USING MICROSOFT MULTIPOINT TECHNOLOGY TO ENHANCE COLLABORATIVE LEARNING IN PRIMARY SCHOOLS

HUSSIN AHMED ABDELKADER

UNIVERSITI UTARA MALAYSIA

2010

USING MICROSOFT MULTIPOINT TECHNOLOGY TO ENHANCE COLLABORATIVE LEARNING IN PRIMARY SCHOOLS

A project report submitted to the Dean of Postgraduate and Research office in partial fulfillment of the requirement for the degree Master of Science

(Information and Communication Technology)

Universiti Utara Malaysia

HUSSIN AHMED ABDELKADER

KOLEJ SASTERA DAN SAINS
(College of Arts and Sciences)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certificate of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

HUSSIN AHMED ABDELKADER
(804062)

calon untuk Ijazah
(candidate for the degree of) **MSc. (Information Communication Technology)**

telah mengemukakan kertas projek yang bertajuk
(has presented his/her project paper of the following title)

USING MICROSOFT MULTIPOINT TECHNOLOGY TO ENHANCE COLLABORATIVE LEARNING IN PRIMARY SCHOOLS

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia Utama
(Name of Main Supervisor): **ASSOC. PROF. DR. SUHAIDI HASSAN**

Tandatangan
(Signature) : _____

ASSOC. PROF. DR. SUHAIDI HASSAN
Assistant Vice Chancellor
College of Arts and Sciences
Universiti Utara Malaysia

Tarikh
(Date) : 20/5/2010

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a postgraduate degree from University Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor or, in his absence by the Dean of the Academic Office. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to University Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to

Dean of Postgraduate and Research Office

UUM CAS

Universiti Utara Malaysia

06010 UUM Sintok

Kedah Darul Aman

ABSTRACT

Researches in psychology, education and pedagogical practices show the stunning impact of collaboration in learning, particularly for young children. However, current design of Personal Computer Applications for collaborative learning is limited and poor. The objective of this research is to use Microsoft Multipoint technology in order to create collaborative quiz game for leveraging collaboration among primary school students. Merging prototyping with Interaction Design is very suitable for the purpose of gaining experience and it identifies new opportunities to customize the interface for collaborative software and make full use of Microsoft Multipoint technology. Measuring usability attributes; Usefulness, Ease of Use, Ease of Learning and Satisfaction is a core step to test the prototype system and approve the successfulness of research objectives. One limitation of this paper is that there is no text-based activity in the prototype. All data was collected via using USE Questionnaire in order to test usability attributes from perspective of users. The results of this research revealed that students found the prototype useful, easy to use, easy to learn and they attained high satisfaction. Most importantly, observations revealed that students were extremely excited and they could collaborate effectively. The value of this project is to leverage collaborative learning in primary schools and clarify the importance of merging Interaction Design and Microsoft Multipoint technology; which in return could encourage collaboration. Placing this project in the public domain will, hopefully enable other researchers and practitioners in similar situations to relate to my experience and gain insights.

ACKNOWLEDGEMENTS

In the Name of ALLAH, the Most Gracious and the Most Merciful

First of all, I give thanks to ALLAH for His guidance, His every blessing I know and His every blessing I don't know and His mercy throughout my life. Peace and Blessing to His last Prophet Muhammad (S.A.W), his household and his companions. My sincere appreciation goes to my beloved parents, my brother and my sisters for everything; their patience, prayers and all kinds of support they have given to me throughout my life.

Secondly, my grateful thanks go to my supervisor, Prof. Dr. Suhaidi Hassan, who had given to me full support and contributed immensely towards the completion of this paper. He encouraged me to the extreme and pushed me forward to learn new things.

Lastly, I appreciate the efforts of all my friends, Staff of Information and Communication Technology, College of Arts and Sciences, University Utara Malaysia and those who contributed directly or indirectly towards the completion of this project. Thanks to all.

HUSSIN AHMED ABDELKADER

College of Arts and Sciences,

Universiti Utara Malaysia,

April 2010

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIXES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Personal computer (PC) is an indispensable tool in every school; it affects the learning performance to the extreme and allows students to access vast amount of educational resources. In particular, the advancement in the capability of PC in terms of high speed and large amount of storage facilitated the process of learning and empowered students' abilities immensely.

It is clear that, in many branches of human knowledge, a lot of problems can be solved by working in group. Particularly for children, who are sociable by nature and take advantage of every opportunity to collaborate with each other.

Collaboration among students has converted learning from traditional way to a new fruitful way that has been proved by education experts to be successful and direct all participants in learning to be fully aware about its impact, and hence, it is important to integrate collaboration in learning process. Lai et al. [1] observed that, there are general skills like collaboration and creativity can be identified as basic skills to support students in learning. Ikeda et al. [2] stated that, collaborative learning also allows students to gain different kinds of skills and leverage learning effectiveness as a whole. Moreover, observations on the use of computers in classrooms strongly clarified that children have

the propensity to work together around a single station [3]. Potential value can be obtained from a children tendency to work together and it is an impetus to drive them to collaborate more.

Research in psychology and education has shown that working in groups can have prodigious effects on development and learning, particularly for young children [4]. Supportive learning environments resulted from a shift in learning trends that demonstrate learning becoming more social and exploratory. Thus, there is a need to make full use of new possibilities of learning using computers, in particular focusing on technologies to support collaboration and creativity in learning environments [5]. In advanced pedagogical practices, using Computer Supported Collaborative Learning (CSCL) has become an essential part of the whole learning environment and the culture of learning [6]. Collaboration has a positive impact on children, they can find it fun to play with others and become engaged when they have the opportunity to share experiences with friends. Thus, experiments show that if you offer five children five computers, within 10 minutes, two to three of those children will be gathered around one computer screen [7]. The implications of social and educational theories emphasize the importance of the role of collaboration in learning and so it is crucial to apply these theories to practical way.

Nowadays, using PC to support collaborative learning might be the ideal solution to limited funding in developing countries; to assign one computer to every child.

Moreover, significant trend of the contemporary educational research on developing countries and developed ones as well, is towards collaborative learning, which has frequently been used as a facsimile for learning success.

1.2 Problem Statement

Nowadays, research in psychology, education and pedagogical practices shows the stunning impact of collaboration in learning, particularly for young children [4, 6]. However, there is lack of educational software that supports collaboration among primary school students.

Although Microsoft created Microsoft Multipoint technology to allow more than one student to collaborate in one PC, but much effort has to be done to address user's needs in terms of collaboration [8]. Furthermore, customizing the interface is a big challenge when it comes to collaborative software.

Collaboration and learning will only occur if the technology is designed to fit within the context of use for which it is intended [9]. Otherwise, the interface may actually be a barrier to learning. In essence, collaboration encouragement is more meaningful than just enable collaboration [10] and it helps to extract the intrinsic value of collaboration.

1.3 Research Questions

The Research Questions are as follow:

- Can we address primary school students' needs in terms of collaboration using Microsoft Multipoint?
- Can we develop windows-based prototype that leverage learning collaboration among primary school students?
- What are the findings of evaluating and measuring usability attributes of the prototype using usability tests?

1.4 Research Objectives

The objectives of the research are as follow:

- To address primary school students needs to enhance collaboration using Microsoft Multipoint technology.
- To develop windows-based prototype using Microsoft Multipoint technology for primary school students to enhance collaborative learning.
- To evaluate the prototype using usability tests.

1.5 Research Scope

Using prototyping is efficient as a practical way used to improve planning and execution of software projects via building executable software systems (prototypes) for experimental objectives. It is very suitable for the purpose of gaining experience in new application areas and for supporting incremental or evolutionary software development [11]. Moreover, combining Interaction Design with prototyping helps to identify stunning opportunities to make a difference to users and then better mix Microsoft Multipoint technology and users activities [12]. Interaction Design will convert

traditional applications interface that serve one user and customize it to multiple users in a proper way.

This research used English as a Second Language (ESL) as an appropriate subject to be the content of collaborative quiz game; based partly on the practical need for ESL instruction in developing regions to both teach and test English vocabulary [13]. The prototype will use multiple choices written in English and students will be required to choose the right word that matches the picture in the screen.

The study will take place in an educational suite of a local primary school, involving the whole population of the class; 36 children from a Year 5 (aged 10); 23 females and 13 males. A similar study focused on year 5, which refers to the capability of children at this age to test the prototype effectively [14]. Children experience in school makes them ready to sit at a task and follow directions from an adult, and they are generally not self-conscious about being observed as they play on the computer [15].

Using usability evaluation is a core step in this research in order to test usability attributes from the perspective of users [16]. The research used USE Questionnaire which stands for Usefulness, Satisfaction, and Ease of Use which we separated it into two factors, Ease of Learning and Ease of Use and the questionnaire statements represented by a five-point Likert scale [17]. Usability evaluation will be a signal of the success of research prototype and attain feed back from users. In essence, to be sure that this research achieves all objectives.

1.6 Significance of Study

This research will be an approach to leverage the collaboration activity among students, in primary schools and address their needs, via using Microsoft Multipoint and create qualified environment, to customize the prototype for the research objectives.

1.1 Summary

This research aims to make full use of Microsoft Multipoint technology to leverage collaboration among primary school students, and help them acquire new skill which in returns will contribute to progress of students' performance in learning.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter identifies the concept of collaboration and reviews literature on initiatives and approaches done in educational software meant for collaborative learning. In essence, to gain working agreements about how we could leverage collaboration among primary school students.

Learning collaboration is considered as an instruction method where students at different performance level can work in a group to achieve mutual goal. In addition, every student will be responsible for his group mates' performance. Thus, collaboration will push learning forward among students and make them successful [18] (see Figure 2.1). This definition gives insight into collaborative learning impacts on children performance in learning, but we have to take into consideration how to orient children to collaborate effectively to gain the intrinsic value of being interact with each other. In this research we claim that, the driven factor in leveraging collaboration is the need to attain a competitive personality; to release the power within a student to do his best, and hence, every student will try to imitate his friend, exchange benefits with each other and interact in a positive way.

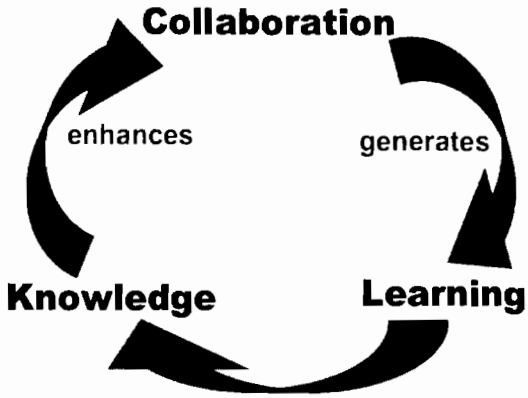


Figure 2.1: Effect of collaboration in learning [18]

Since a child's social identity is enhanced by collaborating in a group, involving children in a social intellectual activity can be a great motivator and can contribute to better learning than depending on individual work [19]. However, it has been proven that much effort has to be done to motivate learning by collaboration [20] and finding opportunities to drive students to consider collaboration as a basic activity in learning process.

Nowadays, PC plays unique role in learning and it makes a big distinguish in learning process. However, there is lack and scarcity of computer-supported social and collaborative aspects in learning [21], [2]. The researcher observation indicates that, the main reason for lacking software that support collaborative learning due to the fact resulted from standard computer applications which were geared to one user. With the emergence of so many technologies that support collaboration in one PC, customizing the interface has received a lot of attention.

2.2 Single Display Groupware

Single Display Groupware (SDG) is a technology that changed the traditional design of computer applications which allow just one user to access and interact with one PC and invented new way to allow more than one user using multiple input devices to stick with just one computer display [22]. SDG has given the ability and the initiative to researchers to utilize it to support collaboration.

SDG applications provide an input channel for every user through usages of a separate input device, but each must share the same single output channel (see Figure 2.2) These are the characteristics that give SDG applications their distinguished character: the combination of multiple independent input devices together with a single shared output channel. The usability studies conducted to evaluate SDG have shown that existing technology has a number of shortcomings when utilized for co-present collaboration [23]. Although SDG has given PC applications the capability of allowing more than one user to access the same computer simultaneously, but interface design still open to question to make any application totally fit to do its functionality.

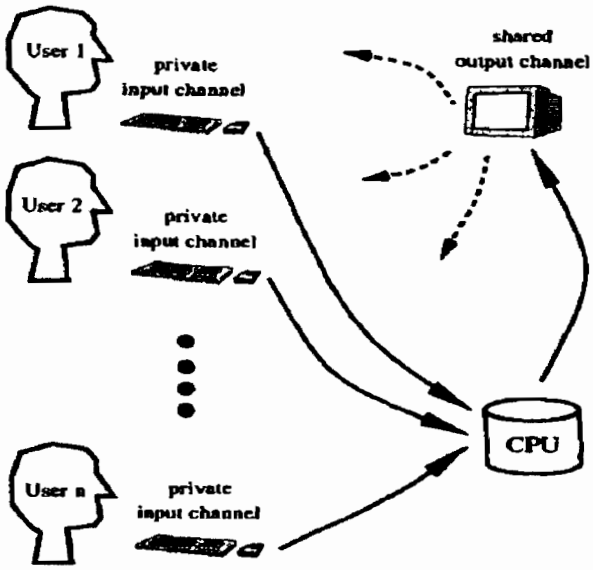


Figure 2.2: Single Display Groupware [23]

2.3 Collaborative Puzzle Game

Our daily life observations indicate that, children love to play with each other, seek the opportunity to collaborate, and they can be oriented in a positive way if we provide to them fun. Thus, we can make advantage of these sociable attributes to be geared to educational purposes.

A study involved pairs of children through playing a puzzle-solving game using three various experimental set-ups: (1) a paper-based style of the game with physical pieces; (2) a computer-based style of the game with one cursor and one mouse; and (3) a computer-based version of the game with two cursors and two mice. The puzzle-solving

game was developed in this study requires placing alien faces with various attributes in a row according to a specific pattern. The software was developed using C++ and Microsoft DirectX and utilized input from one or more Universal Serial Bus (USB) mice [24].

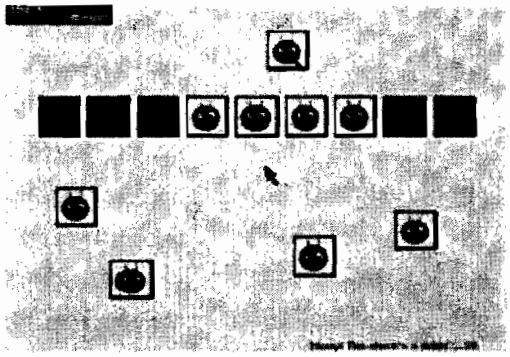


Figure 2.3: Sample puzzle screen [24]

The results of this study show that providing children with support for their collaborative interactions can enhance their performance of activity, engagement, and motivation. Moreover, it is clear that one way to help ensure children are fully engaged in a computer-based learning activity is to allow children to interact simultaneously in the class though multiple mice way. Children's preferences in the study via the three collaborative conditions are shown in Table 2.1.

Table 2.1: Number of children who preferred each collaborative setup [24]

Collaborative Condition	Count	%
Paper-based version	5	12.5%
One-mouse/one-cursor version	7	17.5%
Two-mouse/two-cursors version	28	70.0%
Total	40	100%

The study shows that children who engaged in this study seemed to completely enjoy playing together on a computer with multiple mice (see Figure 2.4).

