THREE-DIMENSIONAL ANIMATION USING IN A COMPUTER-BASED TRAINING OF TAI-CHI-CHUAN

SUPAKIJ RATTANASUWAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2002

ISBN 974-04-1899-6

Thesis
Entitled

THREE-DIMENSIONAL ANIMATION USING IN A COMPUTER-BASED TRAINING OF TAI-CHI-CHUAN

....
Mr. Supakij Rattanasuwan
Candidate

....
Lect. Suttinant Nantachit, M.S.
Major-advisor

....
Assoc.Prof. Panya Kaimuk, M.D.,
FIMS, Board of Orthopedic Surgery
Co-advisor

....
Prof. Liangchai Limlomwongse
Ph.D.
Dean
Faculty of Graduate Studies

....
Lect. Thanakorn Uan-on, D.Engr.
Chair
Master of Science Programme in
Technology of Information System
Management
Faculty of Engineering

Thesis
Entitled

THREE-DIMENSIONAL ANIMATION USING IN A COMPUTER-BASED TRAINING OF TAI-CHI-CHUAN

Was submitted to the Faculty of Graduate Studies, Mahidol University
for the Degree of Master of Science
(Technology of Information System Management)
on
May 21, 2002

....
Mr. Supakij Rattanasuwan
Candidate

....
Lect. Suttinant Nantachit, M.S.
Chair

....
Assoc.Prof. Panya Kaimuk, M.D.,
FIMS, Board of Orthopedic Surgery
Member

....
Lect. Nuttavut Seemontara, M.F.A
Member

....
Prof. Liangchai Limlomwongse
Ph.D.
Dean
Faculty of Graduate Studies
Mahidol University

....
Lect. Thanakorn Uan-on, D.Engr.
Dean
Faculty of Engineering
Mahidol University

4037706 EGTI/M : MAJOR : TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT ; M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
KEY WORDS: COMPUTER-BASED TRAINING / CBT / TAI-CHI / 3D ANIMATION / SOFT-SKINNING / DIRECT3D

SUPAKIJ RATTANASUWAN: THREE-DIMENSIONAL ANIMATION USING IN A COMPUTER-BASED TRAINING OF TAI-CHI-CHUAN. THESIS ADVISORS: SUTTINANT NANTACHIT M.S., PANYA KAIMUK M.D., NUTTAVUT SEEMONTRA M.F.A., 77 p. ISBN 974-04-1899-6

The purpose of this study was to apply 3D animation into computer-based training software to represent a media for self-study learners of Tai-Chi-Chuan. The objectives of the study were to reduce the problems arising from lack of detail in proper representation of animation movement in media. A prototype was developed and the application was implemented. Users' opinions towards the application was then evaluated.

An approach for 3D model rendering utilized the soft-skinning technique, which is usually used in game programs at present.

Software tools for this research were: Microsoft Windows XP an operating system, Microsoft Visual C/C++ 6.0 for programming, Microsoft DirectX 8.1 SDK for multimedia program library, Discreet 3D Studio Max 4.2 for creating 3D model.

Evaluation methods were comprised of evaluation from users (perception and usage) and animation playback evaluation (pictures quality and animation smoothness). Evaluation from users results were from the testers whose are comprised of six users from two groups, experienced and inexperienced in Tai-Chi-Chuan.

The findings of this study show for that the concept of using 3D animation to represent motion was satisfactory users because of the ability to change the position of viewpoint as desired. Application resulted in some incorrect animations while running, because the animation data collection method was not good enough. An improvement of data verification, or use of specific hardware such Motion-capture, may reduce incorrect animation data.

4037706 EGTI/M : สาขาวิชา: เทคโนโลยีการจัดการระบบสารสนเทศ; วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

สุภกิจ รัตนสุวรรณ: การใช้อนิเมชันสามมิติในโปรแกรมช่วยการเรียนมวยไทเก็ก (THREE-DIMENSIONAL ANIMATION USING IN A COMPUTER-BASED TRAINING OF TAI-CHI-CHUAN) คณะกรรมการควบคุมวิทยานิพนธ์: สุทธินันท์ นันทจิต M.S., ปัญญา ไข่มุก M.D., นัฐวุฒิ สีมันตร M.F.A. 77 หน้า. ISBN 974-04-1899-6

การศึกษาวิจัยนี้เป็นการใช้ อนิเมชันสามมิติ ในโปรแกรมช่วยการเรียนสำหรับผู้สนใจเรียนมวยไทเก็กด้วยตนเอง วัตถุประสงค์ของการศึกษาวิจัยนี้เพื่อลดปัญหาเรื่องความไม่ชัดเจนในการแสดงการเคลื่อนไหวของสื่อประเภทอื่นที่มีอยู่ในปัจจุบัน เพื่อพัฒนาโปรแกรมต้นแบบให้ผู้ใช้งานได้ทดลองใช้ และประเมินผลจากความเห็นของผู้ใช้งาน

วิธีการแสดงผลโมเดลสามมิติที่ใช้ในการศึกษาวิจัยนี้ เลือกใช้วิธี Soft-Skinning ซึ่งเป็นวิธีที่นิยมใช้ในโปรแกรมประเภทเกมในปัจจุบัน

เครื่องมือที่ใช้ในงานวิจัยคือ Microsoft Windows XP เป็นระบบปฏิบัติการ Microsoft Visual C/C++ 6.0 เป็นเครื่องมือพัฒนาโปรแกรม Microsoft DirectX 8.1 SDK เป็นเครื่องมือในการพัฒนาโปรแกรมส่วนการแสดงผล Discreet 3D Studio Max 4.2 ใช้เป็นเครื่องมือในการสร้างวัตถุสามมิติ

วิธีการประเมินผลประกอบด้วย 2 ส่วน คือ ผลการประเมินจากผู้ใช้งาน (ในส่วนการแสดงผล และการใช้งาน) ผลการประเมินจากการแสดงอนิเมชัน (ในส่วนคุณภาพของภาพ และความต่อเนื่องของอนิเมชัน) การประเมินผลจากผู้ใช้งาน ได้จากผู้ทดสอบทั้งหมด 6 คน ซึ่งประกอบด้วยผู้มีประสบการณ์ และไม่มีประสบการณ์ในการรำมวยไทเก็ก .

ผลการศึกษาแสดงให้เห็นว่า แนวความคิดในการนำ อนิเมชันสามมิติมาแสดงการเคลื่อนไหว ทำให้ผู้ใช้งานรู้สึกพอใจ เพราะความสามารถในการเปลี่ยนมุมมองได้ตามต้องการ แต่โปรแกรมยังแสดงข้อผิดพลาดในเรื่องการเคลื่อนไหวอยู่บ้าง สาเหตุหนึ่งเป็นเพราะวิธีการเก็บข้อมูลที่ใช้ไม่มีคุณภาพเพียงพอ การเพิ่มการตรวจสอบความถูกต้องของข้อมูล หรือการใช้เครื่องมือเฉพาะทางเช่น Motion-capture อาจช่วยลดข้อผิดพลาดของข้อมูลลงได้

// ACKNOWLEDGEMENTS

I am very grateful and would like to express my special and sincere thanks to Lect.Suttinant Nantachit, chairman of the examination committee and thesis advisor, for his guidance, invaluable advice and positive criticisms. Without his patient and support, this study could not be successful.

I am indebted to Assoc. Prof. Panya Kaimuk for his helpful recommendations, supervision and encouragement through my thesis period. I also indebted to Lect.Nuttavut Seemontara for his helpful information and suggestions related in this study.

I wish to express my thanks to all of my teachers, classmates, and my workplace – OHOGame Co., Ltd.

Finally, I would like to express my deep gracefulness to my beloved parents, and my brother, for their great love, care and moral support. I am indebted to them so much.

Supakij Rattanasuwan

CONTENTS

CONTENTS (cont.)

CONTENTS (cont.)

CONTENTS (cont.)

LIST OF FIGURES

Pages

FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Introduction

An integration of the computer into instruction and learning currently increase in Thailand. Progress has been made from course of study for interesting learners, and becomes to a part of the course of study today. Many schools have purchase and are purchasing computers for their students. Because of inexpensive computers and expanded capabilities in present, computers are increasing in number in both school and home.

Although the mainly trend of computer using in a home is for entertainment. However, in part of computer software, there still are many computer-based training (CBT) software in market. The largest group of CBT is for children, it teaches about characters, shapes, numbers and basic logics. Another groups are concerned about skill improvement, for examples, musical, typing, golf. Although the benefits of learning from a CBT are completed information and no need any instructor while learning. But when the learners found some problems or have any questions, it could not solve all of the questions thus, the information preparation for solving the problems is still the subject for CBT software development in future.

The purpose of this study was to ascertain the usage CBT software for helping the Tai-Chi-Chuan learners. In an addition to the basic information about pictures and descriptions, we will use 3D model animation to simulate Tai-Chi-Chuan postures in any steps; this helps the learner to observe positions and postures clearer, and decreases the problems while learning.

Specific Background

About instruction and learning, computer animation has brought into mechanisms study, and learning about motions. Because it can simulate object's motion thus, it will help users to observe object movement in any motion, have more detail and in any points of view.

An advance of computer technology in present cause the home computers have enough capability to run 3D computer animation in a good quality. Thus, this is possible to bring computer animation to be a tool for helping in motion learning for self study learner. This study use Tai-Chi-Chuan as a case study, it has complicate motions, and possible to misunderstanding while self-study.

There are several techniques used in 3D model animation program, depending on what kinds of program mainly do, and quality or speed it focused on. The techniques that usually used are *segmented-characters* and *single-mesh characters*. Both techniques are useful because they provide adequate quality and speed. Fortunately, an advance of the hardware's capability in present allows us to use an advance animation technique that use on 3D animation movies such as *soft-skinning animation* and *real-time IK animations* (1).

Character animation techniques in this study may have some aspects to consider. If we only focus on an increasing quality of models, overall system performances may be abated, but if concentrate in system speed and performance, the graphic quality may be suffered and the output may not good enough to helps the learners. We should consider the appropriate point that the graphic quality goes along with program speed, or rendering frame rate. The total number of polygons render to the screen must be within the limited number that can be rendered at 15 frames per second at the minimum rate on a PC.

Statement of the Problems

According to the method of self-study Tai-Chi-Chuan in present, there are several problems found while studying and should be concerned. The following are two categories problems found in a CBT of Tai-Chi-Chuan and will be solved in this study.

1. Media formats using in present do not clear enough.

The media used in present such as books and videotapes may not provide clear information enough for self-study learners. Both of media types have it own advantages and disadvantages. If we focus on the detail of movement in each action, both of media types are still indistinct, and may cause misunderstanding on learners while studying.

Advantages and disadvantages of both media types can be described as follows.

1.1 Books

Advantages

- Easy for taking away from home.
- Can be moved to the interesting page immediately, and can be repeatedly read as much as the users wish.
- Cheapness and no need another equipments while using.

Disadvantages

- The pictures in the books are the fixed position pictures. It could not provide enough information for help the learners to understand the movement of each posture.
- Could not show steps and speed on each posture.

1.2 Videotapes

Advantages

- Play of both pictures and sounds synchronously.
- Provides information on any important movements by an instructor.
- Can represent the steps and speed on each movement distinctly.

Disadvantages

- Require a videotape player to playback.
- Difficult to move to the interesting point, or replay only the interesting movement.

- Have a strict viewpoint. Users cannot change it to another point.

2. What is the appropriate method to present Tai-Chi-Chuan in CBT software?

The method that used on a CBT of Tai-Chi-Chuan should reduce the problem from the strict camera's viewpoint. Represent in a 3D animation program can solve this problem, but what is the appropriate technique for this kind of program. There are several techniques using in current 3D real-time rendering programs, each technique has it own advantages and disadvantages depends on the purpose of using.

Objectives

The objectives of this thesis are study and develop the program for self-study learners of Tai-Chi-Chuan. According to the aims of reduce some deficient functions of another media types. The objectives are specifics as follow.

1. Introduce a CBT program of Tai-Chi-Chuan that helps the self-study users.

1.1 Provide complete information for self-study.

1.2 Reduce the unclear information as much as possible.

1.3 Compensating the disadvantage functions on another medias.

2. Study and develop the program function for load and animate 3D models.

2.1 The program function could support the 3D models from 3D Studio Max program.

2.2 The technique should be flexible enough in case of changing a model or animation data without changing any code in program.

2.3 An approach should control and play animation as desire. For example, the animator might be allowed to specify the start and stop points in animation data set to play.

2.4 An approach should play animation fast enough for using in 3D real-time rendering programs

According to the objectives described above, the reason to choose 3D Studio Max as a standard program to create character models and animation data is there is no Motion Capture hardware found in Thailand on the time researching this thesis, and hire a motion capture equipment, or buy animation data from animation data provider is overprice. Although, the animation data used in this thesis is created from 3D Studio Max's plug-in named "Character Studio". It provides several sets of animation data from motion capture hardware that good enough for study and testing in this thesis.

Scope of Works

The study mainly consists of two works as follows:

1. Design and developed a CBT program for helping the Tai-Chi-Chuan self-study users.

1.1 Presentation Tai-Chi-Chuan poses in a real-time 3D character animation system.

1.2 Included explanation both text and voice for any postures and movements.

1.3 Included animation of Tai-Chi-Chuan 24 poses set.

2. Report the evaluation from users.

2.1 Collected the result from experienced users of Tai-Chi-Chuan.

2.2 Collected the result from general users.

2.3 Evaluated and reported the result.

Expected Results

In this study, the author will create:

1. A Prototype of a Tai-Chi-Chuan CBT program.
2. A program function for loading and control 3D animation model.
3. Reports about quality and performance comparison of the chosen technique.
4. Reports about the result and recommendations from users.

CHAPTER II

LITERATURE REVIEW

This chapter described a summary of the reviewed literature concerning history of Tai-Chi-Chuan, and 3D model rendering techniques that are proposed for using and developing in a 3D real-time rendering program. The main modules of the proposed system are also discussed.

