UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

DEVELOPMENT AND IMPLEMENTATION OF
WEB BASED TRAINING FOR TINKER AIR FORCE BASE

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of requirements for the
degree of
MASTER OF SCIENCE
(MECHANICAL ENGINEERING)

By
HENGZHONG WEN
Norman, Oklahoma
2002
DEVELOPMENT AND IMPLEMENTATION OF
WEB BASED TRAINING FOR TINKER AIR FORCE BASE

A THESIS APPROVED FOR
THE SCHOOL OF AEROSPACE AND MACANICAL ENGINEERING

BY

Chair : _________________________________
Dr. Kurt Gramoll

Member : ________________________________
Dr. Kuang-Hua Chang

Member : ________________________________
Dr. Zahed Siddique
Acknowledgements

I would like to express my sincere thanks and appreciation to my advisor and mentor, Dr. Kurt Gramoll, who has provided me guidance, encouragement, support, freedom and patience throughout this research and the process of writing this thesis. I am also indebted to Dr. Kuang-Hua Chang and Dr. Zahed Siddique for their advice, encouragement, and time to serve on my thesis committee.

I wish to thank all my coworkers in Engineering Media Lab (EML), especially to my coworkers in the Tinker research team for their help, encouragement, and friendship.

Above all I want to express my utmost thanks to my dear parents, my brother, and my sister-in-law for their continuous love during the process of my education. Without their advice, encouragement, and support, this thesis can never be done.
List of contents

Acknowledgements ........................................................................................................ iv
List of contents ............................................................................................................... v
List of tables .................................................................................................................. ix
List of figures ................................................................................................................ x
Abstract ......................................................................................................................... xii

Chapter 1 Introduction ................................................................................................. 1

1.1 Online training benefits ...................................................................................... 1
1.2 Online training drawbacks .................................................................................. 3
1.3 Tinker’s purpose to develop online training system ......................................... 4
1.4 Tinker online training requirements overview ............................................... 5
  1.4.1 Tinker online training security requirements ........................................... 5
  1.4.2 Tinker online training management requirements .................................. 7
  1.4.3 Tinker online training content requirements ......................................... 8
  1.4.4 Tinker online training teaching method requirements .......................... 8
1.5 Research objective ............................................................................................. 8
1.6 Thesis outline ...................................................................................................... 10

Chapter 2 Literature survey ..................................................................................... 11

2.1 Distance training overview ............................................................................. 11
2.2 Online training overview ................................................................................. 12
  2.2.1 Synchronous online training ................................................................ 13
  2.2.2 Asynchronous online training ............................................................ 15
  2.2.3 Summary ............................................................................................... 19
Chapter 3 Tinker online system architecture design

3.1 Tinker online training system organization overview

3.2 Tinker online training system architecture overview

3.3 Server system design

3.3.1 Why Database Management System (DBMS) is necessary

3.3.2 Overview of RDBMS

3.3.3 Chosen DBMS for Tinker Online Training System

3.3.4 Tinker online training database design process

3.3.4.1 E-R diagram creation

3.3.4.2 Table creation

3.4 Front-end system design

3.4.1 Web page layout design

3.4.1.1 System layout template design

3.4.1.2 Objective layout template design

3.4.2 Web content presentation design

3.4.2.1 Media types selection

3.4.2.2 Evaluation frame design

3.4.3 Training content organization

3.4.4 Navigation design

3.4.4.1 System navigation design

3.4.4.2 Objective navigation design

3.5 Requirements needed for web site browser

3.5.1 Plug-in requirement
3.5.2 System requirement ................................................................. 54

Chapter 4 Tinker online training implementation ............................ 56

4.1 Security protection ..................................................................... 56

4.1.1 User account security .............................................................. 57

4.1.2 User group security and management ..................................... 58

4.1.2.1 Administrator features ......................................................... 59

4.1.2.2 Manager features ................................................................. 65

4.1.2.3 Normal user features ............................................................ 66

4.1.2.4 Guest features ..................................................................... 67

4.2 Technology selection .................................................................. 67

4.2.1 Active Server Pages ................................................................. 68

4.2.2 VBScript ................................................................................. 71

4.2.3 JavaScript ............................................................................... 72

4.2.4 Macromedia Director ............................................................... 72

4.2.4.1 Lingo programming ............................................................. 73

4.2.4.2 Rich media ........................................................................... 74

4.2.4.3 3D feature ............................................................................ 76

4.2.4.4 Compression capacity .......................................................... 76

4.2.4.5 Performance optimization ..................................................... 77

4.2.4.5.1 Shared cast ...................................................................... 78

4.2.4.5.2 Movie playback option ....................................................... 78

4.2.4.5.3 Video streaming ................................................................. 79

4.2.4.5.4 Audio streaming ................................................................. 79
List of tables

Table 3-1 Summary of notation in E-R diagram .................................................. 30
Table 3-2 Data structure........................................................................................ 31
Table 3-3 Web content presentation style............................................................ 37
Table 3-4 Recommended system requirements .................................................... 55
Table 4-1 File size comparison............................................................................. 75
List of figures

Figure 1-1 Network connection relationship .......................................................... 7
Figure 2-1 Synchronous online training example – Webex........................................ 14
Figure 2-2 Asynchronous online training example – VTC...................................... 16
Figure 2-3 Asynchronous online training example – esafetyonline ...................... 18
Figure 3-1 Tinker online training system structure .............................................. 21
Figure 3-2 System architecture ......................................................................... 22
Figure 3-3 E-R diagram .................................................................................... 29
Figure 3-4 System layout template .................................................................... 34
Figure 3-5 Typical objective browser window ................................................... 36
Figure 3-6 2D simulation example .................................................................... 40
Figure 3-7 3D simulation example .................................................................... 41
Figure 3-8 Animation example .......................................................................... 42
Figure 3-9 Video example .................................................................................. 43
Figure 3-10 User progress page ......................................................................... 45
Figure 3-11 C135 Lesson description page as seen by an administrator user ...... 47
Figure 3-12 C135 Module introduction page as seen by an user.......................... 48
Figure 3-13 Objective navigation system ............................................................ 49
Figure 3-14 Activated administrator button (administrator view) ....................... 50
Figure 3-15 Visit an objective for the first time .................................................... 51
Figure 3-16 Visit an objective for the second time .............................................. 51
Figure 3-17 Deactivated next button (user view) ............................................... 52
Figure 3-18 Activated next button (user view) .................................................... 52
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3-19</td>
<td>Plug-in detecting window</td>
<td>53</td>
</tr>
<tr>
<td>Figure 4-1</td>
<td>New user registration form</td>
<td>57</td>
</tr>
<tr>
<td>Figure 4-2</td>
<td>Guest registration form</td>
<td>58</td>
</tr>
<tr>
<td>Figure 4-3</td>
<td>User group relationship</td>
<td>59</td>
</tr>
<tr>
<td>Figure 4-4</td>
<td>Administrator feature</td>
<td>60</td>
</tr>
<tr>
<td>Figure 4-5</td>
<td>Preference items</td>
<td>61</td>
</tr>
<tr>
<td>Figure 4-6</td>
<td>Manage group</td>
<td>62</td>
</tr>
<tr>
<td>Figure 4-7</td>
<td>Manage directorate</td>
<td>62</td>
</tr>
<tr>
<td>Figure 4-8</td>
<td>User progress list</td>
<td>64</td>
</tr>
<tr>
<td>Figure 4-9</td>
<td>Objective description page for administrator</td>
<td>65</td>
</tr>
<tr>
<td>Figure 4-10</td>
<td>Objective description page for normal user</td>
<td>66</td>
</tr>
<tr>
<td>Figure 4-11</td>
<td>Tinker web site display process</td>
<td>69</td>
</tr>
<tr>
<td>Figure 4-12</td>
<td>Lesson 1 description page</td>
<td>70</td>
</tr>
<tr>
<td>Figure 4-13</td>
<td>Lesson 2 description page</td>
<td>70</td>
</tr>
<tr>
<td>Figure 4-14</td>
<td>C135 welcome page</td>
<td>71</td>
</tr>
<tr>
<td>Figure 4-15</td>
<td>EM welcome page</td>
<td>71</td>
</tr>
<tr>
<td>Figure 4-16</td>
<td>Four answers evaluation frame</td>
<td>74</td>
</tr>
<tr>
<td>Figure 4-17</td>
<td>Two answers evaluation frame</td>
<td>74</td>
</tr>
<tr>
<td>Figure 4-18</td>
<td>An example of Lingo controlled animation</td>
<td>82</td>
</tr>
</tbody>
</table>
Abstract

Information technology has made rapid strides over the last several years, including the availability of broadband services, improved web-development software, and firewall guaranty. These technologies now make it possible to offer fast transmission speed through Internet and high security web based training with streaming audio and video to enhance the learning experience for industry. This research focuses on the Tinker online training system (http://tinker.ou.edu) that has been developed and implemented for Tinker Air Force Base (Tinker AFB) behind their firewall. This developed web based system includes both the training management tools and database to administer the training process via web site, and over 40 lessons to instruct base personnel in either C135 aircraft maintain or environmental management.

The main goal of the online training system was to educate the staff and contract workers on the precautions, procedures, and environmental regulations at Tinker AFB. By having this training available on the web, the system administrators are able to manage the courses and monitor the performance of learners instantly. Meanwhile, thousands of personnel can take the required courses simultaneously. This system offers the flexibility of being able to take the courses anywhere, anytime. Various unique features have been incorporated into this web based training program, such as security protection, scalable database, comprehensive management tools for both course materials and user information, informative simulation, user-friendly interface, and flexible navigation.
In order to provide a web training system for Tinker AFB and meet their high security requirements, methods including controlling traffic, account authorization, and server side script was integrated in the training. A server system is developed to manage the training process. It was composed of an extendable program database and Active Server Pages (ASP). The database stores user information and course content outlines. ASP was employed to pass parameters among database tables, and between the learners’ browser and database tables.

In order to present the training content, a front-end system was developed. It includes interface, navigation, and course presentation design. An attention-grabbing interface was created for the training. Flexible and constrained navigation force the users to learn. Interactive presentation is formed by integrating 3 dimensional (3D) and 2 dimensional (2D) simulations, videos, animations, narrations, texts, and graphics with the support of multimedia technologies, such as Director, Flash, Freehand Photoshop, and Premiere. C135 aircraft maintain procedure (C135) and environment management rules (EM) are the two central materials in this training.

This thesis focuses on the protected method employed in this training, the multimedia technologies used to design and implement the training content, and the server system used to deliver the training content.
Chapter 1 Introduction

Since the beginning of education and training, face-to-face teaching has always been the most universal method, yet more methods are becoming available for education and training with the continuous improvement of technology. Before the 1800’s, the most common education and training method was face-to-face teaching; the only necessary technology was the knowledge of the teacher. As radio developed during the First World War, radio courses emerged rapidly and became the mainstream of distance training; at that time radio technology and printed material were needed in the process of learning; When satellite technology appeared, radio-based courses shrunk and televised courses took the place of radio-based courses as the most popular distance learning method because of its interactive presentation format. From the time when the computer appeared and as the speed of CD-ROM increased, computer-based courses and CD-based courses gradually occupied the distance education market. After Internet technology arrived, online training attracted the world’s attention as the bandwidth improved and multimedia tools developed.

1.1 Online training benefits

Web based education and training has been gaining worldwide popularity for a number of reasons. Online training gives learners the ability of access to training courses via World Wide Web (www) from anywhere in the world, at any time. It is convenient and easy to access. As less traveling between the education agency, office and home is needed, fewer fees are spent on the road
and learners worry about neither the location of the university/school nor the weather. Although the budget for building online education and training systems is high, a course system can be reused as many times as the teachers want and the system can serve an unlimited number of users, thus the cost for each user is low. During the online training, group studies may be employed. In this case, other group members are brought from other countries right in front of the learner, thus it makes international colleagueships easy to form [1]. Another benefit is, time for digesting the knowledge learned from the web is fully controlled by the learners themselves; the slowest learners are no longer constraining the learning pace. Furthermore, students can review the course materials whenever and wherever they are.

With the rapid advancement of communication technologies, Internet Service Providers (ISP) are able to offer higher bandwidth. This opens up the door of online education, which was previously limited to text and graphic, to include other multimedia technologies. The embedding of other multimedia technologies in e-training enables the learner to have a more interactive and eye-catching training environment [1]. Audio, video, animation, simulations, and 3D objects and graphics are integrated to the training to raise the interest of learners and catch their attention. A unique feature for e-training is that resources related to course materials are provided to students as hyperlinks, so that learners can delve deeper into any topic they are interested in with less material searching time.
The rapid spread of web based education and training has proved to provide greater convenience for teachers. Teachers have the ability to update training contents without the worry of several different versions of the contents being available simultaneously. Once the teachers update the training content, students all over the world are able to view the latest version of the courses immediately. No traditional mail correspondences or printed materials are needed. Class managements can be done by the teachers via websites too [1, 2]. For example, class assignments can be posted on the web and students can submit their homework to the class database. When compared with the traditional education, Internet based courses can serve an unlimited number of students, while face-to-face teaching can only accommodate a limited number of students. Furthermore, a database management system makes web based class enrollment more flexible. There will be no need to wait in line to get enrolled and an up-to-date number of available seating for each class is displayed. As the online course is a prerecorded presentation format, the course content and expression is optimized, so that it is easy for the users to learn and all the knowledge presented is focused on the main topics of the training.

1.2 Online training drawbacks

In spite of all its obvious brightness, online training still has some drawbacks that prevent it from being applied even more. The most challengeable fact is the bandwidth limit. Two methods can bypass this problem. The first solution is dependent on the development of the new technologies for hardware such as Digital Subscriber Line (DSL) or cable modems. The second solution
focuses on the software aspect. By carefully selecting the presentation media types, the limited bandwidth can smoothly deliver web based courses. To be an expert media type selector, a teacher will have to invest time learning many multimedia tools to build an interactive learning environment. Since the tools keep changing to add features and become more convenient, there is going to be a slope learning curve for the teachers to learn all the tools. Although online training is excellent for 90% of Internet based courses like concept explanation, data analysis, and software utilization, it is impossible to learn physical skills online. For example, even though persons can learn how to weld in an electronic course, they can never be a welder unless they practice welding in the real world [1]. Additionally, technical support is needed to deal with hardware and software technologies during the development and utilization of web based training. Furthermore, firewalls installed by some government departments will prevent the proper display of training content. Regardless all these disadvantages Tinker employed online training method.

1.3 Tinker’s purpose to develop online training system

Tinker Air Force Base (Tinker AFB) decided to develop an Internet based class to instruct base personnel in C135 Cabin Pressurization Course (C135) and Environmental Management Training (EM) on account of the following facts. Tinker AFB intends the learning system to serve 5,000 people simultaneously, which is nearly impossible for face-to-face teaching. An Internet based course seemed practical because of its support for an unlimited number of assessments at the same time. Meanwhile, online training can be used for
employers in large aerospace companies as well. Since base officials do expect the base personnel to go through the course content and learn, a user-tracking feature, which monitors the students’ progress and performance was needed. Additionally, Tinker administrators want to manage the training courses. As many rules and laws in the C135 aircraft maintenance training and environmental management course are taught, catching the attention of personnel is one of their utmost concerns. Since the interactive Internet based learning environment can solve these problems, Tinker decided to adopt it for training. Tinker AFB funded the Engineering Media Lab (EML) in the School of Aerospace and Mechanical Engineering at the University of Oklahoma to develop this online training system.

1.4 Tinker online training requirements overview

The Tinker online training system is a research topic funded by Tinker AFB. The requirements for this system consist of four parts including security, management, course content, and teaching method. Although these requirements are posted by Tinker AFB, they are general enough for all kinds of web based training system. For example, these requirements can also be used to teach collage students.

1.4.1 Tinker online training security requirements

The issue of Internet security is Tinker AFB’s main concern because the methods of web attack are becoming more concealed and advanced, and the number of web attacks is increasing. Since Tinker AFB is a military base, it is
one of the most favorite attack objects by various hackers. Additionally, millions of computers in Tinker AFB are connected together to form a local area network (LAN), which is connected to other government and military bases’ LANs, so if one computer or server is hacked, other computers or servers may be affected. Therefore, the base officials decided to set up the server permanently in a controlled area inside the Tinker AFB firewall. By doing this, the server can be monitored twenty-four hour a day and problems can be solved more quickly. The server is a key component for this training system since it stores user information and all the training content.

