



ปีการศึกษา 2539

มัลติมีเดียช่วยสอนการเขียนภาษา HTML



โดย
นางสาว อุษณีย์ เหลืองอ่อน รหัส 35104564

วัน เดือน ปี ๑๑ ๑๑ ๒๕๓๙
เลขทะเบียน ๐๓๘๓๒๑
เลขเรียกหนังสือ T 3๓341 ๑๘๖๖๘

อาจารย์ที่ปรึกษา
ดร. วรวัฒน์ ลิ่มโกคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง
038321

มัลติมีเดียช่วยสอนการเขียนภาษา HTML

นางสาว อุษณีย์ เหลืองอ่อน

ดร. วรวัฒน์ ลิ้มโกทา อาจารย์ที่ปรึกษา

บทคัดย่อ

ปฏิญานิทรรศการฉบับนี้ จะกล่าวถึง การนำเสนอข้อมูลที่ช่วยในการเรียนการสอน ในรูปแบบของ ระบบมัลติมีเดีย โดยสื่อออกมาเป็น โครงงานช่วยสอนการเขียนภาษา HTML อันเป็นภาษาพื้นฐานที่ใช้กันโดยทั่วไปในการเขียน Home Page ซึ่งตัวโปรแกรมนี้ จะนำเสนอข้อมูลที่เป็นภาพเคลื่อนไหว และ เสียงเพื่อให้ผู้เรียนสามารถเข้าใจการเขียนภาษา HTML ได้อย่างชัดเจนเป็นขั้นตอน เหมือนกับการเรียน โดยตรงกับครูผู้สอนจริง โดยโปรแกรมจะแสดงรายละเอียดของหน้าจอคอมพิวเตอร์จริงที่สอนนั้น พร้อมกับคำบรรยายประกอบ ตัวโปรแกรมในโครงงานนี้ถูกพัฒนาขึ้นโดยใช้ ภาษา Borland Delphi 2.0 เป็นหลัก เพื่อเรียกการทำงานภายใต้ระบบปฏิบัติการของ Microsoft Windows 95 รวมทั้งการเรียกใช้โปรแกรม Lotus ScreenCam 97 เพื่อบันทึกหน้าจอที่ทำการสอนและจัดลำดับขั้นในการนำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MULTIMEDIA COMPUTER AID INSTRUCTOR FOR HTML

Miss. Usanee Luangon

DR. Voravat Limpoka (Advisor)

ABSTRACT

This thesis presents about presentation of data which use computer program to help teaching in multimedia system. It's present a media for teaching program HTML language (the basic language for making Home Page) in animation and effect sounds , so user can understand HTML language step by step . This program will present the real screen of teaching HTML language while learning and consist of description (voice). The evolution of program used Borland Delphi 2.0 language with Microsoft Windows 95 system and Lotus ScreenCam 97 to capture screen about teaching HTML language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ

1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	2
1.4 แนวทางในการพัฒนา	2

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับมัลติมีเดีย

2.1 ปัจจัยสำคัญในการนำเสนอและเรียนรู้ข้อมูลในยุคสารสนเทศ	3
2.2 ระบบมัลติมีเดียคืออะไร	3
2.3 ระบบมัลติมีเดียใช้ทำงานอะไรได้บ้าง	3
2.4 ระบบมัลติมีเดียมีอะไรบ้าง	4
2.5 ระบบมัลติมีเดียกับการใช้งาน Web ในระบบเครือข่ายอินเทอร์เน็ต	4
2.6 เทคโนโลยีที่เกี่ยวข้องกับมัลติมีเดีย	5
2.7 เรื่องของภาพนิ่ง	6
2.8 การพิจารณาเลือกใช้รูปแบบของไฟล์สำหรับงานมัลติมีเดีย	7
2.9 เรื่องของภาพเคลื่อนไหว	7
2.10 เรื่องของเสียง	8
2.11 การตอบโต้ซึ่งกันและกัน	9
2.12 ไฮเปอร์เท็กคืออะไร	9
2.13 มัลติมีเดียในประเทศไทย	9
2.14 บทสรุป	10

บทที่ 3 ความรู้เกี่ยวกับ Delphi

3.1 ชนิดของเคลไฟ	11
3.2 สิ่งที่เพิ่มขึ้นในเคลไฟ 2.0	11
3.3 แบบข้อมูลใหม่	12
3.4 เมนู	13
3.5 คอมโปเนนท์ใหม่	13
3.6 Object Repository	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การสืบสกุลของหน้าต่าง	28
3.8 ฟอรัมต้นแบบ	28
3.9 รันโปรแกรมทดสอบและยุติการรัน	34
3.10 การเก็บลงไฟล์	36
3.11 การใช้งานเคลไฟ	45
3.12 OBCD	49
3.13 การสร้าง table ด้วย SQL	62
3.14 หลักการพัฒนาแอปพลิเคชันพื้นฐานข้อมูล	62
3.15 การส่งมอบแอปพลิเคชันเพื่อนำไปใช้จริง	64
บทที่ 4 ScreenCam 97	
4.1 คำนำ	66
4.2 การทำงานของโปรแกรม	66
4.3 การบันทึกภาพ	67
4.4 การเก็บข้อมูลลงไฟล์	67
4.5 การแสดงค่าโดยใช้โปรแกรม ScreenCam Player	67
4.6 การใช้งาน ScreenCam ร่วมกับมัลติมีเดีย	68
4.7 การใช้งานใน ScreenCam ที่มีพีเจอาร์ใหม่	69
บทที่ 5 เริ่มต้นกับ HTML	
5.1 ความรู้ทั่วไปของภาษา HTML	72
5.2 HTML ทำงานอย่างไร	72
5.3 พัฒนาการทางภาษา HTML	73
5.4 การใช้ NotePad	74
5.5 การใช้ Markup Tag	75
5.6 ทำความเข้าใจขั้นพื้นฐานของ HTML	75
5.7 รูปแบบตัวอักษร	76
5.8 Hypertext Links	76
5.9 การใส่รูปภาพลงในเว็บเพจ	77
5.10 การใส่ข้อความสำหรับการแสดงผลแบบ Text	77
5.11 การวางตำแหน่งรูปภาพ	78
5.12 Width และ Height	78

5.13 การปรับขนาดรูปภาพให้ได้สัดส่วน	79
5.14 BORDER	79
5.15 BSPACE และ HSPACE	79
5.16 การแสดงภาพฉากหลัง	79
5.17 การสร้างภาพเคลื่อนไหว	80
5.18 การสร้างแบบสอบถาม	80
บทที่ 6 Director 5.0	
6.1 ส่วนประกอบสำคัญในการสร้าง	82
6.2 Case Windows	82
6.3 Score Windows	83
6.4 Channel	83
6.5 Ink Pop Menu	86
6.6 Special	87
6.7 Film Loop	89
6.8 Real Time Recording	90
บทที่ 7 Crystal Flying Fonts	
7.1 ส่วนประกอบในการสร้าง	92
7.2 Menu ที่สำคัญ	93
7.3 ตัวอย่างการ Render	93
7.4 Bevel	94
7.5 Apply Material	94
7.6 Motion	95
7.7 Make Movie	96
บทที่ 8 การออกแบบ	97
8.1 หลักในการออกแบบ	97
8.2 ขั้นตอนในการพัฒนา	99
8.3 อุปกรณ์และระบบที่ต้องการ	100
8.4 โครงสร้างโดยรวมของ โปรแกรม	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9 การทดลอง

