely cost efficient method especially in circumstances where limited time or budgetary resources exist [26].

The primary personas performed the most common scenarios outlined in Chapter 3 with the user interface. My supervisors role played as Matthew the casual game player for this heuristic evaluation, whilst I role played as TJ the competitive game player. The evaluation was conducted on a Dell Optiplex 990 machine running Linux in CO232. Each of us, performed the primary actions of our user and raised any user interface issues that breaks any of Jakob Nielsen's usability heuristics [26]. When issues arose, users rated the level of criticality on a scale from one to three. A value of one measured low, two measured high, whilst three represented critical.

6.1.1 Casual Game Player

The scenarios modelled by the casual game players were registering a user and playing a game.

Register User

A casual game player registered a new user into the system successfully. When initially viewing the system, the casual player was not immediately looking at the Create Account tab. This is a violation of the heuristic to provide a user interface that is flexible and efficient to use. Adequate guidance to setup an account would have been beneficial (rating two). An

automatic login feature was also pointed out as missing and this does not meet Nielsen's heuristic for a user interface that is flexible and efficient to use. Though at the same time, users control and freedom are limited since they are not informed why they should register themself as a user (rating two). A security point was also raised that there should be special password types allowed for users such as passwords needing to start with capital letters (rating one). The need for such exists to make it harder to brute force a password. Allowing for such capability will allow Nielsen's heuristic for a match between the system and the real world to occur. In addition when presented with the Home page, the link to the Nanowasp website should have opened in a separate tab rather than in the same tab (rating one). Smoother flow would have been provided to allow for consistency and standards according to Nielsen's heuristic. Error prevention was possible as users were notified when incorrect login details were provided. Also whilst creating a new user, if passwords do not match users are aided in helping diagnose their problems. But such error could potentially present a problem since THPGame does not automatically display recently inputted usernames.

Upon logging onto the system, text was provided about the Microbee computers providing documentation according to Nielsen's heuristic. However to make the user interface more aesthetic having a visualisation of a Microbee computer on the Login and Home page would have been useful (rating two) .

Play Game

Providing the users had logged on in the earlier scenario, they were able to successfully navigate to the Games page. It was useful to see the game descriptions were simple to understand. Problems that were found included that there were links that appeared clickable which was misleading: 'Home' and 'Logged in as'(rating two). A breaking of the heuristic for consistency and standards occurred. In addition, the hover game descriptions obscured the next element in the list of available games (rating two). This effectively violated the heuristic for users control and freedom. However whilst saying this, the design of the user interface was fairly consistent and standard.

6.1.2 Competitive Game Player

The scenarios modelled by the competitive game players were registering a user, playing a game and viewing player statistics.

Register User

Using the user interface appeared easy to a competitive game player as they knew they had to register a new user when using the system. Automatically the instinct was to look for the Create Account section of the Login page. It was easy to tell whether a user was logged on by inspecting the top right of the navigation bar. As a result, this met the heuristic for visibility of system status. In addition, a match is provided between the system and the real world as users are presented with a typical scenario in regards to setting up new logins. In such circumstances after successfully registering, new users are asked to login with their new details despite it seeming tedious. But unlike real systems, the user account creation form was inflexible as users had to scroll down to fill all fields. In addition they were not told if there were optional fields which did not match a real world system (rating one).

Play Game

Playing the games were easy as the list of available games were conveniently displayed on the page. The game descriptions was a neat feature to give information about the games to those less familiar. A bigger canvas would have been better to play the games however (rating two). Effectively this would have satisfied the requirement for flexible and efficiency of use of the user interface. A valid comment is made about the canvas size in order to make the maximum use of the screen size. When viewing the game descriptions it was noticed that the Camel Race game is selected on the left hand bar on the Games page however no tape name appeared next to current tape (rating two). This breaks the heuristic for recognition rather than recall. But on the other hand it is possible for easy navigation through the user interface.

View Player Statistics

The fast retrieval of data for the player statistics page was pleasing. Needs as a competitive player were served as the top scoring player was listed on the top of the page. This conveys the most important information to a competitive game player in one glance. It would have however been good if scores for individual games were recorded rather than a cumulative score being recorded for all the games (rating one). This would have allowed for a better match between the system and the real world.

6.1.3 Results Discussion

From the results obtained, it seems fair that most of the objectives of creating a usable user interface have been achieved. Though further enhancements can be made to add value to the user interface. The issues with a rating of two are summarised in the following table: 6.1

Issue	Rating
Guidance to setup a user account	2
Reasoning for registering	2
Visualisation of Microbee computer	2
Clickable links not clickable	2
Obscured placement of game descriptions	2
Bigger Canvas for Game	2
Contradicting current tape indicator	2

Table 6.1: Issues raised from the Heuristic Evaluation

6.2 Performance Testing

Performance testing was conducted to test the scalability of the solution produced. We hypothesised that if THPGame was not running fast enough locally then there would be low chances it would perform better whilst running on the server. For this reason I solely completed the performance testing on a Dell Optiplex 990 machine running Linux in CO240.