In whatis.com, the definition of a firewall is:

A set of related programs, located at a network gateway server, which protects the resources of a private network from users from other networks. (The term also implies the security policy that is used with the programs.) An enterprise with an intranet that allows its workers access to the wider Internet installs a firewall to prevent outsiders from accessing its own private data resources and for controlling what outside resources its own users have access to [3].

Figure 1-1 shows the attribution of the Tinker firewall. This firewall prevents the outside harmful program from entering the Tinker LAN. Yet the judgment of which program is unsafe is hard. Sometimes it blocks useful programs and information as well. For example, when QuickTime is playing in Shockwave Director, Director needs QuickTime Xtra to support this media type. And this Xtra does not contain any dangerous programs or information, but the
Tinker firewall prevents its downloading. Therefore, Tinker’s requirement posted a new research topic – select the media types properly so that the web content can pass the Tinker firewall and display correctly. This issue will be further discussed in Chapter 3 and 4.

1.4.2 Tinker online training management requirements

Tinker online training is supposed to serve 5,000 persons simultaneously. In order to manage such a large number of people easily and systematically, all the users are grouped into teams according to the department. For C135 Cabin Pressurization Course, the users are grouped as 10.9 team, 8.6 team, Manager and Administrator. In EM Training, they are Depainter, Manager, Painter, Supervisor-uec, Wipers, and Administrator. The administrators in the system are responsible for the management of all the managers and all the users. The managers need to supervise the users in their own departments. Both
administrators and managers are able to monitor the users’ performance and assign training materials for them. Furthermore, different user groups need to be assigned to different course contents.

1.4.3 Tinker online training content requirements

The course material that will be taught on the web site needs to follow the source materials provided by Tinker AFB on two topics: C135 Cabin Pressurization Procedure and Environment Management.

1.4.4 Tinker online training teaching method requirements

Since the reality is the best training material, documentary videos need to be heavily employed in the course. In order to attract the learner’s attention, interactive learning environment is prefer. Less text is appropriated to the current learners.

1.5 Research objective

A key objective of this research is to create an online training system according to the requirements of Tinker Air Force Base. In order to carry out this objective, five major features need to be created and implemented.

- Firewall solution – Tinker firewall prevents the QuickTime video from playing inside Director movie. Flash video is created to integrate video in Director movie and passes through the firewall.
- User tracking – in order to monitor learners’ performance, a user tracking system is established.
• Protected web site – security is a critical issue for this training not only because military web sites are among the most favorite hacking objects by curious persons but also because the Tinker officials expect to control the whole training process. Therefore database management system is carefully selected, user accounts and user group accounts are set up to manage the users and protect the various course content.

• Scaleable database – database stores and retrieves data correctly and rapidly; meanwhile an extendable database provides convenient further development of the entire system. Furthermore, the data stored are the basis according to which future improvement will be done.

• Multiple media type – by carefully selecting the media types to present the course content, an interactive learning environment, meets the Tinker’s high security requirement, is formed.

• Multiple technologies – with the utilization of various technologies, the Internet delivers the training material smoothly under the current bandwidth limitation.

• Flexible navigation – flexibility provides a user with a fully controlled navigation system in order to make the learning experience convenient and constrained.

• Attention grabbing interface – interface attracts the user the first time they visit the training while saving the developer’s time.

Although the training system is designed for Tinker AFB, it can be used in general engineering training for industry and formal education. 3D
simulations and animations developed in these online courses can help explain complex concepts more clearly for technical persons and university students. Flash videos can pass firewalls better and thus fulfill security requirements. The idea of design Flash video can apply to other departments whose Internet are protected by various kind of firewalls. The management tools can be used to instantly monitor the users’ progress and dynamically administrate web based industry training and formal education.

1.6 Thesis outline

This thesis covers all the topics about design and implementation of the Tinker online training system. The present chapter is the overview of the online training and the foci of this thesis. The following chapter will discuss the background of the distance training and several similar training systems that have used in Internet based training. Chapter 3 covers the topic on the training system design, which is composed of database design, interface design, media type selection, and navigation design. Chapter 4 goes over the technologies used in the implementation of this Internet based course. In Chapter 5, contribution of this research to this field is described and future work to improve this web based training course is recommended.
Chapter 2 Literature survey

Distance training has been a research topic for many years; countless research has been done in this area and numerous papers and books have been published. This chapter will cover the background of the distance training and the previous work done in this area.

2.1 Distance training overview

Education is the process of learning knowledge about concepts, principles, and truth. Training is different from education in that it concentrates more on gaining skills. There are two types of training, face-to-face training and distance training. Face-to-face training is a traditional training method. The teachers explain the skills in front of the students in the classroom with their body language, narration and handwriting. Distance training can be described as students and teachers who are physically separated and learning material, which is delivered through media; furthermore, the communications between the student and the teacher are via media [4]. Radio course exemplify distance training.

As technologies continuously improve, more and more employers and employees are willing to update their knowledge, skills, and degrees. Still full-time jobs, schedules, distances, and costs can confine going to universities or schools for face-to-face training. Therefore, distance training is becoming more and more popular throughout the world. Six different methods are available for distance training, they are radio, televised, videotape, computer based, CD-ROM
based and web based course. Each method has its own advantages and disadvantages. The strong point of a radio program lies in the low cost for a radio and its portability. However, when a radio program is adopted as a training method, the courses taught are normally focused on languages because in the lecture process, no pictures or diagrams can help the teachers to explain complex concepts. Meanwhile, the course can only be taken when it is being broadcast. A televised course is a better method of distance training than a radio program, because the content can be visualized. The diagrams are available and experiments can be presented during the classes to help the learners gain better understanding of the course contents. Compared with radio and televised courses, videotape, computer and CD-ROM based courses are not constrained by the program schedule. With a computer and CD-ROM, the training courses can be taken and reviewed whenever the learners have time and as many times as they want. At this point, Internet based training is the same as computer and CD-ROM based training. As the database utilization technology matures in the web based training, monitoring the learners’ behaviors is possible. Furthermore, updating course materials and managing users are easier than any of the other methods. Of course web based training still has drawbacks, such as it directly depends on bandwidth and the modems the learners use.

2.2 Online training overview

The expanding of new information and knowledge, the need of providing equal access to education and training opportunities, as well as the development of technology are the three main reasons for distance training. Radio courses,
videotape courses, televised courses, computer based courses, CD-ROM based courses and Internet based courses are all available in the current knowledge-expanding world for distance training. The Web based medium is the product of technological improvement and it is a powerful and popular distance training method in the world. Web based training is also referred to as online training, e-training, e-learning, and Internet based training. It can be delivered in either synchronous mode or asynchronous mode.

2.2.1 Synchronous online training

Synchronous, which literally means “at the same time,” is similar to face-to-face teaching except that the students listen to and respond to the teacher’s lecture through media. During the class the teacher presents course material through the Internet, the students ask questions and the teacher gives an answer immediately [5]. Telephone conversation belongs to synchronous online training; http://www.webex.com is an example of a synchronous training center as shown in Figure 2-1.
The Webex training center has been developed to integrate features of both traditional face-to-face teaching and distance training. All students need to have a username and password to access the training. The teachers can block a student or students to take a class, and talk to individual student or all students. The teachers are able to present their training content in the info area. Sharing application will allow the students to work on hands-on practices even though the software has not been installed in students’ local computers. Through participants & communication window, the persons who are in the class and the learners listening to the narration (using teleconference or voice-over IP) are dynamically displayed in all participants’ screens. The students can call the teachers’ attention by clicking on the button of Raise hand to ask questions or answer a question by clicking the feedback button [6]. Meanwhile, the learners can communicate with the teachers or other students through chat individually or
simultaneously. All these communications are done instantly and they make both the teachers and students feel like they were in the same classroom.

The drawbacks of this system lie on several points. First of all, the Webex plug-in needs to be installed in the user’s computer. Only by doing this, the students can see the course content correctly. Although this plug-in is free, it is not a commonly used plug in; thus some learners may not want to install it and the firewall of the Internet may prevent the downloading of this plug-in. Secondly, teachers can control the users’ desktops remotely by using sharing application. Some computers that belong to a government or a secret department with high security requirements would not allow any remote control. Thirdly, the students are confined by the course schedule; they have to stay in front of the computers when the teachers are presenting, and when one student is asking questions, all the others have to wait for him/her. So it is not a real self-paced study. Finally, after the learners log in, on the way to the course content, windows pop up one after another, and even during the presentation process, windows focus and disappear frequently. These windows confuse the learners. Actually, some of these drawbacks are the common negative parts of any synchronous online training. Therefore, Tinker web based training adopts an asynchronous training method.

2.2.2 Asynchronous online training

Asynchronous online training (also known as just-in-time and on-demand training method) is commonly used now. It is different from synchronous distance training in that the teachers do not deliver the learning material and
communicate with all students instantly [5]. To successfully implement asynchronous distance training, mails, emails, phone calls, or other means are needed for asking and answering questions. These additional communication tools are essential for the accomplishment of the course. As shown in Figure 2-2, http://www.vtc.com is an asynchronous training system developed by Virtual Training Company (VTC).

![Figure 2-2 Asynchronous online training example – VTC](image)

VCT online university is a valuable training resource on the web. The main purpose of this web based training university is to offer online software training for Mac computer. In this training some user-friendly features have been created. Like other web based training, all members need to log in the training system. Being an asynchronous learning method, the course can be viewed any time, any where with an Internet connection. The training material is grouped into modules; under each module level, a set of objectives is available. These
objectives are developed as QuickTime movies with narrations in the background. The movies are streamed, so the downloading time is minimized, and the audio is smooth and clear. The audio matches with the mouse action well. The buttons like rewind, stop, and volume control are provided for users’ convenience. There is no sequence requirement or constraint for the learning process.

Additionally a series of attractive features are under construction. An online test system that allows the lecturers to create tests and the learners to take the tests through their web browser is being developed. A management and reporting system about the course taken by the learners is being developed. Furthermore, a template that allows content designers to build learning material is under construction [7].

However, this online training still has shortcomings. It needs two plug-ins including VTC and QuickTime plug-in. Although QuickTime plug-in is a popular free plug-in, the VTC plug-in is not common. Another negative aspect is that the objective is recorded in QuickTime, which makes the movie performance slow. This may be due to the large file size and the bandwidth limit. The third downside is even with the rewind and stop buttons, it is hard to find a specific step taught previously. This is because the percentage or the time the movie has played is not available in the screen. The users are not able to get exact information about percentage or time played.

http://www.esafetyonline.com is another example for asynchronous online training. This training system provides employee safety training for
industry. It is composed of two parts, the server system and the presentation. The server system is developed to manage the user accounts and assign different training material to different company users. Test can be taken online and the result would be reported to the administrators of the training system [8]. The presentation is designed as Flash movies. Flash plug-in is the only necessary player for viewing this web site. The technical procedures and concepts are explained with animations, text and audio [8]. Interactive learning environment is created with fast rewind, rewind, pause, stop and play buttons as shown in Figure 2-3.

Figure 2-3 Asynchronous online training example – esafetyonline

The drawback of this training system is obvious. The biggest problem is the long downloading time. This disappoints the users. Another disadvantage is its text is hard to read for the users.
2.2.3 Summary

Through reviewing different online training systems, various advantages and disadvantages were identified as:

- Only single or double media types were employed to explain course content
- No simulations were available for users to practice
- 3D animation and simulation are not available to explain complex concepts
- The navigation system only allowed the users to view the course content step by step
- The training result could not be viewed by the course administrators via the Internet. In some cases only paper-based reports were available
- Few consider firewall issues
- Long downloading time

These issues invoke the primary thought of integrating multiple media types in a single learning environment with the support of database in the company of high security requirement and current bandwidth limitation.
Chapter 3 Tinker online system architecture design

3.1 Tinker online training system organization overview

Tinker AFB online training (www.tinker.ou.edu) is a web based engineering training system for both the people on and off Tinker AFB Internet LAN. It has two branches, C135 Cabin Pressurization Course (C135) and Environmental Management Training (EM). On the C135/EM introduction page, a brief preface about each course is given. The C135 Cabin Pressurization Course provides accurate pressurization maintenance procedures for C135 series aircraft. The Environmental Management Training describes the standard National Emission Standard for Hazardous Air Pollutants (NESHAP) requirements and environmental regulations. In the introduction page, authorized user name and password are needed to enter the training system. After the users successfully log into the system, the C135/EM welcome page comes up. Here, the training content of each section (C135 and EM) is first grouped into modules (module introduction page), and then subdivided into lessons (lesson description page); each lesson is composed of different numbers of objectives (Typical objective window), which are presented in a series of frames. The organization of the Tinker online training system is shown in Figure 3-1.
Figure 3-1 Tinker online training system structure
3.2 Tinker online training system architecture overview

Both C135 and EM are different groups at Tinker AFB, and thus they are two different branches in the Tinker AFB training system and their training materials delivery rules are different. Therefore, two independent systems are designed for Tinker online training. Despite being individual training systems, the system architecture is similar. The server system is composed of database and ASP files located at the server. Front-end system is composed of interface, navigation, media selection, and system organization. The layout of the system architecture is shown in Figure 3-2. Each topic will be discussed in detail in the following sections.

![Figure 3-2 System architecture](image-url)
The design of the videos is a solution to work with the Tinker firewall and it can be extended to other kind firewalls. The design of course material presentation style, shared script and cast is a practical way for developing constant interface, changing styles quickly and maintaining the web site easily. The design of navigation integrates database technology with the front-end system design and provides the users a flexible routing. Each of these issues will be discussed in this chapter.

3.3 Server system design

Each sever based training system is composed of a database and scripts. The database is the information center for each web based course. The scripts are responsible for transmitting data within database tables and between the users’ browser and the database tables. The design of the databases will be covered in this chapter. The scripts will be discussed in Chapter 4.

3.3.1 Why Database Management System (DBMS) is necessary

A database-management system consists of a collection of interrelated data and a set of programs to store, retrieve, and manipulate the data information conveniently and efficiently [9]. This is a centralized data management system. Compared with the file processing system, a DBMS stores data only in a prescribed format [10], which makes the data succinct and consistent. On creation, the data in DBMS is interconnected making it easy to access and retrieve [11]. This is one of DBMS’s most important benefits. Another significant feature for DBMS is it supports multiple users operating the data
simultaneously. With the primary key defined in the DBMS, the computer system can find the corresponding entity easily and maneuver the entity and the related entities exactly [9]. There are two kinds of DBMS composed of Relational Database Management System (RDBMS) and Object-based database system (ODBMS).

3.3.2 Overview of RDBMS

Even though both RDBMS and ODBMS are powerful DBMS, RDBMS is more traditional and occupies nearly 80% of all new developing database applications. It is composed of a series of tables with certain interrelationships by specifying match fields or columns; in other words, RDBMS creates the new table from the old one or relates to other tables with predefined columns or fields [11]. This characteristic makes the data sharing a reality. Another attribute of RDBMS is that the data in the table can be administered in various methods without changing the structure of the database [9]. Informix, Oracle, Sybase, SQL Server, MySQL, and Microsoft Access are the best known RDBMSs for database management system designing.

3.3.3 Chosen DBMS for Tinker Online Training System

The Tinker Training Database Management System was implemented using the SQL Server program by considering the following factors. First of all, it is designed and optimized for Windows NT in which the Tinker training system is installed. Secondly, SQL Server program databases’ capacity is large. SQL Server program can support as many as 32,767 different databases. About
two billion tables can be inserted into each database; furthermore 1,024 columns and unlimited rows can be defined for each table [11]. This feature leaves plenty of space to store the information for Tinker training material and also can be used to trace the users inside Tinker and outside Tinker. Thirdly, the SQL Server program database can process data fast and effectively because of its multiprocessing, multitasking, multithreading, multiarchitecture, and multiuser characteristics. Compared with Microsoft Access that can serve only 64 connections simultaneously, SQL Server program is capable for supporting 1,000 current connections. This feature fits Tinker’s large number of learners’ requirements well. Finally and most importantly, the SQL Server program database system can be easily transformed from the Microsoft Access database system, on which the Tinker DBMS was originally implemented.