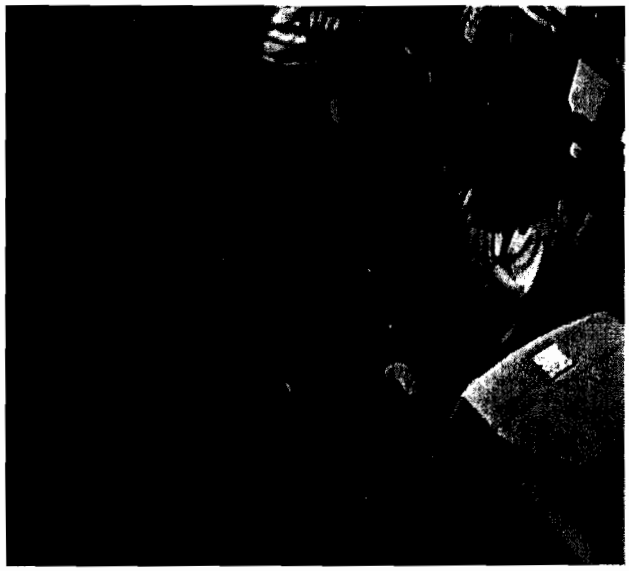


Figure 2.4: "This is fun. We're all best friends and we're all playing!" [24]

PC has a lot of features; appealing graphics and high speed to do functions which contributed to attract children, they can consume a long time for the purpose of using

computer games or any educational software. In particular, when it comes to collaboration they prefer computer as an ideal environment to interact with each other.

2.4 Multiple Users Prototype

Pawar et al. [25] created prototype in order to allow multiple users with multiple inputs and stated that, it is vital to take into consideration that Human - Computer Interaction (HCI) design can extract beneficial aspects of interaction between multiple children at each computer and make sure that a component of that is retained in a new system with a better individualized child-machine interaction (see Figure 2.5).

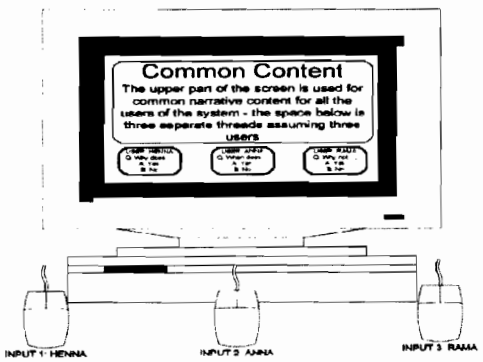


Figure 2.5: Sample prototype of multi-user content with multiple inputs [25]

Design is so important in software development, especially in applications that support collaborative interaction; it has to satiate users' needs and give them the same ability that they used to have in traditional applications.

2.5 Multimouse Technology

Pawar et al. [26] have conducted an experiment with 238 school children in rural India to examine retention of English vocabulary. Given that, PCs in developing countries especially rural ones used in a one-to-many style due to the lack of funding and economic problems. In some cases it is noticeable that ten students gathered around one PC (see Figure 2.6).

Figure 2.6: Eight boys crowded around one PC in rural India [26]

Multimouse is one application of Single Display Groupware (SDG) design, which requires a mouse in addition to separate on-screen cursor for every user. Multiple mice can connect to every PC by using Universal Serial Bus (USB) ports and we can extend the number of mice theoretically via USP hubs, Multimouse is directed to just one application at a time and is not considered a part in the shell of GUI itself [26]. Multimouse Software Development Kit (SDK) created via C Sharp programming language in Windows by using the Raw Input API within Win32. The API provides a distinguished identifier for every device, and gives us the ability to isolate every device-specific events. It is worth mentioning that Multimouse technology is totally based on

software, thus, not expensive and easy to spread and extend to a large scale. Consequently, this extremely fit to developing-nations scenario which lack funding to support using technology in schools.

An application known as Multimouse racing mode (**MM-R**), used to raise competition among students through multiple-choice question to push students to choose the correct answer via every student cursor. In particular, inform students that the first click among them to choose the right answer will be considered and any further will not be considered (see Figure 2.7).

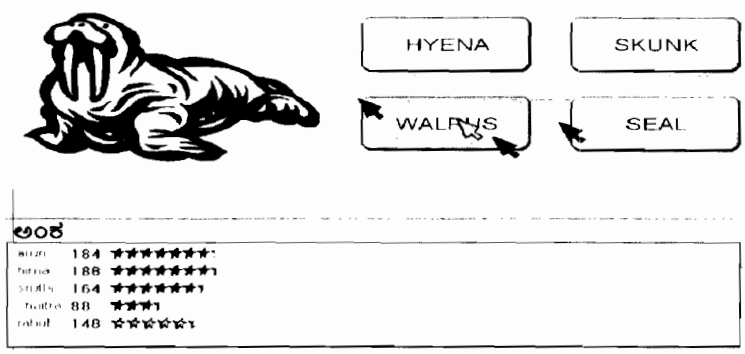


Figure 2.7: Screenshot describes Multimouse Racing Mode [26]

MM-R was designed to image-word association among students; however it has been discovered that this software does not attain the benefits as it supposed to be due to the reason that students tried to click randomly in order to click on the right answer. In addition, the faster turnaround of answers and fast convert to the next screen prohibited students who may have learnt in terms of visual recall from scoring for the reason of the short screen time of every question. It is noticeable that collaboration is not just allowing

every student to access the mouse among group but also give them the opportunity to get benefit, interact, to be truly effective among group.

In the same context, a new application known as **multiple-mice voting** (MM-V) has been designed to achieve the same goal [26]. The color of every choice changes to the cursor color as soon as a child clicks to choose an answer and operations goes on if another child clicks the same choice color will change to indicate the two cursors colors and so on. However, results were different due to the reason that the application does not shift to the next step without clicks from every mouse. Furthermore, quantitative data suggests that the collaboration achieved by MM-V is recognizable, in comparison with MM-R (see Figure 2.8).

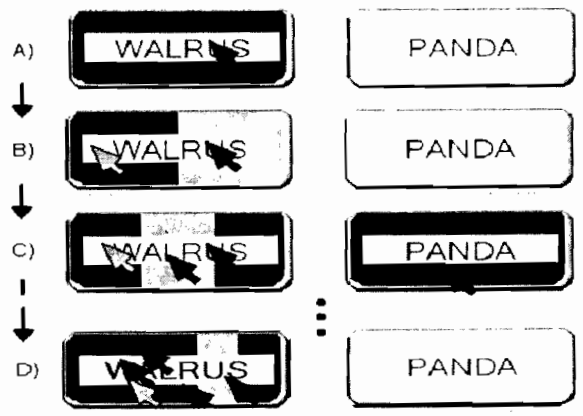


Figure 2.8: Screenshot describes Multiple-Mice Voting Mode [26]

It is clear that Interaction Design can address the need of primary school students and access the whole benefit of Microsoft Multipoint technology. Moreover, understanding

the nature of collaboration activities among them will give powerful interface and stunning functionality.

2.6 Multi Mouse Quiz Systems

Attracting children is considered as an important strategy to encourage them to collaborate and drive them to remain on task. Most importantly, giving children a bit of challenge which matches their abilities will strongly influence their desire to collaborate.

A multi-user quiz system named Multi Mouse Quiz (MMQ) used along with the concept of Socialized Computer (SC) which creates an atmosphere for the purpose of collaborative work via giving multiple users some control of the computer which has one single display to share information among the users [2]. Quiz is an important method to attract children to answer questions and solve problems. In essence, it stimulates them to think and in the same time they have fun.

MMQ was implemented in Visual C# using SDG toolkit which is a middleware for rapid prototyping of single display groupware, and it allows to treat information from connected multiple mice separately. The prototype assumes 1 to 4 users to play simultaneously. As a quiz, a series of questions were assumed, each question consists of question texts, one picture and multiple choices [2] and quiz itself helps to make student excited during collaboration activity.

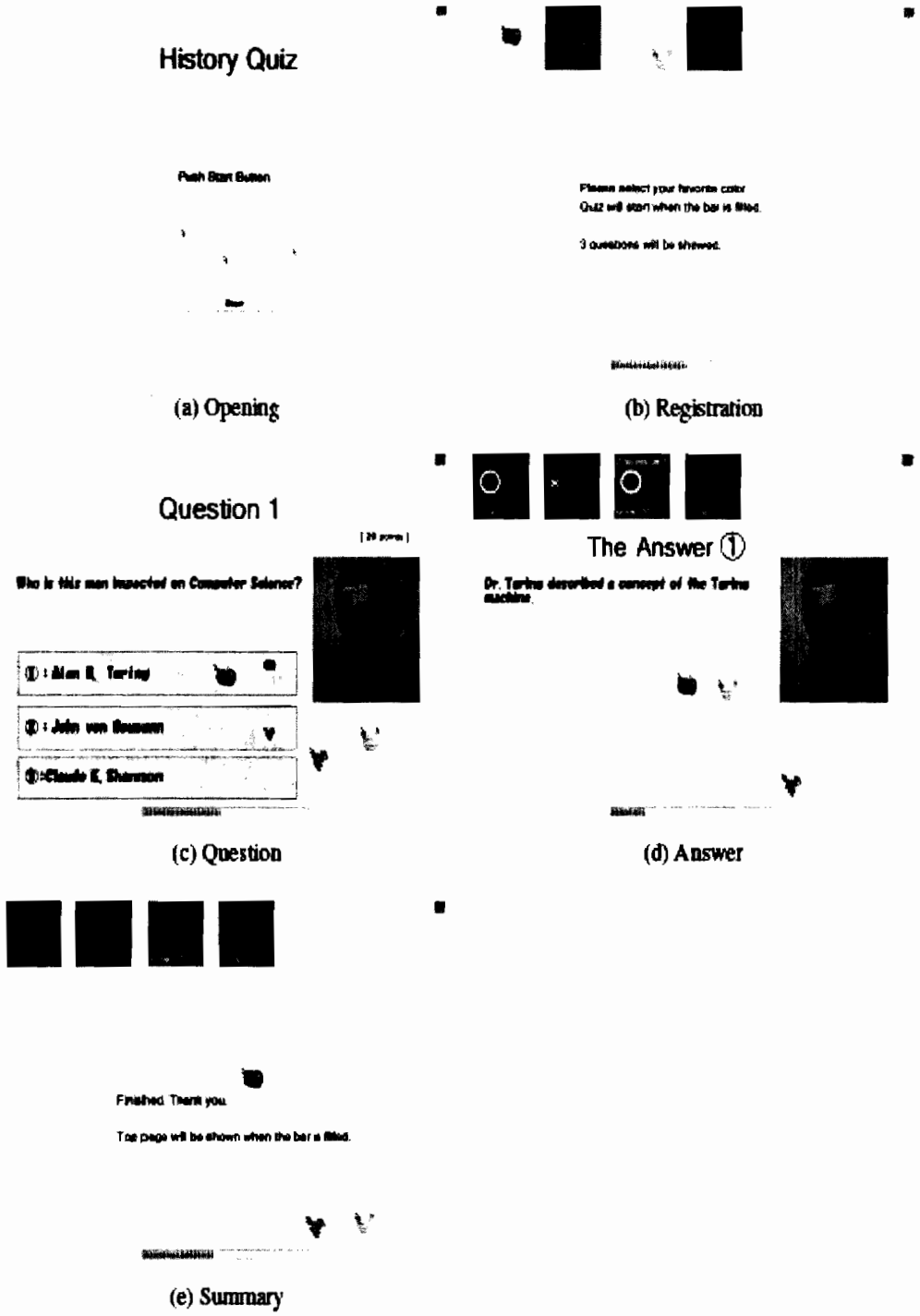


Figure 2.9: Screen shots of the prototype MMQ [2]

In the post-experiment questionnaires, students were asked a question of whether playing MMQ was fun to discover encouragement and motivation of playing. The result is shown in Figure 2.10, which was greatly positive.

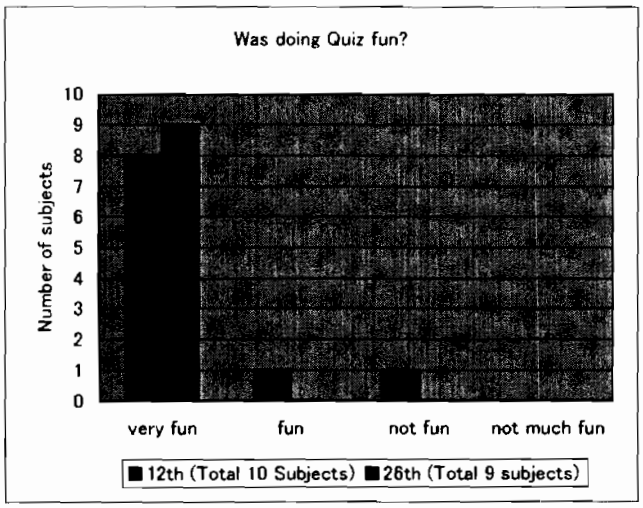


Figure 2.10: Result of a questionnaire asking whether MMQ was fun [2]

2.7 Summary

This Chapter highlighted practical applications that designed to allow users to share one computer screen via using more than one mouse and demonstrated the effect of this new approach on users. Moreover, it refers to the shortcoming of some of these approaches in terms of customizing design to be fully fit to users needs.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This Chapter demonstrates research methodology for the findings of this research project. Using prototyping method allows users to evaluate the proposed system and sheds light on their opinions regarding Usefulness, Ease of Use, Ease of Learning and Satisfaction. Interaction design used in prototyping in order to address users' aesthetic needs and identification of value in addition to other human emotional measures. Data collection methods involve questionnaire and observation which help to analyze data and extract outcomes.

Rapid Application Development (RAD) is a set of various methodologies which its main purpose to use special techniques and tools to accelerate the process of analysis, design and implementation in order to get some portion of the system developed fast and to be available to users for evaluation and feedback [27]. System Prototyping one of (RAD) methodologies which goes through all system development phases quickly including analysis, design, and implementation concurrently in order to quickly develop a simplified version of the system to allow users to have the opportunity to evaluate it [27]. Prototyping is a practical way used to improve planning and execution of software projects via building executable software systems (prototypes) for experimental objectives. It is very suitable for the purpose of gaining experience in new application areas and for supporting incremental or evolutionary software development [28]. In

essence, prototyping will help to visualize the research approach and make it tangible for usability evaluation.

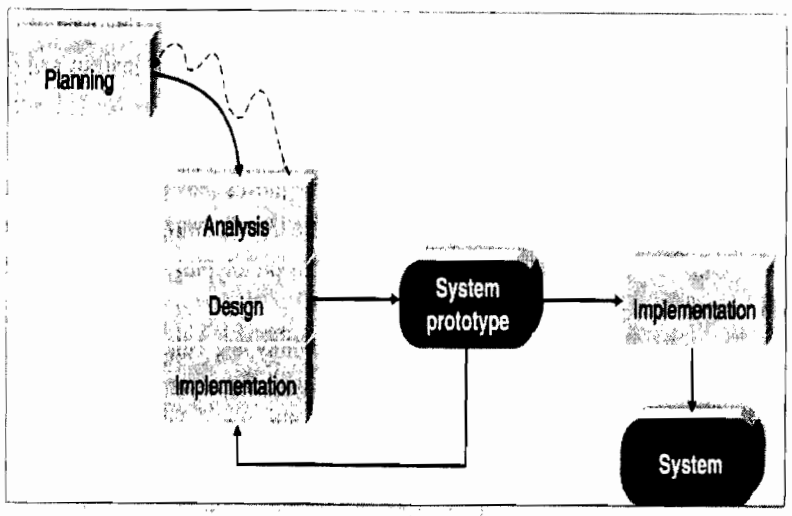


Figure 3.1: System Prototyping [27]

The planning is the corner stone in software development, it includes technical feasibility and economic feasibility in order to answer two questions at the very beginning why an information system should be built and how the system will be built. In essence, planning will identify the opportunity to build the prototype successfully.