History of Tai-Chi-Chuan

Tai-Chi-Chuan is a combination of martial arts, sports, and philosophy, in a harmonious mixing. It appropriate for interesting learners in any age, and any sex. Moreover, it will improve healthy, and concentration, for a person whose training in regularly.

Tai-Chi-Chuan originated in Chinese for about 700 years ago by the grand master *Chang San Feng*. It was inherited and adapted in every generations, and are separated into three families – Chen, Yang, Wu (5).

In present, Tai-Chi-Chuan is appreciated in world wide, and has been put into the Olympic competition. The competition style of Tai-Chi-Chuan is difference from the exercise style that used in this study in the detail. Even in the exercise style of Tai-Chi-Chuan, it still be separated into a number of sets, for example:

- The shorten set of 24 postures
- The standard set of 42 postures
- The 81 postures of Yang Family
- The 101 postures set

In this study we focuses on the shorten set of 24 postures, because this set is appropriate for the beginners, and it take time about 5 minutes to finished them.

Computer Animation Techniques in 3D Real-Time Rendering

Computer animation can be categorized by the type and nature of the objects that are going to be animated and the programming technique used to achieve the animation as follows (2):

- Rigid body animation
- Articulated structure animation
- Dynamic simulation
- Particle animation
- Behavioral animation
- Deformable object animation

There are not restrict to use an individual technique as described above, some animation may be produced using mixture of the above techniques. In this thesis we will focus on only rigid body animation and articulated structure animation because of these two techniques related to the techniques we will approach. Other techniques are all explained in detail in reference text listing in an appendix.

1. Rigid Body Animation

Rigid body animation is the simplest form of computer animation to implement and is the most widely used. We can produce animated sequences by collects each animation frame from rendered scene of an object in different positions. There are two approaches to specify and control the movement of objects in each scene – keyframing or interpolation systems and explicit scripting systems.

- Keyframing or interpolation systems are based on a well-known production technique in film animation. The animators specify a sequence by drawing keyframes at certain intervals and then the computer will interpolate in-between scenes automatically.
- Explicit scripting systems are an idea of using cubic parametric as a script form. A curve can be used as a path over which the reference point or origin of the object to move.

2. Articulated Structure Animation

Articulated structure animation also called linked structures and hierarchical motion. It is a set of rigid objects, or links, connected to each other by joints, which enable the various parts of the structure to move with respect to each other. There are two approaches to solve motion problems in this technique – forward kinematics and inverse kinematics.

- Forward kinematics is a low-level approach where the animator has to specify explicitly all the motions of every part of the articulated structure. The approach manipulates the model from the top of hierarchy on down. For example, if we move the pelvis, the whole body moves; if we rotate the elbow, the wrist moves.
- Inverse kinematics is an approach works out a precise script for all the parts of the structure so that the whole body will perform the desired action. Inverse kinematics is exact opposite of forward kinematics in that moving the children moves the parents. For example, if we raise the model's hand up, the rest of the arm automatically follows. Unfortunately, the software must determine exactly how to bend the rest parts to make everything looks natural thus. We need to tell the software the exact way the joints are constrained, or limited in their motions. An example of rotational limits of human joints shown in Table 2.1.

Table 2.1 Rotational Limits of Human Joints
(Assumed the Z-axis is oriented along the bon of the joint)

Segment	**Joint**	**Type**	**X**	**Y**	**Z**
Foot	Ankle	Rotational	65°	30°	0°
Shin	Knee	Hinge	135°	0°	0°
Thigh	Hip	Ball/Socket	120°	20°	10°
Spine	Hip/Spine	Rotational	15°	10°	0°
Shoulder	Spine	Rotational	20°	20°	0°
Bicep	Shoulder	Ball/Socket	180°	105°	10°
Forearm	Elbow	Hinge	150°	0°	0°
Hand	Wrist	Ball/Socket	180°	30°	120°

An interesting related word with the articulated structure animation is *kinematics animation* (3). Kinematics animation is concerned only with the specification of joint angles and angular velocities over time. It does not deal with the forces and torques acting on or within a creature or their effect on the creature's motion. Motion capture is a special case of the kinematics approach in which the joint angles and/or velocity data are measured from a real motion and then re-used on an animated model. The most common way of capturing a motion at present is to attach a series of markers to various points on the subject's body and to use multiple video cameras or other sensory devices to record the motion of the markers. The subject's motions are mapped directly onto the animated model, thereby ensuring that the animated motion will be realistic. The ability to modify, blend and transition between pre-recorded motions is important to provide the animator with sufficient control over the final motion. However, results based on modifications of captured motions are not guaranteed to remain realistic.

Models Controlling Techniques in 3D Real-Time Rendering

There are several popular model-controlling techniques using in 3D animation real-time rendering in present. We will focus on four popular techniques – segmented models, single-mesh models and soft-skinning models.

1. Segmented Models

A segmented model also called a hierarchy articulated objects, it is a three-dimensional model with separate components for all the articulated body parts, stored in a hierarchy and joined to each other at pivot points. The flexibility and adaptability of this method are its main benefits:

- Animation data can be generated while running as an Inverse Kinematics then applied to the model.
- Memory usage is also small, because the vertex and transformation information for each object contained in the model needs to be stored only once.

However this method also has a number of drawbacks:

- As the objects in the hierarchy are all separate, it is inevitable that gaps between these objects will appear when the model is animated. Although it is possible to hide these gaps by overlapping the objects that make up the model, this again will result in visible seams.

We can directly manipulate each part of the model by using the transformations. For an example, we can move a model's arm up and down by rotating the shoulder joint the desire number of degrees. All the attached body parts (the lower arm, the hand, and so on) will move with the upper arm as expected because of the hierarchical nature of the transformations.

2. Single-Mesh Models

A single-mesh model is a very popular in 3D real-time rendering software. The concept of this technique derives from the film animation. To using this technique, we will store a set of three-dimensional models that represent each movement made in a complete animation sequence then play each step to produce an animation. For an example, to replicate the motion of walking, we will store a set of models that represent the body starting to take a step of walking animation, models in the various positions that the body assumes as it's progressing through the step, and finally models representing the body after the step is complete.

Between each animation frame, interpolating the corresponding vertex position between the different objects and blending the two objects realize the animation. The result is a smooth animation with no gaps or visible seams in the model's geometry. Also, as long as a not too complex interpolation method like linear interpolation is used, the amount of calculations required for the animation of the mesh is very small.

The downside of this technique is that key-frames have to be very frequent to effectively control the animation, which means that the amount of data that has to be stored is very large. This has an effect on memory usage as all object copies that are needed in an animation sequence will have to be kept in memory for at least as long as it takes the animation sequence to finish. Models that are animated with this method are also far less flexible than models which are based on a hierarchic model representation, as the whole of the model animation has to be pre-defined off-line and can not generate an additional animation data on runtime.

3. Soft-Skinning Models

Soft-Skinning models or skeletal animation was developed to simplify the animation process for dealing with articulated objects (models like bipeds, for example) and to provide more realism through improving the looks of animated objects by making them more life-like. It is an improvement on both of the previously mentioned techniques:

It uses an endoskeleton - a hierarchic structure of joints - which drives a skin - a vertex mesh representing the shape of the object. It is this splitting of mesh data and hierarchic position information into two separate data-structures, which makes skeletal animation and mesh skinning superior to the previously mentioned techniques.

A bone is simply a transformation matrix, determining the position of the bone in relation to its parent bone, and all the bones of the articulated object together form the skeleton. Explicitly only the skeleton is animated, using an algorithm similar to that used for animating a hierarchic articulated object, which in turn implicitly animates the skin. Memory usage for skeletal animation is small, as all skin vertices have to be stored only once. It requires a significantly lower amount of information to be stored than the single-mesh models (or blending mesh models) technique discussed above.

4. Real-Time Inverse Kinematics Models

Inverse Kinematics is a technology that originated in robotics. It has rarely been used for real-time animation because of the number of calculations which have to be carried out for IK algorithms is too great to be use in a real-time animation software, such as a computer game. The main reason for this is that the IK algorithms direct copied from robotics IK algorithms. There are only few industrial robots have more than six joints, but a humanoid model in computer animation can easily have one hundred or more joints, which can slow down these IK algorithms considerably.

Related Technology

Microsoft Direct3D API

Direct3D is a part of DirectX that introduced by Microsoft. The DirectX application programming interface (API) was developed to provide a set of interfaces that provide extremely efficient control of multimedia hardware on a computer running Microsoft Windows. DirectX lets programmers work with commands and data structures that are very close to those that the hardware can natively process, without being so low level that code has to be written differently for each device. By writing device-independent code, programmers can create software that will always perform at its best – even as users enhance their systems with new and improve 3D graphics accelerators, sound, cards, input devices, and so on.

DirectX was designed to give developers an environment similar in performance to MS-DOS, which was historically much faster than Windows-based code because of the overhead imposed by earlier Windows multimedia APIs. By supporting all hardware features as they become available, however, DirectX code runs faster than would be possible in most MS-DOS applications. The DirectX API is built on a *hardware abstraction layer* (HAL) that hides the device-specific dependencies of the hardware (4). Because it is designed for future extensibility, DirectX defines a number of hardware acceleration support features that are not available on much of the hardware built today. Such features are emulated through the *hardware emulation layer* (HEL) or ignored if the HEL does not support them either. When a device that accelerates a DirectX feature is introduced, the developers can replace their old device with the new one for instantly use the acceleration features the new hardware supplies.

Direct3D API can be used to write programs that use 3D graphics and take advantage of hardware acceleration of 3D operations. Almost all graphics cards now being sold support 3D acceleration and most 3D games now available for Windows were implemented using Direct3D.

CHAPTER III

MATERIALS AND METHODS

This chapter separated into two main parts –Materials and Methods. The Materials portion describes about the tools for developing the software. The Methods portion describes the methods used in this study, which followed the classical System Development Life Cycle (SDLC) (6,7).

MATERIALS

This portion enumerates the tools for system development. It can be separated into three subparts – Data Collection Tools, Design Tools and Development Tools. The detail are described as follows:

1. Data Collection Tools

The data collection tools described the tools used in the data gathering process.

1.1 Tai-Chi-Chuan video CD

The data source contained the Tai-Chi-Chuan postures, which be copied to the 3D animation data in a Biped movement format. The video source is licensed of Photo House Camera and Videos Co., Ltd (8). This media was developed for the self-study Tai-Chi-Chuan learners in video format.

1.2 MS Movie Maker 1.1

This software was developed and licensed by Microsoft, and already included with MS Windows XP Professional Edition. The software was used for playback the Tai-Chi-Chuan video CD.

1.3 Discreet 3D Studio Max 4.2

This software was developed and licensed by Autodesk. The software was used for creating an animation data in the 3D animation format.

1.4 Character Studio 3.1

This software was developed and license by Autodesk, it used as a plug-in program for 3D Studio Max. This plug-in was used for creating and control the Biped animation movement data.

2. Design Tools

The tools used while designing the screens, interfaces and textures.

2.1 Adobe PhotoShop 6.0

This software was developed and licensed by Adobe. It was used for designed and created textures used in this study.

2.2 Discreet 3D Studio Max 4.2

This software was developed and licensed by Autodesk. It was used for create any object used in this study which based on 3D object model.

3. Development Tools

The Development Tools can be separated into two parts – Software Engineering Tools and Hardware Tools. The detail described as follows:

3.1 Software Engineering Tools

The software used for developing the application.

3.1.1 MS Windows XP Professional Edition

3.1.2 MS Visual C/C++ 6.0

3.1.3 MS DirectX 8.1 SDK

3.2 Hardware Tools

The hardware used for developing the application. The hardware used in this study meets the PC2001 desktop PC standard, the detail are as follows:

3.2.1 Processor Intel Pentium 4 1.6GHz

3.2.2 Main memory 256 MB

3.2.3 Hard disk capacity 20 GB

3.2.4 CD-ROM 40X speed

3.2.5 Display adapter - GeForce2 MX 400 32MB memory, supported DirectX 7.0

3.2.6 Digital Camera for created textures

METHODS

This portion follows the classical Software Development Life Cycle (SDLC) steps. The detail described as follows:

1. Preliminary Investigation

The subject is concerned in this step are the Feasibility Study. The feasibility study will determine that the system is feasible. The aspects in this portion are as follows:

1.1 Technical Feasibility

Even the project is completed but it needed the hi-speed computer or advance technology hardware to run it, it may out of reach for general users. This step will ensure that the project will be supported by the existing technologies. The technical issues during the feasibility stage include:

1.1.1 Review the software proposed in this study, and specifies the graphic functions used, for examples Alpha Blending, Matrix Blending, etc.

1.1.2 Review the chosen graphic functions, and find out the approached that supported by current hardware abilities.

1.1.3 Ensure that the chosen functions would be supported by either hardware or software display drivers.

1.1.4 Specify the standard hardware capabilities used in this study.

Since the proposed software present in a real-time 3D animation format, the software must deal with the display adapter hardware frequently. The examine of another study in this criteria would help for specifying which functions should be implemented, or removed before the project has been start.

2. Determination of System Requirements

The detailed understanding of all facets of the existing system is the heart of system analysis. This step will answer the key questions as follows:

2.1 Basic Requirements

Structure the investigation by seeking answers the major questions about the basic process.

2.1.1 Determine the basic processes on the general CBT software.

2.1.2 Find out how the media used for the Tai-Chi-Chuan self-study learners.

2.1.3 Discussed with the users for identified what the information required, and how to present the information for the Tai-Chi-Chuan self-study learners.

2.1.4 Specify an appropriate function should be implemented to support the users.

2.2 The Existing Problems

Identified the existing problems found on the current system used. The result will help the developers to find out the clear point to present the new system. The key activities are:

2.2.1 Determine the existing problems found on current media used.

2.2.2 Discussed with users, how the serious on the existing problems on the current media used.

2.2.3 Specify the problems found, and collected the requirements from users.

3. Design of the System

The design of the system produces the details that state how a system will meet the requirements identified during system analysis. The system design also describes the data to be input, calculated, or stored. In addition, the system design also providing programmers with complete and clearly outlined software specifications.