103

บทที่ 10 สรุปและวิจารณ์

104

ภาคผนวก

Source Program

กิตติกรรมประกาศ

หนังสืออ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในยุคปัจจุบันนี้ได้ขึ้นชื่อว่าเป็นยุคของเทคโนโลยีการสื่อสารข้อมูล (IT Information System) มีความรู้ใหม่ ๆ เกิดขึ้นมากมาย การเรียนรู้ความรู้ใหม่ ๆ ที่เกิดขึ้นและการติดต่อสื่อสารข้อมูลที่รวดเร็วและมีประสิทธิภาพจึงเป็นสิ่งจำเป็นอย่างยิ่งสำหรับคนในยุคปัจจุบันนี้

เดิมทีการเรียนรู้โปรแกรมใหม่ๆ ในคอมพิวเตอร์ เราจะสามารถเรียนรู้ได้โดยการอ่านหนังสือ เกี่ยวกับโปรแกรมนั้นๆ การเปิดดูใน Help ของโปรแกรมนั้นควบคู่กันไปในขณะที่เขียนโปรแกรม หรือการไปเข้าเรียนใน Course เรียนพิเศษต่างๆ ที่เปิดสอน ซึ่งอาจจะไม่ได้รับความสะดวกในบางกรณี

เดิมทีการสื่อสารข้อมูลต่างๆ มักใช้การส่งเอกสาร แต่ในปัจจุบันกลายมาเป็นการส่งและสื่อสารข้อมูลผ่านทางคอมพิวเตอร์ โดยนำเอาการสื่อสารบนอินเทอร์เน็ตมาใช้กันอย่างกว้างขวาง และหัวใจในการแสดงข้อมูลข่าวสารบนอินเทอร์เน็ต ที่เรียกว่า Home Page ซึ่งเป็นข้อมูลที่เกิดจากกรไฟล์เอกสารเอชทีเอ็มแอล (HTML) ดังนั้นการเรียนรู้ภาษา HTML จึงเป็นสิ่งจำเป็นสำหรับการสร้าง Home Page อย่างหลีกเลี่ยงไม่ได้

ฉะนั้นจากเหตุผลดังกล่าวมาแล้วข้างต้น การที่เราจะสร้างโปรแกรมช่วยสอนการเขียนภาษา HTML จึงน่าจะเป็นสิ่งที่เหมาะสมที่สุดที่จะทำในโครงการนี้

1.2 วัตถุประสงค์

1. สามารถเป็นแนวทางในการเขียนโปรแกรมเพื่อช่วยสอนในเรื่องอื่น ๆ ได้
2. ตัวโปรแกรมเมื่อทำงาน จะสามารถเรียนรู้และทำความเข้าใจเกี่ยวกับการเข้าใจเกี่ยวกับการเขียนภาษา HTML ได้โดยง่าย
3. เรียนรู้การเขียนโปรแกรม Borland Delphi เพื่อใช้ในการดึง Application ต่างๆ ของ Microsoft Windows มาใช้ได้ เช่น การดึงเอาไฟล์ execute ต่างๆ มาเชื่อมต่อ เพื่อสามารถใช้งานได้
4. เรียนรู้การทำงานของ Lotus ScreenCam และสามารถนำมาประยุกต์การใช้งานได้
5. เรียนรู้การเปลี่ยนจากไฟล์ avi เป็นไฟล์ execute และดูข้อแตกต่างในด้านต่างๆ ของการใช้ไฟล์ avt กับการใช้ไฟล์ execute ในกรณีที่ทำให้การแสดงผลออกมาเหมือนกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขต

1. โปรแกรมนี้สามารถทำงานบน Microsoft Windows 95
2. โปรแกรมนี้สามารถสอนให้ทำความเข้าใจและเรียนรู้เกี่ยวกับการเขียนภาษา HTML อย่างง่ายๆ ได้
3. สามารถสั่งให้นำเสนอและยกเลิกการนำเสนออื่นๆ ได้
4. สามารถอธิบายขั้นตอนการทำงานของโปรแกรมโดยการอัดเสียงพูดบรรยายพร้อมทั้งบันทึกการเคลื่อนไหวที่หน้าจอขณะกระทำกรนั้น ๆ ได้

1.4 แนวทางในการพัฒนา

โปรแกรมนี้ สามารถได้รับการพัฒนาได้คือ โดยใช้ Borland Delphi ทำงานภายใต้ระบบปฏิบัติการ Microsoft Windows 95 รวมทั้งได้ใช้โปรแกรม Lotus ScreenCam 97 , Director 5.0 , Crystal Flying Font ซึ่งเป็น Application ที่ทำงานบน windows และข้อมูลสามารถหาได้ โดยง่ายสรุปโดยรวมหลังจากที่เราได้เลือกเครื่องมือในการทำงานเรียบร้อยแล้ว เราจึงได้วางแผนและขั้นตอนในการพัฒนาดังนี้ คือ

1. ทดลองใช้โปรแกรม Lotus ScreenCam ว่า ทำงานอย่างไร และสามารถติดต่อกับโปรแกรมอื่นอย่างไร
2. ทดลองเขียนโปรแกรมอย่างง่าย ในการเข้าควบคุมการทำงานของ Lotus ScreenCam
3. ศึกษาและเปรียบเทียบการอัดให้อยู่ในรูปของ ไฟล์ avi และไฟล์ execute เพื่อสามารถนำไปใช้งานได้
4. ทำความเข้าใจเรื่องการเขียนโปรแกรมภาษา HTML และจัดแบ่งเนื้อหาออกเป็นหมวดหมู่เป็นบท เพื่อให้ง่ายต่อการเรียนรู้
5. เรียนรู้แอปพลิเคชันอื่นๆ ที่ช่วยในการสร้างโปรแกรม เช่น Director 5.0 , Crystal Flying Font เป็นต้น
6. ทดลองเขียนโปรแกรมอย่างง่าย ในการเข้าควบคุมการทำงานของไฟล์ execute อื่นๆ
7. รวมโปรแกรมทั้งหมดเข้าด้วยกันและทดสอบการทำงาน
8. แก้ไขและดัดแปลงแอปพลิเคชันเป็นครั้งสุดท้าย เพื่อให้สมบูรณ์

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับมัลติมีเดีย

2.1 ปัจจัยสำคัญ ในการนำเสนอและเรียนรู้ข้อมูลในยุคแห่งสารสนเทศ

การเรียนรู้เป็นสิ่งสำคัญมากในยุคแห่งเทคโนโลยีสารสนเทศนี้ เพราะมีการเคลื่อนไหวเปลี่ยนแปลงและพัฒนาข้อมูลและการจัดการข้อมูลต่างๆ ให้อยู่ในรูปแบบที่สามารถนำมาประมวลผลและนำมาใช้จริงมากยิ่งขึ้น ทำให้การนำเสนอข้อมูล มีการแข่งขันกันอย่างมาก เพื่อตอบสนองความต้องการของผู้บริโภค หากรูปแบบในการนำเสนอข้อมูล สามารถเข้าใจได้ง่าย หรือ สามารถโต้ตอบ สื่อสารกับผู้รับได้ สื่อนั้นจะน่าสนใจมากทีเดียว

การนำเสนอข้อมูลในรูปแบบมัลติมีเดีย ในปัจจุบันกำลังเป็นที่นิยมกันอย่างมาก ไม่ว่าจะเป็นด้านการเรียนการสอน (ที่กล่าวถึงเป็นหัวข้อในวิทยานิพนธ์นี้) การโฆษณาสินค้า ความบันเทิง โดยตัวกลางของสื่อนี้มีมักจะอยู่ในรูปของ CD-ROM หรือไฟล์ข้อมูลบนเครื่องคอมพิวเตอร์ หรืออาจจะหาซื้อในรูปแบบนี้ได้ ทางอินเทอร์เน็ต

ในบทนี้ จะกล่าวถึงเรื่องเกี่ยวกับมัลติมีเดีย ไม่ว่าจะเป็นความเข้าใจเกี่ยวกับข้อมูลต่างๆ อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการสร้างงานมัลติมีเดีย