The first set of tests involved measuring the memory usage whilst running NodeJS on the computer. Ten runs were conducted of this experiment using a predefined sequence and the results are displayed in Table 6.2:

Action	Memory Size (KB)	Difference	Memory Peak (KB)	Difference
Post Server Startup	754661.2		755480.4	
Post Login	958016	+203354.8	958084.4	+202604
Post Player Statistics Loading	961713.6	+3697	970742.4	+12657.6
Post Games Page Loading	977628	+15914.4	979232	+8489.6
Post Logout	985521.2	+7893.2	993978.8	+14746.8

Table 6.2: Virtual Memory Size and Virtual Memory Peak of THPGame over a set of system actions. Booting up the server used up the most memory.

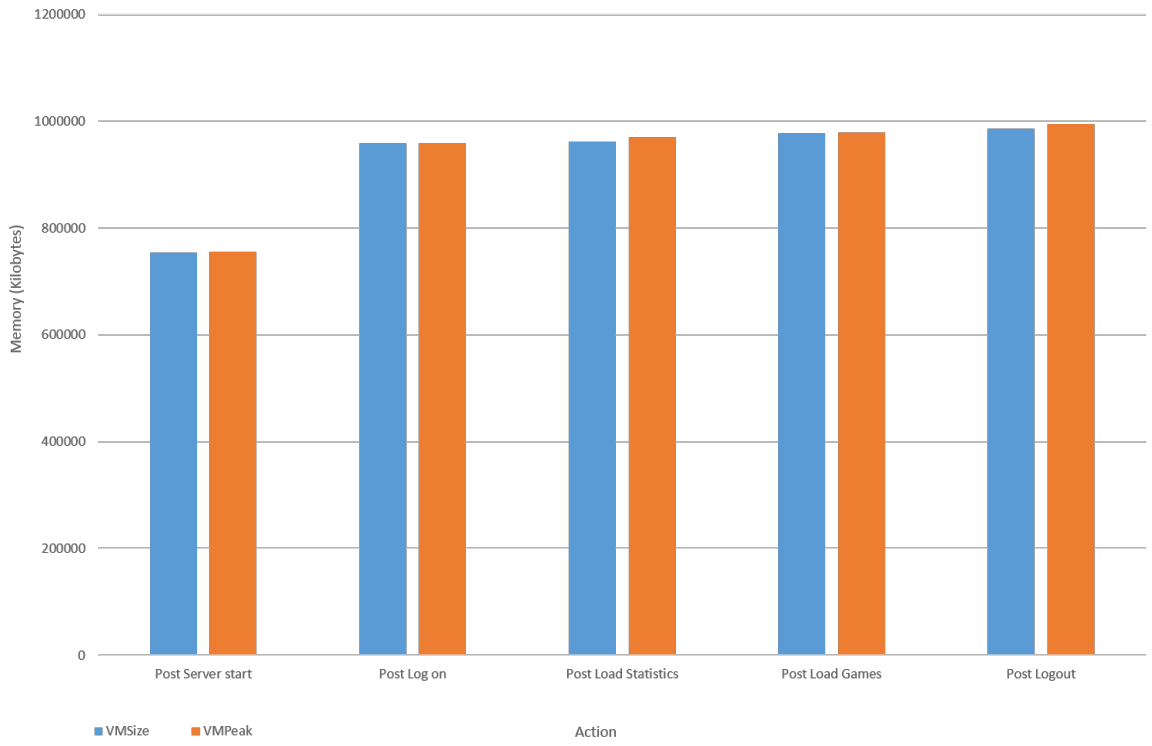


Figure 6.1: Virtual Memory Size and Virtual Memory Peak of THPGame over a set of system actions. The blue bars represent the Virtual Memory Size and the orange bars represent the Virtual Memory Peak.

Measuring the efficiency of characters being sent from the socket was the next metric. Acceptable values from these results will determine the playability of the system. Such experiment was conducted over twenty runs.

Run	Average time (ms)
1	0.0669
2	0.0685
3	0.0648
4	0.076
5	0.0766
6	0.0715
7	0.0361
8	0.0772
9	0.0823
10	0.0766
11	0.0714
12	0.0866
13	0.0786
14	0.0805
15	0.0778
16	0.0819
17	0.0891
18	0.0663
19	0.0372
20	0.0792

Table 6.3: Average time taken to receive a Character Image from the server and display it on the user interface (ms)

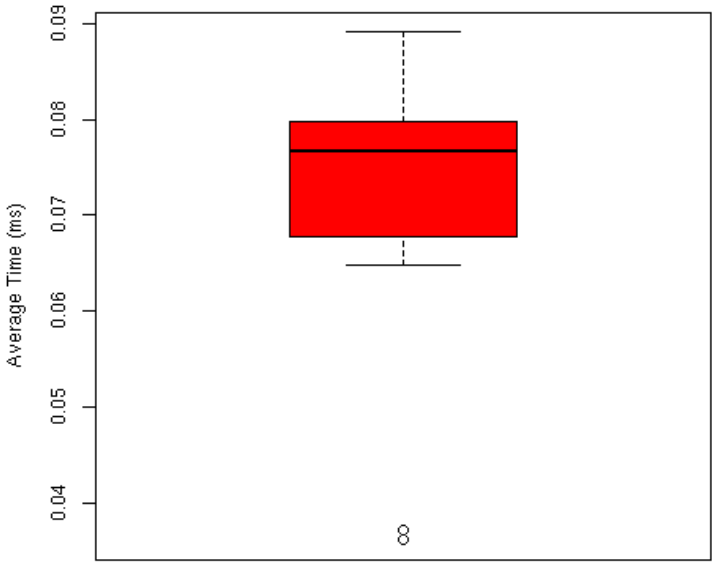


Figure 6.2: Average time taken to receive a Character Image from the server and display it on the user interface (ms). Two outliers exist as shown at the bottom of the graph.