3.3.4 Tinker online training database design process

Tinker Online training database is the control center of Tinker’s web based training. Various data is stored in this database; for instance, a user’s job information, user groups, lesson groups, module description, module names, module numbers, lesson descriptions and so on. Therefore, sorting and organization of this data can be done quickly and correctly. The design of this database consists of three steps, they are database requirements collecting and analysis, an Entity-Relationship (E-R) diagram creation and Table creation. The requirements of the management and course material have already been discussed in Chapter 1. The E-R diagram creation and table creation will be discussed in detail in the following sections.
3.3.4.1 E-R diagram creation

After Tinker requirements were analyzed, an E-R diagram was created for Tinker online training to show the logical relationship among the information stored in the database. The E-R diagram is used to convert real-life data and requirements conveniently into a conceptual logical database. This diagram is composed of entities, relationships and their attributions; in it the logical relationships among the entities are shown in detail.

An entity usually is a set of objects or persons, which share the same characteristics and exist in Tinker AFB. These objects’ and persons’ information are needed to be stored in database. For example, the Tinkeruser entity is composed of a group of Tinker users. Each entity has a set of attributes. An attribute is the property of an entity or a relationship. For example, the Module entity has three attributes, module number (\textit{mnum}), module name (\textit{mname}) and the maximum number of lessons in this module (\textit{maxlesson}). An attribute can either be an identifier or a descriptor. An identifier (also known as key or primary key) is used to identify the data stored in the table; therefore, its value has to be unique within the table. The descriptor is used to store the characteristics of an entity: duplicate values are allowed in the table. For example, \textit{username} is the key in the user entity: it is the identifier attribute for all users of this web based course, while two or more users may have same first name (\textit{fname}), as \textit{fname} is a descriptor attribute. There are two different types of entities: strong entity and weak entity (they are also termed as independent and dependent respectively). The difference between them lies in that the strong
entity has a primary key. It exists independently of other entities. But the weak entity does not have a primary key: its existence depends on some other entity. For instance, in Tinker’s web based training E-R diagram, the Module is a strong entity because module number (mnum) is its primary key; it can associate with the system by itself, while the lesson is a weak entity because it has to relate to module number (mnum) to connect to the system.

Relationship connects two or more entity types. For instance, an administrator in the Tinkeruser entity can post messages for other users in the system. In this case, the relationship between Tinkeruser entity and Announcement entity is posts. There are two types of relationships: strong (identifying) and weak (non-identifying) relationships. A strong relationship connects two strong entities together while the weak relationship ties to at least one weak entity. Userassigns and contains can serve as an example of strong and weak relationship in Tinker online training E-R diagram correspondingly. There are two terms usually related to a relationship, namely degree and connectivity. The degree of the relationship is the number of the entity associated with this relationship. The most commonly used relationships are unary, binary and ternary. In the Tinker online training database, binary relationships are mostly employed. The term connectivity is used to describe the relationship of classification, like one-to-one, one-to-many, many to one or many-to-many relationships. For example, a user “Jim” in Tinkeruser entity works for only one Department entity, like wiper, in the meanwhile one department like wiper can have more than one users, therefore, the connectivity between Tinkeruser entity
and Department entity is one to many. Connectivity has significant influence on the direction of a relationship. The direction always flows from the start entity (parent entity) to the end entity (child entity) in a binary relationship. In a one-to-one relationship, the direction goes from the strong entity to the weak entity. If both entities are strong, the direction is arbitrary. In one-to-many relationships, the entity occurring once is the start entity. The direction of many-to-many relationships is arbitrary. For instance, the direction of Tinkeruser and Department is arbitrary while the direction of Module and Lesson is one-to-many.

Figure 3-3 is the E-R Diagram for Tinker online training. In this E-R diagram rectangular, a double rectangular, oval, diamond and double diamond, line and dash line are employed to represent particular meaning. Table 3-1 is a list of explanation for all the notations used in the diagram.
Figure 3-3 E-R diagram
<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Strong entity</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Weak entity</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Strong relationship</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Weak relationship</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Attribute</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Key attribute for strong entity</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Key attribute for weak entity</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Connectivity</td>
<td>1/m</td>
</tr>
</tbody>
</table>

Table 3-1 Summary of notation in E-R diagram

3.3.4.2 Table creation

According to E-R diagram, the database table is created as shown in Table 3-2. States and Preferences are two tables without key attributes. This is because these two tables are used to store instead of store and retrieve data.
<table>
<thead>
<tr>
<th>Number</th>
<th>Table name</th>
<th>Attribute</th>
<th>Key attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tinkeruser</td>
<td>username, password, fname, lname, email, city, street, state, country, dname, phone, type zip gender, dob</td>
<td>username</td>
</tr>
<tr>
<td>2</td>
<td>Guest</td>
<td>username, password, fname, lname, email, organization, expirtyear, expirmonth, expirday</td>
<td>username</td>
</tr>
<tr>
<td>3</td>
<td>Userlist</td>
<td>adminuser, myuser</td>
<td>Adminuser, myuser</td>
</tr>
<tr>
<td>4</td>
<td>Belong</td>
<td>username, gname</td>
<td>username, gname</td>
</tr>
<tr>
<td>5</td>
<td>Groupinfo</td>
<td>gname</td>
<td>gname</td>
</tr>
<tr>
<td>6</td>
<td>Groupassign</td>
<td>gname, mnum, lessons</td>
<td>gname, mnum</td>
</tr>
<tr>
<td>7</td>
<td>Userassign</td>
<td>username, mnum, lessons</td>
<td>username, mnum</td>
</tr>
<tr>
<td>8</td>
<td>Department</td>
<td>dname, location, phone, division</td>
<td>dname</td>
</tr>
<tr>
<td>9</td>
<td>Tinkercompletes</td>
<td>username, mnum, Inum, score, time, date, viewed, complete, firstcomplete</td>
<td>username, mnum, Inum</td>
</tr>
<tr>
<td>10</td>
<td>Guestcompletes</td>
<td>username, mnum, Inum, score, time, date, viewed, complete, firstcomplete</td>
<td>username, mnum, Inum</td>
</tr>
<tr>
<td>11</td>
<td>Announcement</td>
<td>username, myorder, mydate, message</td>
<td>myorder</td>
</tr>
<tr>
<td>12</td>
<td>Module</td>
<td>mnum, mname, maxlesson</td>
<td>mnum</td>
</tr>
<tr>
<td>13</td>
<td>Lesson</td>
<td>mnum, Inum, displayorder, lname, maxobj, maxscore, intro</td>
<td>Mnum, Inum</td>
</tr>
<tr>
<td>14</td>
<td>Objective</td>
<td>mnum, Inum, onum, oname</td>
<td>mnum, Inum, onum</td>
</tr>
<tr>
<td>15</td>
<td>States</td>
<td>states</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Preferences</td>
<td>emailusermod, emailmanagermod, emailadminmod, emailuserlesson, emailmanagerlesson, emailadminlesson, changepassword, changephone, changeemail, lessonorder, expire, updatescore, guestregister</td>
<td></td>
</tr>
</tbody>
</table>

Table 3-2 Data structure

3.4 Front-end system design

Front-end system design is the interface, navigation and training material presentation design. Short downloading time, attractive visual appearance
template, easy navigation and lively content explanations are the goals of front-end system design [12].

Downloading time is seriously considered in the design process of this web site. According to the research about download time of various web sites and people’s interest, if a web page takes more than 10 seconds to download, learners will be disappointed and leave the web site [13]. Therefore various methods are employed to shorten the loading time, such as streaming technology and the organization of training materials. The implementation of technologies like audio and video streaming will be discussed in Chapter 4. The organization of the training material will be covered in this chapter.

The interface is one of the first factors the users notice. Therefore, it is especially important to create an attractive visual appearance while keeping it simple. A consistent visual appearance distinguishes the Tinker online training system web site from other web sites and makes it easy to navigate. In the design of the interface, system layout template and objective template are formed to facilitate new web page creations and original web page maintenance.

Easy navigation is an essential ingredient for the success of a web site. Due to more than 40 lessons available on this web site, a navigation system, which clearly contains marked destination information is needed. This information should include the following three aspects.

- Current location
- Previous location
- Further location
In order for easy browsing, two sets of navigation systems are utilized in this training system; they are system navigation and objective navigation. Both of them will be covered in this chapter.

To present the training content, 3D and 2D simulation, video, animation, narration, text and graphics are integrated in the training system. Also a web content presentation style is designed to dynamically control the arrangement of the training content. The training content, the media types used to present the training material, as well as, the web content presentation style will be discussed in this chapter.

3.4.1 Web page layout design

A successful web site is accompanied with an impressive web page layout. The web page layout is the arrangement of the navigation buttons and various media types like text, graphics and animations. The purpose of web page layout design is to make Tinker AFB online training flexible to navigation, easy to read, and consistent. The navigation will be discussed later in this chapter. As many aspects of facts may affect the layout of a web page, such as different monitor size, screen resolution, operation system and browser, clean and simple and consistent layouts are designed for Tinker AFB online training. In the web page layout design process, layout templates like module layout template are developed to set standardized Tinker web pages.
3.4.1.1 System layout template design

In the Tinker AFB e-training system, a user must move through five pages before the training material is viewed; they are Tinker introduction page (home page), C135/EM introduction page, C135/EM welcome page, Module introduction page and lesson description page as shown in Figure 3-1.

The home page is the Tinker introduction page located at http://tinker.ou.edu. It is composed of four parts, the top, the left, the content and the bottom part as shown in Figure 3-4. The top part consists of a top graphic and a top menu. The top menu will be discussed in the navigation section. The left part is a menu that will also be covered in the navigation section. The content part explains the two sections of Tinker’s AFB online training that is composed of C135 Cabin Pressurization Course (C135) and Tinker AFB training (EM). The bottom part is a background color.

Figure 3-4 System layout template
These four parts are the basic elements of the system layout template that is also used in the C135/EM introduction page, C135/EM welcome page, Module introduction page and Lesson description page. In the template, the position of the four parts as well as the graphic in the top and the bottom remain the same to keep the interface consistent and visually attractive, while the top menu, the left menu, and the content part change from page to page. The top and left menu provide flexible navigation and give clear instructions about the current page. The content part displays important information such as the module name, lesson name, lesson introduction and objective name that are loaded from the database.

3.4.1.2 Objective layout template design

As Internet learners, the users have full control of their computer. If the students click on a hyperlink or type an address in the address bar of the browser, it is easy to link the learners out of the training system. This is an issue no web based training developers want to see. Therefore, in the Tinker online training, the menu bar and the tool bar are turned off in a typical objective browser window serving as the training content display page. Additionally, the status bar is turned off because it releases messages when the learners’ mouse rollovers a button. As shown in Figure 3-5, this window is designed using an objective template to give a fixed online training presentation format. Furthermore, it provides the web site developer with a reusable and labor saving container to hold all the training media [14]. The objective template is composed
of a training material present area (the top blue area and the white area) and the navigation bar (the bottom blue area). The navigation bar will be discussed later.

![Figure 3-5 Typical objective browser window](image)

The training material present area is the place to show the training contents, which are explained with simulation, video, animation, narration, text and graphics. A brief overview of the particular objective is provided in the top blue area. The main course description is presented in the middle white area. In order to offer an active appearance for the training, a web content presentation style shown in Table 3-3 is developed. The purpose of this style is to avoid the repetition presentation format and. To make it easy for the website creation and maintenance, the position of all the media elements in different styles is dynamically controlled by Lingo script. This script is developed as shared script, which can be used in multiple objectives. The changes made in this shared script will affect every objective. With this shared script the objectives developed by
different people can achieve a constant interface without pay any efforts on the arrangement of media elements. The Lingo script is shown in Appendix A.

<table>
<thead>
<tr>
<th>Number</th>
<th>Describe</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Text only</td>
<td><img src="image1.png" alt="Example" /></td>
</tr>
<tr>
<td>2</td>
<td>One picture at center with only a line of text under the image</td>
<td><img src="image2.png" alt="Example" /></td>
</tr>
<tr>
<td>3</td>
<td>Two images side by side</td>
<td><img src="image3.png" alt="Example" /></td>
</tr>
<tr>
<td>4</td>
<td>Text on the left and two images at right</td>
<td><img src="image4.png" alt="Example" /></td>
</tr>
<tr>
<td>5</td>
<td>One image at the center</td>
<td><img src="image5.png" alt="Example" /></td>
</tr>
<tr>
<td>6</td>
<td>Text on the left and an image at right</td>
<td><img src="image6.png" alt="Example" /></td>
</tr>
<tr>
<td>7</td>
<td>Two images at the top and a line of text at the bottom</td>
<td><img src="image7.png" alt="Example" /></td>
</tr>
<tr>
<td>8</td>
<td>One image at the top and more than one line of text at the bottom</td>
<td><img src="image8.png" alt="Example" /></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><img src="image9.png" alt="Example" /></td>
</tr>
</tbody>
</table>

Table 3-3 Web content presentation style

37
3.4.2 Web content presentation design

The effectiveness of the web training is directly related to the level of interactivity, which can capture the learners’ attention and invoke their learning interest. Considering the low resolution of the users’ monitor, less text, more informative and active media types are employed for course presentation. In this training, the content presentation is composed of two parts: information frames and evaluation frames. Information frames are the frames that point out components, post precaution and describe procedures. The layout of these frames was discussed in the last section. Evaluation frames are used to measure the learning results and update the performance information to the database. Media types selection, which is the foundation of presentation design and the evaluation frames, will be covered in this section.

3.4.2.1 Media types selection

One of the advantages of using online training is its capacity to integrate many media types into the course. The challenge is how to overcome the bandwidth limit to provide responsive presentation. This problem can be partly solved by carefully selecting text, audio, simulation, animation, video and image to present the course content.

Text is an important media type in this web based training, and it is directly supported by the Macromedia Director. It is used for four purposes. First of all, it gives a comprehensive outline of a frame, and its highlighting is synchronized with the audio. By doing this, the learner should be able to
understand the main idea of this frame easily. Second, text serves as labels of parts. For example, in Cabin Pressure Control System operations description, in order to show a panoramic view of this device, text is used together with arrows to point out all the components. Third, it is employed as feedback for animations and simulations, so that the learners can learn the correct way. Finally, it is used for some questions to evaluate learning results and describes the correct answers. The result is passed to the database. This makes user tracking possible.

Audio is an essential item of this Internet based course and used in almost every frame. It is used to give a detail explanation of the concepts in a frame, complete instructions of a simulation or a brief summary of evaluation questions. The changing of tones in the narration emphases the important ideas in the frame and catches the attention of the learners. The audio is recorded in wav format, and then converted to swa format by Macromedia Director, later it is linked and streamed by Director to synchronize with other media elements.

Simulation is an informative and attractive part of the lessons. The main purpose of employing simulation in training is that it explains technical technologies by imitating the real life environment and situations. It also gives instant feedback to instruct the learners’ behaviors. Another advantage of using simulations is that the users can try different options without the risk of damaging costly equipments and components or getting some unexpected results. In these training lessons, some of the simulations are developed in 2D and others are created as 3D simulations.
As an example, 2D simulations are employed to give warnings and explanation of rules and procedures in the C135 aircraft maintenance and environment management. Some of the 2D simulations are even designed as games; such as hang man, matching game, Tic-tac-toe and Jeopardy. These simulations improve the learners’ interest and engage the students with the activity. Regardless if the simulation is designed as a game or as normal theory explanation, the goal of simulation is to clarify a technical concept. Normally, 2D simulations are developed in Macromedia Director and are fully controlled by the build-in Lingo script. Figure 3-6 is an example of a 2D simulation used to identify C135 Cabin Pressurization components. The Lingo script, which controls this simulation with respect to the users actions, is shown in Appendix B.