2.2 ระบบมัลติมีเดีย คืออะไร

การให้คำจำกัดความของคำว่ามัลติมีเดีย ในขณะนี้ยังไม่สิ้นสุด ทั้งนี้เนื่องมาจากการพัฒนาเทคโนโลยีต่างๆ ยังไม่สิ้นสุด จึงขออนุญาตอย่างกว้างๆ ว่าเป็น การรวบรวมสื่อต่างๆ ที่มีองค์ประกอบสำคัญใหญ่ๆ 3 อย่าง คือ

1. ภาพ
2. เสียง
3. การโต้ตอบซึ่งกันและกัน

2.3 ระบบมัลติมีเดีย ใช้กับงานอะไรได้บ้าง

ขอบเขตของระบบมัลติมีเดีย ได้รับการพัฒนากว้างขวางมากขึ้นทำให้ข้อจำกัดในการใช้งานมีน้อยลง เดิมคิดกันแต่เพียงว่า ระบบมัลติมีเดียจะเหมาะสำหรับงานเสนอเท่านั้น แต่ความจริงเราสามารถนำมัลติมีเดียมาประยุกต์ใช้งานได้มากมาย ได้แก่ ประยุกต์ใช้ในการศึกษา ด้านธุรกิจ ด้านการโฆษณา ด้านความบันเทิง ด้านการให้ข้อมูลตามสถานที่ต่าง ๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบมัลติมีเดีย มีอะไรบ้าง

สิ่งที่จำเป็นสำหรับระบบมัลติมีเดีย แยกเป็น 2 ส่วนคือ

1. ฮาร์ดแวร์ องค์ประกอบที่สำคัญของระบบมัลติมีเดียคือ ภาพ เสียง และการตอบโต้กันได้ ส่วนที่เป็นฮาร์ดแวร์จึงต้องจำแนกออกไปตามองค์ประกอบดังกล่าวโดยมีคอมพิวเตอร์เป็นส่วนสำคัญ คอมพิวเตอร์ที่ใช้งานในระบบมัลติมีเดียแยกจากกันระหว่างการสร้างงานจำเป็นต้องใช้คอมพิวเตอร์ที่มีสมรรถนะสูง ถ้าเป็นระดับเวิร์กสเตชัน จะดีมาก ซึ่งจะเป็นตัวเชื่อมโยงไปหาฮาร์ดแวร์อื่น ๆ
2. ซอฟต์แวร์มัลติมีเดีย แยกออกเป็น 2 ตอน คือ ตอนสร้างกับตอนใช้งาน

2.1 ซอฟต์แวร์สร้างงานระบบมัลติมีเดีย แยกออกเป็นสองส่วนคือ ซอฟต์แวร์สำหรับสร้างภาพและเสียง กับซอฟต์แวร์จัดระบบ

ซอฟต์แวร์สำหรับสร้างภาพและเสียง มักมีการใช้และเป็นที่คุ้นเคยของผู้ใช้คอมพิวเตอร์อยู่แล้วมีการตกแต่ง ตัดต่อ คัดแปลง สร้างใหม่ ซึ่งแยกกันทำเป็นชิ้นเป็นตอนได้

ซอฟต์แวร์จัดระบบมัลติมีเดีย เป็นซอฟต์แวร์ที่ควบคุมงานส่วนต่าง ๆ มาจัดลำดับ เพื่อให้มีการตอบโต้กันได้ เช่น บอกให้รู้ว่าถึงตอนนี้จะมีภาพต่อไปจะมีเสียง หรือถ้ากดตรงนี้จะได้ภาพนั้นภาพนี้ออกมา หรือมีเสียงนั้น เสียงนี้เป็นต้น ซอฟต์แวร์ประเภทนี้ยังไม่เป็นที่คุ้นเคยมากนัก ทำให้ยังมีน้อยราคาจึงสูงและที่สำคัญยังต้องใช้กับคอมพิวเตอร์สมรรถนะสูงด้วย

2.2 ซอฟต์แวร์ใช้งานระบบมัลติมีเดีย ระบบมัลติมีเดียนั้นถูกสร้างขึ้นมาจากผ่านระบบคอมพิวเตอร์ ทำให้ไม่เป็นการยากนักหากจะใช้งาน แต่มีข้อจำกัดอยู่ที่ความง่ายในการใช้งาน ถ้าหากใช้คอมพิวเตอร์ไม่เป็นจะทำอย่างไร หรือมาตรฐานของระบบคอมพิวเตอร์จากหลายค่าย ซึ่งจะถูกนำไปสู่การพัฒนาระบบอื่นๆ ที่ไม่ต้องใช้คอมพิวเตอร์ เพื่อรองรับกับตลาดผู้บริโภคมากที่สุดเท่าที่จะทำได้ โดยในปัจจุบันนี้มีการพัฒนาเครื่องรับโทรทัศน์ ให้สามารถใช้งานอินเทอร์เน็ตได้ นั้นหมายความว่าระบบมัลติมีเดียนั้นกำลังได้รับการพัฒนา ให้ง่ายในการใช้งานต่อผู้บริโภคมากยิ่งขึ้น

2.5 ระบบมัลติมีเดีย กับการใช้งาน Web ในระบบเครือข่ายอินเทอร์เน็ต

ระบบเครือข่าย Web เป็นการนำเสนอข้อมูลแบบไม่จำกัดบริเวณไม่ว่าจะอยู่ที่ใดในโลกแห่งอินเทอร์เน็ต เราจะสามารถค้นหาข้อมูลต่างๆ ที่มีอยู่ได้ การใช้งานทั้งสองระบบร่วมกัน ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันสามารถทำได้ แต่ยังมีข้อจำกัดอยู่ที่เรื่อง ความเร็วในการรับส่งข้อมูล หากในอนาคตสามารถแก้ปัญหานี้ได้ จะทำให้การพัฒนาเทคโนโลยีการนำเสนอข้อมูลเป็นไปอย่างไม่มีที่สิ้นสุด

2.6 เทคโนโลยีที่เกี่ยวข้องกับมัลติมีเดีย

เนื่องจากระบบคอมพิวเตอร์มัลติมีเดียเป็นการรวบรวมเทคโนโลยีหลายอย่างเข้าด้วยกัน เพื่อให้เกิดความสมบูรณ์ในการทำงานเทคโนโลยีเหล่านั้น ได้แก่