Technical feasibility explores if the system could be successfully designed, developed and installed properly. Microsoft Multipoint Software Development Kit (SDK) version 1.1 gives the ability to interact with more than one mouse in one PC. Microsoft Multipoint is a development framework that allows developers to build applications that enable multiple mouse devices to work simultaneously on a single computer. Developer can use the Multipoint SDK to build educational applications for schools with limited

technological infrastructure, hence, increasing time access to every student in order to maximize the benefit of computer. Initial pilot programs conducted in India by Microsoft Research clearly denote that for specific subjects, collaborative learning technologies like Multipoint improve learning in comparison to a 1:1 computing scenario. The Multipoint SDK is essentially written for programming in Visual C#; hence, a lot of applications developed with it are written in Visual C#. However, using Visual Studio, it is possible to build Multipoint applications with either Visual C++ or Visual Basic [29]. Technical feasibility allows use to identify physically the opportunity to carry out collaborative software in one PC.

Economic feasibility analysis identifies the financial risk associated with the system in order to build Microsoft Multipoint technology based applications, thus, it is worth mentioning that there is no extra overhead cost except for using multiple mouse connecting via hub to successfully use the system.

The analysis phase main concern to answer essential questions of who is going to use the system, what the system will do, and where and when it will be used. Moreover, gathering hardware and software requirements help to build the system in qualified environment.

Identifying hardware and software requirements is crucial to check the availability to build and use the system. Table 3.1 shows hardware requirements which shed light on the

required processor, screen resolution, size of Random Access Memory (RAM), Universal Serial Bus (USB) and two to four mice for testing the system.

Table 3.1: Hardware Requirements

Hardware Requirements
• . Computer (a computer with a Pentium 4 processor has been tested successfully. • Two to four mice devices for testing • USB ports on the computer. • 128 megabytes (MB) of RAM (256 MB or higher is recommended) • 16-MB of video RAM (32 MB or higher is recommended) • 800x600 resolution set in video card (32-bit color is recommended)

Table 3.2 identifies software requirements to build the prototype such as the required Operating System (OP), Dot Net Framework (DNF) 3.0 or higher, Microsoft Expression Blend for interface design and Visual Studio 2005 with Windows Presentation Foundation (WPF) extensions or Visual Studio 2008.

Table 3.2: Software Requirements

Software Requirements
• Windows XP SP2 is recommended. The Multipoint SDK will also run on Windows Vista. • .NET Framework version 3.0 or higher. • Microsoft Expression Blend. • Visual Studio 2005 with WPF extensions installed (Visual Studio without WPF extensions is not supported by Multipoint), or Visual Studio 2008.

3.2 Prototyping and Interaction Design

Interaction Design, like other kinds of design, is more a process of creativity and novelty, that is, designers identify stunning opportunities where they could make a difference to people, and then better mix the various technologies and our daily life. Interaction Design borrows essential concepts from usability engineering. But other than that, Interaction Design also tends to concern for the users' aesthetic needs, identification of value and other human emotional measures [30] in addition to reduce users' cognitive load [31]. Developing Microsoft Multipoint applications faces a challenge of harnessing the interface to be fully fit for collaboration as a social activity and Interaction Design can play an important role to tackle this issue.

It is worth mentioning that by emphasizing the basic aspects of Interaction Design, designers would have clearer understanding of each of the objectives of prototyping, and hence, they would be able to make clear decision about which prototype to use. Such basic aspects include several questions which support prototyping process to achieve its goals: (1) who is the user and what is the usage scenario of the software? And what role is the software going to play in the user' life. (2) What kinds of sense and pleasure the software can bring to users. (3) How is the software to be structured? With what technology and approach? In essence, to better demonstrate these three aspects, they can be summarized as "scenario and character", "sense and pleasure" and "technology and structure" [30]. These aspects will lay down markers to build successful collaborative software, in terms of addressing the available technology and how we could customize it to satiate users' needs and make them attain high value.

3.2.1 Scenario and Character

Scenario has received big interest and attention of most designers. To review a design in an organized way, it is better to put it into a real and alive scenario instead of just independently analyze its various factors. Scenario is a process where all the stories unfolded on it. Through the software, designers can interact with other entities whether they are people or objects on the process. Thus, the concentration of the scenario would be the focus of design step. "Character" really involves two important parts: the persona related to the user of software, and the role the software plays in users' (persona) life. Persona is not an actual person, but being a hypothetical prototype of the actual user, it

shows a real person in the design process. Particularly, persona identifies needs and tasks related to user to attain their goals. In Interaction Design, when suitable "scenario and character" are prototyped; designers will discover it easier to demonstrate a stay point, and to avoid deviation as the process of design progresses.

To help this research makes persona real researcher sheds light on personal attributes and personal goals of children. This could also help to creating intrinsic value for children regarding collaboration activity [32].

The most vital step is the Interaction Design of the educational software in which Students concentrate on interactivity, and they are bored by explicitly instructional material. If there is a tutorial video to watch (for instance), followed by a quiz, students generally love to jump straight to the quiz [26]. The rational behind such fact was that children tend to be active, thus, applying this fact in software design will affect children response to collaborate.

An objective of educational software for children is to provide an engaging learning environment, keeping children's attention by providing fun. This is usually achieved through games [33], [34]. Moreover, Research data also showed that the use of games in education is perceived as useful for learning and helped to engage students in educational experiences towards achieving specific learning goals and outcomes [35] [36]. When game is interesting and successfully adopted in educational software, students' motivation is likely to rise.

Experiments show that for children point-and-click is a more impressive and effective mouse interaction style than drag-and-drop, both in terms of accuracy and speed. Moreover, children have propensity to prefer the point-and-click interaction style. It is worth mentioning that a point-and-click interaction style, used in an interactive learning environment, can be more effective from the perspective of motivation and performance than drag-and-drop interaction style [37]. It is so important to address available ways to allow children to interact with software contents and choose the most comfortable and preferable way to them.

3.2.2 Sense and Pleasure

This step involves visual and sense related to touch and hearing in addition to other sensual interaction experience. Through prototyping process of these related issues, the design would be focusing on simulating the potential sensual requirements of the target user, to demonstrate or record the users' interactive experience in the scenario, and to discover sensual elements that may have been ignored. Ultimately, the design would have to build up a visualized prototype in order to simulate the various senses aroused by the software.

Children's emotional state affects their skill acquisition. It is difficult to control over factors outside game environment that may affect children's emotional state, but using game elements to evoke emotions that may enhance skill development [37]. Particularly, fun is a perfect way to make positive impact on emotional state of children.

Having fun by playing the game is a form of collaboration; players could not compete if there were no one to play with. Collaborations with friends lend the game sense of novelty, surprise, and variety [39]. It is worth mentioning that fun is the most important goal for computer games. If players do not enjoy the game, they will not play it. Games create fun by challenging players, often testing out the limits of their memory and performance [40]. Consequently, by making a software fun, it is likely that children will effectively remain on task [41] and collaborate more and more.

3.2.3 Technology and Structure

Some prototypes are specifically for giving answers to technical requirements of the future software. Using Microsoft Multipoint SDK 1.1 with the hardware and software requirements sheds light on discovering practical design specifications and attains great feedback from the interaction with potential users.

Nowadays, games are designed to have positive impact on "social skills, education achievement and stunning effect on socialization and family functioning. In particular, games have prodigious effect in improving collaboration [42]. Making full use of games impact on children and merging it in a collaborative quiz will, hopefully maximize fun and drive students to be immersed in collaboration.

3.3 Flow of the Prototype System Usage

Figures 3.2 through Figure 3.5 are screen shots of the prototype system. Figure 3.2 shows the initial screen waiting for players. Two mouse pointers are represented by two arrow cursors at this time, one is blue and the other is green. When any one click "Start Game" button on Figure 3.3, children begin to solve the quiz. Figure 3.4 shows the progress of playing the game, where students get feedback of their scores and horses move across the track, according to the score achieved. Figure 3.5 is the final screen, where students will be notified about the winner of the game.

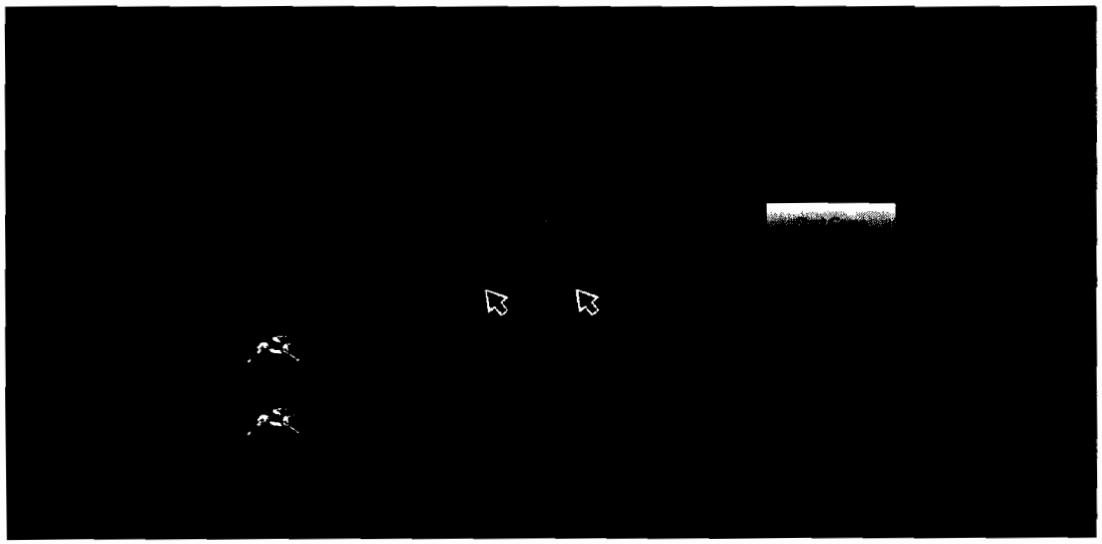


Figure 3.2: Initial screen of collaborative quiz game

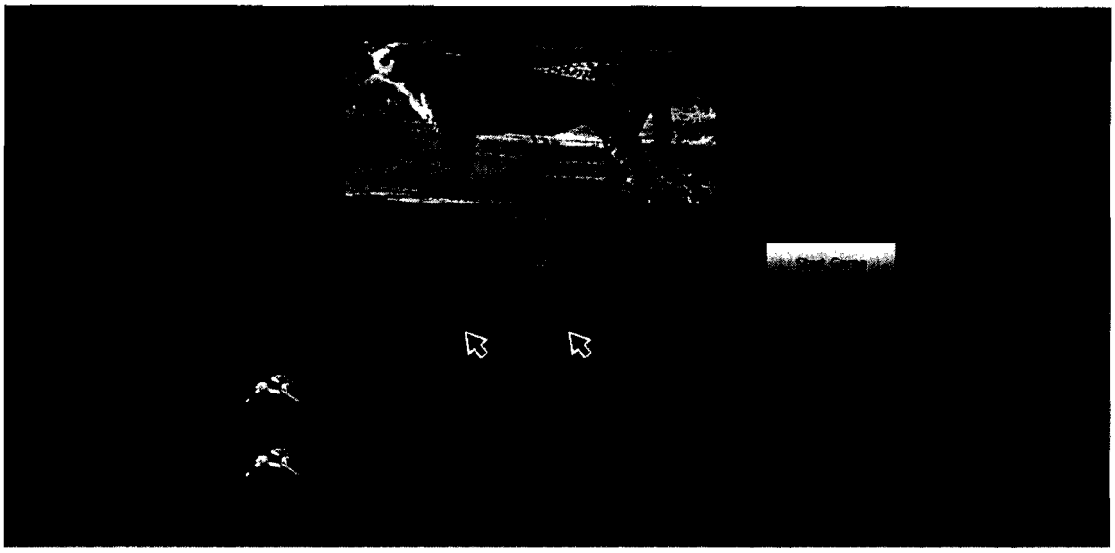


Figure 3.3: Students start playing the game

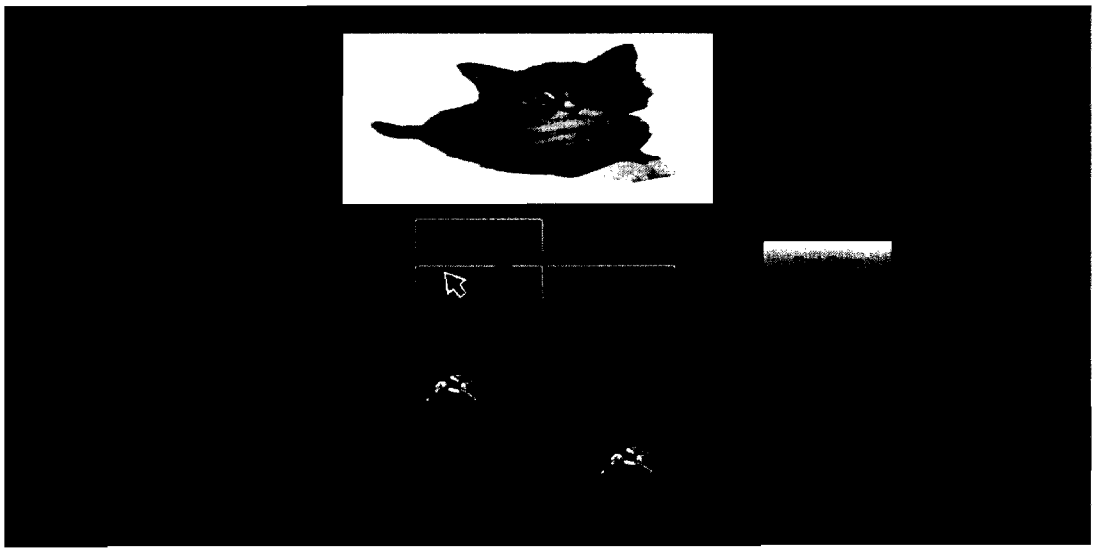


Figure 3.4: Progress of playing the game

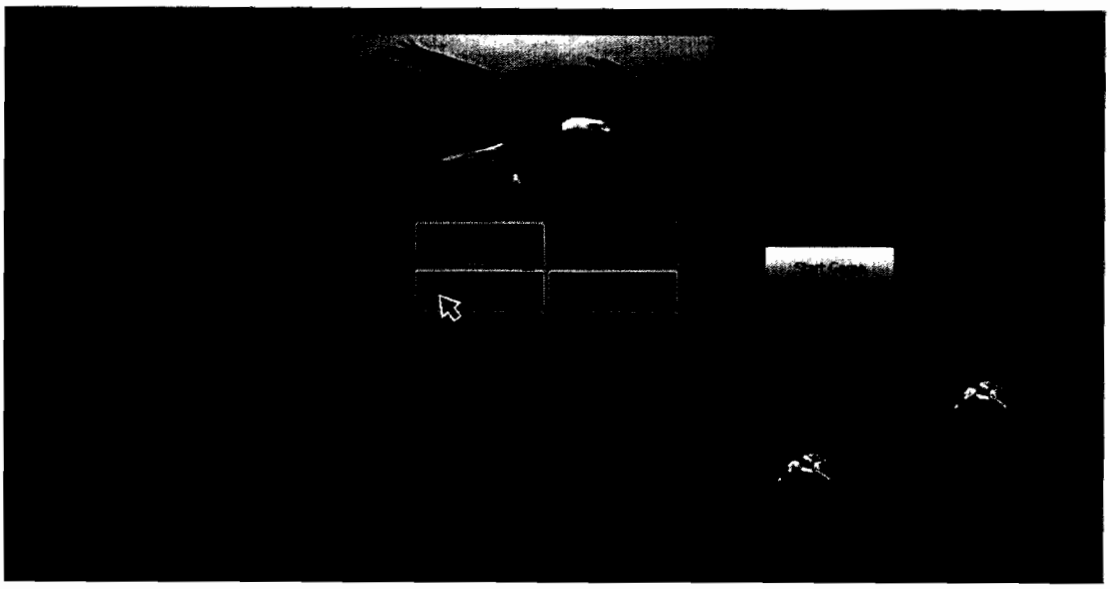


Figure 3.5: Final screen shows the winner of the game

3.4 Data Collection

Usability is an essential factor in establishing whether educational software will facilitate the acquisition of knowledge. In particular, ISO 9241-11 demonstrates usability as the extent to which software can be used by specific users to attain specific objectives with efficiency, effectiveness and satisfaction in a specific context of use. If users perceive that a system is very difficult to use, the perception may affect their ability to absorb material given by the system [33]. Usability testing has been studied extensively and is generally acknowledged to identify some of the key interaction problems in user interfaces [43] which in return identifies opportunities to make the system more appropriate to achieve its objectives.