3.1 Specifying the Software Requirements

In the Determination of System Requirement portion, discussed methods for assemble data that describe how the system operates. This section considers how to assemble those details to identify the requirements a new application must meet. The important subjects to focus in this section are:

3.1.1 Review the Requirement Investigation Objectives

This section will provide important information assistant to analyst for understanding ongoing processes, including why and how they are performed. The subjects would be reviews are:

3.1.1.1 Users requirements from the last process.

3.1.1.2 Important information that the users needed.

3.1.1.3 Scope of the media presentation that should be cover.

3.1.2 Identification of Requirements

This section will summarize the questions asked during an analysis of the system and identify the requirements for the new system. The main activity to done in this step is specify and identify the users requirements on the Tai-Chi-Chuan software.

3.2 Objectives in Design the Software

Requirements identified from the last section are translated into design specifications. This section describes the objectives of system design and explores the nature of the design as well as the explicit results expected. The detail describes as follows:

3.2.1 Specify the Logical Design Elements

3.2.2 Ensure that the Application Features Meet the Requirements

3.2.3 Provide Detailed Software Development Specifications

3.3 Elements of the Design

The components of information described during requirement analysis are the focal point in systems design. The elements must be addressed are as follows:

3.3.1 Design of Input

3.3.2 Design of Output

3.3.3 Design of Control

3.3.4 Design of Program Specifications

4. Development of the Software

Information from the system design guides the programmer to develop the application with the right functions and objectives. The steps in this portion are as follows:

4.1 Data Gathering

This step considers on the capturing data from data source into a pre-designed data format that specified in the Design of Input in last portion. The important processes in this step are:

4.1.1 Capturing Data

Capture data from the data source specified in the Materials section (Tai-Chi-Chuan video CD). The detail in capturing data process will be described later.

4.1.2 Verifying Data

This step ensures the animation data captured this way is correctly enough for used in this study.

4.2 Developing the Software

The programmers develop the software follow by the program specifications document from the Design of the Software portion. The main process in this step is:

4.2.1 Implement an Approach

The approach for solving the problems identified in the Program Specification document is implement in this step. There are several testing processes on the implemented approaches here as well, but it is only the testing for find out improper functions in the implemented approaches.

5. Software Testing

The testing process will ensure that the software will run well and will not fail while the users using. This step also ensure that the application meet the software specification document and requirements.

The testing strategies used in this study can be identified as follow:

5.1 Code Testing

The code testing strategy examines the logic of the program. To follow these testing methods, the test cases would be created, and the result in executing in every instruction in the program or module would be reported (9). This examine may be similar to the functional testing while the software is developed, but may be difference in detail and path of testing. The activities in this step are:

5.1.1 Created the simple animation data and 3D model for testing.

5.1.2 Tested with the created data for finding the error in functions and controlling methods.

5.2 Specification Testing

To perform this testing strategy, the specifications should be examined, and stating what the program should do and how it should perform under various conditions. This strategy treats the program as if it were a black box (9). The assumption is that, if the program meets the specifications, it will not fail. In this step, the activities are as follows:

5.2.1 Testing on the program flows level, finding for an error.

5.2.2 Ensure that the software functions follow the requirements specification report.

6. Implementation and Evaluation

On the evaluations portion in this study, we could break into two subjects, Evaluation from Users, and Animation Playback Evaluation. The Evaluation from Users is concerned about the methods for evaluation while using the program. The Animation Playback Evaluation is concerned about how to measure the quality of an approach used in this study. The detail described as follows:

6.1 Evaluation From Users

We let the users to use and evaluate the program follow by the criteria as follow.

6.1.1 Completed information for self-study – evaluate about the completion and correction of the data using in program, it should be enough for the learners while self-study. (The data used in this study is a 24 postures set of Tai-Chi-Chuan)

6.1.2 Clarification and friendly user interface – an inexpert computer user can easily use the program. The program should describe each pose of Tai-Chi-Chuan in detail.

6.1.3 Comparison with another medias – The users evaluate and compare with another medias, both advantages and disadvantages.

6.2 Animation Playback Evaluation

We can measure the qualities of animation playback follow by these criteria.

6.2.1 Animation smoothness – contiguous and smooth movement of animation will affect to the users' perception, it depending on two matters as follow.

- FPS – the number of frames can be rendered in 1 second. Lower FPS will cause the display to run roughly. The lowest FPS should be 30 frames per second.
- Animation detail – if the detail of movement is low, the display will run roughly even if the FPS over than 30 frames per second.

6.2.2 Display qualities – quality of the model is directly affect to the user's perception, it make the model looks realistic and clearly on each pose. A low quality model may cause the users unable to know what Tai-Chi-Chuan posture is the model doing. The qualities of the model depending on two matters.

- Detail of the model – the higher number of polygons using in a model will provide more detail, this included the number of bones and vertices linked as well.
- Detail of the texture – the texture used in a model will affect to the model realistic, the quality depends on the detail, size of texture, and the filtering methods. The hardware filtering capability is also help in this matter.

We could notice that the animation smoothness and the model qualities are related each other. The higher qualities model will require more resource while rendering, and cause the frames rate automatically dropdown. We can constrain this balancing by use the hardware capabilities help for rendering methods.

CHAPTER IV

RESULT

Results of the study were the system analysis and design using classical System Development Life Cycle (SDLC) methodology (6,7), the CBT of Tai-Chi-Chuan application, and evaluation report of the program usage.

The System Analysis and Design Using SDLC Methodology

1. Feasibility Study

Feasibility study is the process to ensure that the project will be run well and possible to complete. In this study, the main subjects to test the feasibility are:

1.1 Existing hardware technology

We used the standard PC 2001 desktop computer specification is the based computer for developing and testing the application instead of used the standard PC 2002. The reason was the lower standard ensures that the application can run well on the most of computers using today.

The standard PC 2001 main components are as follows:

1) Processor speed at 600MHz – 1.0GHz

2) Main memory 64 MB

3) Hard disk capacity 20 GB

4) CD-ROM, Sound card

5) Display adapter supported DirectX 7.0 or above

1.2 Supporting technology for the implemented approach

Implemented Algorithms in this study are developed based on MS DirectX technology, which is a standard library for multimedia software development in present. Most of the games developing today use this technology as a tool for develop their products.

The Soft-Skinning Model technique used in this study is built on the DirectX Hardware Abstraction Layer (HAL), these mean that there are not any function in directly access to the hardware (4). Moreover, every display adapter, which supported the DirectX HAL, will be able to run the implemented application.

2. Determination of System Requirements

This step comprised of two portions; identify the basic requirements, and identify the existing problems. The results are as follows:

2.1 Basic Requirements

The basic requirements for the media used for self-study of Tai-Chi-Chuan are:

2.1.1 Clear information and enough for self-study

2.1.2 Ease to use

2.1.3 There are not many additional types of equipment needed for using the application.

The basic requirements also included the standard features of the media used in present, such as text describe, pictures, clearly movement display.

The minimum hardware requirement for the application is follow the standard PC 2001 specification that mentioned above.

2.2 Existing Problems

The problems found on the existing media used in present and be solved in this study are:

2.2.1 The information of movement is not clear enough.

2.2.2 Books are easy to use but have not shown the clearly path of animation, videotapes are smooth and realistic animation movement but not easy to point at only an interesting posture.

2.2.3 The current media unable to set the point of view as desire.

3. Design of the Software

Elements of the design consist of

3.1 Design of Input

3.2 Design of Output

3.3 Design of User Control

3.4 Design of Program

3.1 Design of Input

3.1.1 File Format Considerations

While the more data is available, the fewer calculations are required. Although the hard disk is cheap and file sizes are no longer has a significant role, because of an advance of computer technology in present. It still takes a relatively long time to transport the necessary data from a file on disk into the main memory of the computer. It is therefore advisable to only store on disk the data that cannot be generated while running by the program. This section will try to provide a simple analysis describing what data has to be stored in a 3D model file to support the animation of that model.

3.1.2 Requirements

Requirement components in the file format are:

3.1.2.1 Skeleton

3.1.2.2 Skin

3.1.2.3 Animation Set

3.1.2.1 Skeleton

The data required to store a skeleton for an articulated object on disk, apart from positional information for its joints itself, is the relationship between those joints. By far the easiest way to do this is by nesting the information for joints of a lower order in the hierarchy just below the joint, which they are supposed to be connected to (**Table 4.1**). For the joints themselves, all that needs to be known is the relative position of the joint, which occupies the next higher order in the joint hierarchy of the skeleton. The exception to this is the highest order joint in the hierarchy, the root of the skeleton, which should contain global positioning data for the whole object instead of the local positioning data.

Table 4.1 Hierarchy Skeleton sample

```
HEAD          <- file header
|
+- TEXT       <- global texture chunk
+- MATE       <- global material chunk
+- OBJE       <- object chunk - highest chunk in object hierarchy
|
+- NAME       <- the name of the object
+- VECT       <- the vector pool
+- LODD       <- LOD information of sequence in which
|                   vectors will be collapsed
+- SKIN       <- a skinned sub-object as sub-chunk of object chunk
||
| +- NAME     <- For example, 'Upper left arm'
| +- MATE     <- local material chunk
| +- POLY     <- geometry for 'Upper left arm'
| +- PIVO     <- joint data for 'Upper left arm'
| +- SKIN     <- a skinned sub-object as sub-chunk of object chunk
||
| +- NAME     <- For example, 'Lower left arm'
| +- MATE     <- local material chunk
| +- POLY     <- geometry for 'Lower left arm'
| +- PIVO     <- joint data for 'Lower left arm'
```

In addition to that, it may be useful to store constraints for each joint that limit the freedom of transformations, which can be applied to the joint. Then the animations that are generated while run-time can distinguish illegal movements that might cause the object to collapse. Bones (connections between the joints – vectors pointing from a joint to that joint's child joints) do not have to be explicitly saved in the file, as the joints that they connect implicitly define them.

3.1.2.2 Skin

The information that has to be stored for the skin of an articulated object, are the vertices that make up the skin. Each vertex structure has to contain data regarding the untransformed position of the vertex itself, the vertex normal, the UV texture coordinates of the vertex for texturing the model, a list of bones and skin weights. The skin weights defined which of the joints of the skeleton are able to influence the vertex and by how much each of these joints influences the vertex. IK chains to use IK with an articulated object, that object will have to include IK chains, defining which will be affected by the IK. The information required for this would be some sort of list, describing which of the joints in the IK chain is the start of the chain, and which of the joints is the end-effectors on the other side of the IK chain. That information could easily be stored inside the bone structures themselves, provided that the joints can point towards its parent joints, so that the IK chain can be back-tracked from its end-effectors to its starting joint. For more complex IK operations, one could also introduce a weight or sensitivity value that would define by how much a bone in the IK chain would be affected by the IK (10).

3.1.2.3 Animation Set

The easiest solution by far for this is the one implemented in id Software's Quake II file format: All models have an identical number of frames, every single frame is a key-frame and the length of each animation cycle is pre-defined. While this reduces the calculations that are necessary for displaying the animation to a minimum, it also reduces the flexibility of the file format. Usually though, transformation information for all the joints of the articulated structure is stored in each key frame of an animation.

3.1.3 A Look At The DirectX X-File Format

The X-File format is based on templates, which makes it very flexible and adaptable to specific problems. There are already a number of pre-defined templates, which allow the specific requirement data to be stored in the X-File. Adding custom templates to the file format can easily include the unfulfilled requirements. A full description of this file format and its capabilities can be found in Skeleton.

The DirectX X-file format supports skeletons in hierarchic articulated objects in the form called *Frames* of reference, which are the X-File equivalent for bones. Each Frame can contain a *FrameTransformMatrix*, containing local transformations for that Frame. A Frame can also contain *Mesh* objects defining the vertices that form a 3D model, and child Frames as described in Table 4.2.

3.1.3.1 Skin Weight

Skin weight is a part of the X-file format, although the modular and expandable nature of the file format, which allows the addition of new *templates*, would make it relatively easy to create an extension to the file format that could then be parsed and interpreted by the loading program (**Table 4.3**).

3.1.3.2 Animation Set

In the X-File format animation set are saved in the AnimationSet structure. Within an AnimationSet one can define a separate Animation for each part of the model that animates within the time frame of that particular animation cycle. Each Animation contains an AnimationKey structure that in turn contains a list of timed key transformations, which will affect the part of the model referenced by the Animation.

Table 4.2 Sample X-File

```
//  a sample X-File
// assumption: parent_mesh and child_mesh,
// parent_matrix and child_matrix
// are already defined

Frame Object_Root {
        {parent_mesh}
        {parent_matrix}

        Frame Object_Child {
                {child_mesh}
                {child_matrix}
        }
}
```

Table 4.3 Skin Weight template sample

```
template SkinWeights {
        <6f0d123b-bad2-4167-a0d0-80224f25fabb>
        STRING                  transformNodeName;
        DWORD                   nWeights;
        array DWORD     vertexIndices[nWeights];
        array FLOAT     weights[nWeights];
        Matrix4x4               matrixOffset;
}

SkinWeights {
    [Bone Name];
    Number of vertices;
          Vertex indices {
        }
        Vertex Weight values {
        }
   }
```

3.2 Design of Output

The screen functional design from figure 4.1 showed the main functions and screens flows. Main screen shown on the left side, it has three main buttons - "Start", "Credits" and "Quit". The results from the selected button are shown as screens on the right side. The "Start" button takes user to the Tai-Chi-Chuan training screen. The "Credits" button takes user to the credits list sub-screen, which accepts any mouse clicked for taking the user back to main screen.

Figure 4.1 Screen Functional Designs.

On the Tai-Chi-Chuan training screen have blue characters A, B, C, and D that shows the each functional part on screen.

A Animation rendering screen – this are is for rendering a 3D model with Tai-Chi-Chuan animation.

B Environment Control – this help user for rotation the viewpoint, another functions such zoom-in and zoom-out are around this circle also.