Figure 3-6 2D simulation example
One of the unique features of this online training is 3D simulation. Compared with 2D, 3D simulation better mimics the real world and is able to explain concepts more clearly. Figure 3-7 is an example of 3D simulation used to position the equipment and aircraft for a cabin pressurization test. The simulation requires the users to go around the 3D world, and identify objects and the functions of these objects. In this simulation, the aircraft and the human beings were first modeled in Eovia Carrara, then converted and imported into Director as a 3D cast member. The other objects are created by Lingo script shown in Appendix C. Lingo script is also used to control the movement of 3D aircraft and the other components.

![Figure 3-7 3D simulation example](image)

Animation is a key media since it can raise the interest and understanding of the user. It can also explain technical concepts clearly. Most animations were created in Macromedia Flash because its file size is small, and it is easy to
control in Macromedia Director with lingo script. An example is shown in Figure 3-8. In this frame, the different components and their labels are highlighted according to the audio to instruct the learner to identify the composition of the device. The Lingo script used to control the highlight of each component is shown in Appendix D.

![Figure 3-8 Animation example](image)

Video is integrated in the course because it can explain complex procedures with real and accurate graphics. In order to make it convenient for the learners to watch and review the video, Play, Stop, Rewind, and Slide buttons were created. The original video clip is obtained from the previous works recorded by Tinker AFB with digital video camera. The video was imported from the camera to a computer and stored in QuickTime format. QuickTime movies can stream themselves, yet the download time is still too long for web based training. In order to stream the video even more, QuickTime
movies are integrated into the Director file. Although each of QuickTime movies and Director shockwave movies can display correctly inside the Tinker firewall, the users inside the Tinker firewall will only see a red cross when QuickTime movie is embedded into the Director file. This is a problem caused by Tinker firewall. The firewall blocks QuickTime Xtra, which is a necessary programming for the QuickTime playing inside Director. The solution for this problem is to convert the QuickTime movie into a Flash movie. Details are explained in Chapter 4 video creation.

![Video Example](image)

Figure 3-9 Video example

Since images can add color and visual impact, as well as information to the web site, it is the most commonly used media type in the training course. The resolution of the images is restricted to 240x180 pixels to reduce the file size and to allow room to place two images on a single page. The source of the images was from clip art, videotapes, and CDs that were provided by Tinker AFB. After being edited in Photoshop, they were imported into Flash or Director.
3.4.2.2 Evaluation frame design

Normally an evaluation frame is a frame composed of only text using Single- or multiple-choice questions. These frames are important part for the system because it tracks the learners’ performance. After the learner finishes one objective, data such as last time to access, score, module number, lesson number and objective number as well as the other information like the user’s name, will be sent to the database located on the server. (The Lingo script – check answer and asp script – progress.asp are used to send data to a database as shown in Appendix E and Appendix F respectively). The user himself, a manager and an administrator can view the stored data in the system as shown in Figure 3-10. Each learner’s progress and performance is monitored. Another strong point of the evaluation frame is that by tracking a learner’s performance, the lesson can provide dynamic guidance on how best to process the lesson for the learner, meanwhile, assisting the administrator and the manager to evaluate the result of teaching effectiveness. The number of answers and the length of the questions and the answers dynamically control the position of each text line.
3.4.3 Training content organization

The course material for C135/EM is grouped as modules, which is further divided into lessons that are composed of one or more objectives. Furthermore, each objective consists of a numbers of frames. Figure 3-1 shows the structure of training material organization.

In C135 branch, there are four modules. The first module, C135 Cabin Pressure Control System Familiarization, focuses on the introduction of the system component and the tools required for C135 aircraft maintenance. The second module, C135 Fuselage Pressure Proof Testing, describes the preparations and procedures for 0 to 10.9 psi proof test in detail. The third module, Operational and Outflow Safety Valve Checkout Procedures for 0 to
9.42 psi, explains the purpose, coordination, precautions and procedures of operational checkouts. The last module, Cabin Pressure Control system malfunction Analysis, analyses the failure of cabin pressure control system operational checkout and repeats some breakdown operations.

In EM branch, six modules focus on the regulations about National Emission Standard for Hazardous Air Pollutants (NESHAP). Module one introduces the maintenance, record keeping, responsibility, operations, and procedures of aerospace NESHAP. Module two discusses the criteria of air pollutants and the way to limit air pollution. Module three covers the emissions, operations, maintenance, and record keeping of Halogenated NESHAP. Module four talks about maintenance and operations requirement for Chrome NESHP as well as the responsibility for compliance. Module five discusses Tinker AFB pollution prevention plan. Module six focuses on the recycling of solid waste in Tinker AFB.

### 3.4.4 Navigation design

Flexible navigation is one of the more important features of a successful training system. For this online training web site, navigation is designed to be simple and straightforward so that the Tinker personal pay more attention to the web content instead of trying to figure out confusing or convoluted road map.

#### 3.4.4.1 System navigation design

In system layout template, the top menu and the left part are for navigation. The top menu gives global tools and sources material for the
corresponding type of user. For example, Figure 3-11 is a C135 Lesson description page for an administrator and Figure 3-12 is C135 Module introduction page for a normal user. Comparing the two figures, it is easy to find that administrator button is shown for administrators and managers, while it does not show up for a normal user.

Figure 3-11 C135 Lesson description page as seen by an administrator user
The left part provides the routing to different pages. For example, after the users successfully log into C135 section, a pull down menu in the C135 welcome page provides the links to different modules. After a module is selected, separate lessons can be chosen in the module introduction page. Clicking on a lesson, the corresponding objectives are shown in the content part of the lesson description page. This can be seen in Figure 3-12.

### 3.4.4.2 Objective navigation design

Within an objective, the navigation buttons are located at the bottom of the window as shown in Figure 3-5. This navigation system is composed of three parts; they are the left part that is administrator buttons, the middle part that is location buttons, and the right part that is user buttons. All types of learners use
this navigation system, but it works differently for different types of users. Administrators and managers belong to the administrator group who can use all three parts of the buttons. However normal users and guests belong to a user group who can see and use the middle and right part buttons only. Different navigation buttons are designed for these two groups to provide special routings for each group users. The buttons in the middle is convenient for all types of users. The database can remember the objectives and frames, which the users have already viewed and the users can jump to these objectives or frames without listening to the narrations.

Figure 3-13 Objective navigation system

In order for administrators and managers to go through objectives quickly, back and forward administrator buttons are activated on the left continually as shown in Figure 3-14. The administrators and managers do not need to complete the narrations to activate these buttons. For normal users, these buttons do not show up.
The buttons in the middle provide much information. The numbers on the buttons show the users exactly how many frames in this particular objective and how many objectives in this lesson. For example, in Figure 3-14, there are a total of fourteen frames in this objective and two objectives in this lesson. The yellow buttons indicate the users the current location. In Figure 3-13, the user is in objective one, frame seven. The blue buttons indicate the frames and objectives that have been learned. The gray buttons mean that these frames and the objectives that have never been visited. The dimmed buttons can be turned to blue color only after the corresponding objective or frames are learned and the evaluation results are updated to the database. Figure 3-15 and Figure 3-16 show the difference between the first and the second time to visit the same objective. The user can go to any other objective or frame by clicking the blue buttons.
The buttons on the right are designed for step-by-step learners. The previous and repeat buttons are always activated. These buttons allow the learners to review the previous frames or the current frame anytime. The activation of next button is constrained by the completion of the narration. Before the audio is completed, it can not be activated. So the user is forced to hear the entire narrations. Figure 3-17 and Figure 3-18 show the next button before and after the completion of narration.
3.5 Requirements needed for web site browser

Regardless of the uncontrolled factors, there are two things that affect the smooth viewing of the training. The first is the plug-in. The second is the computer requirement, which includes the network.
3.5.1 Plug-in requirement

A plug-in is software that is installed in the users’ browser and permits the browser to employ corresponding functions that are embedded on the web site [3]. Web based Tinker AFB training system incorporates animations, simulations, video and audio to deliver training content with the support of multimedia technologies, such as Macromedia Director and Macromedia Flash. The Shockwave plug-in and the Flash plug-in are needed to display on the web site. These two plug-ins are developed by the Macromedia Company and can be downloaded for free.

For the convenience of the user, this training system provides plug-in detecting functions, which execute automatically when users log into the system. If the users lack either of these two plug-ins, a browser window will pop up (Figure 3-19) to remind the users to download and install the plug-in. If the users have already installed the two plug-ins, this popup window will not appear.

![Figure 3-19 Plug-in detecting window](image)

Figure 3-19 Plug-in detecting window
3.5.2 System requirement

To view the web based Tinker AFB training system, the system requirement including hardware and software constraints for the users’ computer and network is crucial because it determines the download time. Currently a desktop with 1.8 GHz CPU, 20 Gigabytes hard disk, and 256 Megabytes RAM is popular in the computer market. Video card, audio card, and speaker are available for all computers. The monitor color is up to 32 bit and its resolution can go as high as 1600 pixel x 1200 pixel. The commonly used network connection is 56Kbps modem or faster LAN. Windows 95, 98, 2000, NT4.0, and XP are the available operation systems. Internet Explore and Netscape Communicator are two main browsers. These settings are over qualified to browser Tinker online training system. Table 3-4 lists the recommended requirements for visiting this web site.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>400 MHz Intel Pentium II processor with a current hardware accelerated 3D graphics card with 8 MB or more of VRAM.</td>
</tr>
<tr>
<td>RAM</td>
<td>128MB or above</td>
</tr>
<tr>
<td>Hard disk</td>
<td>100 MB of available disk space</td>
</tr>
<tr>
<td>Monitor</td>
<td>16 bit color</td>
</tr>
<tr>
<td>Video card</td>
<td>available</td>
</tr>
<tr>
<td>Sound card</td>
<td>Windows compatible sound card</td>
</tr>
<tr>
<td>Speaker</td>
<td>available</td>
</tr>
<tr>
<td>Network connection</td>
<td>Modem: 56Kbps or higher</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>Operation system</td>
<td>Windows98 or higher</td>
</tr>
<tr>
<td>Browser</td>
<td>Internet explore 5.0 or above</td>
</tr>
<tr>
<td></td>
<td>Netscape communicator 4.76</td>
</tr>
<tr>
<td>Plug-in</td>
<td>Shockwave plug-in</td>
</tr>
<tr>
<td></td>
<td>Flash plug-in</td>
</tr>
</tbody>
</table>

Table 3-4 Recommended system requirements
Chapter 4 Tinker online training implementation

In order to implement the Tinker online training system, the following three issues were considered: the security issue, interactive course presentation format and download speed. With the intention of providing a safe Internet based training course, website content protection is provided. With the purpose of delivering active web content, software such as Active Server Pages, Macromedia Director, Macromedia Flash, are employed. To reduce download time, many techniques like streaming are applied. In this chapter, the implementation of security protection and the selection of technology will be discussed.

4.1 Security protection

As mentioned previously, the Tinker server is located inside Tinker’s firewall, which works twenty-four hours a day. A second method to protect the training system is turn off the status bar of a typical objective window. This denies hackers get information about the file name executed and the parameters transmitted. The third manner utilized in this system is implement web site content security. The purpose of implement site content security is to address the identities of the users and the courses taken. User accounts and user groups are the two parts of web site content protection. Each part will be discussed in detail below. The fourth mode is control of the traffic. This will be discussed in User group security and management section.
4.1.1 User account security

To access Tinker AFB online training system, a unique user name and password are needed. Tinker server access is controlled by a user name and password authentication. Administrators and managers are the only people responsible for filling out a New User Registration form shown in Figure 4-1. By doing so, a new user name and password are assigned to allow access to the http://tinker.ou.edu web site. All users are assigned to one or multiple departments.

Figure 4-1 New user registration form

A unique feature in EM training is that it does not only allow administrators and managers to create new users’ account, but also permits guests to apply for a guest account via Internet in an unsecured section of
http://tinker.ou.edu/em. This feature can be turned on and off easily by administrators, so that only Tinker AFB authorized users can see the content.

The guest registration form is shown in Figure 4-2.

![Guest Registration Form](image)

*Figure 4-2 Guest registration form*

The difference between the user and the guest registration is that Tinker users (normal users) must be in a user type (supervisor or user), lesson group (Admin, Manager, painter, wiper and etc..) and directorate.

### 4.1.2 User group security and management

In the Tinker online training system, different authorized users are restricted to certain lessons within the web site. Two types of users are defined in the training system. They are supervisors including all administrators and managers, and users including all normal users and guests. The user group and their relationships are shown in Figure 4-3. The administrators can manage all
the administrators, managers, normal users and guests; managers can administer all managers, normal users in their group; normal users and guests have no control of others.

![Diagram of user group relationship]

**Figure 4-3 User group relationship**

### 4.1.2.1 Administrator features

Administrators have full control to use any of the following eleven features in the training system. Some of these features are shown in Figure 4-4. All of these features can be fully controlled via a web page. Any administrator with no database and programming knowledge can load or sent data to or from database without opening database tables. This makes the life of administrators and managers easier.
The “Announcement” link allows the posting of announcements in the training system, and anyone who accesses the C135/EM welcome page can read the current news.

The “Preferences” link allows the changes of the system options for all the users. Options, for example, the system automatically sends an email to the users who have finished a module, can be managed by the administrator. The sending email feature is implemented for the purpose of encouraging the learners. This option can be disabled if more learners do not want to get such emails. The objectives display order modification option gives the opportunity of controlling the learning sequence to the managers. An important protection method is developed here – the administrator can turn the guest registration on and off. This setting can block the non-Tinker user from entering the training system for security and traffic reasons. Another method to reduce the traffic is to minimize guest expiration days. The control of guests’ registrations and
expiration days are done for reasons of both security and management. Some of the system options can be seen in Figure 4-5.

“Manage Group” allows the managers to create multiple lesson groups. Each lesson group includes multiple lessons. Once a lesson group is created, it can be assigned to users. This can avoid the situation of assigning multiple individual lessons to each user. A lesson group can be created by assigning a group a name, and then selecting lessons by checking the check box shown in Figure 4-6. An existing lesson group can be edited and deleted by clicking on Group Assignment and Delete Group buttons.
“Manage Directorate” is implemented to provide the opportunity of grouping users according to the directorate. The users in the same directorate can then be manipulated together. With this feature, a directorate can be created by providing the information shown in Figure 4-7. Also a directorate can be edited and deleted.
“Manage User” is developed for the administrators to manage users. When a new user is created, directorate, user type, lesson group, user name, and password are assigned to the user as shown in Figure 4-1. By assigning the user a user type, a user’s control ability to this training system is decided. By assigning a or multiple lesson groups to a user, the lessons that can be viewed by this user is determined. With a unique user name and password, a user can access to the training. Since this feature is provided for the administrator instead of normal user, the options like directorate, user type, and lesson group can not be changed by the normal user himself. In order to convenient the administrators to edit users, three methods are provided including name, lesson group and directorate.

“User Progress” provides the link to monitor all users’ performance. Thus, this training system can be surveyed and future improvement decision can be made accordingly. Figure 4-8 is a user progress list sorted by user’s name. In this list, all users including managers, normal users and guests are listed. If the administrator clicks on a user link, for example, 1234, the corresponding user’s progress page shown in Figure 3-10 appears. Directorate and lesson group can also sort the user progress list. In Figure 3-10, the administrator can gain users’ performance and progress information; such as, user names, the score, the time spent in each objective, last dates to access each objective, the completed objective (green color) and the uncompleted objective (blue color).
“User List” link provides easy access to view or modify user information. If administrators want to monitor several users, they can add them to the list. Once the users are added, they can be accessed directly in the future without necessary of finding them each time.

“Manage Guest” link allows the administrators to edit guest information, view guest progress, and delete guests. A guest’s account can be changed to a user’s account.

“Help” section is provided for the administrators to master the management tools, so that the management process is easy and fast. The normal users and guests cannot view this help section.

Administrators have the ability of access all objectives nonlinearly. This can help them to find the course objective quickly. Administrators can select any objective in any module and lesson anytime without going through other
modules, lessons or objectives. The red check mark shown in Figure 4-9 indicates the activated objectives. For administrator accounts, the check marks are always activated no matter if this objective has been viewed or not.

Figure 4-9 Objective description page for administrator

In each objective, Administrator accounts can go through it without necessary to listen to the narration and answering the evaluation question. Thus, the frames they want to review can be quickly found.