To validate the server response time to accept a request from the client for an image, the time taken for a response to be received after sending a request was also measured. The results are as follows:

Run	Socket Response Time (ms)
1	1004.365
2	1004.77
3	1007.074
4	1005.236
5	1006.504
6	1005.476
7	1006.76
8	1004.766
9	1006.034
10	1005.643
11	1006.394
12	1004.36
13	1004.951
14	1005.025
15	1014.222
16	1005.532
17	1006.087
18	1001.997
19	1006.278
20	1005.166

Table 6.4: Socket Response Time to respond to request (ms)

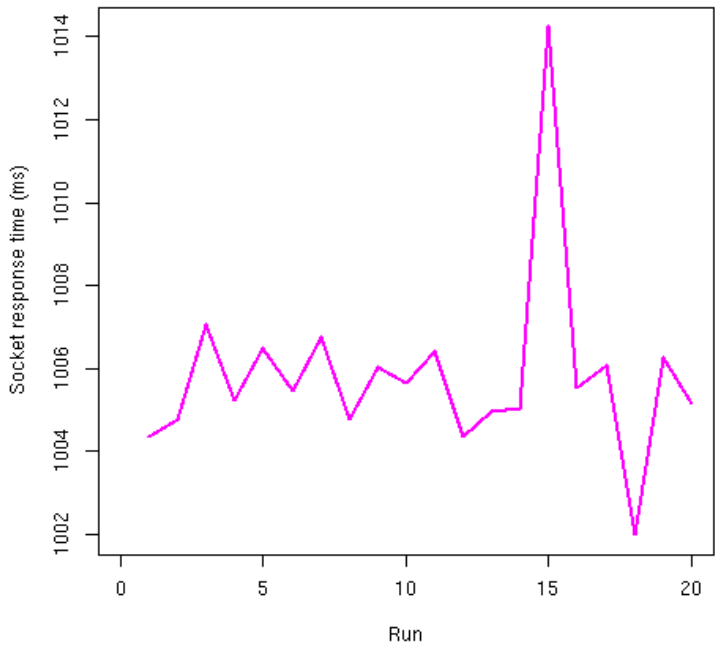


Figure 6.3: Socket Response Time to respond to request (ms)

Results Discussion

Looking at the results the virtual memory size relative to the virtual memory peak were fairly close. The largest increment in memory usage is evident when users successfully login to the system. But after this generally, the memory usage remains relatively stable and has small increments except when choosing to view the Games page. A greater memory usage when loading the Games page is expected due to the sending and receiving of data through the socket. A future test could involve testing with concurrent users.

The results also show that receiving a single character and displaying it on the client side typically takes one second to complete. An outlier is displayed in the data but this could have been affected by the current performance of computers in the Engineering and Computer Science network.

On average it was found that it took 1005.832 milliseconds for the client to send a request to the server to access the emulator and send back a character image that is to be rendered on screen. If we take into account that 1024 characters make up a Microbee screen then it takes one minute to render a screen. This is slow compared to the Doom system which renders 30 frames per second (roughly 0.33 seconds to render a screen) [3].

Past studies have compared the efficiency of running a NodeJS application with other technology. Using Apache Bench, 10 000 requests were made with 1000 happening concurrently. At the same time NodeJS handled each of these requests in either zero or one millisecond [21]. In other studies, NodeJS scaled well compared to PHP [15].

Whilst NodeJS proves to be efficient and scalable, THPGame may not be as efficient as expected. So this will need to be addressed in the future.The efficiency of THPGame is therefore questionable. However an improvement to allow for quicker rendering would be through developing a custom protocol which reduces the overhead in sending data as binary values in a header as opposed to JSON. Storing and exchanging text information much like XML is possible through using JSON.

6.3 Future Experiments

If more interesting games are found to run in THPGame, user testing could be conducted. Such testing was chosen to be performed to see if participants notice they are playing games locally or over Wi-Fi. Participants for the testing would have been obtained from a subset of the pre-study questionnaire participants that indicated interest in helping me test. An equal proportion of 18- 25 year old males and females from a variety of academic backgrounds would be the test participants. Some of these people will play games with the emulator running locally. Whilst others will play games with the emulator running on the server. The user testing would have involved roughly three minute game runs(repeated about 20 times). During each run, network performance and actions performed by users will be automatically recorded in a log. Upon the completion of each run, participants would get a popup appear on screen. Within this popup window, players would need to indicate whether they think the game was run locally or via Wi-Fi. If test participants did not notice if games were played locally or not, then a new technique would have been found for undetectable network deployment of games. The supporting practices that helped in the success of the

project will now be outlined.