C Description Text box – this box is for describing text while animation running.

D Exit Button – this button let the users exit from training screen to main screen.

3.3 Design of User Control

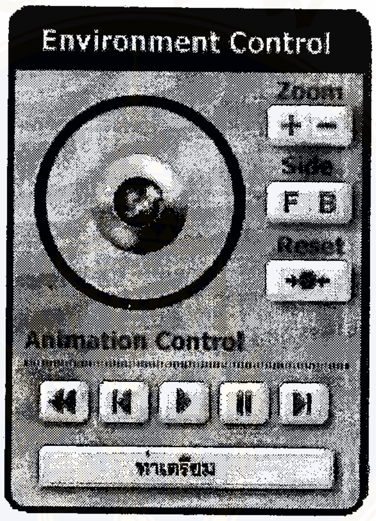

Figure 4.2 Main control panel.

The main control panel **(Figure 4.2)** can be break into 2 main parts – View Control and Animation Control.

View Control – consist of function for controlling the viewpoints. On the left side of the panel, a circle area contains a spherical orb and a smaller earth color orb. The earth orb represents the eye of view and the larger orb represents the actor. The users can drag the earth orb to any point in the circle area for changing the viewpoint. Any movement on the earth orb will affect to the render area immediately.

On the right side of the panel, there are three buttons – Zoom, Side and Reset. The Zoom "+" button helps the users change the viewpoint closer to the actor, the Zoom "-" button helps the users change the viewpoint out of the actor. The Side "F" button let the users change side of view from back to front, besides the Side "B" button let users change the viewpoint to the back side of the actor. The Reset button helps the users reset the movement of viewpoint they made to the pre-set value. This button does not effect to animation control, it only reset the eye of view.

On the bottom of panel that separated by dotted line and words "Animation Control". There are five buttons for control animation playback that similar to the function of another media playback application. The buttons are for – back to the begin of animation set, back to the last pose, play current pose, pause the current pose and skip to the next pose. The longer button below the control buttons is the pose name text. It will show the current playback pose act by the actor.

The view control panel can be drag to any portion of screen. This ability will let the users can move the panel away when it is on the line of sight to the actor.

Figure 4.3 Description text panel and additional buttons.

The bottom side of the screen shows the Description Text panel (**Figure 4.3**) and the exit button on the bottom right of screen. The Description Text panel contains the text described about the current playback animation. The title bar of the panel shows "Description Text", the users can drag the panel to any portion of the screen by dragging on this area. On the most right side of the title bar shows a white down direction triangle, this sign represent the minimize function. The users can click mouse at this point to minimize the Description Text panel, the panel will be change to a small icon "Text" as shown on the figure 4.3. The users can click at the "Text" icon again to restore the Description Text panel back.

The exit button helps the users exit from Training Screen and back to the main screen.

3.4 Design of the Software

The designs of the software described results of the design are as follows:

3.4.1 State Model

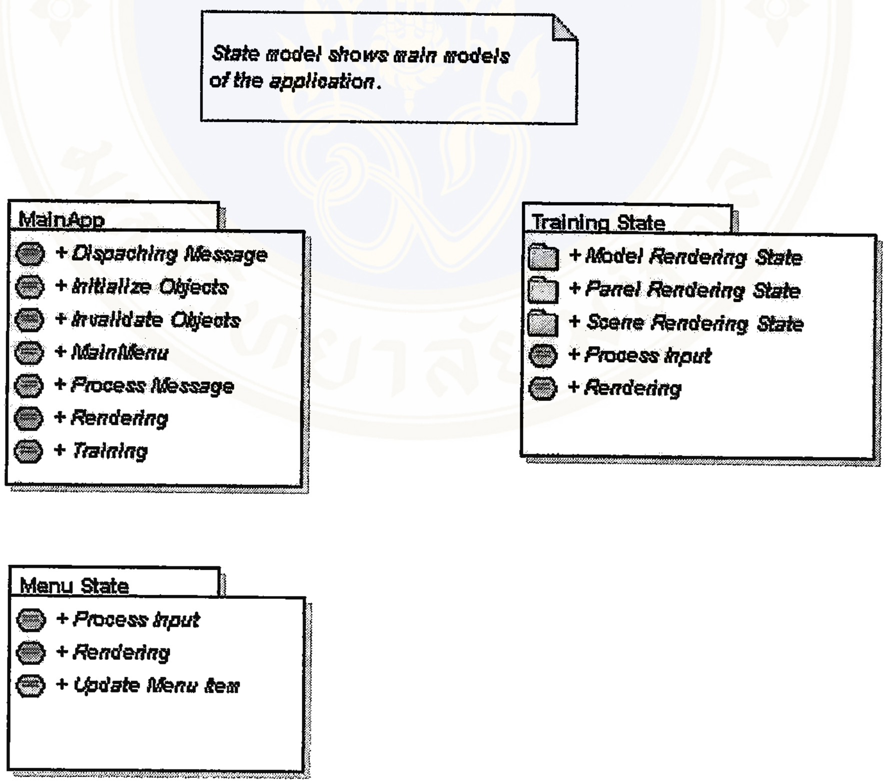

Figure 4.4 State model of the application

State model (Figure 4.4) showed that the application could be separated into three states - MainApp state, Menu state, and Training state.

MainApp described the states changed in the main application object. Menu state described the states changed in the main menu object. Training state describe the states change in training object.