4.1.2.2 Manager features

Managers have less administrative control when compared to administrators. Managers can use the above eleven items for the normal users in their department and all guests, but no changes related to administrators can be made by managers. In other words, managers have restricted power over the training system.
4.1.2.3 Normal user features

Normal users are the main accounts in the training system. Administrators or managers assign a single lesson group or multiple lesson groups to them. This limits normal users access to only lessons assigned to that group when they log in and they have to study those lessons and objectives in sequence. For instance, in Figure 4-10 the user has completed objective 1, so a check mark appears beside objective 1. At this time, the system will respond to the user’s click on objective 2 only. In this case, lesson 2 button is dimmed out, so the user has to finish lesson 1 before going to lesson 2.

![Figure 4-10 Objective description page for normal user](image)

As mentioned in 3.4.4.2, in each objective, the administrator buttons do not show up for normal users, so users are forced to go through all the material in the training course. When one objective is completed, data about the score, question number, module number, lesson number, objective number, user name,
cumulated time (in minutes) spent in the objective and last access date is updated to the database. Only in this way can the check mark beside the objective show up. The start button in Figure 4-10 takes the users to the last uncompleted objective.

4.1.2.4 Guest features

Guests are the fourth type of users in the training system. They do not belong to any of the lesson groups and they have to view all the modules sequentially. As one objective is completed, data about the score, question number, module number, lesson number, objective number, user name, cumulated time (in minutes) spent in the objective and last accessed date will be updated to the database. As guests, their accounts will expire after the duration set by administrators or managers in Preferences passed. If an account is expired, all information related to the particular guest will be deleted from the system. Administrators or managers can change a guest’s account into another type of account like administrator, manager or normal user.

4.2 Technology selection

With the rapid progress of technology, more and more software are suitable for web design and implementation has also grown. Thus the importance of selecting proper tools to develop a web site is increasing. In order to implement Tinker online class, Active Server Pages (ASP) and JavaScript serve as the communication language between the database and the web browser.
Macromedia Director, Adobe Photoshop and other software are used to create content.

4.2.1 Active Server Pages

Active Server Pages (ASP), which is developed by the Microsoft Company to enhance the capacity of writing web pages, is a server side text scripting technology. It is the combination of programmatic code and embedded HTML. The programming code can be written in a number of scripting languages like VBScript, Jscript, PerlScript, and Python. In Tinker training, VBScript is used to generate the dynamic web page. The function defined with VBScript is placed outside the HTML tag. The embedded HTML defines the static the web page [15]. ASP is employed extensively in this system because of its security, dynamics, browser independence and cookie setting features.

The ASP code resides and executes on the server. From the view source menu command of the browser, the user is able to see the running result of ASP files and the static HTML code. In this code, all the functions defined outside the HTML tag can not be displayed on the screen. So it is difficult to get information to hack this online class.

ASP generates a customized web page for individual Tinker training users. The process for viewing a web site is composed of the following five steps. First, the user calls for Tinker web page by typing the address or clicking on the hyperlink in the browser. Second, the web page calls the ASP files on the server, such as login.asp or load_objective.asp. Third, the server processes a set of programmatic code in ASP files, and retrieves the necessary data from the
database. Fourth, ASP embeds process results, such as data from the database, in HTML. Finally, the HTML page displays in the browser [15]. These steps are shown in Figure 4-11. In this procedure, the second and the third step are executed according to the users’ input in the first step, and the display page is the result page. For example, when the user 1234 visits lesson description page of module 2, the display page is different as shown in Figure 4-12 and Figure 4-13 only because the user specified two different lessons – lesson 1 or lesson 2.

Figure 4-11 Tinker web site display process
One of the unique features in the training system is the ability to monitor users. ASP implements this feature by using cookies and updated information to the database. A cookie is a small piece of information that is stored in the client’s hard drive [16]. When the user logs into the system in C135/EM introduction page as shown in Figure 4-14 and Figure 4-15, the cookie that contains user name, user type, and time of access are saved on the computer via the web browser. The cookie passes this information from page to page, until the objective page and then updates this information to the database. If the cookie is accepted, this process can be completed. The group managers and the administrators can see the users’ progress whenever necessary. Even after the users visit other web sites or shut down their computer and returned to the training some days later.
Another benefit of using ASP is its browser independence feature. Since it runs on the server instead of the user’s browser, browser incompatibility will not affect the running result of ASP files.

4.2.2 VBScript

Microsoft Visual Basic Scripting Edition (VBScript) is a subset of Microsoft Visual Basic Programming Language but is much easier than Visual Basic to learn. It can be used as scripting language for either ASP programmatic code to create a dynamic HTML page in the server side or to manipulate Microsoft Internet Explore (IE) for the client’s browser [17][18]. In this Internet based training system, VBScript is mainly used as the script environment for ASP. VBScript is the default language for ASP to work with Microsoft Internet Information Server (IIS), which provides run-time support for VBScript. It can check errors in the scripting while viewing the web page, which makes it easier for web developers to debug. Another purpose of employing VBScript is that it
is easy to learn and use. Furthermore, plenty of reference and learning materials are available on the web.

Although IE supports VBScript, Netscape Navigator does not. Since IE and Netscape are both required to operate with the Tinker system, another scripting language, JavaScript, is employed to serve as the client side scripting language.

### 4.2.3 JavaScript

JavaScript is a scripting language developed by Netscape Communications Corporation to create interactive HTML pages. It can open a pop up window with the click of a mouse, display information based on time, and can change graphics on the web page. It is embedded directly in the HTML page. Unlike VBScript, JavaScript is supported by both Internet Explore and Netscape, and this is the main reason to employ JavaScript to specify different kinds of pop up windows in this training for learning convenience.

### 4.2.4 Macromedia Director

Developed by the Macromedia Company, Director is one of the most commonly used authoring tools to develop sophisticated and interactive presentations for online training course. For the Tinker training system, it is employed as a container to hold the lesson content. The decision to use this multimedia tool is because of its strong scripting language, numerous media type support, powerful compression ability, performance optimization, and attractive 3D features.
4.2.4.1 Lingo programming

Lingo is the object-oriented programming language incorporated in Director, which controls the playback, the media elements’ position, streaming, animation, simulation, and the variables passing in the Director movie or to the ASP file in this course [19][20][21].

In order to provide convenience for users, nonlinear navigation is adapted in this training. Lingo is the script language used to manage the navigation system in objective window with respect to a user’s mouse click and completion of the navigation.

With the purpose of providing a time and labor saving presentation format, as well as easy maintenance for the future, lingo script is used to dynamically decide the position of each element in the scene. For example, in the evaluation frame, the white area between the lines in the frame that has two answers and four answers, is different so that all the lines have equal space, as shown in Figure 4-16 and Figure 4-17. It is fully driven by lingo script.
4.2.4.2 Rich media

Director supports bitmap, vector, shape, and real media directly, and its development environment can be extended to support gif, jpeg, text, sound, Flash movie, QuickTime movie, AVI video, RealAudio, RealVideo, Shockwave 3D and many other media types by including specific Xtras. In order to develop an interactive and rich training content web site, text, Flash movie, sound, and
Shockwave 3D are employed to create text, animation, narration, simulation, and video.

Text is a media type supported directly by Director and works well with lingo. The background color, width, font, font size, alignment, and color of the text are fully controlled by frame lingo script. By doing this, modification becomes easy, because changes to the script will affect all the texts and keep them consistent.

Flash movie is a media integrated into Director. There are many reasons for employing Flash movie. When comparing the file size of an image with the same quality, Flash is smaller than bitmap or jpeg, as shown in Table 4-1. Flash movie can be scaled without a significant influence on the image quality, which is convenient for web developers to fit movies into a given space. The animation created in Flash displays correctly in Director. With several lines of lingo code, a Flash animation can be easily controlled in Director, dramatically saving time of creating the Flash movie. The most important feature for the Flash movie is that it can be streamed to reduce the movie downloading time.

<table>
<thead>
<tr>
<th></th>
<th>Flash movie</th>
<th>Bitmap</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>File size</td>
<td>75.4k</td>
<td>2560k</td>
<td>153k</td>
</tr>
</tbody>
</table>

Table 4-1 File size comparison

Although Director supports various formats of sound, swa format is employed in this training because of its small file size. Since the sound is long and increases the file size of a the Director movie, it is linked to the Director file as an external sound. Meanwhile, this format of external sound can be easily
controlled by lingo script with the response of a user’s click or the completion of the sound. Furthermore, swa format external sound can be streamed frame by frame while the Shockwave file is playing, so the sound does not need to be completely loaded into the memory of the user’s computer. All of these procedures minimize the time period before the training content appears on the screen.

4.2.4.3 3D feature

Shockwave 3D is a brand new feature for Director version 8.5. Simple 3D models, like equipment carts and cabin pressure tester are created with lingo script in director to minimize file size. Complex 3D models can be originated in many popular 3D development platforms, including Eovia Carrara, 3D Studio Max, Alias|Wavefront, Ashlar-Vellum, Caligari, D Vision Works, MAXON Computer, NewTek, TEALVIZ and Softimage/Avid. The 3D model of a C135 aircraft and the human beings were developed in Carrara. Lingo scripts are attached to the sophisticated 3D models to control the movement and feedback of a user’s input behavior. By adding 3D models to e-learning systems, real life is simulated on screen.

4.2.4.4 Compression capacity

Compression and decompression, also referred to as CODEC, can be defined as the process to minimize storing or transmitting space by eliminating redundant information. Although this can be done with both hardware and software, the software method is employed because of its easy and inexpensive.
Director provides compression for a movie, including audio and image compression, to keep the file size deliverable for low bandwidth network.

Audio is an essential part of the training and it is expensive in terms of download time. The sound is first recorded in wav format, where the file size for a thirty-eight second narration is 1.61 megabytes. Due to the undeliverable wav format file size, the wav file sound is converted to swa format, and the compression ratio can be as high as 20:1.

Images are also an important part of this online course. Image compression takes effect when publishing a Director movie. The compression percentage of JPEG can be set by the developers at 80% for all cast members and the entire movie to balance the image quality and file size.

4.2.4.5 Performance optimization

Many factors affect the performance of this training course including hardware, software, and network. Processor, cache, RAM, virtual memory, disk space, bus speed, video card, sound card, and speakers are the main considerations for hardware; operating system, extension, drivers, display settings, and browser are the considerations for software; and modem, network card, internet, and servers are the network considerations [19]. In spite of so many issues, those that can be controlled by web developers are limited. In order to improve the performance, shared cast, playback option, audio streaming and video streaming are incorporated in this online education course.
4.2.4.5.1 Shared cast

Shared cast is a group of cast members commonly used for all objectives, including the buttons, icons, and scripts. The shared cast is downloaded to the cache of a learner’s computer and this fifty-kilobyte file can be reused by every objective, without downloading repeatedly. It dramatically saves loading time. Another advantage of using shared cast is that once a modification is done in shared cast, it will affect all the objectives. For example, currently the width of the text is set at 420 pixel for a text only frame. This is controlled by the behavior start, as shown in Appendix H. If the width is changed to 400 pixel, the space between the lines will be recalculated accordingly and the width of the text in each of the frames with text only format will be set to 400 pixel. This will keep virtual appearance consistent and save time in making changes.

4.2.4.5.2 Movie playback option

Downloading an entire movie means loading the entire movie to the memory of a local computer. This always involves large data loading, long period of time waiting, and slow performance. To avoid these issues, Shockwave movies are streamed. In other words, users only need to wait for the downloading of the first thirty frames before the movie starts to play. Although the number, thirty frames, can be changed to any other number of frames, depending on the movie size, it is set to as few as possible because it relates directly to downloading time. The rest of frames of the movie will be loaded as the playback goes to the frame. This method splits an entire download movie time into small pieces of loading time and decreases downloading time.
4.2.4.5.3 Video streaming

Various issues affect the optimization of the download speed. Small file size will definitely accelerate the loading speed, while another notable choice is streaming the movie. Streaming is a technology that downloads a movie via the Internet while the movie is displayed [19].

Video is always related to large file size, sometime more than 2 megabytes for a twenty-second movie. Therefore, instead of embedding the video in Director, it is linked as an external cast member. It will start to download when the movie is accessed, and when the movie starts to play it is still downloading without affecting the playing of the movie. When the movie goes to the video frame, the download of the movie has already finished.

4.2.4.5.4 Audio streaming

Despite the audio compression in Director, the file size is still too large for low bandwidth, thus sound streaming is incorporated into these lessons. The implementation of sound streaming is fully controlled by Lingo script.

As audio is used in every frame, the narration is subdivided into topics corresponding to the text shown on the screen. The sound is then downloaded to a learner’s computer just before frame is viewed.

4.2.5 Macromedia Flash

Macromedia Flash is a popular authoring tool for web site design. It creates vector-based drawing by defining graphics using straight or curved lines with color fills. This results in smaller file size and vector files can easily be
enlarged without degrading quality. Flash supports rich media types like bitmap, Windows Metafile (WMF), JPEG, and so on. Thus, rich source format file can be imported into Flash and converted to Flash format movie. The embedded scripting language, Action Script, dynamically manages the animation with respect to a user’s action. In this training system, Flash is used mainly for three purposes, including compression of images, creating video and developing animations.

4.2.5.1 Video creation

As mentioned in the previous chapter, video is employed to explain complex technology with the real facts. However the long downloading time prevents video from being broadly applied on web courses. Therefore, several technologies are integrated in the Tinker training system to solve this problem.

Large file size is undeliverable for web based training. So in Tinker training, the original video clip is recorded QuickTime (mov) format. There are three methods that can be used to deliver the QuickTime video. The first is linking the video as an independent web page. The second is implanting the video into Director as an internal cast member. The third is embedding the video into Director and linking it as an external cast member. By using the first method, the web page that contains the video will not start to stream until the users enters that page. Therefore, it is discarded because it takes long to load the movie. Since the second method causes the entire Director movie file size to be large, it is not feasible. The third method is used because the file size remains small and the video is streamed in the previous frames. Although the third
method works fine when the training content is tested in Engineering Media Lab (EML), Tinker officials could not view the videos. The problem lies in the Xtra needed by Director to play QuickTime video. This Xtra is small software that can be embedded in Shockwave plug-in. It should be downloaded from the Macromedia Company’s web site before the video start to play, however, the Tinker firewall prevents its downloading.

The solution to this problem is converting QuickTime video to Flash video. Although the file size for Flash video is almost twice the size of QuickTime video, it displays smoothly in this training because double streaming technologies are utilized. The first streaming technology: the scripting language, Action Script, can stream the Flash video in Flash movie. In this training, the script streams the video 80 percent of the total Flash video frame number. The second streaming technology, Director, starts the download of Flash video from the beginning of the Shockwave file download. This was mentioned in section 4.2.4.5.3.

4.2.5.2 Animation development

Animation development is an important application of Flash. Flash is utilized to implement animation development because it has a set of powerful tools that makes the animation building process easy. For example, normally the media in Flash movie will appear in sequence according to the main timeline. However, if Action Script is applied to the symbols, frames, buttons, then the programming will control the display of the movie.
Another benefit of employing Flash as the authoring tool is the easy communication between Flash and Director [20][22][23]. In the case of learning the components of a C135 aircraft, an animation is created. In this animation, 10 symbols like `to_outside_air`, `low_safty_connection` and `upper_chamber_line` are created in the Flash movie. These symbols can be called by Lingo script in Director with the commands shown in Appendix D. In these commands, symbols whose transparence is set as 100 percent, will be highlighted, while the symbols with 30 percent transparency will dim out, as shown in Figure 4-18.

Figure 4-18 An example of Lingo controlled animation

### 4.2.5.3 Image compression

Images are useful in training because of visual information. Image format is the first concern to adding pictures to the course content because the format chosen determines the file size and the download time. The decision to use Flash depends on its double compression abilities. For instance, when a bitmap image
is imported into Flash and exported as Flash format with a *swf* extension, the file size will be smaller than the original bitmap format. When the Flash file is imported to Director and published as Shockwave with a *dcr* extension, the size of the *dcr* file will be less than the *swf* file.