1. การพัฒนาเทคโนโลยีในการบันทึกข้อมูลการทำงานของมัลติมีเดียประกอบไปด้วย ภาพและเสียง
2. การพัฒนาด้านระบบคอมพิวเตอร์เครือข่าย สิ่งทีระบบคอมพิวเตอร์มัลติมีเดียเข้าไปมีบทบาทร่วมกับระบบคอมพิวเตอร์เครือข่าย เช่น การติดต่อสื่อสารด้วยระบบ Electronics Mail ซึ่งเดิมเป็นการติดต่อที่เป็นลักษณะ Text Base เท่านั้น เป็นการนำสองเทคโนโลยีมารวมกัน ทำให้การติดต่อสื่อสารในระบบเครือข่ายคอมพิวเตอร์ทำได้ทั้งที่เป็นภาพและเสียง
3. การพัฒนาเทคนิคการย่อหรือบีบอัดขนาดข้อมูล การย่อข้อมูลที่มีประสิทธิภาพจะเป็นปัจจัยสำคัญอย่างหนึ่ง ในการทำงานของระบบคอมพิวเตอร์มัลติมีเดีย
4. การพัฒนาไมโครคอมพิวเตอร์ การทำงานของคอมพิวเตอร์มัลติมีเดียเป็นการทำงานที่เกี่ยวข้องกับข้อมูลในปริมาณมหาศาล กระบวนการย่อและขยายขนาดข้อมูลจะต้องเกิดอย่างรวดเร็วและมากพอที่จะท
‘หยุดชะงัก เพราะถ้าเกิดเหตุการณ์เช่นนี้จะทำให้การแสดงผลทั้งภาพและเสียงอาจเพิ่มไปมาก ของจริงได้
5. การพัฒนาของจอภาพ
6. การพัฒนาอุปกรณ์ป้อนข้อมูล
7. การพัฒนาซอฟต์แวร์ ส่วนหนึ่งที่ทำให้โลกของคอมพิวเตอร์มัลติมีเดียเป็นจริงก็คือ การพัฒนาซอฟต์แวร์ ให้มีประสิทธิภาพสูงและมีการใช้งานได้ง่ายขึ้น การพิจารณาเลือกซอฟต์แวร์เพื่อมาทำงานด้านมัลติมีเดีย อาจพิจารณาได้จาก
 - ความง่ายในการใช้งาน
 - ความสามารถในการนำเสนองาน
 - ความสามารถในการติดต่อกับผู้ใช้
 - ความสามารถในการใช้ตัวแปรและฟังก์ชันในการคำนวณ
 - ความสามารถในการใช้งานร่วมกับโปรแกรมอื่น ๆ
 - มี Library ของโปรแกรมย่อยต่างๆ สนับสนุนการทำงาน
 - ความสามารถในการทำเอกสารประกอบ โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไฟล์ที่สามารถย่อและขยายภาพได้โดยมีการสูญเสียรายละเอียดน้อยมาก แต่เรามักใช้กับภาพที่ไม่มีรายละเอียดมาก ลักษณะของภาพที่มีการย่อ-ขยายขนาด เช่น โลโก้(Logo) และภาพในงานเขียนแบบต่างๆ เป็นต้น

2.8 การพิจารณาเลือกใช้รูปแบบของไฟล์สำหรับงานมัลติมีเดีย

1. BMP

BMP เป็นรูปแบบไฟล์มาตรฐานของ Windows โปรแกรมส่วนใหญ่ของ Windows สามารถทำงานกับ ไฟล์ภาพในรูปแบบนี้ได้ ซึ่งสามารถแสดงจำนวนสีได้มากถึง 16.7 ล้านสี (True Color 24 bits) ข้อดีของไฟล์รูปแบบนี้คือ ไม่มีการใช้เทคโนโลยีการบีบข้อมูลเลย ทำให้ไฟล์มีขนาดค่อนข้างใหญ่มาก เมื่อเปรียบเทียบกับ ไฟล์ในรูปแบบอื่น

2. GIF (Graphics Image Format)

GIF เป็นไฟล์รูปแบบของ CompuServe ซึ่งเป็นรูปแบบที่ใช้ในการสื่อสารข้อมูล (Down Load, Up Load) ไฟล์รูปแบบนี้จะสามารถเรียกใช้ได้โดยโปรแกรมตกแต่งภาพ (Image Retouching) หลายโปรแกรม

สามารถแสดงจำนวนสีได้ 256 สี (8 bits) มีการใช้เทคโนโลยีการบีบข้อมูลเข้ามาช่วย เพื่อให้ขนาดของ File ลดลงเพื่อประโยชน์ ในการส่งผ่านข้อมูล คุณภาพของภาพหลังจากบีบข้อมูลแล้วไม่มีการสูญเสียรายละเอียดของภาพแต่อย่างใด

3. JPEG (Joint Photographic Experts Group)

เป็นรูปแบบของภาพหนึ่งที่กำลังเป็นที่นิยมมากในการเก็บข้อมูลขณะนี้ เพราะสามารถบีบข้อมูลได้มากมีจำนวนสี 16.7 ล้านสี สามารถกำหนด Factor ของการบีบขนาดภาพได้

ข้อเสียคือมีการสูญเสียรายละเอียดของภาพบางส่วนตามอัตราการ บีบข้อมูล

ในการทำโครงการมัลติมีเดียเราจะควรใช้รูปแบบไฟล์ JPEG เพื่อประหยัดเนื้อที่ในการเก็บภาพ แต่ในบางครั้งเราจำเป็นต้องใช้รูปแบบไฟล์ BMP หรือรูปแบบไฟล์ GIF เนื่องจากซอฟต์แวร์ที่ใช้ในการจัดการระบบมัลติมีเดีย ไม่สามารถเรียกใช้ไฟล์ในรูปแบบ JPEG ได้

2.9 เรื่องของภาพเคลื่อนไหว (Animation Image)

ในงานมัลติมีเดียข้อมูลประเภทภาพเคลื่อนไหวเป็นข้อมูลที่น่าสนใจและสร้างความประทับใจให้กับผู้ใช้ได้มาก เพราะเป็นการผสมผสานระหว่างข้อมูลภาพและเสียง

ข้อมูลประเภทภาพเคลื่อนไหวจะมีวิธีการเก็บได้หลายวิธี มีทั้งชนิดที่ไม่ได้มีการบีบขนาดข้อมูลและที่มีการบีบขนาดข้อมูลด้วยการใช้ซอฟต์แวร์และฮาร์ดแวร์ ทำให้เกิดมีมาตรฐาน

2.11 การตอบโต้ซึ่งกันและกัน

คำว่า ตอบโต้กันได้ (Interactive) หมายถึง แสดงผลตอบสนองการทำงานว่าถูกต้องหรือไม่ ใช้ได้หรือไม่ได้ในขณะนั้น ระบบมัลติมีเดียต้องมีการตอบโต้กันได้จึงจะสมบูรณ์ จากลักษณะอันนี้เองที่ทำให้ขอบเขตของมัลติมีเดียกว้างขวางออกไปจำแนกออกได้ดังนี้

1. การตอบโต้กันได้ ในลักษณะตัวเลือก มีการจำลองสถานการณ์ (Simulation) จากปัญหาหนึ่งไว้หลายรูปแบบ โดยพิจารณาจากเงื่อนไขที่แตกต่างกัน หากสามารถจำลองสถานการณ์ได้มากและซับซ้อนครอบคลุมได้มากเท่าไร จะทำให้ระบบมัลติมีเดียที่สมบูรณ์มากขึ้นเท่านั้น

2. การตอบโต้กันได้ ในลักษณะการติดต่อสื่อสารใช้สมรรถนะของการคมนาคมที่ทันสมัยเข้าช่วยผลลัพธ์ที่ได้จะปรับเปลี่ยนตามเหตุการณ์ที่เปลี่ยนไป เช่น เดิมถ้าต้องการทราบข้อมูลเพียงว่า มีโรงแรมอะไรบ้างเท่านั้น แต่ปัจจุบันเมื่อเราเลือกโรงแรมใดแล้ว เราจะทราบถึงขนาดว่าโรงแรมนั้นมีห้องพักกี่ห้องและมีห้องว่างหรือไม่ด้วย เป็นต้น

สำหรับในงานมัลติมีเดียเราควรจะทำความรู้จักกับ ไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นรูปแบบของการจัดข้อมูล แต่จะมีความแตกต่างจาก Database โดยทั่วไปในด้านรูปแบบ และหน้าที่การทำงานต่างๆ เช่น การ Browsing, nodes และ การเชื่อมต่อ (Link)

2.12 ไฮเปอร์เท็กซ์ คืออะไร

ไฮเปอร์เท็กซ์ คือ สารบัญญัตินค้นหาในเทคโนโลยีใหม่ ที่สามารถกระโดดไปยังส่วนที่เกี่ยวข้องกัน เพื่อแสดงความหมาย ความสัมพันธ์ หรือ แสดงการตอบโต้ (Interactive) กับผู้ใช้ เช่นใน On-line Help ในวินโดว