3.4.2 State Diagram

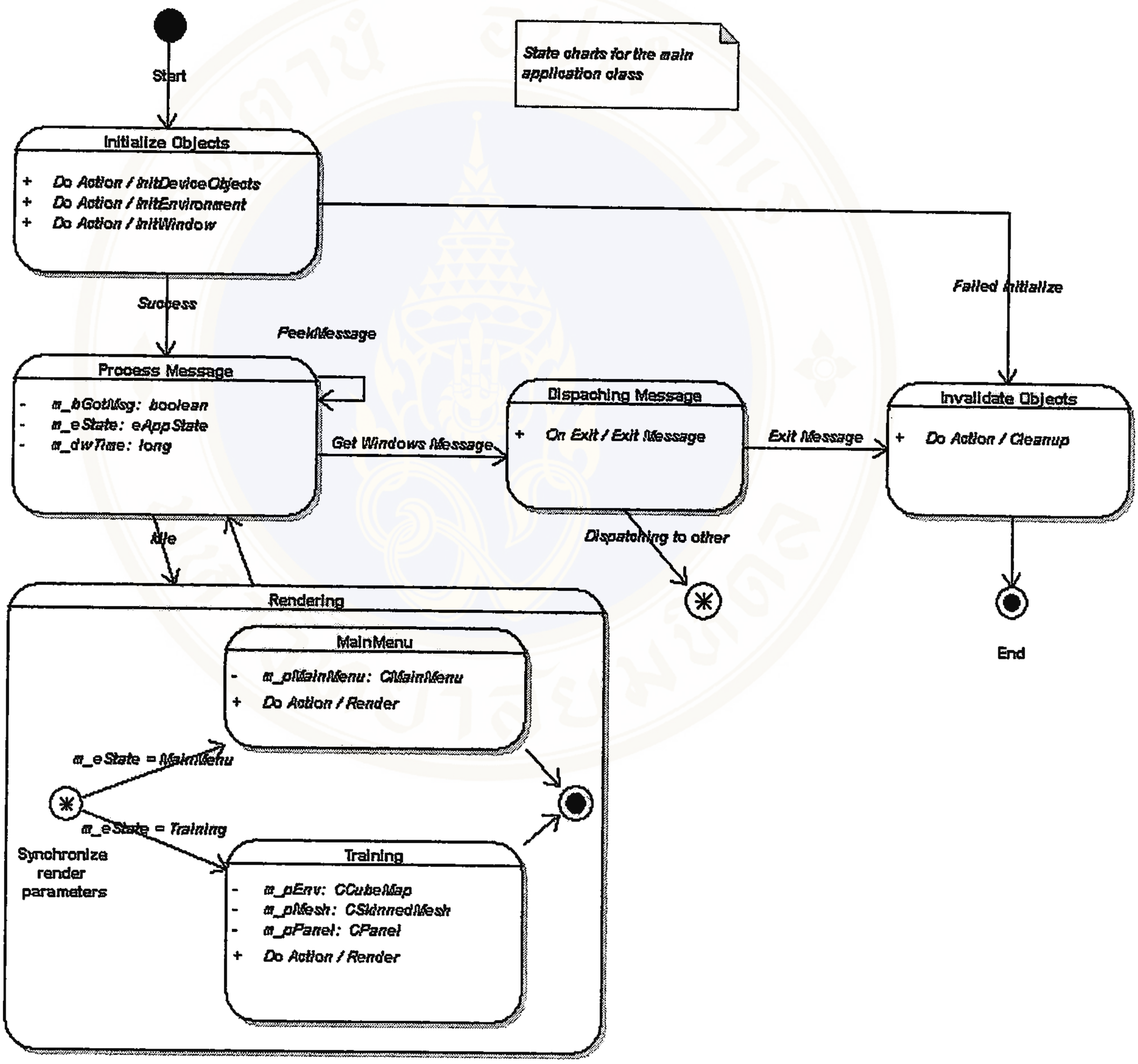

Figure 4.5 State charts of the main application state

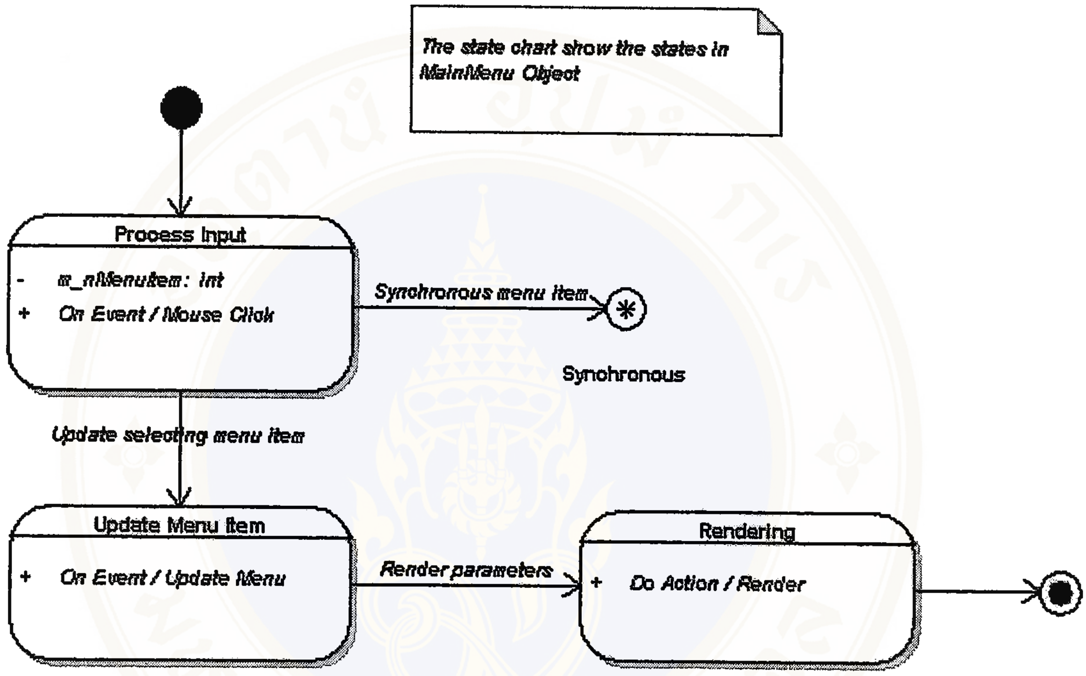

Figure 4.6 State charts of main menu state

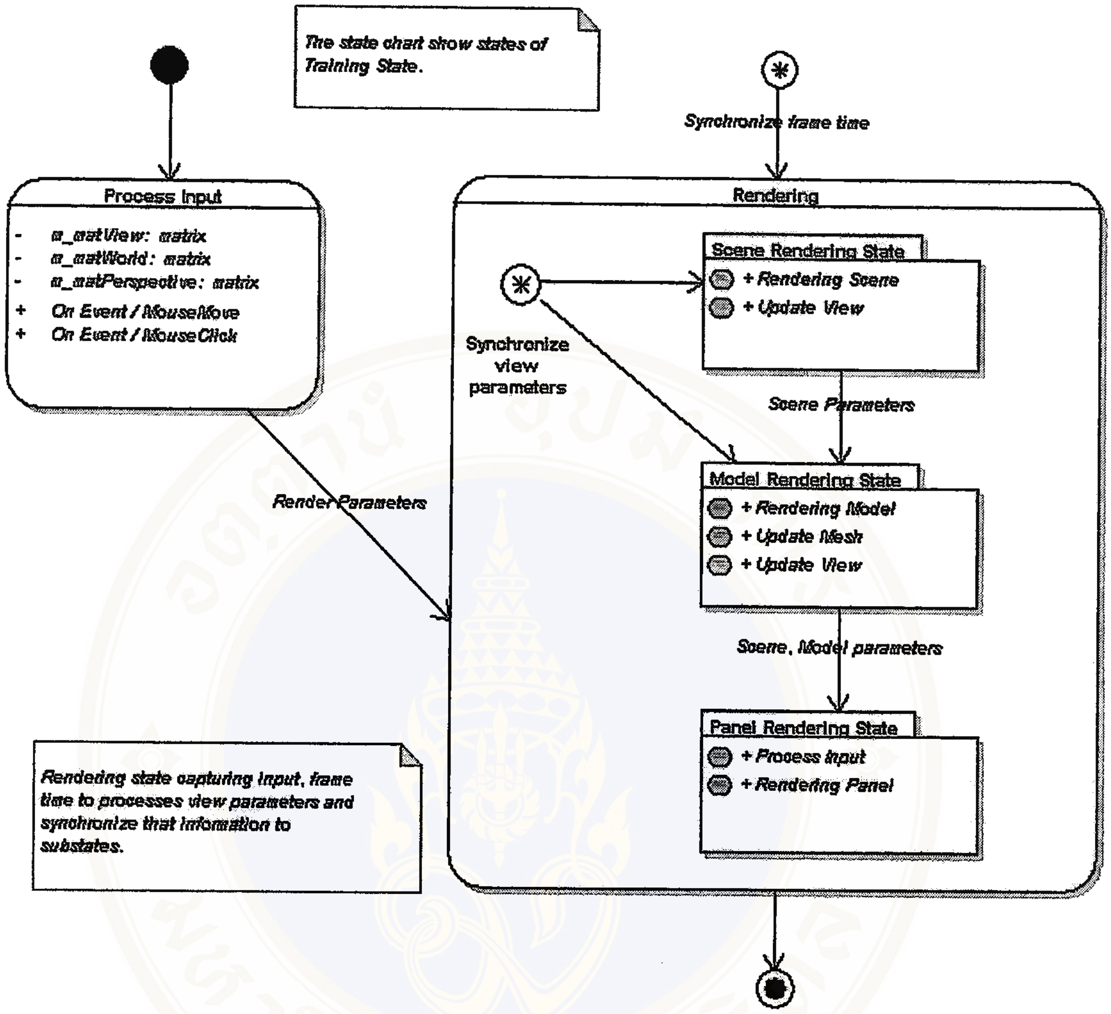

Figure 4.7 State charts of training state

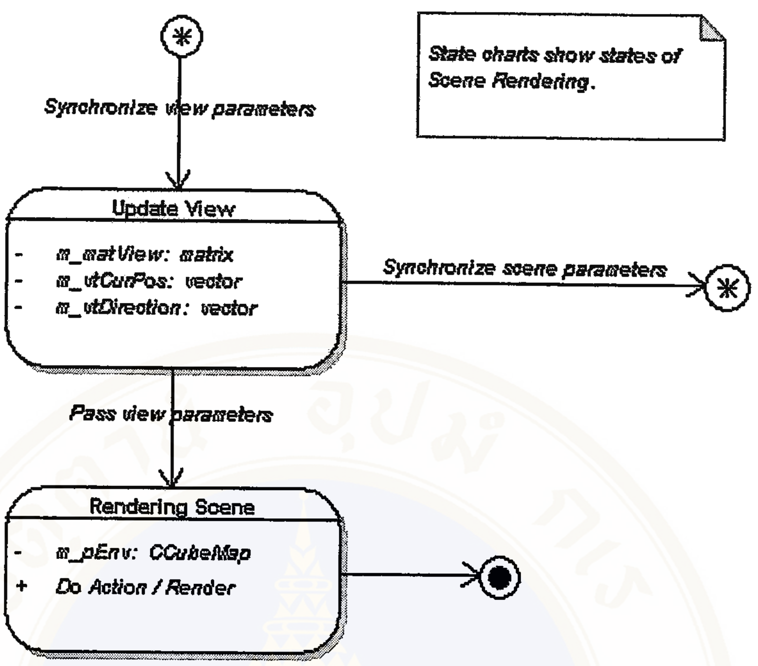

Figure 4.8 State charts of Scene Rendering state

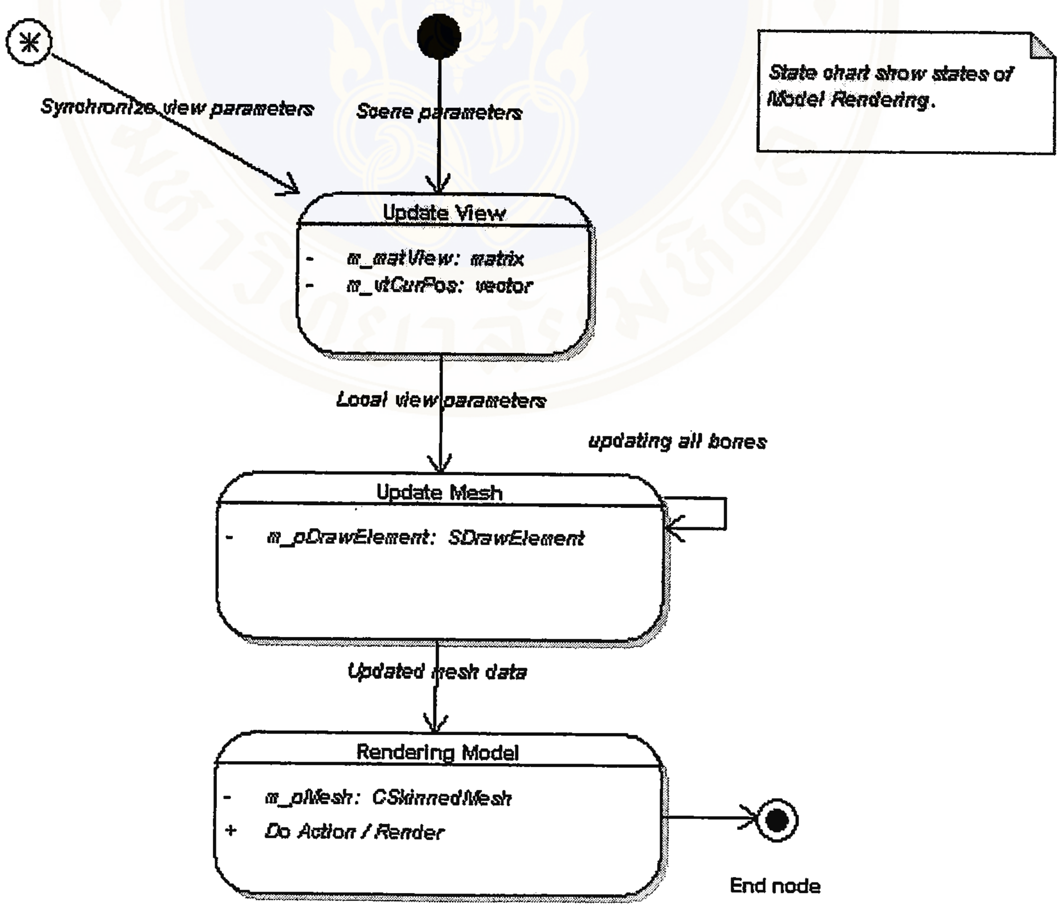

Figure 4.9 State charts of Model Rendering state

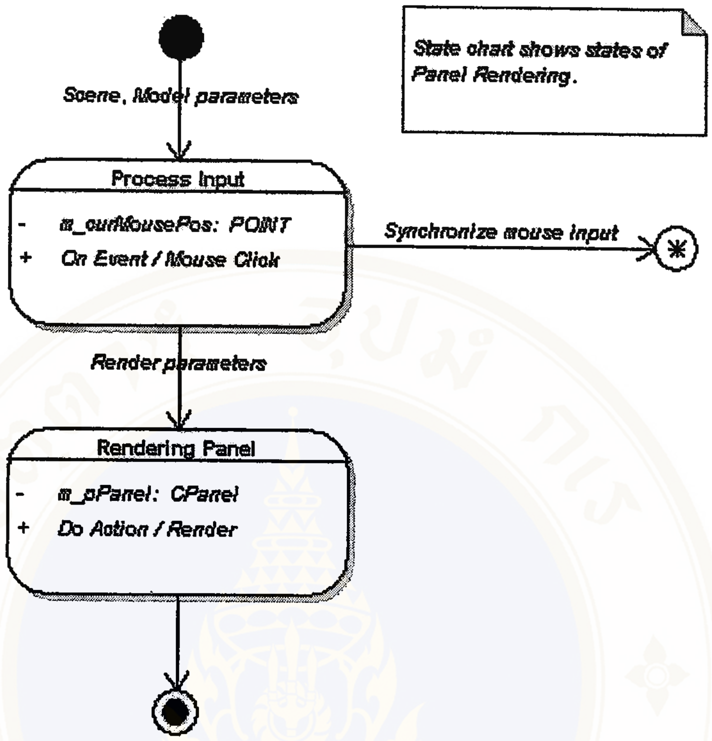

Figure 4.10 State charts of Panel Rendering state

3.4.3 Class Diagram

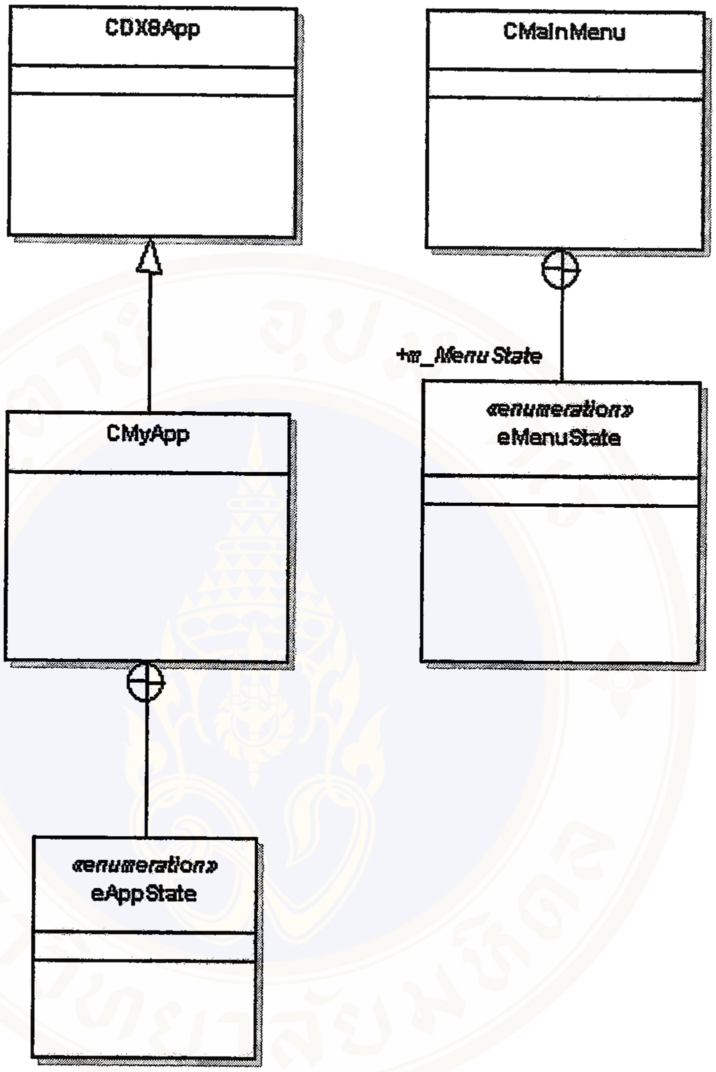

Figure 4.11 Classes relationships (1)

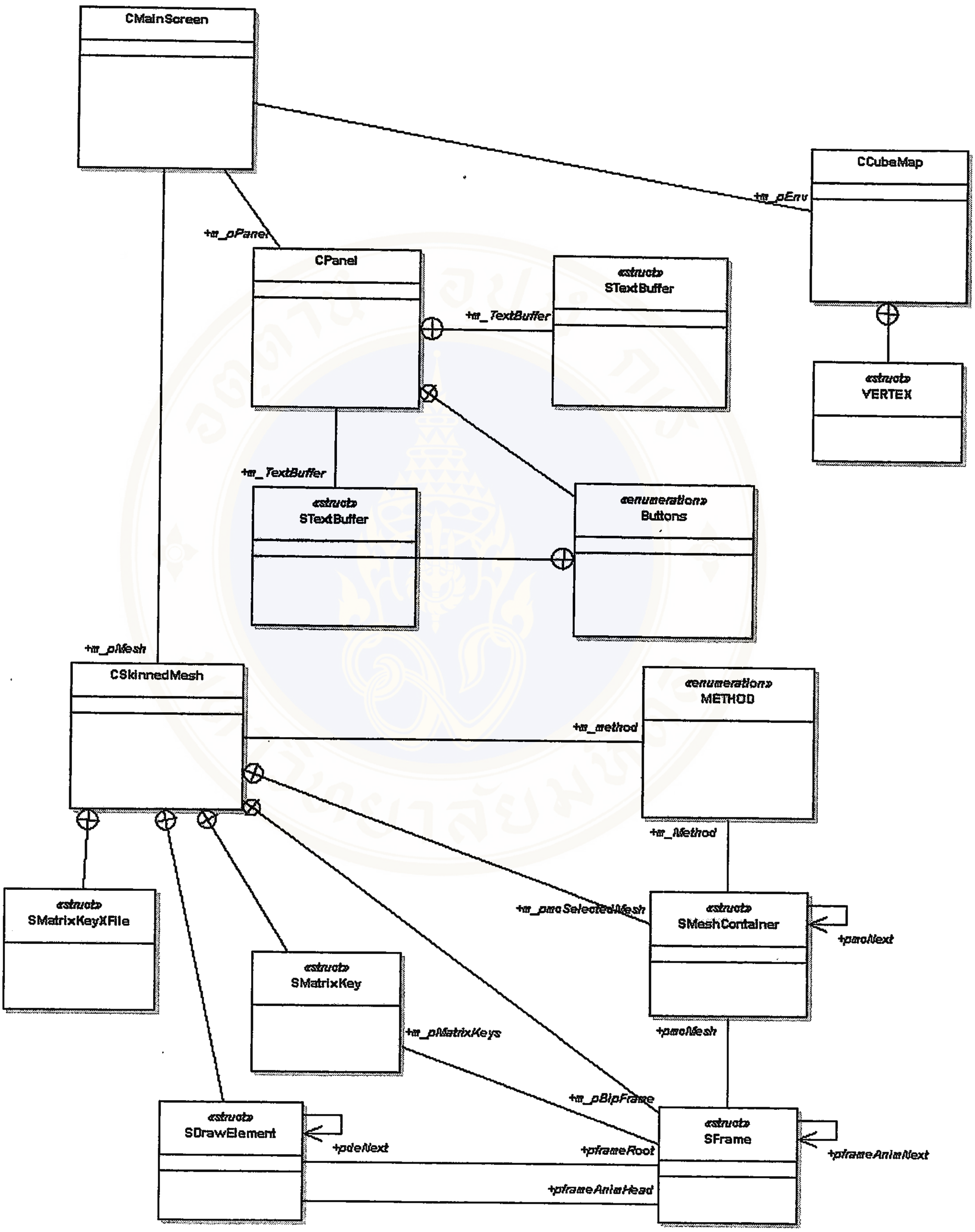

Figure 4.12 Classes relationships (2)