### 4.2.6 Macromedia Freehand

Macromedia Freehand provides a set of tools for creating diagrams and illustrations, which can be integrated into Flash. The purpose of utilizing Freehand is due in part being a flexible drawing tool. The graphic shown in Figure 4-18 is a complex cut view of an upper chamber with straight lines, curves, and fills. The connection between the lines and the curves, lines and lines, or curves and curves is difficult to line up perfectly in other software, even though they have alignment tools. However, with the Pen Tool provided in Freehand, two points can be easily snapped together. Since the graphic drawn in Freehand is a vector based graphic, small file size is gained. Meanwhile, scaling the graphic will not dramatically impact the graphic quality. A convenient feature of this illustration builder is that when all the lines, curves, and the fills created in Freehand are copied and pasted into Flash, all the elements in the same layer created in Freehand will be grouped together as one object in Flash. This makes the symbol defining process simple and time saving.
4.2.7 Adobe Photoshop

Adobe Photoshop is a professional image editing software. There are three main issues concerning the use of Photoshop within this research. They are image conversion ability, size reduction capacity, and editing functions.

As mentioned in Chapter 3, the source images were obtained from clip art, videos, scanned images, digital cameras and Tinker CDs. Various formats of these images are included, such as wmf, JPEG, and bitmap. However, the format that Flash and Director can highly compress is bitmap. Thus, a format converter is needed. As Photoshop can easily convert image from one format to another one, for example, from gif, JPEG to bitmap, it is selected to realize the conversion.

Since the prototypes of the images for this training are from different kinds of sources, the physical size of the images are varied from 1024 pixel x 768 pixel to 300 pixel x 256 pixel, and their file size are varied from 1.5 megabytes to 100 kilobytes. These file sizes are unacceptable for the Tinker training content style and are undeliverable for many type of network connections. If these images are resized in Flash or Director, the quality of the images will drop, yet the file size will still remain unchanged. Therefore, Photoshop is used to reduce the physical size and the file size of the image.

Although the resolution of a monitor is less than a printed copy, the images shown on the screen have to be clear. Some of the source images for this training were not good quality due to poor lighting, distance to the object, and contrast between the object and the background. Thus, Photoshop is selected to
serve as a powerful image-editing tool to sharpen the image, and increase both brightness and contrast by selecting a part of the image or adjusting the level or contrast of the image.

4.2.8 Adobe Premiere

Adobe Premiere is the standard video-editing software in the multimedia world. It can deal with video clips from Vertical Helix Scan (VHS) tape, Digital Video (DV) camera, and Digital Versatile Disk (DVD). Premiere is used in this training because of the following four reasons. First, Premiere can capture videos easily. A simple import feature can bring the recorded video into Premiere directly. Second, convenient editing tools are embedded in Premiere, including cut and compress. For example, a 500 Megabytes video can be reduced to 200 kilobytes by doing the following steps. First, the video is cut into small pieces, and then all the unnecessary parts of the video are removed from the timeline. Afterwards, the movie is exported as a self-container video. The third step is important because by doing this the new video will be independent to the original video. Fourth, Premiere can reduce file size. For instance, in this training, MPEG videos are converted to QuickTime Video. Meanwhile, some options like Compressor, Frame Rate, Frame Size, and Quality can be chosen to minimize the file size. Finally, Premiere can export movies not only as video clips but also as frame images in a variety of formats. This feature makes exporting the images obtained from video simple and effective.
4.2.9 Eovia Carrara

Eovia Carrara is a good, all purpose 3D modeling program. The decision of employing Carrara as the 3D C135 aircraft modeling tool is due to many facts. Carrara is a stable and solid application tool that would not crash due to long time of application or large file size. Its user-friendly interface concentrates the user on the function of the software instead of figuring out the position of the buttons. Spline and Vertex modelers are the two methods to construct complex models with substantial features. Carrara supports many export format, such as WaveFront (obj), 3D Studio (3ds), VRML (wrl) and Carrara File (car). Obj format model can be converted directly to w3d format by Director. This w3d is the only 3D model format supported by Director. This makes the cross platform application easy.
Chapter 5 Summary and conclusion

5.1 Summary

Web based engineering training tends to be the mainstream of distance learning because it is easy to access, low cost, self-paced, facilitates updating course material, nearly unlimited simultaneous learners, and user tracking. The delivered web based Tinker training is created to train Tinker AFB personnel in two topics: C135 Cabin Pressurization Course and Environmental Management Training. C135 explains the process of testing the integrity of a fuselage and ensures that the cabin pressurization system operates properly for C135 aircraft. EM provides detail instruction of environmental management. Each of the two parts of this training system is composed of a server system and a front-end delivery system.

The server system consists of an online database and ASP scripts. The online database is created to dynamically manage the course material, user information, user accounts, and user group accounts. This database has been implemented using the Microsoft SQL Server. ASP scripts exchange data between a user’s browser and database table or among database tables. These scripts execute on the server increasing the chance of web security.

The front-end system is composed of an interface, navigation, and training material presentation. The interface is designed to be attractive to keep the learners interested in the training. The navigation is simple, straightforward and constrained, so that a user has to learn the content in a predefined sequence.
Text, animation, audio, video, 2D and 3D simulation are selected as the media types to explain the technologies and concepts. Flash video is developed to pass the Tinker firewall and correctly display. The presentation is fitted in a presentation style to provide an active learning environment and the arrangement of each element is dynamically controlled by Lingo script. Director, Flash, Freehand, Photoshop, Premiere, and Carrara are employed to prepare the media elements for the course content. Compression and streaming technologies integrated to overcome bandwidth limitation. JavaScript is used to develop pop up windows.

In order to meet Tinker’s high security requirement, the Tinker server is located inside the Tinker firewall. User accounts and user group accounts are set up to protect the training content. An administrator or manager can manage the system via the Tinker online training system without opening the database table. Administrators and managers of the system have higher control ability than the user group people. Management tools such as user progress, lesson group, and preference are developed to make it easy to control the training system via Tinker web site. An unconstrained navigation system is provided for administrator and manager users. Basic users, who are composed of normal users and guests, are required to view the training content in sequence.

This research contributed to the following:

- User account was developed to protect the training content.
- User group was developed to manage learners and training material throughout the web site.
• Extendable database was created for learner tracking and future expansion of this training system.

• Flash video was incorporated in a single interactive learning environment with text, audio, simulation, animation, and image without conflict with Tinker’s firewall.

• Three-dimensional simulation was integrated in the training in order to clearly explain technologies and concepts.

• Shared cast technology was employed to share common source material, keep consistency and reduce file size.

• Various image, video, audio, Flash, Shockwave compression and streaming technologies were incorporated.

• Reusable Lingo scripts were created for future course material control.

• Templates were designed for objective development to provide a reusable presentation format for future training content development.

• Template layout design provided a user-friendly interface.

• Navigation systems were formed to provide learners with a constrained, yet convenient, learning environment.

5.2 Future work

After two years research, the Tinker online training system was become fairly powerful. However, there are still several research topics that can improve the performance of this training system. A few recommendations are proposed for future Tinker web training development.
Currently, more than 40 lessons have been developed in this training, and as the training system expands, more lessons will be added to it. As a result, administrators and managers have a hard time remembering all lesson names and the lesson content. Therefore, a search engine is recommended for quick searching of data in the database.

By now, flexible navigation has been implemented in the typical objective window, and the highlighted frame and objective buttons allow learners to jump to frames and objectives without any constraint. However, these buttons permit jumps to recent lessons only. An improved navigation that provides jumps among different lessons and different modules would serve learners better.

Although one 3D simulation is integrated into this training to imitate real life, most simulations and animations are developed in a 2D world. These media are not in a stereoscopic world and limit the mastery of complex technology and concepts. Therefore, more 3D simulations and animations are suggested for the future development of training content.

In this training the activated blue button reveals how many frames have been played in the objective, but the percentage of audio completed is still unknown to the user. Therefore, an indication button for the percentage of audio played in each frame is suggested.

When a user name and password are transferring though the Internet, hackers may be able to get this data, and go into this training. Thus, a user name and password encryption system is advised to avoid password stealing.
Appendix A Script for media element position controlling

Global StepNo, totalNo, LineNo, styleNo, totalLineNo
Global gHiliteColor
global currentNumber
global textHeightList, textWidth, startPositionX, startPositionY, sceneHeight

on beginsprite me
    startPositionX = 0
    startPositionY = 0
    textWidth = 0
    StepNo = word 2 of the framellabel
    totalNo = word 3 of the framellabel
    LineNo = integer (totalNo.char[1..2])
    styleNo = integer (totalNo.char[3..3])
    totalLineNo = integer (totalNo.char[4..4])
    member("Frame" & StepNo & "TB").color = color(#rgb,238,238,238)
    member("Frame" & StepNo & "TB").width = 625
    member("Frame" & StepNo & "TB").fontsize = 14
    member("Frame" & StepNo & "TB").fontstyle = #[BOLD]
    member("Frame" & StepNo & "TB").alignment = #Right
    sprite(12).loc = point(0,20)
    sprite(12).ink = 36
    member("Frame" & StepNo & "Title").color = color(#rgb,0,0,0)
    member("Frame" & StepNo & "Title").width = 620
    member("Frame" & StepNo & "Title").fontsize = 18
    member("Frame" & StepNo & "Title").fontstyle = #[BOLD]
    member("Frame" & StepNo & "Title").alignment = #Center
    sprite(24).ink = 36
    sprite(24).loc = point(10,75)
    case styleNo of
        1: startPositionX = 40
           startPositionY = 100
           textWidth = 280
           sceneHeight = 300
           giveposition
           sprite(14).loc = point(475,200)
           sprite(14).ink = 36
           sprite(15).loc = point(475,354)
           sprite(15).ink = 36
        2: startPositionX = 150
           startPositionY = 100
           textWidth = 430
           sceneHeight = 300
           giveposition
           sprite(14).loc = point(315,247)
           sprite(14).ink = 36
           4: sprite(14).loc = point(183,262)
           sprite(15).loc = point(463,262)
           sprite(14).ink = 36
           sprite(15).ink = 36
5: startpositionX = 40  
   startpositionY = 100  
   textwidth = 280  
   Scenehight = 300  
   giveposition  
     sprite(14).ink = 36  
     sprite(14).loc = point(475, 247)  
6:  
   sprite(14).loc = point(183, 200)  
   sprite(15).loc = point(463, 200)  
   sprite(14).ink = 36  
   sprite(15).ink = 36  
   repeat with i = 1 to 2  
     member("line" & StepNo & i ).width = 280  
     member("line" & StepNo & i ).fontsize = 14  
     member("line" & StepNo & i ).fontstyle = [BOLD]  
   end repeat  
   sprite(26).loc = point(60, 320)  
   sprite(26).ink = 36  
   sprite(27).loc = point(340, 320)  
   sprite(27).ink = 36  
7:  
   sprite(14).loc = point(320, 200)  
   sprite(14).ink = 36  
   Scenehight = 120  
   textwidth = 500  
   startpositionX = 90  
   startpositionY = 280  
   giveposition  
8:  
   sprite(14).loc = point(183, 200)  
   sprite(15).loc = point(463, 200)  
   sprite(14).ink = 36  
   sprite(15).ink = 36  
   Scenehight = 120  
   textwidth = 500  
   startpositionX = 90  
   startpositionY = 300  
   giveposition  
9:  
   sprite(14).loc = point(320, 200)  
   sprite(14).ink = 36  
   i = 1  
   member("line" & StepNo & i ).fontsize = 14  
   member("line" & StepNo & i ).fontstyle = [BOLD]  
   member("line" & StepNo & i ).alignment = #Center  
   Scenehight = 120  
   textwidth = 400  
   startpositionX = 120  
   startpositionY = 280  
   giveposition  
end case  
end

on giveposition  
   spriteHeightList = [0]  
   totalhight = 0  
   texteHeight = 0  
   if member("line" & StepNo & "1" ).text = empty then  
     startline = 2 
   end
else
    startline = 1
end if

repeat with i = startline to totallineNo
    member("line" & StepNo & i ).width = textwidth
    member("line" & StepNo & i ).fontsize = 14
    member("line" & StepNo & i ).fontstyle = [#BOLD]
    spriteheight = member("line" & StepNo & i ).height
    spriteHeightList.addat(i+1,spriteheight)
    totalHeight = totalHeight + spriteheight
end repeat
if startline = 2 then
    spriteHeightList.deleteat(startline)
end if

textSpace = integer((Sceneheight - totalHeight)/(totallineNo + 1 - startline +1))
if startline = 1 then
    repeat with n=startline to totallineNo
        texteHeight = texteHeight + getat(spriteHeightList,(n))
        sprite( n +25).loc = point(startpositionX, startpositionY + textSpace * (n) + texteHeight)
        sprite( n +25).ink = 36
        put styleNo
        if styleNo = 2 then
            sprite( n +25 -12).loc = point(100, startpositionY + textSpace * (n) + texteHeight+9)
            sprite( n +25-12).ink = 36
        end if
    end repeat
else
    repeat with n= 1 to totallineNo-1
        texteHeight = texteHeight + getat(spriteHeightList,(n))
        sprite( n +26).loc = point(startpositionX, startpositionY + textSpace * (n) + texteHeight)
        sprite( n +26).ink = 36
        put styleNo
        if styleNo = 2 then
            sprite( n +26 -13).loc = point(100, startpositionY + textSpace * (n) + texteHeight+9)
            sprite( n +26-13).ink = 36
        end if
    end repeat
end if
end
Appendix B 2D simulation script

property Orgloc, My_sprite
global main1_rect, main2_rect, noose_rect, cockpit_rect
global checklist
global Myloc

on beginsprite me
  definerect
  checklist= [0,0,0,0,0,0,0,0,0]
  Orgloc = sprite(me.spriteNum).loc
  sprite(me.spriteNum).moveableSprite = true
  timetotal1 = 0
  timetotal2 = 0
end

on mousedown me
  cursor -1
end

on mouseup me
  cursor -1
  My_sprite = me.spriteNum
  if sprite(My_sprite).moveableSprite = true then
    Myloc = the mouseloc
    sprite(My_sprite).loc = Myloc
    repeat with i = 70 to 79
      sprite(i).loc = point(800,800)
    end repeat
    if inside(Myloc, cockpit_rect) then
      if My_sprite = 51 or My_sprite = 52 or My_sprite = 53 then
        correctfeedback
        sprite(My_sprite + 20).loc = point(403,285)
        sprite(My_sprite).moveableSprite = false
      else
        flyback
      end if
    else if inside(Myloc, main1_rect) then
      checkrepeat(main1_rect)
    else if inside(Myloc, main2_rect) then
      checkrepeat(main2_rect)
    else if inside(Myloc, noose_rect) then
      checkrepeat(noose_rect)
    else
      flyback
    end if
  else--corresponding to moveableSprite if
  exit
end if
end

on checkrepeate pos
  count1 = 0
  repeat with check1sprite = 54 to 56
    if inside(sprite(check1sprite).loc, pos) then
      count1 = count1 + 1
    end if
  end repeat
  count2 = 0
  repeat with check2sprite = 57 to 59
    if inside(sprite(check2sprite).loc, pos) then
      count2 = count2 + 1
    end if
  end repeat

  if count1 > 1 or count2 > 1 then
    flyback--there is an overlap
  else
    if pos = main1_rect then
      if My_sprite >= 54 and My_sprite <= 56 then
        sprite(87).LOC = point(68,291)
        sprite(88).LOC = point(50,218)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else if My_sprite >= 57 and My_sprite <= 59 then
        sprite(89).LOC = point(61,354)
        sprite(90).LOC = point(37,383)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else
        flyback
      end if
    else if pos = main2_rect then
      if My_sprite >= 54 and My_sprite <= 56 then
        sprite(93).LOC = point(137,348)
        sprite(94).LOC = point(167,334)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else if My_sprite >= 57 and My_sprite <= 59 then
        sprite(91).LOC = point(90,363)
        sprite(90).LOC = point(37,383)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else
        flyback
      end if
    else if pos = noose_rect then
      if My_sprite >= 54 and My_sprite <= 56 then
        sprite(97).LOC = point(315,253)
        sprite(98).LOC = point(318,276)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else if My_sprite >= 57 and My_sprite <= 59 then
        sprite(99).LOC = point(312,273)
        sprite(98).LOC = point(318,276)
        sprite(My_sprite).moveableSprite = false
        sprite(My_sprite+20).loc = point(403,285)
      else
        flyback
      end if
else if My_sprite >= 57 and My_sprite <= 59 then
    sprite(95).LOC = point(288,254)
sprite(96).LOC = point(247,278)
sprite(My_sprite).moveableSprite = false
    sprite(My_sprite+20).loc = point(403,285)
else
    flyback
end if
else
    flyback
end if
end if