ในปัจจุบันไฮเปอร์เท็กซ์ถูกนำเสนอในรูปแบบของหน้าตาที่สามารถจะใส่ข้อมูล เช่น ตัวอักษร (text), กราฟิก (graphic), รูปภาพ, เสียง, ภาพเคลื่อนไหว ซึ่งได้มีการรวบรวมมาตรฐานต่างๆ ของไฮเปอร์เท็กซ์ เรียกว่า "Standard General Markup Language (SGML)" ซึ่งจะอนุญาตให้โปรแกรมสามารถเชื่อมต่อไปยัง แอปพลิเคชัน ต่างๆ ได้

2.13 มัลติมีเดียในประเทศไทย

มัลติมีเดียได้เข้ามาในประเทศไทยนานแล้วแต่การใช้งานยังจำกัดอยู่แค่การใช้เพื่อการศึกษาและใช้เพื่อความบันเทิงเท่านั้น ทั้งนี้อาจเป็นเพราะระบบสร้างงานมัลติมีเดียต้องประกอบด้วยเครื่องคอมพิวเตอร์ความเร็วสูง ซอฟต์แวร์โปรแกรมกราฟิก และซอฟต์แวร์สำหรับใช้งานด้านมัลติมีเดียที่มีราคาสูง อีกทั้งผู้ค้าชาวไทยเกรงว่า หากพัฒนาระบบมัลติมีเดียเสร็จแล้ว จะมีผู้อื่นนำไปลอกเลียนแบบอีกทอดหนึ่ง อย่างไรก็ตามปัจจุบัน เมื่อพิจารณาจากยอดขายสินค้าระบบมัลติมีเดียแล้วจะพบว่าระบบมัลติมีเดียบนเครื่องพีซีนั้น มีแนวโน้มของความเป็นไปได้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขยายตัวของตลาดมัลติมีเดียในเมืองไทยสูง ในช่วงที่ผ่านมาขยายเพิ่มขึ้นเรื่อย ๆ นี้เองที่ส่งผลให้อัตราการใช้งานด้านมัลติมีเดียข้อมมีมากขึ้นเป็นเงาตามตัวด้วย ดังนั้นไม่ใช่เรื่องง่ายเลยที่จะทำนายทิศทางในอนาคตของระบบมัลติมีเดียว่า จะเป็นอย่างไร แต่พอจะกล่าวได้ว่า ระบบมัลติมีเดียจะเข้าสู่ตลาดเพื่อการอุปโภคและบริโภคอย่างแน่นอน

2.14 บทสรุป

ระบบมัลติมีเดียนับได้ว่าเป็นเทคโนโลยีที่เราควรจับตามอง เพราะเราจะสามารถนำภาพเสียงกราฟิก วิดีโอและข้อความมาผสมผสานเพื่อนำเสนอข้อมูลให้ดึงดูดความสนใจได้อย่างมากและถูกนำไปใช้ในวงการต่างๆ เช่น การเรียนการสอน ความบันเทิง ธุรกิจรูปแบบต่างๆ และนับวันยิ่งจะเป็นที่นิยมขึ้นไปเรื่อย ๆ แม้ระบบที่มีอยู่ในปัจจุบันจะยังมีข้อจำกัดอยู่มาก แต่ทางผู้ผลิตก็กำลังพัฒนาเทคโนโลยีใหม่ ๆ ขึ้นมา เพื่อแก้ปัญหาที่มีอยู่ในขณะนี้ อีกทั้งบริษัทซอฟต์แวร์ก็มีการผลิตซอฟต์แวร์ของมัลติมีเดียออกสู่ตลาดมากขึ้นด้วย ดังนั้น ตลาดของมัลติมีเดียซึ่งเคยจำกัดอยู่ในวงแคบ ๆ ก็กำลังขยายตัวออกไปอย่างกว้างขวาง ระบบมัลติมีเดียจึงเป็นสิ่งที่เข้ามามีบทบาทอย่างมากต่อวงการคอมพิวเตอร์ แม้การพัฒนาเทคโนโลยีมัลติมีเดียจะเพิ่งเริ่มต้น แต่เชื่อแน่ว่า ยังมีการพัฒนาไปอีกไกลอย่างแน่นอน

บทที่ 3

ความรู้เกี่ยวกับ Borland Delphi 2.0

3.1 ชุดของเดลไฟ 2.0

หลังจากที่ไม่โครซอฟท์ได้ออกวินโดวส์ 95 อย่างเป็นทางการแล้ว บอร์แลนด์ได้ออก Delphi 2.0 เพื่อใช้กับวินโดวส์ 95 และวินโดวส์เอ็นที

โดยจะกล่าวเรื่องของ เดลไฟ 2.0 ในเรื่อง/ข้อแตกต่าง/แบบข้อมูลใหม่/เมนู/คอมโปเนนต์ใหม่/Object Repository/ การสืบสกุลของหน้าต่างและ finalization

สำหรับเดลไฟ 2.0 คือเดลไฟที่ใช้สำหรับติดตั้งกับวินโดวส์ 95 หรือวินโดวส์เอ็นที ซึ่งจะขอรวมเรียกเป็นวินโดวส์ 95/เอ็นที เพื่อใช้พัฒนาโปรแกรมเป็นแอฟพลิเคชัน 32 บิต และใช้สภาพแวดล้อมเช่นไดอะล็อกบ็อกซ์ตามแบบของวินโดวส์ 95 เพื่อนำไปใช้รันบนวินโดวส์ 95 /เอ็นที

เดลไฟ 2.0 ประกอบด้วยเดลไฟชุดต่าง ๆ รวม 3 ชุด คือ

1. Delphi Desktop 2.0 ซึ่งเดิมเรียกว่า Delphi คือเดลไฟรุ่นแรกตามที่กล่าวในหนังสือนี้ มาแต่ต้น และเมื่อได้ออกรุ่น 2.0 แล้วได้เปลี่ยนชื่อเป็น Delphi Desktop 1.0

2. Delphi Developer 2.0 คือ Delphi Desktop 2.0 ที่มีโปรแกรมยูทิลิตี้ที่ช่วยในการพัฒนาโปรแกรมเพิ่มขึ้น

3. Delphi Client/Server 2.0 มีข้อแตกต่างกับชุด Desktop ในเรื่องงานฐานข้อมูลเท่านั้น ชุดที่ใช้ในการเขียน

ในการเขียนหนังสือเล่มนี้ ผู้เขียนใช้ Delphi Desktop 2.0 บนวินโดวส์ 95 เป็นหลัก นำซอร์สโปรแกรมทั้งหมดไปคอมไพล์ด้วยเดลไฟ 2.0 เป็นแอฟพลิเคชัน 32 บิต ใช้สภาพแวดล้อมของวินโดวส์ 95 และทำให้ไฟล์ .EXE มีขนาดเล็กและรันได้เร็วขึ้นมีเรื่องที่เกี่ยวข้องกับหนังสือนี้ ตามที่จะนำมากล่าว (เช่นไม่กล่าวถึงเรื่อง OLE ซึ่งจะอยู่ในเล่มสาม) สามเรื่องคือ

3.2 สิ่งที่เพิ่มขึ้นในเดลไฟ 2.0

สิ่งที่เพิ่มขึ้นในเดลไฟ 2.0 (เดลไฟ 1.0 เป็นซัพเซตของเดลไฟ 2.0) โดยหลักใหญ่คือทำให้ตนเองและโปรแกรมที่คอมไพล์ด้วยเดลไฟ 2.0 เป็นแอฟพลิเคชัน 32 บิต ใช้สภาพแวดล้อมของวินโดวส์ 95 และทำให้ไฟล์ .EXE มีขนาดเล็กและรันได้เร็วขึ้นมีเรื่องที่เกี่ยวข้องกับหนังสือนี้ ตามที่จะนำมากล่าว (เช่นไม่กล่าวถึงเรื่อง OLE ซึ่งจะอยู่ในเล่มสาม) สามเรื่องคือ

1. ทางด้านภาษา ตามที่จะกล่าวในหัวข้อ “แบบข้อมูลใหม่” และ เทียบ “โปรแกรม”