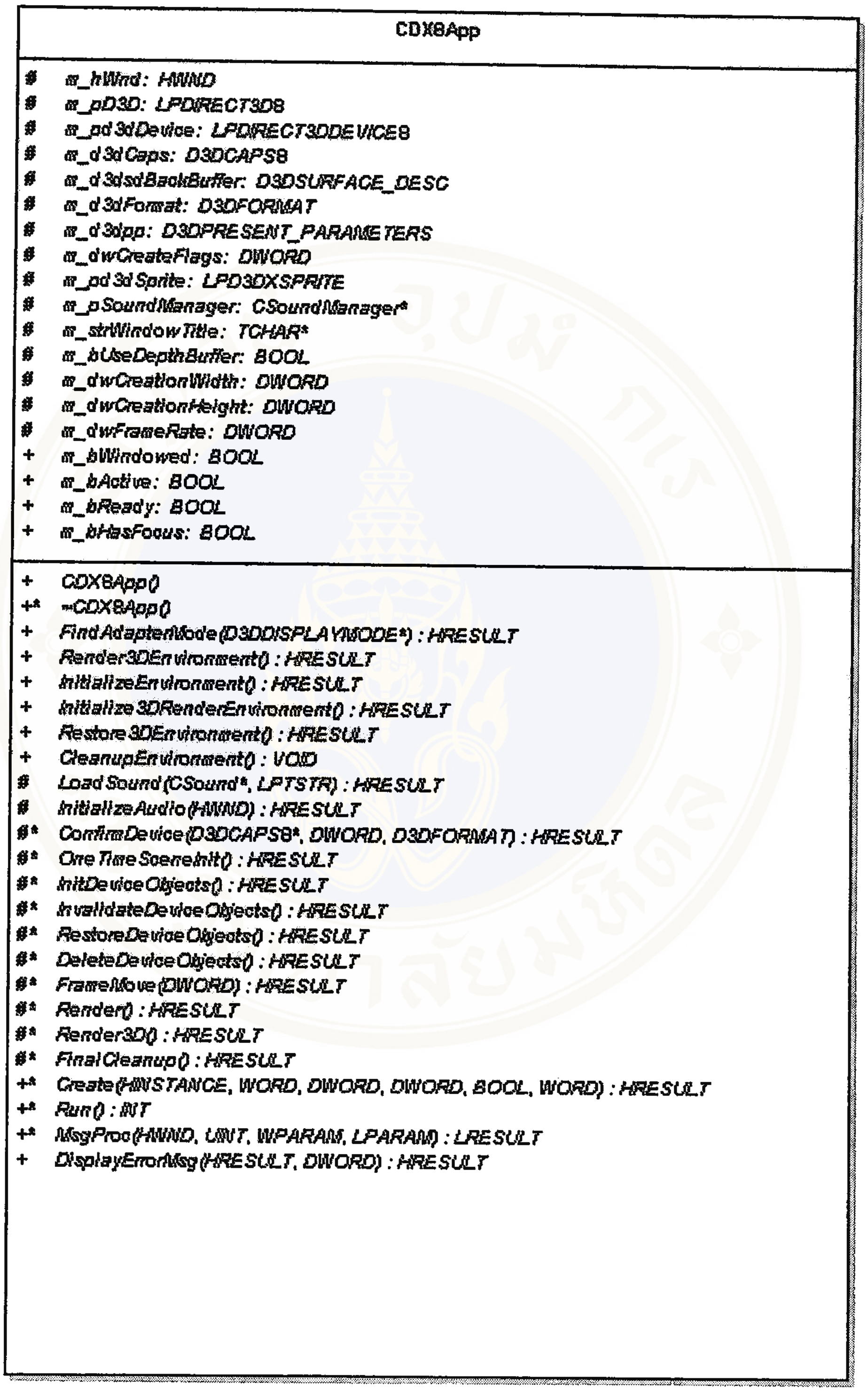

Figure 4.13 Class CD3DX8App

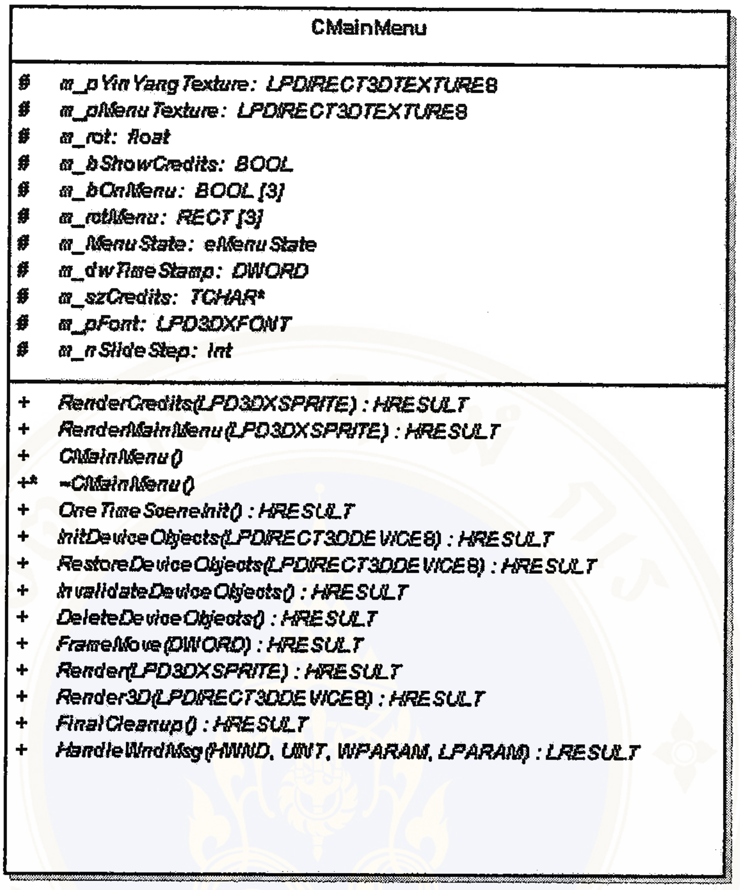

Figure 4.14 Class CMainMenu

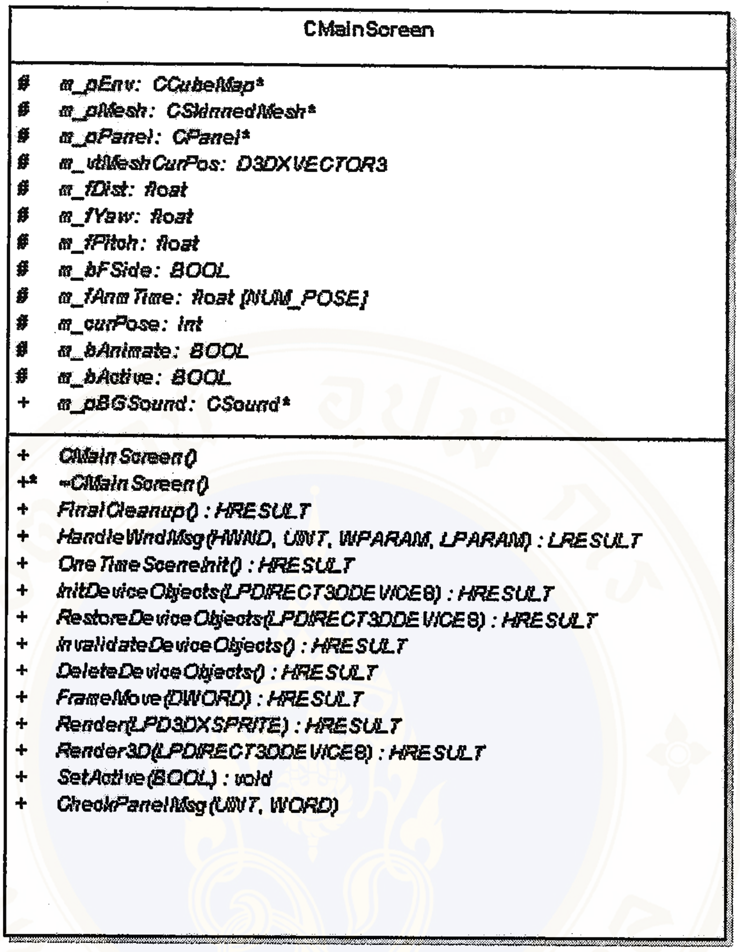

Figure 4.15 Class CMainScreen

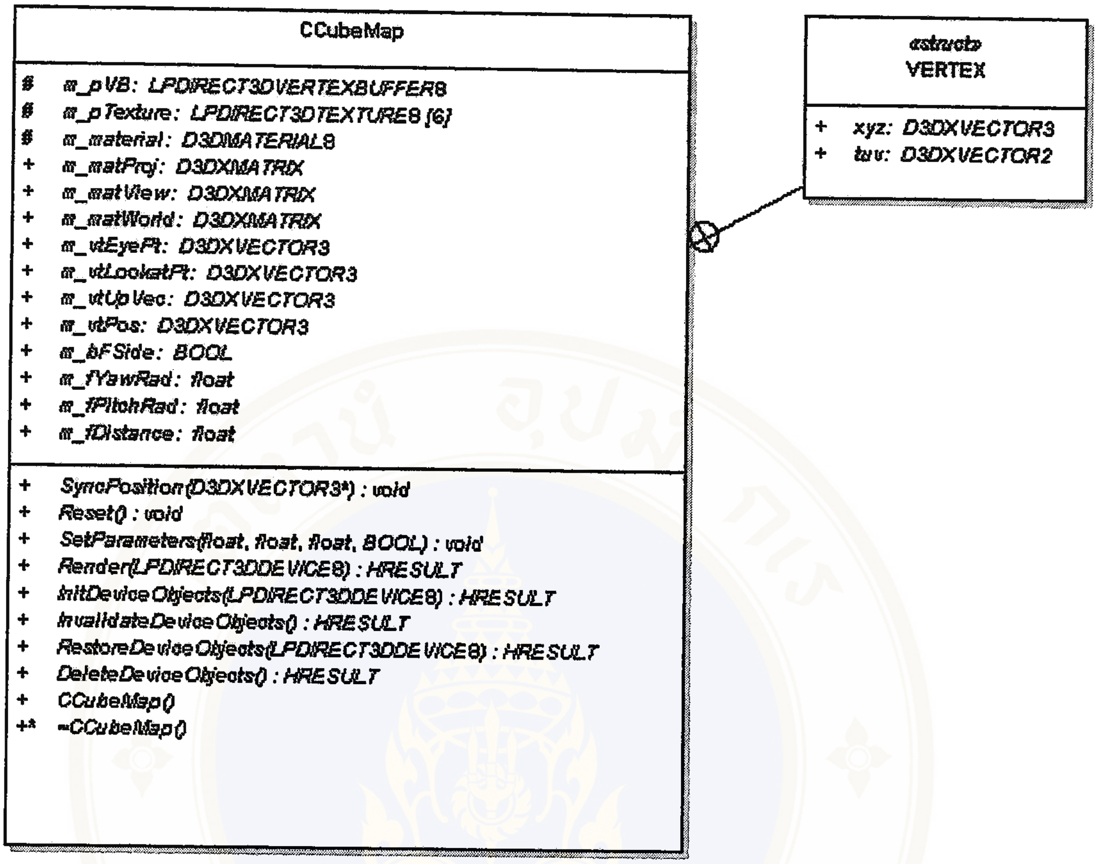

Figure 4.16 Class CCubeMap

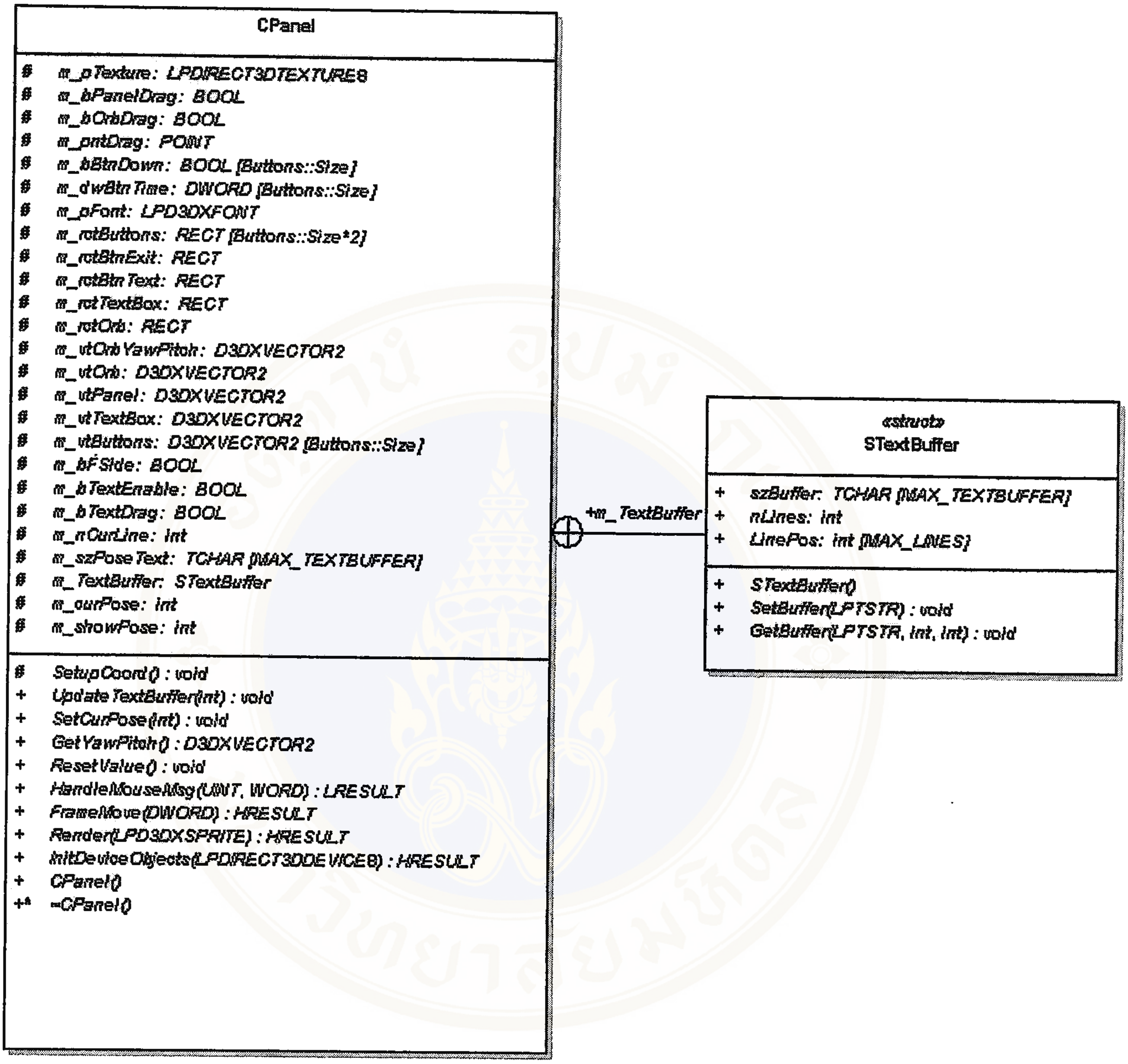

Figure 4.17 Class CPanel

	CSkinnedMesh
+	m_dwFVF: DWORD
+	m_pd3dDevice: LPDIRECT3DDEVICE8
+	m_pdeHead: SDrawElement*
+	m_pdeSelected: SDrawElement*
+	m_pmcSelectedMesh: SMeshContainer*
+	m_pframeSelected: SFrame*
+	m_pBoneMatrices: LPD3DXMATRIX
+	m_maxBones: DWORD
+	m_method: METHOD
+	m_matWorld: D3DXMATRIX
+	m_matView: D3DXMATRIX
+	m_matViewRot: D3DXMATRIX
+	m_vtEyePt: D3DXVECTOR3
+	m_vtLookatPt: D3DXVECTOR3
+	m_vtUpVec: D3DXVECTOR3
+	m_dwTotalTime: DWORD
+	m_bAnimate: BOOL
+	m_bFSide: BOOL
+	m_fDistance: float
+	m_fYawRad: float
+	m_fPitchRad: float
+	m_pBipFrame: SFrame*
#	CSkinnedMesh()
#*	~CSkinnedMesh()
#	Create(LPDIRECT3DDEVICE8, LPTSTR) : HRESULT
#	InvalidateDeviceObjects() : HRESULT
#	DeleteDeviceObjects() : HRESULT
#	FrameMove(DWORD) : HRESULT
#	Render() : HRESULT
#	GetMeshPos() : D3DXVECTOR3
#	ReleaseDeviceDependentMeshes(CSkinnedMesh::SFrame*) : void
#	SetProjectionMatrix() : HRESULT
#	CalculateSum(SFrame*, D3DXMATRIX*, D3DXVECTOR3*, UINT*) : HRESULT
#	CalculateRadius(SFrame*, D3DXMATRIX*, D3DXVECTOR3*, float*) : HRESULT
#	CalculateBoundingSphere(SDrawElement*) : HRESULT
#	FindBones(SFrame*, SDrawElement*) : HRESULT
#	LoadMeshHierarchy(LPTSTR) : HRESULT
#	LoadMesh(LPDIRECTXFILEDATA, DWORD, DWORD, LPDIRECT3DDEVICE8, SFrame*) : HRESULT
#	LoadAnimation(LPDIRECTXFILEDATA, SDrawElement*, DWORD, DWORD, LPDIRECT3DDEVICE8, SFrame*) : HRESULT
#	LoadAnimationSet(LPDIRECTXFILEDATA, SDrawElement*, DWORD, DWORD, LPDIRECT3DDEVICE8, SFrame*) : HRESULT
#	LoadFrames(LPDIRECTXFILEDATA, SDrawElement*, DWORD, DWORD, LPDIRECT3DDEVICE8, SFrame*) : HRESULT
#	SplitMesh(LPD3DXMESH, DWORD, DWORD*, DWORD, DWORD, LPD3DXMESH*, LPD3DXMESH*) : HRESULT
#	GenerateMesh(SMeshContainer*) : HRESULT
#	DeleteSelectedMesh() : HRESULT
#	DrawFrames(SFrame*, UINT&) : HRESULT
#	DrawMeshContainer(SMeshContainer*) : HRESULT
#	UpdateFrames(SFrame*, D3DXMATRIX&) : HRESULT
+	SetParameters(float, float, float, BOOL) : void
+	SetAnimationTime(float) : void
+	GetCurAnimationTime() : float
+	Animate(BOOL) : void

Figure 4.18 Class CSkinnedMesh (1)

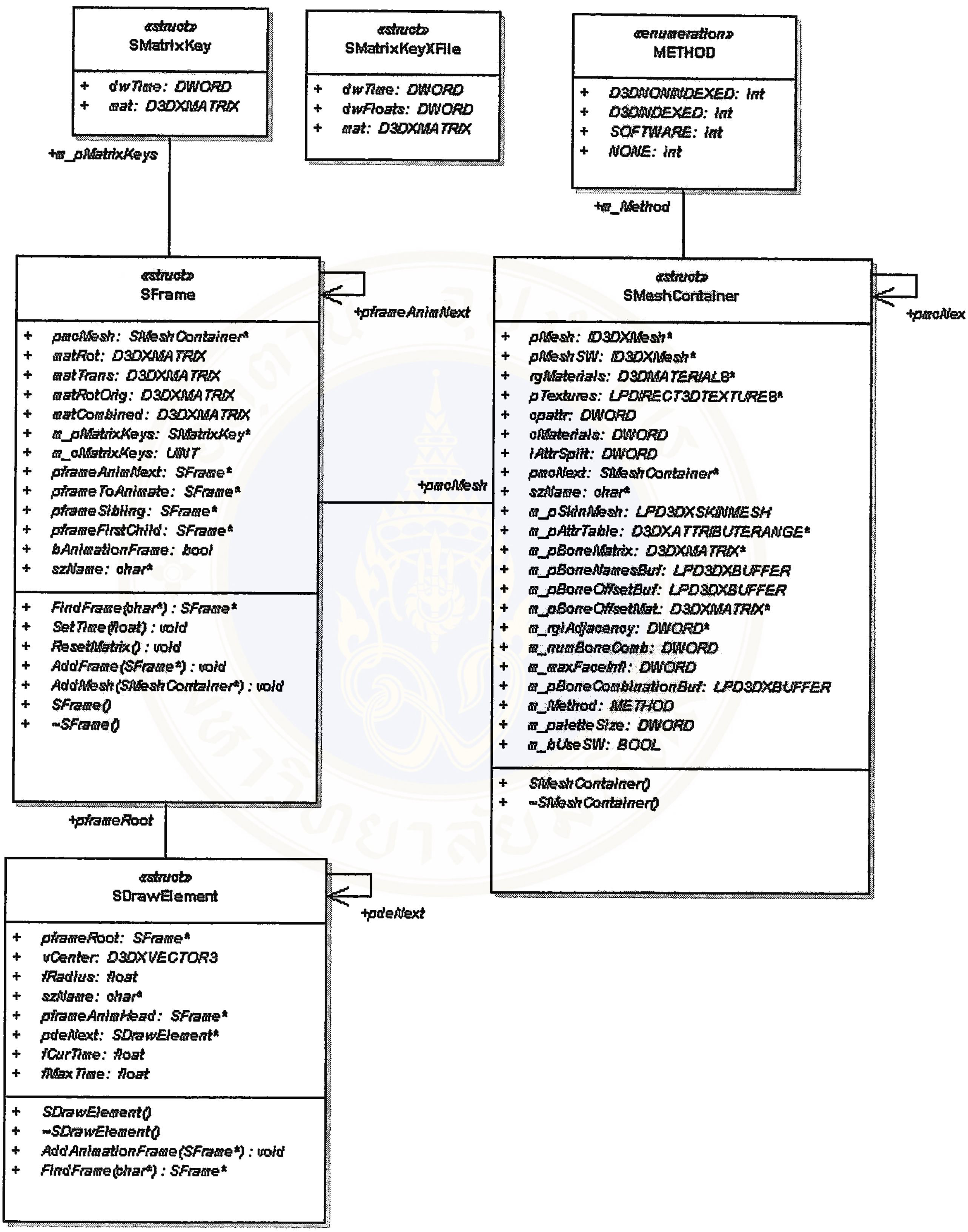

Figure 4.19 Class CSkinnedMesh (2)

4. Development of the Software

4.1 Data Gathering

Specifications of the 3D model and Biped using in this study are:

- The model does not have polygons over than 1000 polygons.
- The texture used with model should be square and its size does not over than 1024 x 1024 pixels.
- Total bones using in the Biped should not over than 50 pieces.

These specifications will directly effect to the speed of rendering. The over amount of polygons or bones cause the program need more calculate, and the rendering speed will slowdown (11). In the animation data collection process, we used 3D Studio Max with Biped as a tool for copying the posture from video source.

A Biped comprised of polygonal parts, its parts amount depending on our required. Each polygonal part stands for a body part of the Biped, and it was been linked with the related part followed the IK linked concept. The movement of any part will affect to the related parts, for example, rotation of its waist cause its hip followed, its chest and both hands will rotate to the direction of its hip also.

The subjects should be concerned while setting the Biped poses is the direction of each body part, balancing and feeling of postures. On some cases, the video source may not clear enough for copying its pose, especially the unseen parts, and cause the Biped poses an incorrect posture. The animation data collector could reduce this problem by trying to pose his body follows the Biped, and checks his feeling and balancing, or asked for a recommend from experienced Tai-Chi-Chuan users.