The research used USE Questionnaire which stands for Usefulness, Satisfaction, and Ease of Use which could be separated into two factors, Ease of Learning and Ease of Use. These are the three dimensions that emerged most strongly in the early development of the USE Questionnaire. For many applications, Usability appears to consist of Usefulness and Ease of Use, which affects one another. Each in turn drives user Satisfaction and frequency of use. Users appear to have a good sense of what is usable and what is not, and can apply their internal metrics across domains. USE Questionnaire has proven useful in assessing several dimensions of usability across a wide variety of domains. The questionnaires were constructed as 5 point Likert rating scales, raging from strongly disagree to strongly agree. Various forms of the questionnaires were used to evaluate user attitudes towards the prototype.

The study took place in an educational suite of a local primary school in Malaysia involving 36 children from a Year 5 (aged 10) classroom. Children in this age range are relatively easy to include in software usability testing. Their experience in school makes them ready to sit at a task and follow directions from an adult, and they are generally not self-conscious about being observed as they play on the computer. Moreover, they will answer questions and try new things with ease. In particular, ten-year-old children may have extensive computer experience and be ready to critique research prototype [15]. Children participated to the activities of testing the prototype and filling in the questionnaire during a normal school day.

Groups of two children at a time were brought into the room and asked to play together the game set up for the study. Following the games, they were asked to fill in the evaluation sheet, consisting of questionnaire prepared for usability test.

3.5 Process of Conducting Usability Test

Rubin et al. [44] states that, the process for conducting a usability test consists of the following procedures:

1. Develop the Test Plan
2. Set Up a Testing Environment
3. Find and Select Participants
4. Prepare Test Materials
5. Conduct the Test Sessions
6. Debrief the Participant and Observers
7. Analyze Data and Observations
8. Report Findings and Recommendations

3.5.1 Develop the Test Plan

The test plan is the foundation for the entire test. It is meant for answering several questions the how, when, where, who, why and what of usability test. Test plan should communicate the resources that are required to complete the test successfully. Test plan formats will include the following parts:

1. Purpose, goals, and objectives of the test

2. Research questions
3. Participant characteristics
4. Method
5. Task List
6. Test environment, equipment, and logistics
7. Test moderator role
8. Data to be collected and evaluation measures
9. Report contents and presentation

3.5.1.1 Purpose, goals, and objectives of the test

It is so important to describe at a high level the reasons for doing this test at this time. In essence, the major focus or impetus of the research which in return will clarify specific procedures taken to achieve the research objectives. This research main objective to use Microsoft Multipoint technology to create collaborative quiz game in order to leverage collaborative learning and find opportunities to encourage them to be tottaly involved in collaboration.

3.5.1.2 Identify Research Questions

This section is the most essential part in test plan, because it identifies the issues and questions that need to be resolved, as well as the rest of the activities associated with planning, designing and conducting the test. It is of high importance that the research questions be as precise, accurate, clear, and measurable or observable as possible.

3.5.1.3 Participants characteristics

Describing children characteristics who are the end users of research prototype is considered a vital step in test plan. Moreover, choosing the right number of participants helps to achieve statistically valid results. The researcher took the whole population of the class (36 students) in a local primary school in Malaysia, they are in the fifth year and they are familiar about using computer and ready to critique the prototype without being shy.

3.5.1.4 Method

The researcher has to determine how to carry out the research with the participants, and how the test session will unfold. Essentially, to provide an overview of each facet of the test from the moment children begin the test until the end of the test and draw the road map for conducting the test. Research used questionnaire as an inquiry method to test the prototype and to attain feed back from children.

3.5.1.5 Task List

Task list comprises tasks that children will perform during the test. The list should consist of using the prototype and filling out the questionnaire. Task list has to be organized sequentially and in a proper way and assign suitable time for every step involved in the task list.

3.5.1.6 Test environment, equipment, and logistics

Describing the environment that the researcher will attempt to simulate during the test, and equipments like computers, in addition to data collection tools. This will ensure that the environment will be suitable and equipped with all required equipments like computers for running the prototype and whiteboard for the purpose of explanation the experiment to students.

3.5.1.7 Test moderator role

Moderator plays an important role to introduce the session, and introduce the tasks as appropriate in addition to taking detailed notes and record the participants' behavior and comments. The moderator of this research is Malaysian teacher who could contact with students effectively and explain every procedure in an understandable way.

3.5.1.8 Data to be collected and evaluation measures

This section of the test plan provides an overview of the types of measures the researcher will collect during the test, both performance and preference data. Performance data, representing measures of children behavior and preference data, representing children opinion and thoughts.

3.5.1.9 Report contents and presentation

This step meant for presenting findings for the original questions to investigate and giving quantitative results and discussing specifics as appropriate to the questions and data. Observations have to be included in the report contents to give clear insights to the experiment.

3.5.2 Set up a Testing Environment

Preparing testing environment by determining the appropriate location of usability testing, items that are particular to a test for instance, basic equipments like laptop in order to test the prototype, forms and note-taking tools.

3.5.3 Find and Select Participants

Selection and acquisition of users is crucial element of the testing process. After all, test results will be valid if the users who participate are typical users of the research prototype. Selecting participants involves identifying and describing the relevant behavior, skills, and knowledge of the users. This description is known as user profile of the target users.

3.5.4 Prepare Test Materials

One of the basic activities required to conduct a usability test is developing test materials that will be used to communicate with the users, collect the data, and satisfy

requirements. Test materials have to be readable and understandable for children in order to recognize every question and remove any ambiguity.

3.5.5 Conduct the Test Sessions

Having completed the basic groundwork and preparation of usability test, it is almost ready to begin testing. Conducting the test session successfully will help to attain meaningful and valid results. During test session the researcher has to remove any ambiguity and solve unexpected problems in a proper way.

3.5.6 Debrief the Participant and Observers

Debriefing refers to exploring and reviewing children actions during the performance portion of a usability test. Ideally, for every test the researcher should conduct to understand why every difficulty, error, and omission occurred for every child for every session. Using "Replay the test" technique is an excellent way to help children remember important points that they would forget to say. By actively recalling events that occurred, children recall their thoughts and feelings at the time of the usability test. Reviewing with observers is helpful to know what happened in each session and they become fully engaged in the usability test.

Analyzing Data and Observations in addition to Report Findings and Recommendations are the last steps in usability testing in order to extract the research outcome and report it,

which will help the researcher to gain insights of the prototype experiment; the researcher will discuss them in details in the following chapter.

3.6 Summary

This Chapter highlighted research methods used to develop windows - based game prototype and identified techniques geared to achieve research objectives. Data analysis and the expected results will be shown in Chapter Four in which more details will be revealed on research findings.

CHAPTER FOUR

RESULTS AND FINDINGS

4.1 Introduction

This chapter identifies results and findings of the research approach. For the purpose of prototype evaluation the researcher used USE Questionnaire which stands for Usefulness, Satisfaction, and Ease of Use which could be separated into two factors, Ease of Learning and Ease of Use. Cluster sampling is a sampling technique where the entire population is divided into groups, or clusters and a random sample of these clusters are selected, cluster sampling is typically used when the researcher cannot get a complete list of the members of a population they wish to study but can get a complete list of groups or 'clusters' of the population [45].

For the purpose of this research we focused on the fifth year cluster of primary schools and took the whole population of one class in it. Number of participants in this study was 36 students; 23 females and 13 males. For the purpose of testing and analyzing data, the research used Statistical Package for the Social Sciences program (SPSS). Observations of the study highlighted children attitudes; towards playing the research collaborative quiz game.

The experiment was done in a local primary school in Malaysia. Firstly, the researcher prepared the required tools such as laptop and questionnaire documents and organized the experiment environment to be sure that it is qualified for prototype testing. Secondly, the

researcher explained to students' teacher the procedures taken to implement the experiment in order to facilitate the process to students and communicate with them effectively. Thirdly, the researcher clarified to students the purpose of doing this research and their role to test the prototype. After that, students knew steps taken to test the prototype and began to play collaborative quiz game. Finally, they filled in questionnaire evaluation after an explanation of every question in the questionnaire list.

4.2 USE Questionnaire Analysis

USE Questionnaire consists of 30 questions; every question represented by a five-point Likert scale, ranges from strongly disagree to strongly agree and every point of scale represented by smiley face to help students recognize their choice; 8 questions to measure Usefulness, 11 questions to measure Ease of Use, 4 questions to measure Ease of Learning and 7 questions to measure Satisfaction.

4.2.1 Usefulness Attribute Analysis

Table 4.1 shows the statistics of the Usefulness Questions; it represents mean, median, mode, standard deviation, minimum and maximum of questions scales. Mean is used to measure the average score in a set of data; it found by adding up all the scores and dividing by the number of scores. Median is the point that divides the scores into two, with half the score below and half above the median. Mode is the score which has occurred the highest number of times in a set of data and standard deviation is a measure

of the spread or dispersion of a set of data. There was one missing values regarding question 3, the minimum chosen scale ranges from 1 (considering the prototype not very useful) to 3 (considering the prototype quite useful) and the maximum chosen scale was 5 (considering the prototype very useful); it revealed that students opinion varies from strongly disagree to strongly agree that the prototype was useful. The most frequently chosen scale was 5 (considering the prototype very useful) in all questions, except for question 2 and 7 it was 4; which reflects the tendency of children to consider the prototype very useful and in question 2 and 7 they consider the prototype useful.

Table 4.1: Statistics of Usefulness Questions

		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
N	Valid	36	36	35	36	36	36	36	36
	Missing	0	0	1	0	0	0	0	0
Mean		4.61	4.14	4.14	4.33	4.44	4.31	4.19	4.25
Median		5.00	4.00	4.00	5.00	5.00	4.50	4.00	4.00
Mode		5	4	5	5	5	5	4	5
Std. Deviation		.599	.723	1.033	1.014	.735	.786	.786	.874
Minimum		3	3	2	1	3	3	2	2
Maximum		5	5	5	5	5	5	5	5

In order to measure Usefulness attribute, the researcher took the average of all questions to every student and analysis of data was shown in Table 4.2. The minimum chosen scale was 3 (considering the prototype quite useful), the maximum was 5 and the most frequently chosen scale was 5 which refers to that students considered the prototype very useful through their experience on playing collaborative quiz game.

Table 4.2: Statistics of Usefulness Attribute

N	Valid	36
	Missing	0
Mean		4.42
Median		5.00
Mode		5
Std. Deviation		.732
Minimum		3
Maximum		5

The result considering Usefulness was mostly positive (see Figure 4.1); 55.6 % of stud*ents* (20 students) considered the prototype very useful, 30.6 % (11 students) considered it useful and the rest of students (5 students) considered it quite useful.

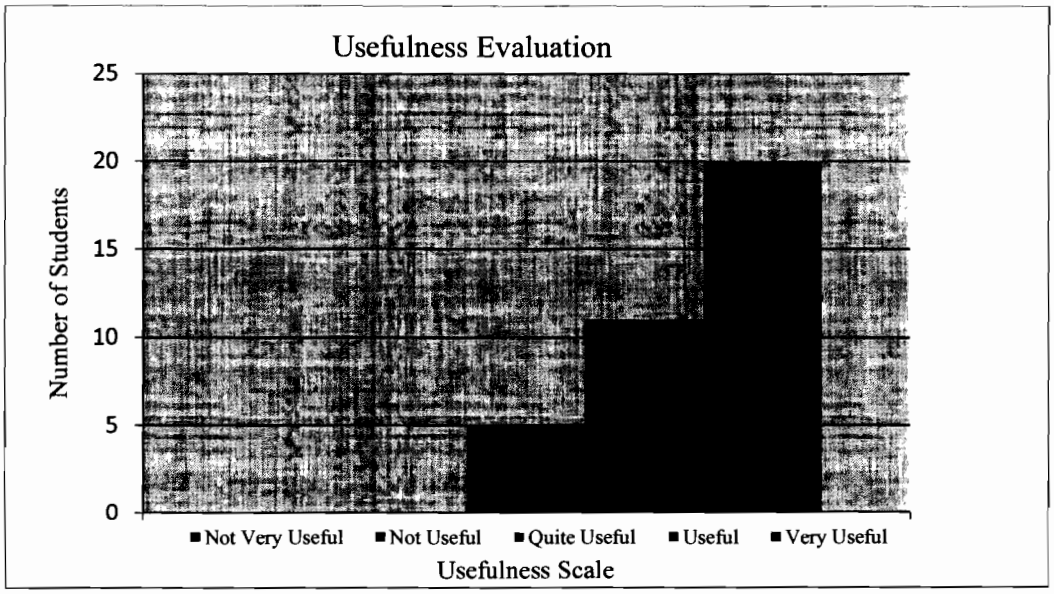


Figure 4.1: Usefulness evaluation

4.2.2 Ease of Use Attribute Analysis

Table 4.3 shows Statistics values of Ease of Use Questions. There was no missing values regarding Ease of Use questions, the minimum chosen scale ranges from 1 (considering the prototype not very easy to use) to 3 (considering the prototype quite easy to use) and the maximum chosen scale was just 5 to all questions; it reveals the variation of students to consider the prototype easy to use. However, the most frequently rate was 5 in all questions, except for question 10 it was 4 which identifies that the vast majority of students considered the collaborative quiz game very easy to use.

Table 4.3: Statistics of Ease of Use Questions

		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
N	Valid	36	36	36	36	36	36	36	36	36	36	36
	Missing	0	0	0	0	0	0	0	0	0	0	0
Mean		4.81	4.69	4.69	4.19	4.56	4.69	4.19	4.03	4.56	4.22	4.61
Median		5.00	5.00	5.00	4.00	5.00	5.00	4.50	4.00	5.00	4.00	5.00
Mode		5	5	5	5	5	5	5	5	5	4	5
Std. Deviation		.467	.577	.525	.920	.843	.577	1.064	1.108	.652	.760	.645
Minimum		3	3	3	2	1	3	1	1	3	2	3
Maximum		5	5	5	5	5	5	5	5	5	5	5

In order to measure Ease of Use attribute, average of all questions to every student was taken and analysis of data was shown in Table 4.4. The minimum chosen scale was 4 (considering the prototype useful), the maximum was 5 (considering the prototype very useful); deduction from student choices refers strongly to their approval of the ease of use related to the prototype and the most frequently rate was 5 (considering the prototype very useful) which refers to that the prototype tends to be easy to use.