on flyback
    sprite(My_sprite).loc = Orgloc
    sprite(70).loc = point(393,285)
end

on correctfeedback
    if My_sprite = 51 then
        sprite(81).LOC = point(300,161)
sprite(82).LOC = point(203,130)
else if My_sprite = 52 then
        sprite(83).LOC = point(307,151)
sprite(84).LOC = point(272,99)
else if My_sprite = 53 then
        sprite(85).LOC = point(296,171)
sprite(86).LOC = point(213,165)
end if
    sprite(My_sprite + 20).loc = point(803,285)
--sprite(My_sprite).moveableSprite = false
end

on checktimes
    counter = My_sprite - 50
    Currenttimes = getat(checklist,counter)
    Currenttimes = Currenttimes + 1
    setAt checklist, counter, Currenttimes
if counter = 1 or counter = 2 or counter = 3 then
    if Currenttimes = 2 then
        if inside(Myloc, cockpit_rect) then
            correctfeedback
        else
            sprite(My_sprite ).loc = point(803,285)
correctfeedback
            sprite(My_sprite + 50).loc = point(319,171)
        end if
else
    exit
end if
else
    timetotal1 = getat(checklist,4) + getat(checklist,5) +getat(checklist,6)
timetotal2 = getat(checklist,7) + getat(checklist,8) +getat(checklist,9)
if counter = 4 or counter = 5 or counter = 6 then
    if timetotal1= 6 then

repeat with i = 70 to 79
    sprite(i).loc = point (800, 800)
end repeat
sprite(104).loc = point(300, 220)
sprite(105).loc = point(65, 325)
sprite(106).loc = point(100, 340)
sprite(97).LOC = point(315, 253)
sprite(98).LOC = point(318, 276)
sprite(87).LOC = point(68, 291)
sprite(88).LOC = point(50, 218)
sprite(93).LOC = point(137, 348)
sprite(94).LOC = point(167, 334)
sprite(74).loc = point(393, 285)
repeat with i = 54 to 56
    sprite(i).loc = point(800, 800)
end repeat
end if
else if counter = 7 or counter = 8 or counter = 9 then
    if timetotal2 = 6 then
        sprite(107).loc = point(306, 226)
sprite(108).loc = point(71, 331)
sprite(109).loc = point(106, 346)
    repeat with i = 70 to 79
        sprite(i).loc = point (800, 800)
    end repeat
    repeat with i = 57 to 59
        sprite(i).loc = point(800, 800)
    end repeat
    sprite(78).loc = point(393, 285)
sprite(95).LOC = point(288, 254)
sprite(96).LOC = point(247, 278)
sprite(89).LOC = point(61, 354)
sprite(90).LOC = point(37, 383)
sprite(91).LOC = point(90, 363)
sprite(90).LOC = point(37, 383)
    end if
else
    exit
end if
end if
end
Appendix C Script for 3D objective creation

global obj
global slot
on beginsprite me
  obj = member("modelF3")
  obj.resetworld()
  obj.deleteTexture("colorOrange")
  obj.deleteTexture("colorWhite")
  obj.deleteTexture("colorGreen")
  obj.deleteTexture("colorBlue")
  obj.deleteTexture("colorBlack")
  obj.deleteTexture("colorRed")
  obj.deleteTexture("colorGrey")
  obj.deleteShader("shaderOrange")
  obj.deleteShader("shaderWhite")
  obj.deleteShader("shaderGreen")
  obj.deleteShader("shaderBlue")
  obj.deleteShader("shaderBlack")
  obj.deleteShader("shaderRed")
  obj.deleteShader("shaderGrey")

  colorOrange    = obj.newTexture("colorOrange", #fromCastmember, member("orange"))
  colorWhite    = obj.newTexture("colorWhite", #fromCastmember, member("white"))
  colorGreen    = obj.newTexture("colorGreen", #fromCastmember, member("green"))
  colorBlue    = obj.newTexture("colorBlue", #fromCastmember, member("blue"))
  colorBlack    = obj.newTexture("colorBlack", #fromCastmember, member("black"))
  colorRed    = obj.newTexture("colorRed", #fromCastmember, member("red"))
  colorGrey    = obj.newTexture("colorGrey", #fromCastmember, member("grey"))

  shaderOrange            = obj.newShader("shaderOrange",#standard)
  shaderOrange.texture    = colorOrange
  shaderWhite           = obj.newShader("shaderWhite",#standard)
  shaderWhite.texture    = colorWhite
  shaderGreen           = obj.newShader("shaderGreen",#standard)
  shaderGreen.texture    = colorGreen
  shaderBlue           = obj.newShader("shaderBlue",#standard)
  shaderBlue.texture    = colorBlue
  shaderBlack           = obj.newShader("shaderBlack",#standard)
  shaderBlack.texture    = colorBlack
  shaderRed           = obj.newShader("shaderRed",#standard)
  shaderRed.texture    = colorRed
  shaderGrey           = obj.newShader("shaderGrey",#standard)
  shaderGrey.texture    = colorRed

  obj.deleteModelResource("ibox")
  obj.deleteModel("ibox")
  ibox = obj.newModelResource("ibox", #box, #both)

  ibox.length    = 1
  ibox.width     = 1
  ibox.height    = 1

  ibox = obj.newModel("ibox")
  ibox.resource = obj.modelResource("ibox")
ibox.visibility = #none

obj.deleteModelResource("cone")
obj.deleteModel("slot")
cone = obj.newModelResource("cone", #Cylinder, #both)
cone.topRadius   = 0
cone.bottomRadius = .8
cone.height       = 4
slot = obj.newModel("slot")
slot.resource = obj.modelResource("cone")
slot.transform.rotation = vector(90,0,0)
slot.transform.position = vector(20,0,0)
slot.shader = shaderOrange
repeat with i = -5 to 5
  obj.deleteModel("cone left"&i)
  slot.clone("cone left"&i)
  obj.model("cone left"&i).transform.position = vector(-20,4*i,0)
  obj.deleteModel("cone right"&i)
  slot.clone("cone right"&i)
  obj.model("cone right"&i).transform.position = vector(20,4*i,0)
  obj.deleteModel("cone top"&i)
  slot.clone("cone top"&i)
  obj.model("cone top"&i).transform.position = vector(4*i,20,0)
  obj.deleteModel("cone bottom"&i)
  slot.clone("cone bottom"&i)
  obj.model("cone bottom"&i).transform.position = vector(4*i,-20,0)
end repeat
obj.deleteModelResource("base")
obj.deleteModel("base")
base = obj.newModelResource("base", #Plane, #both)
base.length   = 50
base.width    = 50
base.resource = obj.modelResource("base")
base.shaderlist = shaderWhite
obj.deleteModelResource("cbox")
obj.deleteModel("cbox")
cbox = obj.newModelResource("cbox", #box, #both)
cbox.length   = .8
cbox.width    = 1
cbox.height   = 2.4
cbox.resource = obj.modelResource("cbox")
cbox.transform.rotation = vector(0,0,90)
cbox.transform.position = vector(6.6,.17,.6)
cbox.shaderlist = shaderGreen
obj.deleteModelResource("wheel")
obj.deleteModel("wheel")
wheel = obj.newModelResource("wheel", #Cylinder, #both)
wheel.topRadius   = .1
wheel.bottomRadius = .1
wheel.height       = .05
wheel = obj.newModel("wheel")
wheel.resource = obj.modelResource("wheel")
wheel.transform.rotation = vector(0,0,0)
wheel.transform.position = vector(5.7,-.15,.1)
wheel.shaderlist = shaderBlack
obj.deleteModel("wheel right")
wheel.clone("wheel right")
obj.model("wheel right").transform.position = vector(7.5,-.15,.1)
obj.deleteModel("wheel top right")
wheel.clone("wheel top right")
obj.model("wheel top right").transform.position = vector(7.5,.5,.1)
obj.deleteModel("wheel top left")
wheel.clone("wheel top left")
obj.model("wheel top left").transform.position = vector(5.7,0.5,.1)
obj.deleteModelResource("manometer")
obj.deleteModel("manometer")
manometer = obj.newModelResource("manometer", #box, #both)
manometer.length = .3
manometer.width = 1
manometer.height = .7
manometer = obj.newModel("manometer")
manometer.resource = obj.modelResource("manometer")
manometer.transform.rotation = vector(0,0,0)
manometer.transform.position = vector(6,7,.5)
manometer.shaderlist = shaderRed
obj.deleteModelResource("ecart")
obj.deleteModel("ecart")
ecart = obj.newModelResource("ecart", #box, #both)
ecart.length = .4
ecart.width = 1.2
ecart.height = 4.2
ecart = obj.newModel("ecart")
ecart.resource = obj.modelResource("ecart")
ecart.transform.rotation = vector(0,0,0)
ecart.transform.position = vector(6,7,.15)
ecart.shaderlist = shaderGreen
obj.deleteModelResource("ewheel")
obj.deleteModel("ewheel")
ewheel = obj.newModelResource("ewheel", #Cylinder, #both)
ewheel.topRadius = .025
ewheel.bottomRadius = .025
ewheel.height = .0125
ewheel = obj.newModel("ewheel")
ewheel.resource = obj.modelResource("ewheel")
ewheel.transform.rotation = vector(0,0,0)
ewheel.transform.position = vector(5.53,6.66,.025)
ewheel.shaderlist = shaderBlack
obj.deleteModel("ewheel right")
ewheel.clone("ewheel right")
obj.model("ewheel right").transform.position = vector(5.7,6.66,.025)
obj.deleteModel("ewheel top right")
ewheel.clone("ewheel top right")
obj.model("ewheel top right").transform.position = vector(5.7,7.35,.025)
obj.deleteModel("ewheel top left")
ewheel.clone("ewheel top left")
obj.model("ewheel top left").transform.position = vector(5.53,7.35,.025)
obj.deleteModelResource("hose")
obj.deleteModel("hose")
hose = obj.newModelResource("hose", #Cylinder, #both)
hose.topRadius  = .1
hose.bottomRadius = .1
hose.height       = 12
hose = obj.newModel("hose")
hose.resource = obj.modelResource("hose")
hose.transform.rotation = vector(0,0,90)
hose.transform.position = vector(-1,0,0.05)
hose.shaderlist = shaderOrange
obj.deleteModelResource("hoset")
obj.deleteModel("hoset")
hoset = obj.newModelResource("hoset", #Cylinder, #both)
hoset.topRadius  = .1
hoset.bottomRadius = .1
hoset.height      = .6
hoset = obj.newModel("hoset")
hoset.resource = obj.modelResource("hoset")
hoset.transform.rotation = vector(45,0,90)
hoset.transform.position = vector(-7.18,0,0.25)
hoset.shaderlist = shaderOrange
obj.deleteModel("hoset clone")
hoset.clone("hoset clone")
obj.model("hoset clone").transform.position = vector(5.23,0,0.2)
obj.model("hoset clone").transform.rotation = vector(30,0,-90)
obj.deleteModelResource("sline")
obj.deleteModel("sline")
sline = obj.newModelResource("sline", #Cylinder, #both)
sline.topRadius   = .08
sline.bottomRadius = .08
sline.height      = 12
sline = obj.newModel("sline")
sline.resource = obj.modelResource("sline")
sline.transform.rotation = vector(0,0,90)
sline.transform.position = vector(-1,2,0.025)
sline.shaderlist = shaderBlack
obj.deleteModelResource("slinet")
obj.deleteModel("slinet")
slinet = obj.newModelResource("slinet", #Cylinder, #both)
slinet.topRadius  = .08
slinet.bottomRadius = .08
slinet.height     = .6
slinet = obj.newModel("slinet")
slinet.resource = obj.modelResource("slinet")
slinet.transform.rotation = vector(45,0,90)
slinet.transform.position = vector(-7.18,2,0.23)
slinet.shaderlist = shaderBlack
obj.deleteModel("slinet clone")
slinet.clone("slinet clone")
slinet.clone("slinet clone").transform.position = vector(5.23,0,0.17)
obj.model("slinet clone").transform.rotation = vector(30,0,-90)
obj.deleteModelResource("sline1")
obj.deleteModel("sline1")
sline1 = obj.newModelResource("sline1", #Cylinder, #both)
sline1.topRadius   = .08
sline1.bottomRadius = .08
sline1.height      = 13
sline1 = obj.newModel("sline1")
sline1.resource = obj.modelResource("sline1")
sline1.transform.rotation = vector(0,0,67)
sline1.transform.position = vector(-1,2.84,0.025)
sline1.shaderlist = shaderBlack
obj.deleteModel("sline1t")
sline.clone("sline1t")
obj.model("sline1t").transform.rotation = vector(45,0,90)
obj.model("sline1t").transform.position = vector(-7.18,5.375,0.23)
obj.deleteModel("sline1t clone")
slinet.clone("sline1 clone")
obj.model("sline1 clone").transform.position = vector(5.23,.3,.17)
obj.model("sline1 clone").transform.rotation = vector(30,0,-90)
obj.deleteModel("slinem")
sline.clone("slinem")
obj.model("slinem").transform.rotation = vector(0,0,90)
obj.model("slinem").transform.position = vector(-1,7,0.025)
obj.deleteModel("sline1m")
slinet.clone("sline1m")
obj.model("sline1m").transform.rotation = vector(45,0,90)
obj.model("sline1m").transform.position = vector(-7.18,7,0.23)
obj.deleteModel("slinem clone")
slinet.clone("slinem clone")
obj.model("slinem clone").transform.position = vector(5.23,7,0.145)
obj.model("slinem clone").transform.rotation = vector(25,0,-90)
p1 = -13
p2 = -25.5
repeat with i = 21 to 38
    obj.model[i].shader = shaderBlue
end repeat
repeat with i = 21 to 38
    obj.model("ChestA").addChild(obj.model[i])
end repeat
obj.model("ChestA").scale(6)
obj.model("ChestA").transform.position = vector(-6+p1,-18+p2,0)
obj.deleteModel("team lead")
obj.model("ChestA").clone("team lead")
obj.model("team lead").transform.position = vector(-6+p1,-18+p2,0)
obj.deleteModel("cabin pressure tester")
obj.model("ChestA").clone("cabin pressure tester")
obj.model("cabin pressure tester").transform.position = vector(-7+p1+1,-25.4+p2,0)
obj.deleteModel("aircraft visual check")
obj.model("ChestA").clone("aircraft visual check")
obj.model("aircraft visual check").transform.position = vector(-18+p1,-25.4+p2,0)
obj.deleteModel("roamer")
obj.model("ChestA").clone("roamer")
obj.model("roamer").transform.position = vector(-30+p1,-25.4+p2,0)
end
Appendix D Lingo controlled animation

sprite (25).setFlashProperty("blue", #alpha, 100)
sprite (25).setFlashProperty("low_safy_connection", #alpha, 30)
sprite (25).setFlashProperty("to_outside_air", #alpha, 30)
sprite (25).setFlashProperty("base", #alpha, 30)
sprite (25).setFlashProperty("needle_valve", #alpha, 30)
sprite (25).setFlashProperty("diaphragm", #alpha, 30)
sprite (25).setFlashProperty("upper_chamber_line", #alpha, 100)
sprite (25).setFlashProperty("saft_valve_line", #alpha, 30)
sprite (25).setFlashProperty("lower_chamber_line", #alpha, 30)
sprite (25).setFlashProperty("to_outside_air_line ", #alpha, 30)
Appendix E Check answer

```plaintext
global corrGroupList, answerList, gMyPath, StepNo
global qScore1,qScore2,qScore3,qScore4,qScore5,qScore6,qScore7
global qStartFrame
on determineScore
if answerList.count = 0 then
  alertMessage
else
  checkList
end if
end

on finishedChkTest
if answerList.count = 0 then
  alertMessage
else
  checkList
  submitScore
end if
end

On alertMessage
  alert "You have not selected an Answer Choice."
end

On checkList
  Q_Numbber = stepNo - qStartFrame + 1
if corrGroupList = answerList then
  sprite(53).member = member("Correct" & StepNo)
  if Q_Numbber = 1 then qScore1 = 1
  else if Q_Numbber =2 then qScore2 = 1
  else if Q_Numbber =3 then qScore3 = 1
  else if Q_Numbber =4 then qScore4 = 1
  else if Q_Numbber =5 then qScore5 = 1
  else if Q_Numbber =6 then qScore6 = 1
  else qScore7 = 1
  else
    if Q_Numbber = 1 then qScore1 = 0
    else if Q_Numbber =2 then qScore2 = 0
    else if Q_Numbber =3 then qScore3 = 0
    else if Q_Numbber =4 then qScore4 = 0
    else if Q_Numbber =5 then qScore5 = 0
    else if Q_Numbber =6 then qScore6 = 0
    else qScore7 = 0
  sprite(53).member = member("Incorrect" & StepNo)
end if
sprite(7).loc = point(1000,300)
sprite(51).loc = point(620,130)
member("Correct" & StepNo).width = 240
member("Incorrect" & StepNo).width = 240
sprite(53).loc = point(330,315)
sprite(54).loc = point(580,380)
sprite(49).loc = point(320,295)
sprite(50).loc = point(320,290)
```

sprite(52).loc = point(320, 290) --small
myColor = color(#rgb,144,144,144)
member("ans" & StepNo &"1c").color = myColor
member("ans" & StepNo &"2c").color = myColor
member("ans" & StepNo &"3c").color = myColor
member("ans" & StepNo &"4c").color = myColor
end

on submitScore
  -- This is an array which is passed to the database to update the score.
  -- Change the Module Number, Lesson Number, and Objective Number depending on your objective.
  qScore = qScore1 + qScore2 + qScore3 + qScore4 + qScore5 + qScore6 + qScore7
  InfoList = ["Score": qScore, "Module" : 1, "Lesson": 1, "Objective" : 1]
  postNetText(gMyPath, InfoList)
end
Appendix F Progress.asp