6.4 Supporting Practices

Maintaining good coding practices was vital whilst completing the project. Storing the project in a repository on a cloud application was necessary. Doing such a task avoided work being lost due to the university system corrupting. It also allowed for the opportunity and collaboration of supervisors to view code. Github was the chosen technology utilised to store files due to past experience and confidence in storing files in such an application.

Keeping a log book was another good practice adopted whilst completing this project. Recording minutes of weekly meetings became a routine. In addition, I utilised it to take notes of new information discovered during the week and to help maintain a weekly to-do list. Keeping a to-do list supported following weekly iterations under iterative development. An electronic copy of the log book was also stored to compensate for the ease in losing the book.

6.5 Improvements

Better adherence with my Gantt Chart would be an improvement for better managing the project. Some tasks in the project took longer than anticipated. To make a better use of a Gantt Chart in future projects, it would make sense to regularly update the chart according to the progress to date. Also iterations could have been controlled to be completed on schedule. Therefore, the project could have been better managed and controlled over the year.

Chapter 7

Conclusions and Future Work

Obsoletion of hardware is one factor that causes games to no longer be runnable. In addition the fact that Microbee computer games are not in the public domain drives the need to develop a client-server model for emulation of old systems such as Microbee.

The architecture for transmitting the synchronised audio and video over the Wi-Fi in a museum has been outlined in Chapter 4 of this report. Design of the system has been based on a user-centered design approach taking into account personas and user scenarios. The three personas considered for the system were a competitive game player, casual game player and system administrator. In addition, software engineering considerations have been put into thought. A prototype has been developed called THPGame with beneficial results useful to the future.

The validation of the project issues will now be discussed.

7.1 I1: No access to play old computer games on the desktop due to intellecutal property rights restrictions

We encourage copyright owners to make their games available on THPGame. This is because the client will not have the entire copy of games. At the same time users are able to run Microbee platform games on the system which is portable to the desktop and works cross platform. So the issue is resolved.

7.2 I2: Games do not have synchronised audio and visual components

Frames have been displayed from the video RAM to the user interface. The audio component of the system remains unimplemented and forms the future work for the project.

7.3 I3: Users need a collective place to share gaming experiences and allow for playing as a social construct

The user interface has been developed to provide a collective place to share the gaming experience and for playing as a social construct. The user interface is highly intuitive to cater for the range of users utilising the system in the museum setting. Tabs and navigation

within the user interface is reasonable according to the results from the heuristic evaluation. The client-server architecture also supports playing as a social construct and allows for the Microbee games to run on a range of platforms. Users are able to play socially on THPGame through the ability of the interface to store persistent game session and user data. Consequently, this issue has been resolved.

7.4 I4: All emulators are not cross platform compatible (run on all operating systems

Utilising Nanowasp helps solve the issue of emulators not being cross platform as it works on all operating systems. A restriction is therefore not placed on the type of computer that THPGame users will need. Though having this emulator allows for Microbee games to be run and for cross platform compatability to occur.

7.5 Future Work

Currently one instance of Microbee can be run in the system. Further enhancing the multiplayer capability of the system would be beneficial in the future. On the other hand, the user interface component of THPGame handles having more than one user logged on so the emulator code would need to be modified to satisfy requirement F4 (multiplayer game handling). By doing this, further work can be done on continually ensuring requirement NF2 (game performance being indistinguishable to human perception) is also met. As well as this the audio component of the system can be implemented. In the future when the full system has been developed, the user testing discussed can be conducted.

7.6 Summary

Evident from above, most of the issues outlined in the introduction have been resolved by my solution. The contributions outlined in Chapter 1 have also been met.

7.6.1 Architecture design for transmitting synchronised audio/video over Wi-Fi in a shared environment such as a museum

An architecture design has been developed for THPGame through utilising a user-centered design approach. Personas and scenarios have been used to design the system. It is the closest de-facto standard in industry for modelling user interfaces.

7.6.2 Proof of concept prototype that demonstrates modifying an existing emulator to support video transfer in a client/server model

A prototype has been developed that modifies the Nanowasp emulator to support video transfer in a client-server architecture to run Microbee games. The Nanowasp emulator is run off a NodeJS server. Such emulator is used to run Microbee games. A canvas is displayed on the front end user-interface with the game display. A successful startup of the server will automatically redirect the user to the Login page of the user interface on the default web browser (ideally Google Chrome). The Games page is the primary page of the user interface but there are other pages that for example display Player Statistics.

7.6.3 Evaluation as to whether performance of the model is noticeably detected by human perception and if a usable user interface has been developed

The evaluation has revealed that the time taken to send a request and receive a response in THPGame is slow in comparison to related work. The solution produced is not as scalable as hoped. As a result it was thought to send the data as binary values rather than JSON over the socket. While on the other hand the heuristic evaluation has revealed a fairly usable user interface but improvements can be made to it.

I am pleased with the system that has been developed and envisage an extension of the prototype to complete more of the architecture.

Bibliography

[1] ABRAS, C., MALONEY-KRICHMAR, D., AND PREECE, J. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications 37*, 4 (2004), 445–56.

[2] BARWICK, J., DEARNLEY, J., AND MUIR, A. Playing games with cultural heritage: a comparative case study analysis of the current status of digital game preservation. *Games and Culture 6*, 4 (2011), 373–390.