Table 4.4: Statistics of Ease of Use Attribute

N	Valid	36
	Missing	0
Mean		4.61
Median		5.00
Mode		5
Std. Deviation		.494
Minimum		4
Maximum		5

The results shown in Figure 4.2 were positive with Ease of Use evaluation where 61.1 % (22 students) assumed that it is very easy to use and the rest (14 students) assumed that it is easy to use.

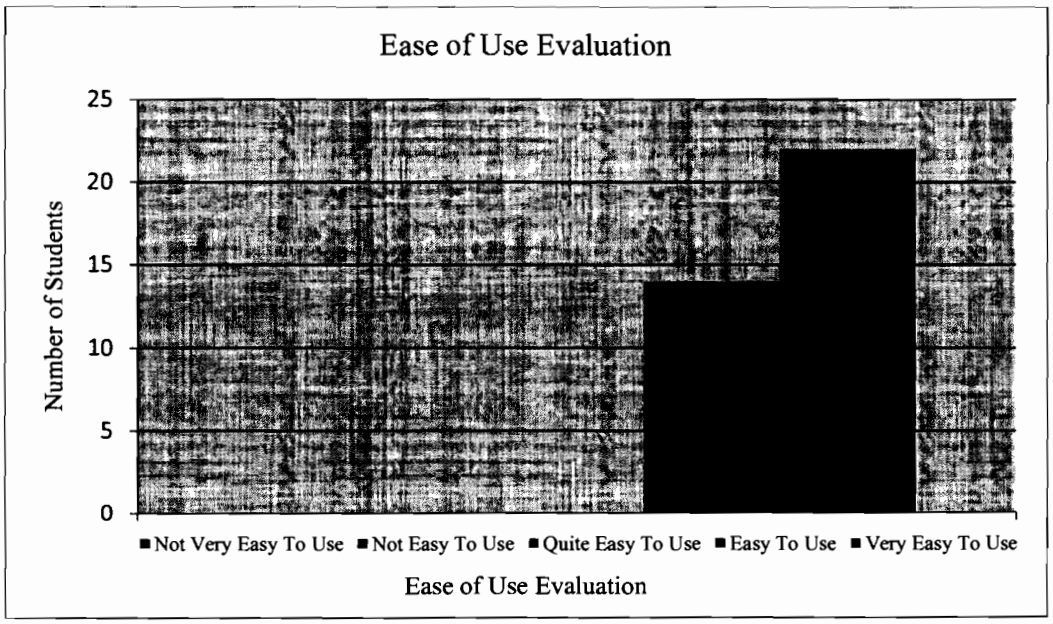


Figure 4.2: Ease of Use evaluation

4.2.3 Ease of Learning Attribute Analysis

Table 4.5 shows Statistics values of Ease of Learning questions. There was no missing values regarding Ease of Use questions, the minimum chosen scale was just 3 (considering the prototype quite easy to learn) in all questions and the maximum chosen scale was 5 (considering the prototype very easy to learn) to all questions; from students' choices we assume that they did not find any difficulty regarding ease of learning. The most frequently chosen scale was 5 in all questions.

Table 4.5: Statistics of Ease of Learning Questions

		Q1	Q2	Q3	Q4
N	Valid	36	36	36	36
	Missing	0	0	0	0
Mean		4.81	4.72	4.72	4.33
Median		5.00	5.00	5.00	4.50
Mode		5	5	5	5
Std. Deviation		.467	.513	.513	.756
Minimum		3	3	3	3
Maximum		5	5	5	5

In order to measure Ease of Learning attribute, average of all questions to every student was taken and analysis of data was shown in Table 4.8. The minimum chosen scale was 4 (considering the prototype useful), the maximum was 5 (considering the prototype very useful) and the most frequently chosen scale was 5; through student experience, the vast majority assumed the ease of learning of the prototype. The results reflect the simplicity

of the collaborative quiz game which derived students to learn it so easily without exerting a lot of effort.

Table 4.6: Statistics of Ease of Learning Attribute

N	Valid	36
	Missing	0
Mean		4.75
Median		5.00
Mode		5
Std. Deviation		.439
Minimum		4
Maximum		5

Regarding Ease of Learning the result was absolutely positive where most of students; about 75% (27 students) have perceived it very easy to learn and 25 % (9 students) of them perceived it as easy to learn (see Figure 4.3).

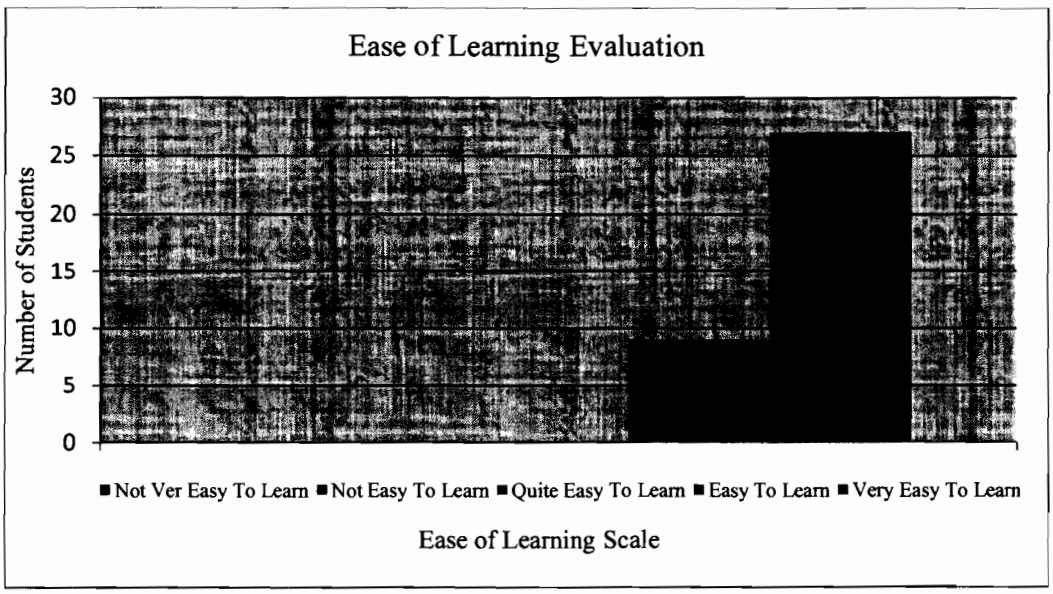


Figure 4.3: Ease of Learning evaluation

4.2.4 Satisfaction Attribute Analysis

Table 4.7 shows Statistics values of Satisfaction questions. There was no missing values regarding Satisfaction questions, the minimum chosen scale ranges from 1 (considering children choice of being not very satisfied) to 3 (considering children choice of being quite satisfied) and the maximum chosen scale was 5 (considering children choice of being very satisfied) to all questions. The most frequently chosen scale was 5 in all questions; this frequently choice reflects the high satisfaction which students got through the experience.

Table 4.7: Statistics of Satisfaction Questions

		Q1	Q2	Q5	Q6	Q7	Q8	Q9
N	Valid	36	36	36	36	36	36	36
	Missing	0	0	0	0	0	0	0
Mean		4.58	4.44	4.75	4.42	4.42	4.39	4.64
Median		5.00	5.00	5.00	5.00	5.00	5.00	5.00
Mode		5	5	5	5	5	5	5
Std. Deviation		.806	.843	.500	.732	.692	.838	.593
Minimum		2	1	3	3	3	2	3
Maximum		5	5	5	5	5	5	5

In order to measure Satisfaction attribute, average of all questions to every student was taken and analysis of data was shown in Table 4.8. The minimum chosen scale was 3 (considering the children choice of being quite satisfied), the maximum was 5 (considering children choice of being very satisfied) and the most frequently chosen scale

was 5 which refers to the high satisfaction students attained through playing collaborative quiz game.

Table 4.8: Statistics of Satisfaction Attribute

N	Valid	36
	Missing	0
Mean		4.56
Median		5.00
Mode		5
Std. Deviation		.607
Minimum		3
Maximum		5

The last attribute was satisfaction where 61.1 % (22 students) attained high satisfaction, 33.3 % (12 students) were satisfied and the rest (2 students) were quite satisfied (see Figure 4.4).

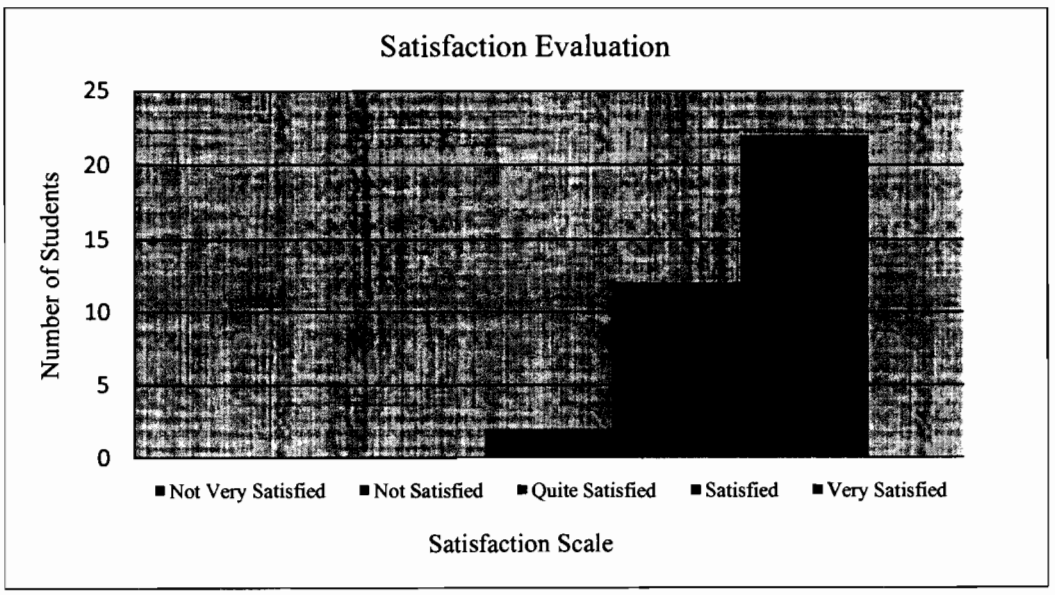


Figure 4.4: Satisfaction evaluation

4.3 Experiment Observation

From the observation during play, students recognized their cursors easily, in addition to the flow of playing the game with just a simple explanation before they began the game. Most of them interacted with their friend's action, like clicking on the right answer so fast, in order to progress and to move their horses on track. Students express their opinions after playing the game; some examples of their opinions were "it teaches us spelling", "it is fun "and "it has animation". Some students suggest putting something else instead of the horse on the track; like smiley faces.

4.4 Summary

This chapter analyzed all data collected during questionnaire sessions. All the data collected was analyzed and the results were interpreted. Based on the analysis of data, most children found the collaborative quiz game easy to use, easy to learn and they were so satisfied. Moreover, they interacted with each other effectively in collaborative way. This demonstrates that the research has successfully achieved its objectives in order to leverage collaborative learning.

CHAPTER FIVE

CONCLUSION

5.1 Introduction

This chapter sheds light on the overall progress of this research. It refers to the limitations encountered during prototype development. Furthermore, recommendations which will identify opportunities for the purpose of future work.

5.2 Overview

This research meant to use Microsoft Multipoint to create collaborative quiz game, for the purpose of supporting collaborative learning. The research used Prototyping to build executable software systems for experimental objectives. Merging Interaction Design with prototyping helped to discover the persona of children, their preferences and identified their needs to achieve the research objectives. Collaborative quiz game; our research prototype was created via using Microsoft Multipoint SDK 1.1 and C Sharp programming language. The prototype matched the research process, to encourage the collaboration among children by customizing the interface to be easy to use, easy to learn and adding appealing animations has given much fun to the collaboration environment. The experiment was done in local primary school in Malaysia, and usability evaluation has shown that children found the prototype useful, easy to use, and easy to learn and they were satisfied.

5.3 Limitations of Collaborative quiz game

Although the research has successfully achieved the proposed objectives, but some limitations have to be taken into consideration. First, some students suggested using different character instead of the horse element in the prototype; for instance, smiley faces. Secondly, the prototype did not use graphics which can maximize the value of fun during the experiment. Finally, collaborative quiz game focused on multiple choice features, but using text- based content can give the collaboration new dimension and new opportunity.

5.4 Recommendations for Future Work

In order to overcome limitations encountered during collaborative quiz game prototype, several suggestions recommended, for the purpose of gaining more progress in the upcoming future. First, it is important to encourage collaboration among children in order to remain on task for a long time. This could be done by adding more attractive features like graphics and more appealing animation. It is noticeable that children prefer some characters to interact with it; hence, identifying children preferences towards graphics elements will add great value to collaboration environment. Secondly, using text- based content in the collaborative game will identify different opportunities and skills attained by children. Finally, increasing number of children to involve in collaboration will help to maximize the value of interacting with each other in class room.

References

[1] Y. Lai & T. Wong. "Developing creativity in computer lessons,"*ACM*. 41(2), June 2009.

[2] K. Ikeda et al. "Development of a Multiple User Quiz System on a Shared Display," *Creating, Connecting and Collaborating through Computing, 2009. C5 '09. Seventh International Conference*. 103-110, January 2009.

[3] D. Africano, S. Berg, K. Lindbergh, P. Lundholm, F. Nilbrink & A. Persson. "Designing tangible interfaces for children's collaboration," *CHI '04 extended abstracts on Human factors in computing systems.ACM*. April 2004.

[4] D. Stanton, H. Neale & V. Bayon." Interfaces to support children's co-present collaboration: multiple mice and tangible technologies," *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community. International Society of the Learning Sciences.* January 2002.

[5] P. Wyeth, C. Diercke & Stephen Viller. "Design for inspiration: children, personal connections and educational technology," *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments. ACM.* November 2006.

[6] L. Lipponen. "The challenges for computer supported collaborative learning in elementary and secondary level: Finnish perspectives," *Proceedings of the 1999*

conference on Computer support for collaborative learning. International Society of the Learning Sciences. December 1999.

[7] A. Druin & K. Inkpen. "When are Personal Technologies for Children?," *Personal and Ubiquitous Computing. Springer-Verlag*. 5 (3). January 2001.

[8] Microsoft News Center. "With Windows MultiPoint, Youths in Developing-World Classrooms Learn 21st-Century Skills". Retreived on 4 January 2010, from http://www.microsoft.com/presspass/features/2006/dec06/12-14MultiPoint.mspx .

[9] H. Neale et al. "Classroom collaboration in the design of tangible interfaces for storytelling," *Proceedings of the SIGCHI conference on Human factors in computing systems. ACM*. March 2001.

[10] S. Benford et al. "Designing storytelling technologies to encouraging collaboration between young children," *Proceedings of the SIGCHI conference on Human factors in computing systems.ACM*. 2000.

[11] D. Bäumer, W. R. Bischofberger, H. Lichter & H. Züllighoven. "User interface prototyping—concepts, tools, and experience," *Proceedings of the 18th international conference on Software engineering. IEEE Computer Society*. May 1996.

[12] M. Hua & H. Qiu. "The prototyping in Interaction Design," *Computer-Aided Industrial Design and Conceptual Design*. 468 – 471, 2008.