<!-- #include file="includes/header.asp" -->
<!-- #include file="includes/colorfile.inc" -->
<%
'*****************************************************************************
'loop module {
' get the lessons assigned to the user in this module
'loop lesson {
' get score from database
'loop module {
'print score based on the data in the db and lessons assignment
'
'
'
'}
'*****************************************************************************

Dim str_username, str_usertype, str_table
str_username = Request.Cookies("userName")
str_usertype = Request.Cookies("userType")

If str_usertype = "0" Then 'guest
str_table = "GuestComplete"
Else
str_table = "TinkerComplete"
End If
%

<html>
<head>
<title>OC-ALC Online Training System</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" topmargin="0">
<table width="600" border="0" cellpadding="1">
<tr bgcolor="#000000">
<td colspan="6" height="20"><font face="Arial, Helvetica, sans-serif" size="2"><b><font color="#FFFFFF">
My Progress <% =str_username %></font></b></font>
</td>
</tr>
<tr height="10"></tr>
<tr>
<td width="61"></td>
<td width="110"><img src="cgibin/image/box2.gif" width="16" height="16" align="absmiddle">
<font face="Arial, Helvetica, sans-serif" size="2">N/A</font></td>
<td width="130"><img src="cgibin/image/box3.gif" width="16" height="16" align="absmiddle">
<font face="Arial, Helvetica, sans-serif" size="2">Incomplete</font></td>
<td width="145"><img src="cgibin/image/box1.gif" width="16" height="16" align="absmiddle">
<font face="Arial, Helvetica, sans-serif" size="2">Completed</font></td>
<td width="76"><img src="cgibin/image/box4.gif" width="70" height="37" align="absmiddle"></td>
</tr>
</table>
</body>
</html>
<!-- start displaying user progress table -->
<%>
Dim obj_connect, obj_retrieve, str_SQL 'for module
Dim obj_retrieve1, str_SQL1 'for lesson
Dim obj_retrieve2, str_SQL2 'for objective
Dim obj_retrieve3, str_SQL3 'for lesson group
Dim arr_groupName(10) 'max of 10 groups/person
Dim int_groupNum 'number of groups of the person
Dim str_module, int_maxLesson, int_moduleLoop, int_lessonLoop, int_objLoop
Dim int_maxObj, int_view, int_lessonNum
Dim str_score, str_time, str_date, str_maxScore 'data retrieved from the database
Dim arr_score, arr_time, arr_date, arr_maxScore
Dim arr_assign(20) 'maximum lessons in a module
Dim bln_assign 'see if this lesson is assigned to the user
Dim obj_retrieve4, str_SQL4, bln_isUserAssign

Set obj_connect = Server.CreateObject("ADODB.Connection")
obj_connect.Open "DSN=", & Application("tinkerDB") & ",UID=", & Application("userDB") & ",password=", & Application("passDB")

str_SQL4 = "SELECT username FROM UserAssign WHERE username = \\
" & str_username & \\
Set obj_retrieve4 = obj_connect.Execute(str_SQL4)
If obj_retrieve4.EOF = True Then
bln_isUserAssign = "False"
Else
bln_isUserAssign = "True"
End If

If Not bln_isUserAssign Then
int_groupNum = 1
If str_usertype = "0" Then
arr_groupName(1) = "Guest"
int_groupNum = 2
Else
str_SQL3 = "SELECT gname FROM Belong WHERE username = \\
" & str_username & \\
Set obj_retrieve3 = obj_connect.Execute(str_SQL3)
While obj_retrieve3.EOF = False
arr_groupName(int_groupNum) = obj_retrieve3("gname")
int_groupNum = int_groupNum + 1
obj_retrieve3.MoveNext
Wend
End If
End If

str_SQL = "SELECT * FROM Module;"
Set obj_retrieve = obj_connect.Execute(str_SQL)
```plaintext
int_moduleLoop = 1
int_lessonLoop = 1
While obj_retrieve.EOF = False
str_module = obj_retrieve("mname")
int_maxLesson = obj_retrieve("maxlesson")

'retrieve lesson group for this particular module
getAssignLesson()

```
int_maxObj = CInt(obj_retrieve1("maxobj"))
str_maxScore = obj_retrieve1("maxscore")
bln_assign = FALSE 'reset variable

If arr_assign(int_lessonNum) = 1 Then
  bln_assign = TRUE
End If

If bln_assign Then

If str_maxScore <> "" Then
  arr_maxScore = Split(str_maxScore,"","",1)
End If

%>
<tr>
  <td rowspan="2" width="44" bgcolor="#FFDCB9">
    <div align="center"><font face="Arial, Helvetica, sans-serif" size="1">Lesson<% =int_lessonLoop %></font></div>
  </td>
%>
<tr>
  <tr>
    <%   printSecondRow() %>
  </tr>
%>
<int_lessonLoop = int_lessonLoop + 1
End If 'bln_assign
obj_retrieve1.MoveNext
Wend 'Lesson Loop

obj_retrieve.MoveNext
int_moduleLoop = int_moduleLoop + 1
Wend 'Module Loop

obj_connect.Close
Set obj_connect = Nothing

Sub printFirstRow()

'See if there is a score in database
If int_view = 0 Then 'no score available in database
If bln_assign Then 'user is assign to this lesson
For int_objLoop = 1 to int_maxObj
printIncomplete()
Next
For int_objLoop = int_maxObj+1 to 9
printNA()
Next
Else
For int_objLoop = 1 to 8
printNA()
Next
End If

Else'score in database
For int_objLoop = 1 to int_view
printComplete()
Next
For int_objLoop = int_view+1 to int_maxObj
printIncomplete()
Next
For int_objLoop = int_maxObj+1 to 9
printNA()
Next
End If

End Sub

Sub printSecondRow()

'See if there is a score in database
If int_view = 0 Then 'no score available in database
If bln_assign Then 'user is assign to this lesson
For int_objLoop = 1 to int_maxObj
printIncomplete2()
Next
For int_objLoop = int_maxObj+1 to 9
printNA2()
Next
Else
For int_objLoop = 1 to 8
printNA2()
Next
End If
Else 'score in database
For int_objLoop = 1 to int_view
printComplete2()
Next
For int_objLoop = int_view + 1 to int_maxObj
printIncomplete2()
Next
For int_objLoop = int_maxObj + 1 to 9
printNA2()
Next
End If

End Sub

'*****************************************************************************
Sub getAssignLesson()
Dim obj_temp, str_temp, arr_temp
Dim i, j, k

'initialize arr_assign
For k = 1 to 20
arr_assign(k) = 0
Next

'------------------------For Individual Assign-----------------------------
If bln_isUserAssign Then
str_SQL = "SELECT lessons FROM UserAssign WHERE username = '" & str_username & "' AND mnum = " & int_moduleLoop & ";"
Set obj_temp = obj_connect.Execute(str_SQL)
str_temp = obj_temp("lessons")
arr_temp = Split(str_temp,"",-1,1)

For j = 1 to int_maxLesson
If arr_temp(j) = "1" Then
arr_assign(j) = 1
End If
Next

'------------------------For Group Assign-------------------------------
Else
For i = 1 to int_groupNum-1
str_SQL = "SELECT lessons FROM GroupAssign WHERE gname = '" & arr_groupName(i) & "' AND mnum = " & int_moduleLoop & ";"
Set obj_temp = obj_connect.Execute(str_SQL)
str_temp = obj_temp("lessons")
arr_temp = Split(str_temp,"",-1,1)

For j = 1 to int_maxLesson
If arr_temp(j) = "1" Then
arr_assign(j) = 1
End If
Next

Else
End If

End Sub
Next
Next
End If
End Sub

Sub printNA() %>
<td width="30" bgcolor="<% =NA_COLOR %>">
   <div align="center"><font face="Arial, Helvetica, sans-serif" size="1">&nbsp;</font></div>
</td>
<td width="30" bgcolor="<% =NA_COLOR %>">
   <div align="center"><font size="1" face="Arial, Helvetica, sans-serif">&nbsp;</font></div>
</td>
<% End Sub %>

Sub printIncomplete() %>
<td width="30" bgcolor="<% =INCOMPLETE_COLOR %>">
   <div align="center"><font face="Arial, Helvetica, sans-serif" size="1">&nbsp;</font></div>
</td>
<td width="30" bgcolor="<% =INCOMPLETE_COLOR %>">
   <div align="center"><font size="1" face="Arial, Helvetica, sans-serif">&nbsp;</font></div>
</td>
<% End Sub %>

Sub printComplete() %>
<td width="30" bgcolor="<% =COMPLETE_COLOR %>">
   <div align="center"><font face="Arial, Helvetica, sans-serif" size="1"><%=arr_score(int_objLoop)%/<%=arr_maxScore(int_objLoop)%></font></div>
</td>
<td width="30" bgcolor="<% =COMPLETE_COLOR %>">
   <div align="center"><font size="1" face="Arial, Helvetica, sans-serif"><%=arr_time(int_objLoop)%></font></div>
</td>
<% End Sub %>

Sub printNA2() %>
<td colspan="2" bgcolor="<% =NA_COLOR %>">
   <div align="center"><font face="Arial, Helvetica, sans-serif" size="1">&nbsp;</font></div>
</td>
<% End Sub %>

Sub printIncomplete2() %>
<td colspan="2" bgcolor="<% =INCOMPLETE_COLOR %>">
   <div align="center"><font face="Arial, Helvetica, sans-serif" size="1">&nbsp;</font></div>
</td>
<% End Sub %>

Sub printComplete2() %>
<td colspan="2" bgcolor="<% =COMPLETE_COLOR %>">
<div align="center"><font face="Arial, Helvetica, sans-serif" size="1"><% =arr_date(int_objLoop) %></font></div>
</td>
<% End Sub %>
Appendix G Position

global currentnumber, StepNo
property orig_color

on mouseup me
  if sprite(me.spriteNum).color = rgb(250, 210, 104) or sprite(me.spriteNum).color = rgb(255, 255, 153) then
    currentnumber = me.spriteNum - 125
    go to "step " & currentnumber & " 0"
  end if
  repeat with i = 126 to 139
    if sprite(i).color = rgb(255, 255, 153) then -- yellow color
      sprite(i).color = rgb(204, 255, 255) -- blue color
    end if
  end repeat
end

on mouseEnter me
  if sprite(me.spriteNum).color = rgb(204, 255, 255) then -- blue color
    orig_color = sprite(me.spriteNum).color
    cursor 280
    sprite(me.spriteNum).color = rgb(250, 210, 104) -- orange color
  else if sprite(me.spriteNum).color = rgb(250, 210, 104) then -- orange color
    cursor 280
  else if sprite(me.spriteNum).color = rgb(255, 255, 153) then -- yellow color
    cursor 280
  else
    cursor -1
  end if
end

on mouseLeave me
  if sprite(me.spriteNum).color = rgb(204, 255, 255) or sprite(me.spriteNum).color = rgb(250, 210, 104) then
    sprite(me.spriteNum).color = rgb(204, 255, 255) -- blue color
  end if
  cursor -1
end
Appendix H Start script

property NumOfAnswer, AnswerSpace, AnswerHeight
global StepNo, lineNumber, AnswerHeightList

on getPropertyDescriptionList me
    description = [:]
    description.addProp (#NumOfAnswer, [#comment: "Number of Answers", #default: "3", #format: #string, #range: ["2", "3", "4", "5"]])
    return description
end

on beginSprite me
    AnswerHeightList = [0]
    totalHeight = 0
    AnswerHeight = 0
    member("Frame" & StepNo & "Title").color = color(#rgb,0,0,0)
    member("Frame" & StepNo & "Title").width = 620
    member("Frame" & StepNo & "Title").fontsize = 18
    member("Frame" & StepNo & "Title").fontstyle = [#BOLD]
    member("Frame" & StepNo & "Title").alignment = #Center
    member("line" & StepNo & "1").color = color(#rgb,0,0,0)
    member("line" & StepNo & "1").width = 545
    member("line" & StepNo & "1").fontsize = 14
    member("line" & StepNo & "1").fontstyle = [#BOLD]
    TitleHeight = member("line" & StepNo & "1").height
    if TitleHeight > 48 then yStartPosition = 160
    else yStartPosition = 140

    sprite(14).loc = point(10,66)
    sprite(14).ink = 36
    sprite(15).loc = point(60,110)
    sprite(15).ink = 36
    sprite(7).loc = point(583,393)
    sprite(7).loc = point(560,393) --move the submit button in
    sprite(51).loc = point(1362, 1) --move the feedback stuff out
    sprite(49).loc = point(900,800)
    sprite(52).loc = point(900, 60)
    sprite(53).loc = point(900, 100)
    sprite(54).loc = point(900, 80)
    sprite(50).loc = point(900, 800)
    repeat with n = 1 to NumOfAnswer
        member("ans" & StepNo & n & "c").color = color(#rgb,0,0,0)
        member("ans" & StepNo & n & "c").width = 420
        member("ans" & StepNo & n & "c").fontsize = 14
        member("ans" & StepNo & n & "c").fontstyle = [#BOLD]
        member("ans" & StepNo & n & "a").hilite = FALSE
        spriteHeight = member("ans" & StepNo & n & "c").height
        AnswerHeightList.add(spriteHeight)
        totalHeight = totalHeight + spriteHeight
    end repeat
    AnswerSpace = integer(260 - totalHeight)/(NumOfAnswer + 1)
    repeat with n = 1 to NumOfAnswer
        AnswerHeight = AnswerHeight + getat(AnswerHeightList,n)
sprite(19 + n).loc = point(130,ystartposition + AnswerSpace * n + AnswerHeight)
sprite(19 + n).ink = 36
sprite(33 + n).loc = point(100,ystartposition + AnswerSpace * n + AnswerHeight)
sprite(33 + n).ink = 36
sprite(26+n).loc = point(100+3,ystartposition + AnswerSpace * n + AnswerHeight+1)
sprite(26 + n).ink = 36
end repeat
end
Reference