A supported tool used with Biped is Footstep. The main concept for using it is the user has to set the Biped's foots positions on each frame time, then the program will calculate the Biped posture between frames automatically. This tool may not help the user while setting posture of Tai-Chi-Chuan, but it can help the user can set beginning and ending point of each movement more easier. Because of we use videotapes as a data source, so we can use each video frame time to set each beginning step of movement, and set the movement between frames in detail later.

The problem found in this step is improper foot touchdown timing, because while the foot touchdown timing that foot would be locked on the floor and unable to slide or rotate that foot. Therefore, we should set the touchdown frame only when the foot on video source does not move or has a little movement.

General concept of making animation data by using Biped is set the beginning and ending points called "set frame", then the 3D Studio Max will automatically calculate model's position between frames with motion parameters. Animation data should be set on every body part, but each body part no need to set on every frame. The number of frames depending on the complex of movement, on some movement may have to set up to 10 frames per second, but on the simple movement may need only 2-3 frames per second. The parameters for controlling the Biped movement are normally present in a curve line. The shape of curve related to the characteristic of movement. These parameters could help an experienced user reduce the frame setting, and help the movement more realistic. Otherwise, an inexperienced user should use the default parameters for more convenience.

After finished the frame setting, next step is adjustment the animation smoothness for the contiguous and realistic movement. The tool that helps in this step is Trajectory. It shows the path of movement on each body part, and we can add or delete frames for smoother movement or adjust to the position, as we desired. On some cases, it can help us to remove unnecessary frames from animation sets. An animation data acquired from this process is for a 3D model that we prepared to use in this study.

Later than assigned an animation data into our model, we need to export all data from the 3D Studio Max to X file format. In an X file contained a model mesh, textures, bones and their position, including the vertices linked, transformation Matrix of each bone and so on.

4.2 Developing the Software

This step is the implement of algorithms process. There are two subjects should be concerned in this application, Animation Controlling Approach and Blending Between Frames Approach. The detail are described as follow:

4.2.1 Animation Controlling Approach

The data used in the model movement calculations are.

- Model data – comprise of all vertices that combine into a model, textures and materials used in the model.
- Bones data – comprise of bones data using in a model, related vertices and each bone influence.

- Animation data – position and movement data of each bone on any animation time periods, stored in a transformation Matrix for each bone.

Begin with finding an appropriate frame to render by compare the render time with time-table in an animation frame data, then find out the closest frame and get that animation data. The data acquire in this way are the position and movement data of each bone called bone transformation or bone matrix, these bones matrices are important for calculation the vertices.

The bone relation of each vertex is up to four bone matrices. We should know how the bones influenced do on a vertex; this data called blending weight and can get from the X file data (12). Unfortunately, we have to make sure that the blending weight use in transformation vertex must combine into 1.0 total. Therefore, we should use the formula as follows.

$$V' = \sum_{i=1}^{n-1} VM_i b_i + VM_n (1.0 - \sum_{i=1}^{n} b_i)$$

V'	Output vertex
V	Source vertex
M	Bone matrix
b	Blending weight

We have to calculate all of the vertices; the result is a transformed vertex that correctly as desired.