[3] BIKKER, J VAN SCHIJNDEL, J. The brigade renderer: A path tracer for real time games. http://www.hindawi.com/journals/ijcgt/2013/578269/.

[4] CHURCHILL, D. nanowaspjs. https://github.com/dgchurchill/nanowaspjs.

[5] EXCHANGE, C. B. C64 emulators. http://www.zzap64.co.uk/c64/c64emulators.html.

[6] FITZSIMMONS, M. Sony and gaikai join forces for social and remote play in ps4. http://www.techradar.com/news/gaming/consoles/sony-and-gaikai-join-forces-for-social-and-remote-play-in-ps4-1131847.

[7] FRIENDS, A. Xampp. http://www.apachefriends.org/en/xampp.html.

[8] GAMACHE, D. What is it? http://www.getskeleton.com/#whatAndWhy.

[9] GROUP, T. P. G. D. Postgresql. http://www.postgresql.org.

[10] GUACAMOLE. Guacamole. http://guac-dev.org/.

[11] GUTTENBRUNNER, M., BECKER, C., AND RAUBER, A. Keeping the game alive: Evaluating strategies for the preservation of console video games. *International Journal of Digital Curation 5*, 1 (2010), 64–90.

[12] HILL, O. The microbee story. http://harveycohen.net/oznaki/microbee.html.

[13] HOLDINGS, D. ubee512. http://freecode.com/projects/ubee512.

[14] HTML5ROCKS. The problem: Low latency client-server and server-client connections. http://www.html5rocks.com/en/tutorials/websockets/basics/http://www.html5rocks.com/en/tutorials/websockets/basics/.

[15] IMPACT, L. Node.js vs php using load impact to visualize node.js efficiency. http://blog.loadimpact.com/2013/02/01/node-js-vs-php-using-load-impact-to-visualize-node-js-efficency.

[16] INC, G. Github. https://github.com/.

[17] INC, J. Node packaged modules. https://npmjs.org/.

[18] INC, J. nodejs. http://nodejs.org/.

[19] KNIGHT, M. Node.js vs php performance - maths. http://www.matt-knight.co.uk/2011/node-js-vs-php-performance-maths/.

[20] LAPLANTE, P. A., AND NEILL, C. J. The demise of the waterfall model is imminent. *Queue 1*, 10 (2004), 10.

[21] LERNER, R. M. At the forge: Node.js. *Linux J. 2011*, 205 (May 2011).

[22] MA, H. Structured methods. University Lecture, 2013.

[23] MARSHALL, S. User interface design. University Lecture, 2012.

[24] MESS, M. Welcome to the mess wiki. http://www.mess.org/.

[25] NEWMAN, J. Illegal deposit game preservation and/as software piracy. *Convergence: The International Journal of Research into New Media Technologies 19*, 1 (2013), 45–61.

[26] NIELSEN, J. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1992), CHI '92, ACM, pp. 373–380.

[27] OF NEW ZEALAND, C. C. Copyright basics. http://www.copyright.org.nz/basics.php.

[28] PROJECT, M. S. P. Microbee software preservation project. http://www.microbee-mspp.org.au/.

[29] PRUITT, J., AND GRUDIN, J. Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences* (New York, NY, USA, 2003), DUX '03, ACM, pp. 1–15.

[30] RECHERT, K., AND VON SUCHODOLETZ, D. Efficient access to emulation-as-a-service-challenges and requirements.

[31] RECHERT, K., VON SUCHODOLETZ, D., AND WELTE, R. Emulation based services in digital preservation. In *Proceedings of the 10th annual joint conference on Digital libraries* (2010), ACM, pp. 365–368.

[32] SONY. Gaikai. http://www.gaikai.com/qa.

[33] STRODL, S., BECKER, C., NEUMAYER, R., AND RAUBER, A. How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries* (2007), ACM, pp. 29–38.

[34] SWALWELL, M. Towards the preservation of local computer game software challenges, strategies, reflections. *Convergence: The International Journal of Research into New Media Technologies 15*, 3 (2009), 263–279.

[35] TAYLOR, T. L. Whose game is this anyway? negotiating corporate ownership in a virtual world. In *Computer Games and Digital Cultures Conference Proceedings* (2002), Tampere University Press Tampere, pp. 227–242.

[36] TECHNOLOGY, M. About us/history. http://www.microbeetechnology.com.au/index_3.htm.

[37] TILKOV, S., AND VINOSKI, S. Node.js: Using javascript to build high-performance network programs. *IEEE Internet Computing 14*, 6 (2010), 8083.

[38] TWITTER. Base css. http://getbootstrap.com/2.3.2/base-css.html.

[39] UNIVERSITY, V. Vuw qualtrics. https://vuw.qualtrics.com/ControlPanel/.

[40] UNIVERSITY, V. Play it again project correspondence email. Private Communication, 2013.

[41] UNKNOWN. Surveymonkey. http://www.surveymonkey.com.

[42] W3SCHOOLS. Browser statistics and trends. http://www.w3schools.com/browsers/browsers_stats.asp.