[13] U. S. Pawar, J. Pal, R. Gupta & K. Toyama. "Multiple mice for retention tasks in disadvantaged schools," *Proceedings of the SIGCHI conference on Human factors in computing systems.ACM*. April 2007.

[14] E. Mazzone, D. Xu & J. C. Read. "Design in evaluation: reflections on designing for children's technology," *Proceedings of the 21st British HCI Group Annual Conference on HCI. British Computer Society*. 2 . September 2007.

[15] L. Hanna, K. Risden & K. Alexander. "Guidelines for usability testing with children," *ACM*. September 1997.

[16] D. M. Hilbert & D. F. Redmiles. "Extracting usability information from user interface events," *ACM*. 32(4). December 2000.

[17] A. A. Lund. "USE Questionnaire Resource Page". Retreived on 23 May 2010, from http://usesurvey.com/.

[18] J. Vicic, B. Kavsek, M. Kljun & Brodnik, A. "Extending Traditional Learning by Enforcing Collaboration and Self-Assessment," *Information Technology Interfaces, 29thInternationalConference*. 387 – 392, June 2007.

[19] F. Garzotto & M. Forfori." Hyperstories and social interaction in 2D and 3D edutainment spaces for children," *Proceedings of the seventeenth conference on Hypertext and hypermedia. ACM.* August 2006.

[20] M. Ratcliffe, J. Holloway & W. Ellis. "Enhancing student learning through collaboration," *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education. ACM.* June 2004.

[21] K. Issroff, E. Scanlon & A. Jones. "Two empirical studies of computer-supported collaborative learning in science: methodological and affective implications," *Proceedings of the 2nd international conference on Computer support for collaborative learning. International Society of the Learning Sciences.* December 1997.

[22] J. E. Stewart. "Single display groupware," *Extended abstracts on Human factors in computing systems: looking to the future. ACM.* March 1997.

[23] J. Stewart, B. B. Bederson, A. Druin. " Single display groupware: a model for co-present collaboration," *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit.ACM.* May 1999.

[24] K. M. Inkpen, W. Ho-Ching, O. Kuederle, S. D. Scott & G. B. D. Shoemaker. "This is fun! we're all best friends and we're all playing: supporting children's synchronous collaboration," *Proceedings of the 1999 conference on Computer support for collaborative learning. International Society of the Learning Sciences.* December 1999.

[25] U.S. Pawar, J. Pal & K. Toyama. "Multiple Mice for Computers in Education in Developing Countries," *Information and Communication Technologies and Development. ICTD '06. International Conference*. 64 – 71, 2006.

[26] U. S. Pawar, J. Pal, R. Gupta & K. Toyama. "Multiple mice for retention tasks in disadvantaged schools," *Proceedings of the SIGCHI conference on Human factors in computing systems.ACM*. April 2007.

[27] A. Dennis, B. H. Wixom & R. M. Roth. *Systems Analysis and Design*. John Wiley & Sons, Inc. 2010.

[28] D. Bäumer, W. R. Bischofberger, H. Lichter & H. Züllighoven. "User interface prototyping—concepts, tools, and experience," *Proceedings of the 18th international conference on Software engineering. IEEE Computer Society*. May 1996.

[29] **Microsoft Download Center.** Microsoft MultiPoint Software Development Kit (SDK) 1.1. Retrieved on 23 May 2010, from **http://www.microsoft.com/downloads/details.aspx?FamilyID=f851122a-4925-4788-bc39-409644ce0f9b&displayLang=en**

[30] M. Hua & H. Qiu. "The prototyping in Interaction Design," *Computer-Aided Industrial Design and Conceptual Design*. 468 – 471, 2008.

[31] J. Ma, X. Wang & F. Wang. "Research of interaction design method based on metaphor," *Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD.* 142 – 145. 2009.

[32] U. Dantin. "Application of personas in user interface design for educational software," *Proceedings of the 7th Australasian conference on Computing education. Australian Computer Society, Inc.* 42. January 2005.

[33] S. MacFarlane, G. Sim & M. Horton. "Assessing usability and fun in educational software," *Proceedings of the 2005 conference on Interaction design and children.ACM.* June 2005.

[34] J. A. Fails et al. "Child's play: a comparison of desktop and physical interactive environments," *Proceedings of the 2005 conference on Interaction design and children. ACM.* June 2005.

[35] W. S. Yue & N. A. M. Zin." Usability evaluation for history educational games," *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human. ACM.* November 2009.

[36] L. Xie, A. N. Antle & N. Motamedi. "Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces,"

Proceedings of the 2nd international conference on Tangible and embedded interaction. ACM. February 2008.

[37] K. M. Inkpen. "Drag-and-drop versus point-and-click mouse interaction styles for children," *Transactions on Computer-Human Interaction (TOCHI). ACM.* 8(1). March 2001.

[38] H. Gelderblom & P. Kotzé. "Designing technology for young children: what we can learn from theories of cognitive development," *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology.ACM.* October 2008.

[39] B. Nardi & Justin Harris. "Strangers and friends: collaborative play in world of warcraft," *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work. ACM.* November 2006.

[40] M. Obrist et al. ""Now you need to laugh!": investigating fun in games with children," *Proceedings of the International Conference on Advances in Computer Entertainment Technology. ACM.* October 2009.

[41] S. Chiasson & C. Gutwin. "Testing the media equation with children," *Proceedings*

of the SIGCHI conference on Human factors in computing systems. ACM. April 2005.

[42] J. A. Polack-Wahl. "Game development, social responsibility, and teaching. Consortium for Computing Sciences in Colleges". 24(2). December 2008.

[43] B. S. Als, J. J. Jensen & M. B. Skov." Comparison of think-aloud and constructive interaction in usability testing with children," *Proceedings of the 2005 conference on Interaction design and children. ACM.* June 2005.

[44] J. Rubin & D. Chisnell. *Handbook of Usability Testing.* Wiley Publishing, Inc. 2008.

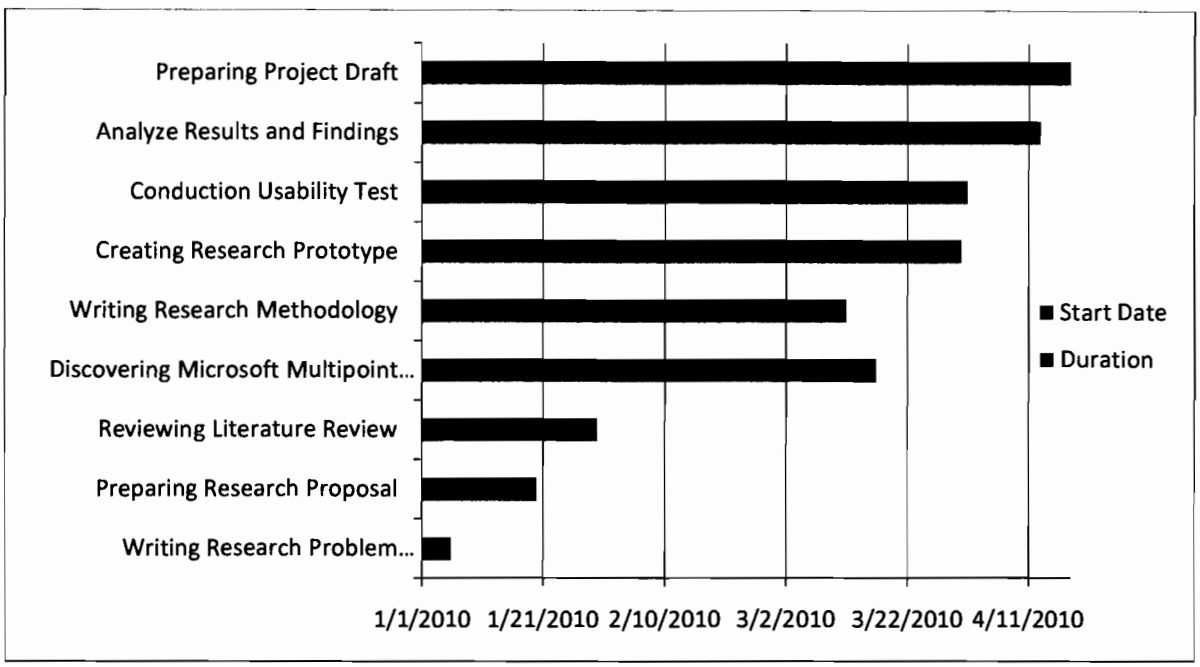
[45] J. Easton. Valerie & H. McColl. John. "Statistics Glossary". Retrieved from http://www.stats.gla.ac.uk/steps/glossary/sampling.html#clustsamp.

APPENDIX

APPENDIX A

Project Gantt Chart

This section highlights research process in order to accomplish all objectives. The following Gant Chart diagram demonstrates procedures taken through the research process.



Project Gantt Chart

APPENDIX B

Usefulness, Ease of Use, Ease of Learning and Satisfaction (USE) Questionnaire

Introduction:

This questionnaire meant to evaluate three attributes of usability; Usefulness, Ease of Use, which separated to two factors Ease of Use, and Ease of Learning and the last attribute, is Satisfaction. The questionnaire consists of 30 questions; Usefulness attribute consists of 8 questions, Ease of Use consists of 11 questions, Ease of Learning consists of 4 questions and satisfaction consists of 7 questions. The scale for the questionnaire ranges from 1 which refers to strongly disagree, to 5 which refers to strongly agree. Children will respond to questions by clicking on smiley face that matches their choice. For the purpose of conducting the test in local primary school in Malaysia we use Malay version of USE questionnaire.

The Questionnaire consists of two sections:

A- Respondent profile

B- USE questionnaire

Objective: measuring usability attributes; Usefulness, Ease of Use, Ease of Learning and Satisfaction

Section A:

Respondent Profile

Please answer the following question:

Gender

a- Male ( )
b- Female ( )

Section B:

This section contains 30 questions for the purpose of usability evaluation.

Please circle the appropriate number which indicates the extent to which you agree or disagree with the statement using the following scale:

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	2	3	4	5

USE Questionnaire (English version)

USEFULNESS			1	2	3	4	5		NA
1.	It helps me be more effective								
		strongly disagree	○	○	○	○	○	strongly agree	○
2.	It helps me be more productive.								
		strongly disagree	○	○	○	○	○	strongly agree	○
3.	It is useful.								
		strongly disagree	○	○	○	○	○	strongly agree	○
4.	It gives me more control over the activities in my life.								
		strongly disagree	○	○	○	○	○	strongly agree	○
5.	It makes the things I want to accomplish easier to get done.								
		strongly disagree	○	○	○	○	○	strongly agree	○
6.	It saves me time when I use it.								
		strongly disagree	○	○	○	○	○	strongly agree	○
7.	It meets my needs.								

		strongly disagree	○	○	○	○	○	strongly agree	○
8.	It does everything I would expect it to do.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
EASE OF USE			**1**	**2**	**3**	**4**	**5**		**NA**
9.	It is easy to use.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
10.	It is simple to use.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
11.	It is user friendly.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
12.	It requires the fewest steps possible to accomplish what I want to do with it.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
13.	It is flexible.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
14.	Using it is effortless.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
15.	I can use it without written instructions.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
16.	I don't notice any inconsistencies as I use it.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○
17.	Both occasional and regular users would like it.		■	■	■	■	■		
		strongly disagree	○	○	○	○	○	strongly agree	○

18.	I can recover from mistakes quickly and easily.								
		strongly disagree	○	○	○	○	○	strongly agree	○
19.	I can use it successfully every time.								
		strongly disagree	○	○	○	○	○	strongly agree	○
EASE OF LEARNING			**1**	**2**	**3**	**4**	**5**		**NA**
20.	I learned to use it quickly.								
		strongly disagree	○	○	○	○	○	strongly agree	○
21.	I easily remember how to use it.								
		strongly disagree	○	○	○	○	○	strongly agree	○
22.	It is easy to learn to use it.								
		strongly disagree	○	○	○	○	○	strongly agree	○
23.	I quickly became skillful with it.								
		strongly disagree	○	○	○	○	○	strongly agree	○
SATISFACTION			**1**	**2**	**3**	**4**	**5**		**NA**
24.	I am satisfied with it								
		strongly disagree	○	○	○	○	○	strongly agree	○
25.	I would recommend it to a friend.								
		strongly disagree	○	○	○	○	○	strongly agree	○
26.	It is fun to use.								
		strongly disagree	○	○	○	○	○	strongly agree	○
27.	It works the way I want it to work.								

		strongly disagree	○	○	○	○	○	strongly agree	○
28.	It is wonderful		●	●	●	●	●		
		strongly disagree	○	○	○	○	○	strongly agree	○
29.	I feel I need to have it.		●	●	●	●	●		
		strongly disagree	○	○	○	○	○	strongly agree	○
30.	It is pleasant to use.		●	●	●	●	●		
		strongly disagree	○	○	○	○	○	strongly agree	○

(USE) Questionnaire (Malay version)

Section A:

Respondent Profile

Sila jawab soalan berikut:

Jantina

1- Lelaki ( )

2- Perempuan ( )

Section B:

Sila bulatkan nombor yang sesuai mengikut sejauh mana anda bersetuju dengan pernyataan yang diberikan mengikut skala berikut:

Sangat tidak setuju	Tidak setuju	Berkecuali	Setuju	Sangat setuju
1	2	3	4	5

USEFULNESS			1	2	3	4	5		NA
1.	Ini membantu saya untuk lebih berkesan		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
2.	Ini membantu saya untuk lebih produktif		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	strongly agree	○
3.	Ia adalah berguna.		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
4.	Ia memberi saya lebih kawalan ke atas aktiviti harian saya.		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
5.	Ia memudahkan saya melaksanakan apa yang ingin saya capai.		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
6.	Ini menjimatkan masa saya apabila saya menggunakannya.		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
7.	Ini memenuhi keperluan saya.		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
8.	Ia melaksanakan semua perkara yang saya harapkan		■	■	■	■	■		
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○

EASE OF USE			1	2	3	4	5		NA
9.	Ia mudah digunakan								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
10.	Ia senang digunakan.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
11.	Ia mesra pengguna.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
12.	Ia hanya memerlukan beberapa langkah untuk mencapai apa yang ingin saya lakukan								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
13.	Ia adalah fleksibel.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
14.	Ia tidak memerlukan usaha yang banyak								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
15.	Saya dapat menggunakannya tanpa arahan bertulis								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
16.	Saya tiada mendapati ketidakkonsistenan semasa menggunakannya								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○

17.	Semua kategori pengguna menyukainya.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
18.	Saya mudah mengesan kesilapan dengan cepat								
.		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
19.	Saya berjaya menggunakannya setiap masa.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
EASE OF LEARNING			**1**	**2**	**3**	**4**	**5**		**NA**
20.	Saya dapat mempelajarinya dengan cepat.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
21.	Saya mudah mengingati bagaimana untuk menggunakannya.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
22.	Saya mudah untuk belajar menggunakannya.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
23.	Saya cepat menjadi mahir dengan menggunakannya.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
SATISFACTION			**1**	**2**	**3**	**4**	**5**		**NA**
24.	Saya berpuas hati dengannya.								
		Sangat tidak	○	○	○	○	○	Sangat setuju	○

		setuju							
25.	Saya akan mengesyorkannya kepada orang lain.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
26.	Ia seronok untuk digunakan								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
27.	Ia beroprasi dengan cara yang saya inginkan.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
28.	Ia sangat menarik.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
29.	Saya merasa perlu untuk memilikinya.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○
30.	Ia menyenangkan untuk digunakan.								
		Sangat tidak setuju	○	○	○	○	○	Sangat setuju	○

APPENDIX C

Questionnaire Answer Sheet

This section shows questionnaire answer sheet which conducted by the whole population of a local primary school in Malaysia for every usability attribute Usefulness, Ease of Use, Ease of Learning and Satisfaction.