2. การสืบสกุลจาก TForm ตามที่จะกล่าวในหัวข้อ “การสืบสกุลของหน้าต่าง”
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการศึกษาเท่านั้น มิใช่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทางด้านคอมโปเนนต์ ตามที่จะกล่าวในหัวข้อ “คอมโปเนนต์ใหม่” และแสดงการใช้ในหัวข้อที่ตามมา

3.3 แบบข้อมูลใหม่

ในเคลฟ 2.0 ได้เพิ่มแบบข้อมูล (type) ขึ้นอีก 4 แบบ คือ

1. Character type แบ่งเป็น

- AnsiChar 8 บิต

- WideChar 16 บิต

ในปัจจุบันให้ Char เท่ากับ AnsiChar

2. Shortring type

- ShortString คือสตริงดั้งเดิม เป็นสตริงอย่างสั้น

- AnsiString เป็นสตริงอย่างยาว ซึ่งยาวได้ถึง 2 กิกะไบต์ (ค่าบวกสูงสุดของ LongInt) เป็นสตริงที่เป็นอะเรย์ของอักษร และเริ่มอักขระแรกที่ช่อง 1 เช่นกัน แต่ไม่มีช่องศูนย์ ซึ่งของสตริงเดิมหรือ ShortString ใช้เก็บความยาวของสตริง ส่วนของ AnsiString จะปิดท้ายด้วยอักขระ null (#0) และเก็บความยาวแยกไว้ต่างหาก

สำหรับคำว่า String ได้ให้คำตีฟอลต์ไว้เป็น AnsiString คือสตริงอย่างยาวโดยที่ถ้ากำหนด ตัวแปร เช่น

```
var astr : String;  
    bstr : String[255];
```

จะได้ astr เป็นสตริงอย่างยาว ซึ่งความยาวจะขึ้นกับข้อความที่จะให้ ส่วน bstr จะเป็นสตริงอย่างสั้นและจะกำหนดความยาวได้ไม่เกิน 255 อักขระ

ซึ่งในเคลฟได้ให้คำแนะนำไว้ว่าในโปรแกรมใหม่ควรใช้สตริงอย่างยาว และใช้ได้เช่นเดียวกับที่เคยใช้ เพียงแต่ไม่ต้องกำหนดความยาวของสตริง

3. Variant type เป็นแบบข้อมูลที่ไม่กำหนดแบบ อาจให้ค่าเป็นเลขจำนวนหรือสตริงหรือวันเวลาได้

4. Currency type เป็นแบบข้อมูลสำหรับการเก็บเลขจำนวนจริงที่มีทศนิยมน้อยตำแหน่ง ใช้หน่วยความจำ 18 ไบต์ เก็บค่าเป็นเลขจำนวนเต็ม โดยแบ่งเป็นเลขทศนิยม 4 ตำแหน่ง จึงเก็บค่าได้ประมาณ 900 ล้านล้าน เช่นบาท

3.4 เมนู

ในเดสก์ทอป 2.0 ได้ปรับปรุงเมนูใหม่ ที่เปลี่ยนแปลงและเพิ่มขึ้นตามที่ใช้ในหนังสือนี้คือ

1. File

- 1.1 New ตามที่จะกล่าวในหัวข้อ “สืบสกุลจาก Tform”
- 1.2 New Application เพื่อการเปิดโปรเจกต์ใหม่ คือ New Project เดิม
- 1.3 Open เพื่อเปิดโปรเจกต์เก่าคือ Open Project เดิมและใช้เปิดไฟล์ทุกชนิด
- 1.4 Reopen เพื่อใช้ในการเปิดโปรเจกต์หรือยูนิตเก่าที่เพิ่งปิด
- 1.5 Save As เพื่อการเก็บยูนิตเป็นไฟล์ .pas คือ Save File เดิม

2. Project เป็นรายการใหม่ คือ

- 2.1 Add to Project เพื่อผนวดยูนิตเข้ากับโปรเจกต์ คือ View/Project Manager เลือกปุ่ม Add unit ทั้งของเดิมและใหม่
- 2.2 Remove from Project เพื่อถอนยูนิตออกจากโปรเจกต์ ใช้ Project Manager เช่นกันทั้งของเดิมและใหม่
- 2.3 Add to Repository ตามที่จะกล่าวในหัวข้อ “การสืบสกุลของหน้าต่าง”

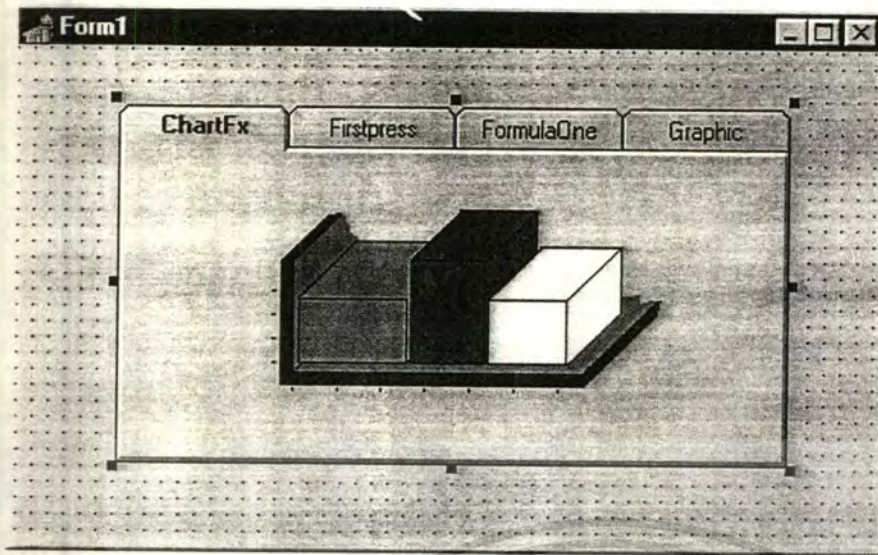
3. Tools

- 3.1 Options เพื่อการเซตค่าต่าง ๆ คือ Options/Environment เดิม
- 3.2 Repository เพื่อการเพิ่มต้นแบบตามที่จะกล่าวในหัวข้อ “การสืบสกุล

หน้าต่าง”

3.5 คอมโปเนนต์ใหม่

ในเดสก์ทอป 2.0 ได้เพิ่มคอมโปเนนต์ในงานทั่วไป (คือเว้นทางด้านฐานข้อมูล) ไว้ในหน้า Win 95 รวม 12 ชนิด เป็นคอนโทรลทั้งหมด เพื่อใช้แทนคอมโปเนนต์เดิมและเพิ่มใหม่ เพื่อให้เป็นคอนโทรลตามรูปแบบของวินโดวส์ 95 และได้นำคอมโปเนนต์เดิมที่มีของ ใหม่ใช้ แทนไปไว้ในหน้า Win 3.1 คอมโปเนนต์ที่เพิ่มใหม่ได้แก่



รูป 1 ภาพใบหน้าของ ChartFX ของ pageControl

TabControl เพื่อใช้แทน Tabset และมีวิธีใช้เช่นเดียวกัน (โปรแกรม Calenda)

PageControl เพื่อใช้แทน TabbedNote Boose และมีวิธีใช้คล้ายกัน แต่ทำให้กำหนดหน้าได้ง่ายกว่า 33 คือ อาจกำหนดหน้าทั้งหมดก่อนแล้วถึงกำหนดคอมโปเนนต์ในแต่ละหน้า หรือกำหนดหน้าและกำหนดคอมโปเนนต์ในแต่ละหน้า ได้กำหนดหน้าไว้ 4 หน้า จะเตรียมโปรแกรมที่หน้าได้ดังนี้