[43] WINGET, M. A. Videogame preservation and massively multiplayer online role-playing games: A review of the literature. *Journal of the American Society for Information Science and Technology 62*, 10 (2011), 1869–1883.

Appendix A

User Interface Additional Screenshots

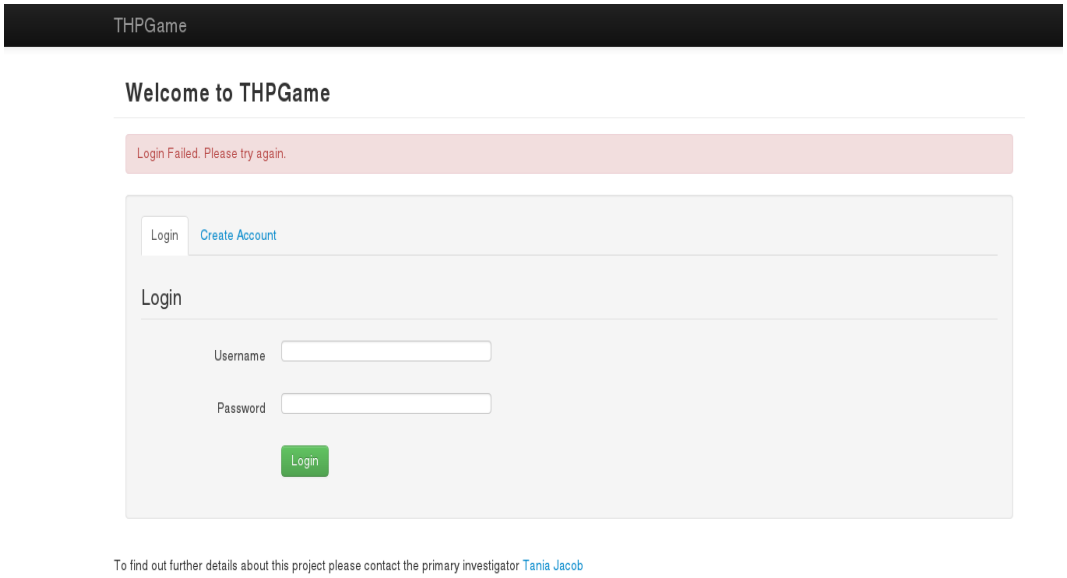


Figure A.1: Login failure to THPGame

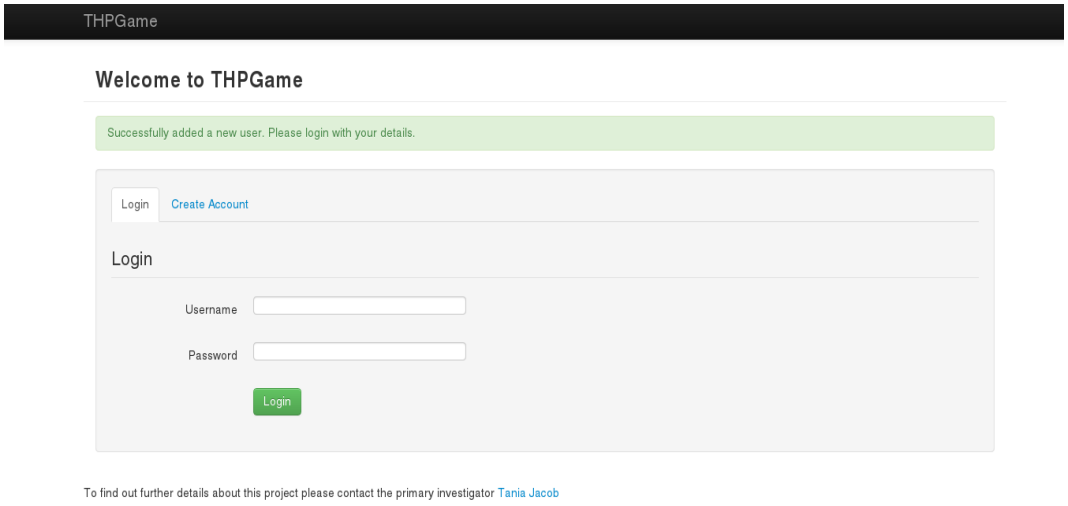


Figure A.2: Successful registration of new THPGame user

Appendix B

Human Ethics Information Sheet

INFORMATION SHEET

ENGR489 Project for Bachelor of Engineering in Software Engineering with Honours

Topic: Client-Server Model for Preserving Old Computer Games

Primary Investigator: Tania Jacob (tania.jacob@vuw.ac.nz)

Supervisors: Stuart Marshall & Ian Welch

Experiment Purpose and Procedure

The purpose of this questionnaire is to help get the right test participants to assist me for testing my ENGR489 project. My project aims to enable people to play old computer games which are currently not able to be played due to outdated operating systems that these run on. Victoria University's ethics committee has been asked for approval on the user testing I am carrying out.

Please answer all questions to the best of your ability. It will only take roughly five minutes to complete.

Confidentiality of Survey Responses

All responses will be collated to help determine the best set of participants to take part in the testing starting in July 2013. Selected test participants will be informed. Confidential information provided in the survey will not be revealed online, but rather presented in an aggregated manner.