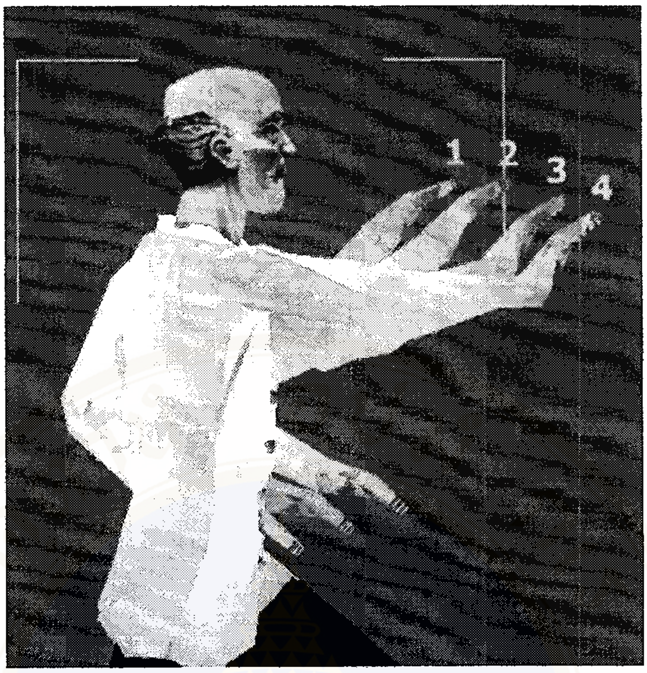

Figure 4.20 Blending between frames

4.2.2 Blending Between Frames Approach

On the last step, we mention about the animation controlling approach and the blending weight on the vertices. Afterwards we should consider on the model rendering between frames. Why this is important? The figure 4.20 shows a set of animation of the right hand. Begin with frame number 1 to frame number 4 takes 0.5 second long. If the animation data is only four frames per second, these mean there are only two frames (1 and 4). Even the computer can render at 100 frames per second, the animation display still rough, because first 0.5 second it rendered frame 1 then the 0.5 second later it will render frame 4.

Improve more detail of animation data is not a good cure. The larger animation data size may reduce the calculation time, but it will need more memory. The larger data size cause a lot of memory footprint occur while running, and may cause the system down from the out of memory. The detail of animation data should suit with the detail of movement, on some movement may need only 4-8 frames per second, and other works may need more over.

However, between frames of movement should display the correct position between those animation data frames. The calculation in this situation called *tweening* (stand for "in-between"). The approach for tweening is depending on the type of animation data format.

In this study, we store the animation data frame in a Transformation Matrix on each frame, so an appropriate approach for tweening is Linear Interpolation (Lerp) between two Transformation Matrices. The usage in this study describe as follow:

Weight Time between frames

M1	Transformation Matrix of frame 1
M2	Transformation Matrix of frame 2
M'	(1.0 – Weight) * M1 + (Weight – 1.0) * M2

5. Software Testing

The software testing strategy used in this study was Bottom-up testing, which appropriate for object-oriented systems (9). In the design state of this software, we separated the software into several small components covered their independent processes. Some data and functions that shared between components were connected via their interfaces.

The testing began with the main component, DirectX graphics component, and Application Framework component. This seems to be a Top-down testing strategy, but we had to test these main components before, because another components need data and interfaces from them. The main components had been tested while the application development step. The testing in this step was for ensure that the components still work well, and ready to test another components.

The animation data used in this testing phase were not the same as animation data used in final product. The data are smaller and simpler, because the reason of testing time. Completed data would be used for testing when the final product completed.

6. Implementation and Evaluation

Users who participate in this study program testing are six people, separated to experienced Tai-Chi-Chuan 2 users and inexperienced Tai-Chi-Chuan 4 users. Workstation Computer speed for the testing is 800MHz – 1.6GHz, and all display adapter supported DirectX 7.0. The study result can separated into two parts as follow:

6.1 Evaluation from Users Results – display the result from program used by six users, evaluation separated into three issues.

6.1.1 Complete information for self-study.

6.1.2 Clarification and friendly user interface.

6.1.3 Comparison with another media.

6.2 Animation Playback Results – display the result from the program functions for loading and controlling 3D model, evaluation separated into 2 issues.

6.2.1 Animation smoothness.

6.2.2 Display quality.

6.1 Evaluation from Users Results

All of the evaluation process done on the system speed at 800 MHz – 1.6GHz, and let the users tried the program for 2 hours. The evaluation result can be enumerated as follow,

6.1.1 Complete information for self-study

The program comprised of text and narrative sound that satisfy the users. Moreover, they can choose to close the sound and choose to playback the animation continuously. These capabilities make the users feel convenience when they want to repeat any animation, or move the viewpoint to any point as desired.

There are some recommendations from experienced users of Tai-Chi-Chuan about the animation movement. They found some incorrect postures and incorrect rhythmical movement on some postures.

6.1.2 Clarification and friendly user interface

Functions displayed on the screen have only necessary functions, the users may not need to choose or set any value for an operation. It is convenience to change the viewpoint, interface neat and easy to understand.

6.1.3 Comparison with another media

Another medias to be compared are books and videotapes, the evaluation result as follow,

6.1.3.1 Comparison with books

Books still more convenience for use, easy to carry and it can explain in detail more than this program. However, the implemented program more clearly postures, the movement details are clearly than books as well.

6.1.3.2 Comparison with videotapes

Videotapes have more clearly and more correctly movement. On the part of the posture movement, it is fulfilled than the implemented program. The main reason is an instructor who acted in videotapes usually is an expert on Tai-Chi-Chuan. That is the reason why the postures from videotapes are more clearly and correctly.

The advantages of this program over the videotapes are the program can change the point of view while the model moving, and can choose to look at any point during playback time. Moreover, it is convenience to playback on any selected postures as desire.

6.2 Animation Playback Results

The method to testing program use computer speed 1.6 GHz, display adapter support DirectX 7.0 functions, main memory 256 MB, display adapter memory is 32 MB. Screen resolution is 1024 x 768 pixels and color dept is 16 bits.

6.2.1 Animation Smoothness

Program can render picture at 80-120 frames per second depends on viewpoint and number of movement in the period. Rendering speed higher than the lowest acceptable rate (30 fps) then display result is softly and smoothly.

Frequently changes the viewpoint will be effect to the rendering speed, and cause the frame rate dropdown, but not badly enough to run roughly.

6.2.2 Display Quality

Qualities of the pictures depend on the detail of model and texture resolution. The model used in this study is a low-polygons model that has less than 1000 polygons, and use texture size at 512x512 pixels. The using of low-polygons model will raise the render rate, but the model will look like solid shape and unrealistic. Using high texture resolution will help the model look more realistic and compensate inferior while using the low-polygons model.

Texture filtering technique used in this research is Bi-Linear Texture Filtering, which help texture look more real and harmonious then display better result.

CHAPTER V

DISCUSSION

Main Purpose of This Study

The main purpose of this study is to represent a new format type for self-study of Tai-Chi-Chuan that different from another ordinary formats used in present. The chosen format is represent in 3D animation with sub-title text and narrative sounds.

The aspect concerned while choosing the 3D presentation format is an approach to create and control 3D animation data. We began with study several techniques used in game program in present, because the display format in this study is 3D real-time rendering program that similar with the characteristic of game program. Then we selected and adapted the soft-skinning technique that appropriate for using in this study. The adaptive focused on balancing of picture quality and display performance.

Review of the Findings

The results from the testers of this program can break into 2 groups. The testers whose do not have experience in Tai-Chi-Chuan satisfied in our 3D animation interface. Because they can change the view point to any points as desired, and that match with the main focus of this program also. Otherwise, the testers whose have experience in Tai-Chi-Chuan have some comments about motions and movements of a character in program. They found many wrong points and missing movement while the character acting. These may occurs because the animation data collection step do not have enough quality, and may not have a good error checking process while collecting animation data.

The program in this study focused on smoothness and speed of display, so we used the low-polygons character model and use medium texture quality to increase the realistic model. Another operations that increase the model quality but consume much more processing time such as light and shadow were ignored. These may cause the users seem the pictures not realistic enough. On the aspect of feeling while using this program, the most of users still more pleasing on use the books. The main reason is the books have more convenient to use than the videotapes and this program.

On the part of the technique used for rendering, we selected the Soft-Skinning Model technique that usually used in 3D game programs in present. And develop program function for controlling 3D model base on the concept of chosen technique. But the quality result is lower than we estimated, and can be described as follows.

- Animation data used in this study is more complex and large. We never optimize animation data before used, and the data format is not fitting for the program, as it should be. Animation data time used in this study is 5 minutes, but its size is over than 35 MB. This may cause the program consume much system memory while running, and take time for calculate.
- Hardware capabilities used in this study not fully supported all of functions used in this program. We have to calculate the rest part by software that makes the program run slower.

Problems Found in This Study

We got incorrect data from animation data collection process, so it took much time for correction. The most incorrect points were the model postures and movement steps of Biped. The main reasons are as follows.

- The data source is 2D pictures, so we cannot see another views except the camera view. And cause the error occurs with an unseen part while posing the Biped.
- Unable to see the correct position and direction of each joint from data source. Because the instructor in data source wear cloths, that hides his joint inside.

We may solve these problems by.

- Use 4-8 video recorders while recording the movement of an instructor. This may reduce the problem of useable parts while copy the posture from data source into 3D Studio Max.
- Put the markers on important joints of the instructor to help the copier know the exact direction of joints and movements.

These guides described above may help to reduce some incorrect animation data, but it will take more time for take care all of these processes. Using a specific tool -- Motion Capture hardware -- is not even to reduce the process time, but the data from this hardware is more accurate and the quality of motion data will depends on the movements of the instructor.

The size of animation data used in this study is rather large. It may cause the program run slower on PC that does not has enough system memory, because it need to swap into its hard disk while running. Because of the plug-in that used to export animation data from 3D Studio Max format to X file format used large data structure and generate many unnecessary records. The exporter will generate a transformation matrix of each bone for every frame movement. So the data will reach 3 millions records if animation data is 5 minutes long, frames rate at 20 frames per second and the Biped has 50 bones in its.

We may solve this problem by optimizing the transformation data. Remove the duplicated records or adjust the data structure to the smaller size.

We also found the rendering problems about the realistic pictures and animation smoothness. Quality of real-time rendering is closely depending on the display adapter hardware. The display adapter used in this study does not support all of functions used in the program. These cause some operations have been operated by software instead such blending the character model. Each vertex has been linked the character's bone, it may linked to a bone or up to 4 bones depending on each part. All of related linked should be processed by display hardware, but the display adapter used in this study can support only 2 linked for each vertex. These cause we have to split the vertex linked into 2 parts, first 2 linked do by hardware and the rest will be done by software. The render speed drops down about 20-30% because of this hardware limitation.

CHAPTER VI

CONCLUSION

Tai-Chi-Chuan is an exercise style that the people favorite in present. There are a lot of media for self-study about Tai-Chi-Chuan, but the most of them seem to lack of the detail in motion movement. In this study try to represent a new self-study media format that can provide more detail in motion movement than another media formats, it is a real-time 3D rendering program. There are several approaches used in real-time 3D rendering programs, we use the technique usually used in game program in present named "Soft-Skinning Model" in this study.

The program design focuses on the capability of changing viewpoint as desire, the friendly user-interface, including text describe and narrative voice for each movement. The movement data used in this study is the 24 poses set of Tai-Chi-Chuan and length about 5 minutes.

The evaluation could be separated into 2 parts – evaluation from users and the animation playback evaluation. The users are satisfied in the concept of bringing the capabilities of 3D animation to represent Tai-Chi-Chuan. Because the users can move the viewpoint as desired. But still have some comments about the correction of animation data, they found some incorrect movement on some poses. On the part of technique used in this study, the result is balancing on pictures quality and speed of display. But it has a hardware limitation on several program functions. These causes the pictures and movement do not realistic enough.

The recommendations for future works are increasing quality and accuracy of animation data collection process. The specific hardware such a Motion Capture could help in this process. On the part of technique used for 3D model rendering, the developers should consider on animation data optimization and using the hardware capabilities as much as possible.

Recommendations for Future Works

The concept of using 3D animation in this CBT of Tai-Chi-Chuan could be developed for a CBT of another movement such as dancing. The rendering technique and program function for loading and controlling 3D animation using in this study could be adapted for a game program also.

Quality of animation could be improved by optimize the data format of animation data. In this study, we store the movement of each body part separated by frame movement, each frame stored in a transformation matrix format. The duplicate records should be discarded and stored only begin point and end point of movement. Then calculate an interpolation between frames for each frame rendered, this will help the animation run more smoothly. Another problem when interpolate between frames is an incorrect transformation from using Matrix called "Gimbals Locked". These may solve by using Quaternion instead of Matrix, then interpolate between Quaternion by using spherical linear interpolation (Slerp).

Performance of a 3D real-time rendering program would depend on the capabilities of the display adapter it running on. Developers should study on the using display adapter about its capabilities and functions in detail, and use the appropriate functions for the processes.

REFERENCES

1. Kovach JP. Inside Direct3D. Redmont (WA). Microsoft Press; 2000.
2. Watt A. 3D Computer Graphics. 3rd ed. Addison-Wesley Publishing; 2000.
3. Pitzel S. Character Animation: Skeletons and Inverse Kinematics. Intel Developer Service.
4. Microsoft DirectX 8.0 Programmer Reference. Microsoft; 2000
5. คณิต คุรฑหงษ์. มวยไทเก็ก. กรุงเทพฯ: สำนักพิมพ์สุขภาพใจ; 2528
6. Writtin, Jeffrey L. System Analysis and Design Methods. Fourth Ed. McGraw-Hill Book: 1989.
7. Senn J. Analysis and Design of Information System. 2nd Ed. McGraw-Hill Book; 1989.
8. ไทเก็ก กายบริหารเพื่อสุขภาพ. บริษัท โฟโต้เฮ้าส์ คาเมรา แอนด์วีดีโอ จำกัด; 2543
9. Sommerville I. Software Engineering 5th ed. Addison-Wesley: 1996
10. Anderson E. Real-Time Character Animation for Computer Games. National Center for Computer Animation, Bournemouth University; 2000.
11. Todd M. 3D Studio Max 2 Fundamentals. New Riders; 1998.
12. Freidlin B. Enhancing Real-Time Character Animation with Matrix Palette Skinning and Vertex Shaders. MSDN Magazine 2001; 6:

BIOGRAPHY

NAME	Mr. Supakij Rattanasuwan
DATE OF BIRTH	25 April 1974
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Rajabhat Institute Suan Dusit, 1992-1996:
	Bachelor of Science (Computer Science)
	Mahidol University, 1997-2002:
	Master of Science (IT)
POSITION & OFFICE	1998-Present, OHOGame, Bangkok, Thailand.
	Position : Software Engineer