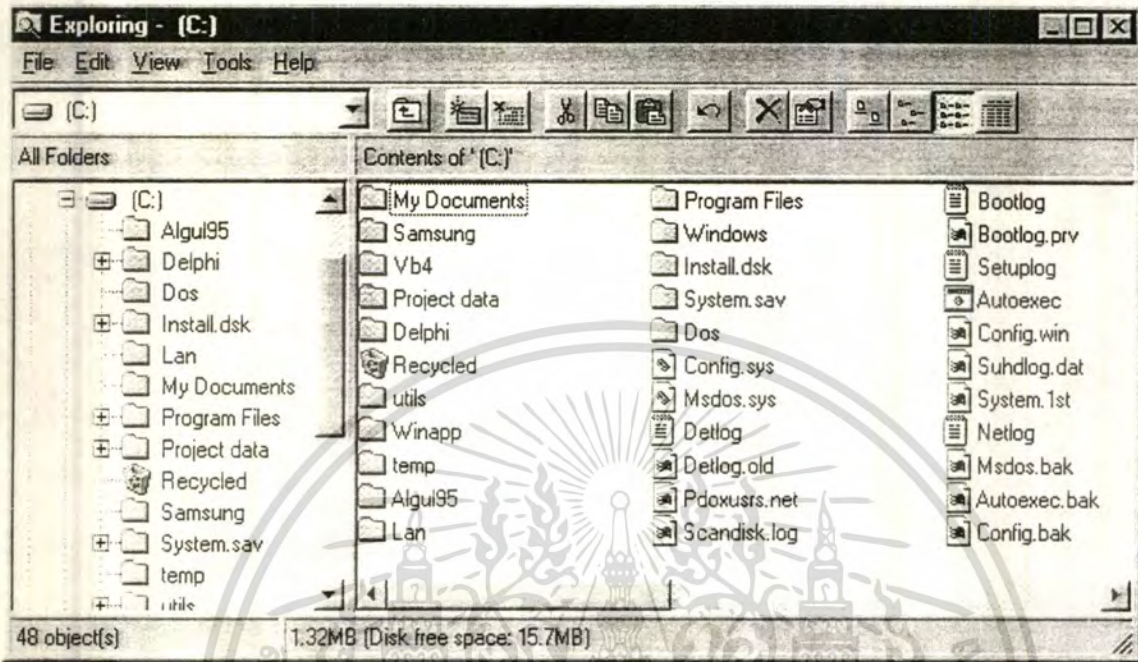
1. กำหนดคอมโปเนนต์เป็น PageControl 1
2. กำหนดหน้าและคอมโปเนนต์ได้โดย
 - คลิกเมาส์ขวาบน PageControl 1 จะแสดงเมนูลอย
 - ให้เลือกรายการ NewPage จะเพิ่มหน้าให้
 - ให้ป้อนชื่อออปเจกต์เป็น ChartFx จะเป็นการให้ค่าแก่ Name ซึ่งเมื่อกดคีย์ Enter จะได้เป็นค่า Caption ด้วย
 - กำหนดคอมโปเนนต์เป็น ChartFX1 (อยู่ในหน้า OCX ซึ่งเป็นแต่เพียงให้มี การกำหนดคอมโปเนนต์และแนะนำคอมโปเนนต์ในหน้า OCX เท่านั้น)
3. ทำซ้ำข้อ 2 โดยให้ชื่อหน้าเป็น Firstmpress, FormulaOne และ Graphic สำหรับการกำหนดคอมโปเนนต์ให้กำหนดเป็น VCFirst Impression 1, VCFormulaOne 1 และ GraphicsServer 1 ตามลำดับ
4. หากต้องการเปลี่ยนแปลงหรือกำหนดคอมโปเนนต์เพิ่มในหน้าใดให้เลือกหน้านั้นได้โดยตรง เมื่อที่เรารันคงเลือกหน้าจากที่คั่นหน้าได้เช่นกัน

TreeView

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ListView

ImageList



รูป 2 หน้าต่างของโปรแกรม Explorer

ทั้งสามคอมพิวเตอร์นี้ได้เตรียมไว้ให้ใช้งานร่วมกัน ดังเช่นในโปรแกรม Explorer ตามรูปข้างบน ได้แบ่งหน้าต่างออกเป็น 2 บาน บานซ้ายคือ Tree pane แสดงรายการให้เลือก โดยที่

- บางรายการมีปุ่มเครื่องหมายบวก เมื่อเลือกจะแสดงรายการรอง และเปลี่ยนเครื่องหมายเป็นลบ

- แต่ละรายการจะมีไอคอนกำกับอยู่ เพื่อบ่งบอกชนิดหรือประเภทของรายการ แต่ถ้าเป็นรูปแฟ้มปิดเมื่อเลือกจะแสดงรายการรองในบาน Contents pane ทางด้านขวา จะแสดงโดยมีไอคอนกำกับเช่นกัน และให้เลือกแสดงได้ 4 รูปแบบ โดยเลือกรายการใน View คือ

- Large Icon แสดงรายการจากซ้ายไปขวาและจากบนลงล่าง โดยแสดงภาพไอคอนขนาดใหญ่

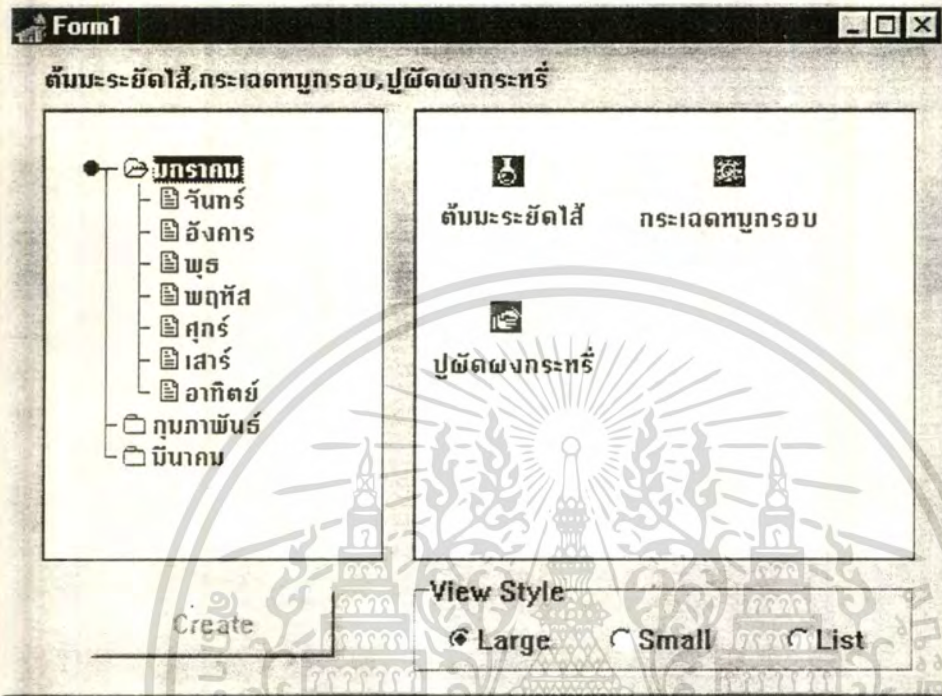
- Small Icon เช่นกันแต่กำกับด้วยภาพไอคอนขนาดเล็ก

- List เรียงลำดับจากบนลงล่างด้วยไอคอนขนาดเล็ก

- Details เช่นเดียวกับ List แต่จะให้รายละเอียด เช่น ขนาดของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเคลฟจึงจัดคอมโปเนนต์ชนิดนี้ให้ เพื่อใช้ TreeView เป็น Tree pane และ ListView เป็น Contents pane ส่วน ImageList เพื่อเก็บภาพเป็นอะเรย์ของรูปภาพ (บิตแม็พ ไอคอน หรือ เมตาไฟล์) เพื่อแสดงเป็น ไอคอน