Appendix C

Human Ethics Consent Form

CONSENT FORM

I have been provided with adequate information relating to the nature and objectives of this research project, I have understood that information and have been given the opportunity to seek further clarification or explanation. I understand that any information or opinions I provide will be kept confidential and reported only in an aggregated/non-attributable form. I also understand that I can withdraw from this study up until data analysis. Also, the information I have provided will only be used for this research project and that any further use will require my written consent.

Participant Name

_____

Participant Signature

_____

Participant Email (optional: for further communication about this research)

_____

Date

_____

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

HUMAN ETHICS COMMITTEE

Application for Approval of Research Projects

Please write legibly or type if possible. Applications must be signed by supervisor (for student projects) and Head of School

Note: The Human Ethics Committee attempts to have all applications approved within three weeks but a longer period may be necessary if applications require substantial revision.

1. NATURE OF PROPOSED RESEARCH:

(a) Student Research

(b) If Student Research: Degree: Bachelor of Engineering. Course Code: ENGR489...

(c) Project Title: Client Server Model for Preserving Old Computer Games

2. INVESTIGATORS:

(a) Principal Investigator

Name: Tania Jacob

Email address: tania.jacob@vuw.ac.nz

School/Dept/Group: School of Engineering & Computer Science

(b) Other Researchers Name Position

.....

.....

(c) Supervisor (in the case of student research projects): Stuart Marshall & Ian Welch

3. DURATION OF RESEARCH

(a) Proposed starting date for data collection: 1 July 2013

(**Note:** that NO part of the research requiring ethical approval may commence prior to approval being given)

(b) Proposed date of completion of project as a whole: 12 November 2013

4. PROPOSED SOURCE/S OF FUNDING AND OTHER ETHICAL CONSIDERATIONS

(a) Sources of funding for the project: School of Engineering & Computer Science

Please indicate any ethical issues or conflicts of interest that may arise because of sources of funding e.g. restrictions on publication of results

N/A

(b) Is any professional code of ethics to be followed **Y** ☒ **N** ☐

If yes, name: ACM Code of Ethics & IITP

(c) Is ethical approval required from any other body **Y** ☐ **N** ☒

If yes, name and indicate when/if approval will be given

.....

5. DETAILS OF PROJECT

Briefly Outline:

(a) The objectives of the project

To see if participants can notice if they are playing multiplayer games over a server as opposed to running the game locally.

(b) Method of data collection

Participants will take part in roughly 3 minute game runs (repeated about 20 times). During each run, network performance and actions performed by users will be automatically recorded in a log. Upon the completion of each run, participants will get a popup on screen in which they have to indicate whether they think the game they just played was run locally or via Wi-Fi.

(c) The benefits and scientific value of the project

If test participants cannot notice if games were played locally or not, then a new technique has been successfully found for undetectable network deployment of systems.

(d) Characteristics of the participants

Male and Female (equal proportion) 18-25 year old students from a wide range of university academic backgrounds to cater for a better representation of the population.

(e) Method of recruitment

Based on personal relationships. A pre-study questionnaire will be distributed electronically via the creation of a private Facebook event to link to a Qualtrics survey.

(f) Payments that are to be made/expenses to be reimbursed to participants

2x $50 vouchers will be awarded to participants. One for the highest scoring participant in games whilst the other winner is selected from a random draw.

(g) Other assistance (e.g. meals, transport) that is to be given to participants

No

(h) Any special hazards and/or inconvenience (including deception) that participants will encounter:

No

(i) State whether consent is for (delete where not applicable):

(i) the collection of data

(ii) attribution of opinions or information

(iii) release of data to others

(iv) use for a conference report or a publication

(iv) use for some particular purpose (specify)

.....

.....

Attach a copy of any questionnaire or interview schedule to the application

(j) How is informed consent to be obtained (see sections 4.1, 4.5(d) and 4.8(g) of the Human Ethics Policy)

(i) the research is strictly anonymous, an information sheet is supplied and informed consent is implied by voluntary participation in filling out a questionnaire for example (include a copy of the information sheet) Y □ N □

(ii) the research is not anonymous but is confidential and informed consent will be obtained through a signed consent form (include a copy of the consent form and information sheet) Y □ N □

(iii) the research is neither anonymous or confidential and informed consent will be obtained through a signed consent form (include a copy of the consent form and information sheet) Y □ N □

(iv) informed consent will be obtained by some other method (please specify and provide details) Y □ N □

.....

.....

With the exception of anonymous research as in (i), if it is proposed that written consent will not be obtained, please explain why

.....

.....

(k) If the research will not be conducted on a strictly anonymous basis state how issues of confidentiality of participants are to be ensured if this is intended. (See section 4..1(e) of the Human Ethics Policy). (E.g. who will listen to tapes, see questionnaires or have access to data). Please ensure that you distinguish clearly between anonymity and confidentiality. Indicate which of these are applicable.

(i) access to the research data will be restricted to the investigator Y □ N □

(ii) access to the research data will be restricted to the investigator and their supervisor (student research) Y □ N □

(iii) all opinions and data will be reported in aggregated form in such a way that individual persons or organisations are not identifiable Y ☒ N ☐

(iv) Other (please specify)

Non anonymous data will be available to just the supervisors and investigator. The anonymous information will be available in the interests of scientific reproducibility.