Usefulness Answer Sheet

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
3	4	4	4	5	5	4	4
4	5	4	5	4	5	4	5
4	3	2	4	3	3	3	3
5	4	2	3	3	5	2	3
4	3	3	4	4	4	4	5
4	3	4	4	4	3	3	4
5	4	5	4	5	5	5	4
5	4	5	5	5	4	3	5
5	5	5	4	4	4	5	5
5	5	4	5	5	5	5	5
5	5	5	5	5	4	4	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	4
5	4	5	5	5	4	4	4
5	4	5	4	4	5	5	5
5	5	5	4	5	5	4	5
4	3	5	5	5	4	5	4
5	4	3	2	3	3	4	5
5	3	2	1	5	4	4	3
5	4	5	5	5	5	5	4
5	4	5	5	5	4	5	4
5	5	4	5	5	4	4	5
5	5	5	5	5	5	5	5
3	4	5	5	5	4	3	2
4	3		3	3	5	3	2
5	4	5	4	4	3	4	4
5	5	3	5	3	3	4	3
5	4	4	5	5	5	5	4
4	4	4	4	4	5	4	4
5	4	4	5	5	5	4	4
5	4	5	5	5	5	5	5
5	5	5	5	5	5	5	5
4	5	4	5	4	4	4	5
4	4	4	5	4	3	4	5
5	4	3	5	5	5	5	5
4	3	2	2	4	3	4	4

Ease of Use Answer Sheet

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
5	5	5	4	5	5	5	3	4	4	5
5	5	5	5	5	4	4	4	5	4	5
4	3	4	2	4	4	3	4	3	4	4
5	3	5	2	5	5	1	3	5	2	3
4	5	5	5	5	5	3	4	4	4	5
4	5	5	4	4	5	4	3	5	4	4
5	5	5	4	5	5	3	5	5	5	5
5	5	4	5	4	5	3	4	5	5	5
3	4	4	2	3	4	5	4	3	5	5
5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	4	5	5	5	5
5	5	5	5	5	5	5	5	4	4	5
5	5	5	5	5	5	5	5	5	4	5
5	5	4	5	5	5	5	4	5	5	5
5	4	5	4	5	5	5	4	4	5	5
5	5	5	4	5	4	5	5	5	4	3
5	5	5	5	4	5	4	5	3	4	5
5	4	4	3	5	3	1	5	5	3	5
5	5	5	4	5	5	5	5	5	5	5
5	5	5	5	4	5	4	4	4	4	4
5	4	5	4	5	5	4	5	5	4	5
5	4	5	4	5	4	4	5	4	5	4
5	5	5	5	5	5	3	5	5	5	5
4	5	5	3	5	5	5	3	4	4	5
4	4	4	5	4	4	5	3	4	3	4
5	5	4	5	4	5	5	5	5	3	5
5	5	3	3	4	3	4	2	5	3	3
5	4	5	4	5	4	4	3	4	4	5
5	5	4	4	3	5	5	5	5	4	4
5	5	5	4	5	5	4	4	5	4	5
5	5	5	5	1	5	5	1	5	5	5
5	5	5	5	5	5	5	1	5	5	5
5	5	4	4	5	5	4	5	5	5	4
5	5	5	4	5	5	5	4	5	5	4
5	5	5	5	5	5	5	4	5	4	5
5	5	5	4	5	5	5	4	4	4	5

Ease of Learning Answer Sheet

Q1	Q2	Q3	Q4
5	5	4	3
5	5	5	4
5	4	4	3
5	4	5	3
4	5	4	5
5	5	5	5
5	4	5	5
5	5	5	5
4	3	4	4
5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	4
5	5	4	4
5	4	5	4
5	5	3	4
5	5	5	5
5	4	5	4
5	5	5	4
5	5	5	4
5	5	5	5
5	5	4	3
5	5	5	4
5	5	5	5
3	4	5	3
4	5	5	3
5	4	5	5
4	5	4	5
5	5	5	5
5	5	5	5
4	5	4	4
5	5	5	4
5	4	5	5
5	5	5	5

Satisfaction Answer Sheet

Q1	Q2	Q3	Q4	Q5	Q6	Q7
4	5	4	5	4	3	5
5	4	5	5	5	5	4
4	4	5	4	4	3	3
5	5	5	4	3	5	4
4	4	4	3	4	5	4
4	4	5	4	5	4	4
5	5	5	5	5	4	5
5	5	5	3	5	5	5
5	5	4	4	5	5	4
5	5	5	5	5	5	5
5	5	5	5	5	5	5
5	5	5	5	5	5	5
4	5	5	4	5	5	5
5	5	5	5	5	5	5
5	3	4	5	5	4	5
5	5	4	4	5	5	5
5	4	5	5	4	3	5
5	4	5	5	5	4	4
5	4	5	5	5	5	5
5	4	5	4	4	4	5
5	5	5	4	4	4	5
5	4	5	5	4	5	5
5	5	5	5	5	5	5
2	5	4	3	4	5	5
2	1	3	5	4	4	4
5	3	5	4	3	3	5
5	4	4	3	3	2	3
4	5	5	5	4	5	5
5	4	5	4	3	3	4
3	5	5	3	4	5	5
5	5	5	5	5	5	5
5	5	5	5	5	5	5
5	5	5	5	4	4	4
5	4	5	4	4	5	5
4	5	5	5	5	5	5
5	5	5	5	5	4	5

APPENDIX D

Collaborative Quiz Game Development Programming Code

This section shows the written code for creating Collaborative Quiz Game. For the purpose of building the research prototype, the researcher used Microsoft Multipoint SDK 1.1, C# language, and Visual Studio 2008.

```
// All namespaces required for developing Microsoft Multipoint Application

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Threading;
using System.Windows.Threading;
using System.Timers;
using System.Data.SqlClient;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using Microsoft.MultiPoint.SDK;
using Microsoft.MultiPoint.CommonTypes;
using Microsoft.MultiPoint.Controls;
using Microsoft.MultiPoint.InputFilter;
using Microsoft.MultiPoint.MousePlugIn;
using Microsoft.MultiPoint.Provider;


namespace MultipointFinal
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {

        // code required for Timer events

            DispatcherTimer dispatcherTimer = new DispatcherTimer();


             private void mpButton_Click(object sender, RoutedEventArgs e)
             {
                 mpButton.Content = "Start Game";

                 dispatcherTimer.Tick += new EventHandler(dispatcherTimer_Tick);

                 dispatcherTimer.Interval = new TimeSpan(0, 0, 4);
```

```
        dispatcherTimer.Start();

    }

    public Window1()
    {
        InitializeComponent();

        this.Loaded += new RoutedEventHandler(this.Window1_Loaded);

        // Declare KeyEventHandler to handle keyboard events

        this.KeyDown += new KeyEventHandler(KeyDown_Event);



        // Declare MultiPoint mouse button handler
        mpButton.MultiPointClick += new RoutedEventHandler(mpButton_Click);

        button1.MultiPointClick += new RoutedEventHandler(button1_Click);
        button2.MultiPointClick += new RoutedEventHandler(button2_Click);
        button3.MultiPointClick += new RoutedEventHandler(button3_Click);
        button4.MultiPointClick += new RoutedEventHandler(button4_Click);
                                button1.MultiPointMouseLeaveEvent += new
          RoutedEventHandler(button1_MultiPointMouseLeaveEvent);
                                button2.MultiPointMouseLeaveEvent += new
          RoutedEventHandler(button2_MultiPointMouseLeaveEvent);

                                button3.MultiPointMouseLeaveEvent += new
          RoutedEventHandler(button3_MultiPointMouseLeaveEvent);

                                button4.MultiPointMouseLeaveEvent += new
          RoutedEventHandler(button4_MultiPointMouseLeaveEvent);




    }



    // 1. Set up the Window Loaded Event Handler, and Initialize the SDK
    private void Window1_Loaded(object sender, EventArgs e)
    {
         MultiPointSDK.Instance.Initialize(this);

        // handling arrival of mice

    MultiPointSDK.Instance.DeviceArrivalEvent += new
   EventHandler<DeviceNotifyEventArgs>(MultiPointObject_DeviceArrivalEvent);

  for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
        {
            // Act on each individual MouseDevice here


              DeviceInfo mouseObject =
            (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                MultiPointMouseDevice mpMouseDevice =
            (MultiPointMouseDevice)mouseObject.DeviceVisual;
                                mpMouseDevice.CursorColor =
           GetCursorColor(i);

        }
    }

    private System.Windows.Media.Color GetCursorColor(int id)
    {
        switch (id)
        {
            case 0:
```

```
                    return Colors.Blue;
                case 1:
                    return Colors.Green;
                case 2:
                    return Colors.Red;

                case 3:
                    return Colors.HotPink;
                default:
                    return Colors.HotPink;
            }
        }

        // Handling Mouse Connection and Disconnection

private void MultiPointObject_DeviceArrivalEvent(object sender, DeviceNotifyEventArgs e)
        {
            // add code to handle the arrival of the device here


            if (e.DeviceInfo.DeviceType == DeviceType.Mouse)
            {

                DeviceInfo mouseObject = e.DeviceInfo;
                 MultiPointMouseDevice mpMouseDevice =
               (MultiPointMouseDevice)mouseObject.DeviceVisual;
                                        Bitmap cursorBitmap =
               GetCursorImage(MultiPointSDK.Instance.MouseDeviceList.Count);
                                        mpMouseDevice.CursorImage =
               ConvertBitmapToBitmapImage(cursorBitmap);
                                    }
        }


        private Bitmap GetCursorImage(int id)
        {
            switch (id)
            {
                case 0:
                    return Properties.Resources.blue;
                case 1:
                    return Properties.Resources.green;
                case 2:
                    return Properties.Resources.red;
                case 3:
                    return Properties.Resources.pink;
                default: return Properties.Resources.blue;
            }
        }


public static BitmapImage ConvertBitmapToBitmapImage(System.Drawing.Bitmap b)
        {
            BitmapImage bmpimg = new BitmapImage();
            System.IO.MemoryStream memStream = new System.IO.MemoryStream();
            bmpimg.BeginInit();
            b.MakeTransparent(System.Drawing.Color.White);
            b.Save(memStream, System.Drawing.Imaging.ImageFormat.Png);
            bmpimg.StreamSource = memStream;
            bmpimg.EndInit(); return bmpimg;
        }

        // Handle keyboard events

        private void KeyDown_Event(object sender, KeyEventArgs e)
        {
            if (e.Key == Key.Escape)
            { App.Current.Shutdown();
            }
        }
```

```
                    // Timer method

                    private void dispatcherTimer_Tick(object sender, EventArgs e)
                    {
                       Random random = new Random();
                        int p = random.Next(1, 4);

                        BitmapImage myBitmapImage = new BitmapImage();
                        myBitmapImage.BeginInit();
                       myBitmapImage.UriSource = new Uri(@"MultipointPictures/" +
p.ToString() + ".jpg", UriKind.RelativeOrAbsolute);
                        myBitmapImage.DecodePixelWidth = 200;
                        myBitmapImage.EndInit();
                        image5.Source = myBitmapImage;


                        switch (p)
                        {
                            case 1:
                                button1.Content = "Cat";
                                button2.Content = "Caat";
                                button3.Content = "Cut";
                                button4.Content = "Coot";

                            break;

                            case 2:
                            button1.Content = "Eagle";
                            button2.Content = "Eogle";
                            button3.Content = "Ewgle";
                            button4.Content = "Eugle";

                            break;

                            case 3:
                            button1.Content = "Leopard";
                            button2.Content = "Lupard";
                            button3.Content = "Liapard";
                            button4.Content = "Luapard";

                            break;
                            case 4:
                            button1.Content = "Lion";
                            button2.Content = "Lian";
                            button3.Content = "Luan";
                            button4.Content = "Lyan";

                            break;
                        }


for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
  {
    DeviceInfo mpDeviceInfo = (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
    MultiPointMouseDevice mpMouseDevice =
 (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;
                                mpMouseDevice.DisableLeftMouseButton = false;
                        }

                    }


                    private void button1_Click(object sender, RoutedEventArgs e)
                    {


 for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
        {
```

```
                                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                 MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                                if (multipointargs != null)
                                {
                                    // perform required action

                                    int current = multipointargs.DeviceInfo.ID;

                                    switch (current)
                                    {

                                        case 0:
          DeviceInfo mouseObject = (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
        MultiPointMouseDevice mpMouseDevice =
        (MultiPointMouseDevice)mouseObject.DeviceVisual;
           Bitmap cursorBitmap = GetCursorImage(0);
            mpMouseDevice.CursorImage = ConvertBitmapToBitmapImage(cursorBitmap);

                                            break;

                                        case 1:
                                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                                            Bitmap cursorBitmap1 = GetCursorImage(1);
                                             mpMouseDevice1.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap1);

                                            break;
                                    }
                                }
                            }

                            if (string.Equals((string)button1.Content,"Cat") ||
string.Equals((string)button1.Content, "Eagle") || string.Equals((string)button1.Content,
"Leopard")|| string.Equals((string)button1.Content, "Lion"))
                            {


                    MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                                if (multipointargs != null)
                                {
                                    // perform required action

                                    int current = multipointargs.DeviceInfo.ID;

                                    switch (current)
                                    {

                                        case 0:
         TranslateTransform translateTransform1 = new TranslateTransform(50, 0);
          double a = Canvas.GetLeft(this.image1);
          translateTransform1.X = a + 100;
                      image1.RenderTransform = translateTransform1;
                      Canvas.SetLeft(this.image1, translateTransform1.X);


                                      if (translateTransform1.X == 106)
                                          {
                                           textBlock1.Text=" Blue Cursor 1 Out of 4";
                                          }

                                           else if (translateTransform1.X == 206)
                                           {
```

```
                                    textBlock1.Text = "Blue Cursor  2 Out of 4";
                                    }
                                    else if (translateTransform1.X == 306)
                                    {
                                   textBlock1.Text = "Blue Cursor  3 Out of 4";
                                    }
                                    else if (translateTransform1.X == 406)
                                    {
                                     textBlock1.Text= "Blue Cursor  4 Out of 4";
                                    }
        for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
              {
                                    DeviceInfo mpDeviceInfo =
             (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                    MultiPointMouseDevice mpMouseDevice =
             (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                        switch (i)
                                        {
                                            case 0:


                                    mpMouseDevice.DisableLeftMouseButton = true;
                                              break;

                                        }
                                    }


                                    if (Canvas.GetLeft(this.image1) == 406)
                                    {
                                        textBlock5.Text="Winner is Blue Cursor";
                                        dispatcherTimer.Stop();

                                    }

                                    break;
                                case 1:
TranslateTransform translateTransform2 = new TranslateTransform(50, 0);
                               double b = Canvas.GetLeft(this.image2);
                               translateTransform2.X = b + 100;
                                image2.RenderTransform = translateTransform2;
                           Canvas.SetLeft(this.image2, translateTransform2.X);
                                    if (translateTransform2.X == 106)
                                    {
                                 textBlock2.Text= "Green Cursor  1 Out of 4";
                                    }

                                    else if (translateTransform2.X == 206)
                                    {
                                  textBlock2.Text = "Green Cursor  2 Out of 4";
                                    }
                                    else if (translateTransform2.X == 306)
                                    {
                                   textBlock2.Text = "Green Cursor  3 Out of 4";
                                    }
                                    else if (translateTransform2.X == 406)
                                    {
                                  textBlock2.Text = "Green Cursor  4 Out of 4";
                                    }
   for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
          {
               DeviceInfo mpDeviceInfo =
              (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];

             MultiPointMouseDevice mpMouseDevice =
             (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;
```

```
                                        switch (i)
                                        {
                                            case 1:


                                        mpMouseDevice.DisableLeftMouseButton = true;
                                                    break;

                                            }
                                        }
                                        if (Canvas.GetLeft(this.image2) == 406)
                                        {
                                    textBlock5.Text="Winner is Green Cursor";
                                            dispatcherTimer.Stop();

                                        }
                                        break;

                                    }


                                    }
                                }

                            ;


                            }

                        private void button2_Click(object sender, RoutedEventArgs e)
                        {

            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
                            {
                                // Act on each individual MouseDevice here

              MultiPointButton btn = (MultiPointButton)sender;
              MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                                if (multipointargs != null)
                                {
                                    // perform required action

                                    int current = multipointargs.DeviceInfo.ID;

                                    switch (current)
                                    {

                                        case 0:
                                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                                            Bitmap cursorBitmap = GetCursorImage(0);
                                            mpMouseDevice.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap);

                                            break;

                                        case 1:
                                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                                            Bitmap cursorBitmap1 = GetCursorImage(1);
                                            mpMouseDevice1.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap1);

                                            break;
```

```
                                    }
                                }
                            }


    if (string.Equals((string)button2.Content, "Cat") ||
string.Equals((string)button2.Content, "Eagle") || string.Equals((string)button2.Content,
"Leopard") || string.Equals((string)button2.Content, "Lion"))
                            {


                                MultiPointButton btn = (MultiPointButton)sender;

                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                                if (multipointargs != null)
                                {
                                    // perform required action

                                    int current = multipointargs.DeviceInfo.ID;

                                    switch (current)
                                    {

                                        case 0:
                    TranslateTransform translateTransform1 = new TranslateTransform(50, 0);
                                            double a = Canvas.GetLeft(this.image1);
                                            translateTransform1.X = a + 100;
                                image1.RenderTransform = translateTransform1;
                               Canvas.SetLeft(this.image1, translateTransform1.X);
                                            if (translateTransform1.X == 106)
                                            {
                                            textBlock1.Text = " Blue Cursor 1 Out of 4";
                                            }

                                            else if (translateTransform1.X == 206)
                                            {
                                           textBlock1.Text = "Blue Cursor  2 Out of 4";
                                            }
                                            else if (translateTransform1.X == 306)
                                            {
                                            textBlock1.Text = "Blue Cursor  3 Out of 4";
                                            }
                                            else if (translateTransform1.X == 406)
                                            {
                                           textBlock1.Text = "Blue Cursor  4 Out of 4";
                                            }
                    for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
                            {
                                    DeviceInfo mpDeviceInfo =
                                (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                    MultiPointMouseDevice mpMouseDevice =
                                (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                            switch (i)
                                            {
                                                case 0:


                                            mpMouseDevice.DisableLeftMouseButton = true;
                                                        break;

                                            }
                                        }
                                        if (Canvas.GetLeft(this.image1) == 406)
```

```
                                        {
                                            textBlock5.Text="Winner is Blue Cursor";
                                            dispatcherTimer.Stop();

                                        }
                                        break;
                                    case 1:
                    TranslateTransform translateTransform2 = new TranslateTransform(50, 0);
                                        double b = Canvas.GetLeft(this.image2);
                                        translateTransform2.X = b + 100;
                                image2.RenderTransform = translateTransform2;
                     Canvas.SetLeft(this.image2, translateTransform2.X);

                                        if (translateTransform2.X == 106)
                                        {
                                    textBlock2.Text = "Green Cursor  1 Out of 4";
                                        }

                                        else if (translateTransform2.X == 206)
                                        {
                                     textBlock2.Text = "Green Cursor  2 Out of 4";
                                        }
                                        else if (translateTransform2.X == 306)
                                        {
                                      textBlock2.Text = "Green Cursor  3 Out of 4";
                                        }
                                        else if (translateTransform2.X == 406)
                                        {
                                    textBlock2.Text = "Green Cursor  4 Out of 4";
                                        }
   for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
          {
                 DeviceInfo mpDeviceInfo =
                (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                   MultiPointMouseDevice mpMouseDevice =
                (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                            switch (i)
                                            {
                                                case 1:

mpMouseDevice.DisableLeftMouseButton = true;
                                                        break;

                                            }
                                        }


                                        if (Canvas.GetLeft(this.image2) == 406)
                                        {
                                            textBlock5.Text="Winner is Green
Cursor";
                                            dispatcherTimer.Stop();

                                        }
                                        break;

                                }


                                }
                            }

                        }

        private void button3_Click(object sender, RoutedEventArgs e)
        {
```

```
            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            Bitmap cursorBitmap = GetCursorImage(0);
                            mpMouseDevice.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            Bitmap cursorBitmap1 = GetCursorImage(1);
                            mpMouseDevice1.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap1);

                            break;

                    }
                }
            }


            if (string.Equals((string)button3.Content, "Cat") ||
string.Equals((string)button3.Content, "Eagle") || string.Equals((string)button3.Content,
"Leopard") || string.Equals((string)button3.Content, "Lion"))
            {


                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            TranslateTransform translateTransform1 = new
TranslateTransform(50, 0);
                            double a = Canvas.GetLeft(this.image1);
                            translateTransform1.X = a + 100;
                            image1.RenderTransform = translateTransform1;
                            Canvas.SetLeft(this.image1, translateTransform1.X);
                            if (translateTransform1.X == 106)
```

```
                                if (translateTransform1.X == 106)
                                {
                                    textBlock1.Text = " Blue Cursor 1 Out of 4";
                                }

                                else if (translateTransform1.X == 206)
                                {
                                    textBlock1.Text = "Blue Cursor  2 Out of 4";
                                }
                                else if (translateTransform1.X == 306)
                                {
                                    textBlock1.Text = "Blue Cursor  3 Out of 4";
                                }
                                else if (translateTransform1.X == 406)
                                {
                                    textBlock1.Text = "Blue Cursor  4 Out of 4";
                                }

    for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                    DeviceInfo mpDeviceInfo =
                    (DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                                MultiPointMouseDevice mpMouseDevice
                    = (MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                            switch (i)
                            {
                                case 0:

                                    mpMouseDevice.DisableLeftMouseButton = true;
                                    break;

                            }
                        }

                        if (Canvas.GetLeft(this.image1) == 406)
                        {
                            textBlock5.Text="Winner is Blue Cursor";
                            dispatcherTimer.Stop();

                        }
                        break;
                     case 1:
                        TranslateTransform translateTransform2 = new
TranslateTransform(50, 0);
                        double b = Canvas.GetLeft(this.image2);
                        translateTransform2.X = b + 100;
                        image2.RenderTransform = translateTransform2;
                        Canvas.SetLeft(this.image2, translateTransform2.X);
                        if (translateTransform2.X == 106)
                        {
                            textBlock2.Text = "Green Cursor  1 Out of 4";
                        }

                        else if (translateTransform2.X == 206)
                        {
                            textBlock2.Text = "Green Cursor  2 Out of 4";
                        }
                        else if (translateTransform2.X == 306)
                        {
                            textBlock2.Text = "Green Cursor  3 Out of 4";
                        }
                        else if (translateTransform2.X == 406)
                        {
                            textBlock2.Text = "Green Cursor  4 Out of 4";
                        }

                        for (int i = 0; i <
MultiPointSDK.Instance.MouseDeviceList.Count; i++)
                        {
```

```
                                DeviceInfo mpDeviceInfo =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                switch (i)
                                {
                                    case 1:

                                        mpMouseDevice.DisableLeftMouseButton = true;
                                        break;

                                }
                            }

                            if (Canvas.GetLeft(this.image2) == 406)
                            {
                                textBlock5.Text="Winner is Green Cursor";
                                dispatcherTimer.Stop();

                            }
                            break;

                    }


                }
            }


        }

        private void button4_Click(object sender, RoutedEventArgs e)
        {
            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            Bitmap cursorBitmap = GetCursorImage(0);
                            mpMouseDevice.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            Bitmap cursorBitmap1 = GetCursorImage(1);
                            mpMouseDevice1.CursorImage =
ConvertBitmapToBitmapImage(cursorBitmap1);
```

```
                        break;


                    }
                }
            }


            if (string.Equals((string)button4.Content, "Cat") ||
string.Equals((string)button4.Content, "Eagle") || string.Equals((string)button4.Content,
"Leopard") || string.Equals((string)button4.Content, "Lion"))
            {


                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                TranslateTransform translateTransform1 = new TranslateTransform(50, 0);
                            double a = Canvas.GetLeft(this.image1);
                            translateTransform1.X = a + 100;
                            image1.RenderTransform = translateTransform1;
                            Canvas.SetLeft(this.image1, translateTransform1.X);
                            if (translateTransform1.X == 106)
                            {
                                textBlock1.Text = " Blue Cursor 1 Out of 4";
                            }

                            else if (translateTransform1.X == 206)
                            {
                                textBlock1.Text = "Blue Cursor  2 Out of 4";
                            }
                            else if (translateTransform1.X == 306)
                            {
                                textBlock1.Text = "Blue Cursor  3 Out of 4";
                            }
                            else if (translateTransform1.X == 406)
                            {
                                textBlock1.Text = "Blue Cursor  4 Out of 4";
                            }

                            for (int i = 0; i <
MultiPointSDK.Instance.MouseDeviceList.Count; i++)
                            {
                                DeviceInfo mpDeviceInfo =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                switch (i)
                                {
                                    case 0:

                                        mpMouseDevice.DisableLeftMouseButton = true;
                                        break;

                                }
                            }
```

```
                            if (Canvas.GetLeft(this.image1) == 406)
                            {
                                textBlock5.Text="Winner is Blue Cursor";

                                dispatcherTimer.Stop();

                            }
                            break;
                        case 1:
        TranslateTransform translateTransform2 = new TranslateTransform(50, 0);
                            double b = Canvas.GetLeft(this.image2);
                            translateTransform2.X = b + 100;
                            image2.RenderTransform = translateTransform2;
                            Canvas.SetLeft(this.image2, translateTransform2.X);
                            if (translateTransform2.X == 106)
                            {
                                textBlock2.Text = "Green Cursor  1 Out of 4";
                            }

                            else if (translateTransform2.X == 206)
                            {
                                textBlock2.Text = "Green Cursor  2 Out of 4";
                            }
                            else if (translateTransform2.X == 306)
                            {
                                textBlock2.Text = "Green Cursor  3 Out of 4";
                            }
                            else if (translateTransform2.X == 406)
                            {
                                textBlock2.Text = "Green Cursor  4 Out of 4";
                            }
for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
                            {
                                DeviceInfo mpDeviceInfo =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[i];
                                MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mpDeviceInfo.DeviceVisual;


                                switch (i)
                                {
                                    case 1:

                                        mpMouseDevice.DisableLeftMouseButton = true;
                                        break;

                                }
                            }



                            if (Canvas.GetLeft(this.image2) == 406)
                            {
                                textBlock5.Text="Winner is Green Cursor";
                                dispatcherTimer.Stop();

                            }
                            break;

                    }


                }
            }

        }
```

```
        private void button1_MultiPointMouseLeaveEvent(object sender, RoutedEventArgs e)
        {

            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            mpMouseDevice.CursorColor = GetCursorColor(0);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            mpMouseDevice1.CursorColor = GetCursorColor(1);

                            break;

                        case 2:
                            DeviceInfo mouseObject2 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice2 =
(MultiPointMouseDevice)mouseObject2.DeviceVisual;
                            mpMouseDevice2.CursorColor = GetCursorColor(2);

                            break;

                        case 3:
                            DeviceInfo mouseObject3 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice3 =
(MultiPointMouseDevice)mouseObject3.DeviceVisual;
                            mpMouseDevice3.CursorColor = GetCursorColor(3);

                            break;
                    }
                }
            }


        }

        private void button3_MultiPointMouseLeaveEvent(object sender, RoutedEventArgs e)
        {

            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
```

```
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            mpMouseDevice.CursorColor = GetCursorColor(0);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            mpMouseDevice1.CursorColor = GetCursorColor(1);

                            break;

                        case 2:
                            DeviceInfo mouseObject2 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice2 =
(MultiPointMouseDevice)mouseObject2.DeviceVisual;
                            mpMouseDevice2.CursorColor = GetCursorColor(2);

                            break;

                        case 3:
                            DeviceInfo mouseObject3 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice3 =
(MultiPointMouseDevice)mouseObject3.DeviceVisual;
                            mpMouseDevice3.CursorColor = GetCursorColor(3);

                            break;
                    }
                }
            }

        }

        private void button2_MultiPointMouseLeaveEvent(object sender, RoutedEventArgs e)
        {
            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
```

```
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            mpMouseDevice.CursorColor = GetCursorColor(0);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            mpMouseDevice1.CursorColor = GetCursorColor(1);

                            break;

                        case 2:
                            DeviceInfo mouseObject2 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice2 =
(MultiPointMouseDevice)mouseObject2.DeviceVisual;
                            mpMouseDevice2.CursorColor = GetCursorColor(2);

                            break;

                        case 3:
                            DeviceInfo mouseObject3 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice3 =
(MultiPointMouseDevice)mouseObject3.DeviceVisual;
                            mpMouseDevice3.CursorColor = GetCursorColor(3);

                            break;
                    }
                }
            }

        }

        private void button4_MultiPointMouseLeaveEvent(object sender, RoutedEventArgs e)
        {
            for (int i = 0; i < MultiPointSDK.Instance.MouseDeviceList.Count; i++)
            {
                // Act on each individual MouseDevice here

                MultiPointButton btn = (MultiPointButton)sender;
                MultiPointMouseEventArgs multipointargs = e as MultiPointMouseEventArgs;

                if (multipointargs != null)
                {
                    // perform required action

                    int current = multipointargs.DeviceInfo.ID;

                    switch (current)
                    {

                        case 0:
                            DeviceInfo mouseObject =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[0];
                            MultiPointMouseDevice mpMouseDevice =
(MultiPointMouseDevice)mouseObject.DeviceVisual;
                            mpMouseDevice.CursorColor = GetCursorColor(0);

                            break;

                        case 1:
                            DeviceInfo mouseObject1 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[1];
```

```
                            MultiPointMouseDevice mpMouseDevice1 =
(MultiPointMouseDevice)mouseObject1.DeviceVisual;
                            mpMouseDevice1.CursorColor = GetCursorColor(1);

                            break;

                        case 2:
                            DeviceInfo mouseObject2 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice2 =
(MultiPointMouseDevice)mouseObject2.DeviceVisual;
                            mpMouseDevice2.CursorColor = GetCursorColor(2);

                            break;

                        case 3:
                            DeviceInfo mouseObject3 =
(DeviceInfo)MultiPointSDK.Instance.MouseDeviceList[3];
                            MultiPointMouseDevice mpMouseDevice3 =
(MultiPointMouseDevice)mouseObject3.DeviceVisual;
                            mpMouseDevice3.CursorColor = GetCursorColor(3);

                            break;
                    }
                }
            }



        }


                    }


            }
```