(l) Procedure for the storage of, access to and disposal of data, both during and at the conclusion of the research. (see section 4.12 of the Human Ethics Policy). Indicate which are applicable:

(i) all written material (questionnaires, interview notes, etc) will be kept in a locked file and access is restricted to the investigator Y ☒ N ☐

(ii) all electronic information will be kept in a password-protected file and access will be restricted to the investigator Y ☒ N ☐

(iii) all questionnaires, interview notes and similar materials will be destroyed:

(a) at the conclusion of the research Y ☐ N ☒

(b) <u>One</u> year after the conclusion of the research; or Y ☒ N ☐

(iv) any audio or video recordings will be returned to participants and/or electronically wiped Y ☐ N ☒

(v) other procedures (please specify):

This applies to the non anonymous data. Whilst the anonymous data will be available online.

If data and material are not to be destroyed please indicate why and the procedures envisaged for ongoing storage and security

N/A

(m) Feedback procedures (See section 7 of Appendix 1 of the Human Ethics Policy). You should indicate whether feedback will be provided to participants and in what form. If feedback will not be given, indicate the reasons why.

Participants can indicate if they wish to be informed about the summary of findings from this research in the consent form attached with the pre-study questionnaire.

(n) Reporting and publication of results. Please indicate which of the following are appropriate. The proposed form of publications should be indicated on the information sheet and/or consent form.

(i) publication in academic or professional journals Y ☒ N ☐

(ii) dissemination at academic or professional conferences Y ☒ N ☐

(iii) deposit of the research paper or thesis in the University Library (student research) Y ☐ N ☒

(iv) other (please specify) Y □ N □

.....

.....

Signature of investigators as listed on page 1 **(including supervisors) and Head of School**.

NB: All investigators and the Head of School must sign before an application is submitted for approval

..... Date.....

..... Date.....

..... Date.....

Head of School:

..... Date.....

Appendix D

Pre-study Questionnaire

Pre-Study Questionnaire for ENGR489 Project

Q1 What is your name?

Q2 What is your gamer tag (if you have one)?

Q3 What gender are you?
- ❍ Male
- ❍ Female

Q4 What do you study?
- ❑ Architecture
- ❑ Art History
- ❑ Biomedical Science
- ❑ Building Science
- ❑ Commerce
- ❑ Criminology
- ❑ Design
- ❑ Earth Sciences/Geology
- ❑ Engineering/Computer Science
- ❑ English
- ❑ Geography
- ❑ Languages
- ❑ Law
- ❑ Mathematics/Operations Research/Statistics
- ❑ Music
- ❑ Politics/International Relations
- ❑ Psychology
- ❑ Science- Biology, Chemistry, Development Studies, Ecology, Environmental Studies, Physics
- ❑ Sociology
- ❑ Teaching
- ❑ Tourism
- ❑ Media/Theatre
- ❑ Nursing
- ❑ HealthSci/Med
- ❑ Other

Q5 What kind of phone do you currently own?
- ❍ Android
- ❍ Blackberry
- ❍ iPhone
- ❍ Non Smartphone

Q6 Please rate your experience in using the following types of phones (where applicable), with 5 being excellent.

_____ Android
_____ Blackberry
_____ iPhone
_____ Non Smartphone

Q7 What version of Android does your current phone run?

- ❍ Android 2.3 (Gingerbread)
- ❍ Android 3 (Honeycomb)
- ❍ Android 4.0 (Icecream)
- ❍ Android 4.2 (Jellybean)

Q8 What version of iPhone do you own?

- ❍ iPhone 3
- ❍ iPhone 4/iPhone4S
- ❍ iPhone 5

Q9 What brand is your phone?

Q10 What do you use your phone for (select all that apply)?

- ❑ Applications
- ❑ Camera
- ❑ Calling
- ❑ Games
- ❑ Internet
- ❑ Music
- ❑ Texting
- ❑ Email

Q11 What web browsers do you use in your smartphone?

- ❑ Default Android Browser
- ❑ Google Chrome
- ❑ Mozilla Firefox
- ❑ Safari
- ❑ Other

Q12 How would you rate the touchscreen experience (5 is excellent)

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q13 Do you play games of any type?

- ❍ Yes
- ❍ No

Q14 What kind of games do you play?

- ❑ Social Games eg Farmville/Angry Birds
- ❑ Competitive Games eg Call of Duty/Dota 2

Q15 What do you play games on (select all that apply)?

- ❑ iPod/Phone
- ❑ Laptop/PC
- ❑ PS2/PS3/Wii/Xbox

Q16 List the most popular games you play on ANY device

Q17 Have you had any experience playing server based games in the past such as multi player pacman (i.e play multiplayer games over internet outside a web browser such as Google Chrome)?

- ❍ Yes
- ❍ No

Q18 Do you have experience in detecting network performance lags (i.e. slow response times) whilst playing such games?

- ❍ Yes
- ❍ No

Q19 Would you be willing to give some time to participate in testing old computer games as part of my ENGR489 project?

- ❍ Yes
- ❍ No

Q20 If you are interested in finding out more information about the conclusions reached or for further information from this research please give your email address.