รูป 3 แสดงการใช้ TreeView กรอบซ้าย ListView กรอบทางขวาและ ImageList ซึ่งไม่แสดงตัวขณะรัน

ดังจะแสดงตัวอย่างโดยสังเขป ซึ่งเมื่อเริ่มต้นจะแสดงรายการเป็นชื่อเดือน (3เดือน) ต่อเมื่อเลือกปุ่มบวกของรายการ “มกราคม” จะแสดงรายการวันและเมื่อเลือกตรง ไอคอน ของ รายการวัน เช่น ตามรูปได้เลือกของวัน “อาทิตย์” จะแสดงรายการอาหารในกรอบ ListView 1 ทางขวา

ดังจะกล่าวถึงการเตรียม โปรแกรมควบคู่ไปกับคำอธิบายถึงคอมโปเนนต์ทั้งสามนี้ ดังนี้

คือ

1. การใช้ ImageList
2. การใช้ TreeView
3. การใช้ ListView

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ ImageList

เป็นคอมโปเนนต์ที่ไม่ใช่คอนโทรล จึงไม่แสดงตัวในขณะรัน ในการใช้งานจะต้องโหลดภาพ (บิตแมป ไอคอน หรือเมตาไฟล์) จำนวนหนึ่งให้แก่ ImageList ซึ่งอาจให้ได้ทั้งขณะเตรียมโปรแกรมและในขณะรันและภาพที่จะให้อาจให้เป็นภาพเล็กหลายภาพหรือให้ภาพที่มีลักษณะเป็นแถบยาวภาพเดียว

ดังจะแสดงตัวอย่างทุกวิธี ขอให้กำหนดคอมโปเนนต์เป็น ImageList รวม 3 ตัวได้เป็น imglTree, imglSmall และ imglLarg3e (ให้ค่าแก่ Name) แล้วดำเนินการดังนี้

1. ของ imglTree ซึ่งจะให้เก็บภาพไอคอนของรายการใน treeView 1 โดยให้ค่าในโปรแกรม จึงต้องเตรียมภาพบิตแมปเก็บไว้ในรีซอร์สคือในไฟล์ view1.res โดยใช้ Image Editor คือ

- เลือกรายการ Tools/Image Editor
- เลือกรายการ File/New/Resource File
- เลือกรายการ Resource/New/Bitmap

ให้ค่า Width เป็น $6 \times 16 = 96$ และค่า Height เป็น 16 จะได้ชื่อบิตแมปรีซอร์สเป็น Bitmap 1 ให้ป้อนแก่เป็น SPOTSSIMG

ดับเบิลคลิกที่ SPOTSIMG จะแสดงหน้าต่างสำหรับวาดภาพซึ่งอาจไม่เห็นต้องขยายหน้าต่าง Image Editor

-เลือกรายการ View/Zoom In เพื่อขยายพื้นที่วาดภาพ ซึ่งเตรียมไว้สำหรับการวาดภาพ 6 ภาพ จึงขอให้กะแบ่งพื้นที่เป็น 6 ช่องเท่า ๆ กัน และตามตัวอย่างเพื่อความง่ายจึงได้วาด เป็นรูป วงกลมสีน้ำเงิน แดง เขียว เหลือง ขาวและดำรวม 6 รูป

-เลือกรายการ File/Save As เก็บไว้ในชื่อ view.res ในไดเรกทอรีเดียวกับที่จะเก็บโปรแกรม
ลิสต์ที่ 1

```
implementation
{ SR *.DFM }
{ SR view1.res }
procedure TForm1.FormCreate ( Sender : TObject );
begin
    oMnuList := TStringList.Create ;
    imglTree.ResourceLoad(rtBitmap, 'SPOTSIMG',c10live); ต่อหน้าถัดไป
end;
procedure TForm1.FormDestroy ( Sender : TObject );
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```
oMnuList.Free;
```

```
end;
```

ซึ่งจะ โหลดมาให้แก่ imglTree ไปด้วย ResourceLoad ดังในลสคิงที่ 3

2. ของ imglSmall ซึ่งจะให้เก็บภาพบิตแม็พขนาดเล็ก เพื่อใช้เป็นไอคอนของรายการใน LisView 1 โดยให้ค่าในขณะเตรียมโปรแกรม ดังนี้คือ

- คลิกเมาส์ขวาที่ imglSmall เพื่อเปิดเมนูลอย แล้วเลือกรายการ ImaglList Editor

- จะแสดงไดอะล็อกบ็อกซ์ให้เลือกภาพ โดยเลือกรายการ Add ขอให้เลือกภาพขนาด 16 x 16 ซึ่งอาจวาดไว้ก่อน หรือ ตามตัวอย่างนี้ได้เลือกภาพในไดเรคทอรี

```
\program File\Borland\delphi 2.0 \Images \Icons \
```

โดยเลือกไฟล์ที่ลงท้ายด้วยเลข 16 ทั้งหมด โดย Add ครั้งละ 1 ไฟล์

3. ของ imglLarge ซึ่งจะให้เก็บภาพบิตแม็พขนาด 32 x 32 ดังนั้นก่อน โหลดภาพขอให้ป้อนแก้ค่าพารามิเตอร์คือ Height และ Width เป็น 32 คู่เสร็จแล้ว ให้โหลดภาพที่เหลือในไดเรคทอรีข้างบนให้ โดย Add ครั้งละ 1 ไฟล์

การใช้ TreeView

TreeView ก็คือ Outline (โปรแกรม BrowseNI ในบทที่ 4) ของวินโดวส์ 95 สิ่งที่เราเพิ่มมาคือให้เรากำหนดภาพที่จะใช้เป็นไอคอนได้ วิธีใช้จึงใกล้เคียงกันซึ่งสามารถให้ค่าได้ในขณะเตรียมโปรแกรมและขณะรัน เป็นการให้เป็นค่ารายการหรือคือเป็นโหนดหนึ่ง ๆ

โดยที่ใน TreeView จะมี ItemS เป็นค่าพารามิเตอร์ของทุก TTreeNodeS ซึ่งเป็นอระเรย์ของ TTreeNode ซึ่งก็คือโหนดหนึ่ง ๆ โดยที่ในแต่ละโหนดนี้จะมี ItemS อีก ทำให้รายการหนึ่ง ๆ สามารถมีรายการรองและมีรายการรอง ๆ ต่อไปอีกได้ ตามที่จะกล่าวต่อไป คือ

หลังจากที่กำหนดคอมโปเนนต์เป็น TreeView แล้ว ให้เลือกหัวข้อ ImaglTree และให้ค่ารายการได้ดังนี้

1. ในขณะเตรียมโปรแกรม (ตามที่จะยกมาเป็นตัวอย่างไม่ได้ใช้วิธีนี้) ให้เลือกหัวข้อ ItemS แล้วเลือกปุ่มอิลิป จะแสดงไดอะล็อกบ็อกซ์ให้ป้อนชื่อรายการ โดย

- เลือกปุ่ม New Item เพื่อป้อนหัวข้อใหญ่

- เลือกหัวข้อใหญ่ (ใดก็ได้หากมีมาก) แล้วเลือกปุ่ม New Item เพื่อป้อนหัวข้อใหญ่ใน

ระดับเดียวกัน เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- เลือกหัวข้อใหญ่ แล้วเลือกปุ่ม New SubItem เพื่อหัวข้อรองของหัวข้อใหญ่นั้น

- ให้หมายเลขภาพปิดแม่พิน Image List นับจากศูนย์ แก่ Image Index ให้เป็นภาพที่แสดงว่าไม่ถูกเลือก และแก่ Selected Index เพื่อแสดงว่าถูกเลือก

ซึ่งเมื่อเลือกปุ่ม OK จะแสดงรายการหัวข้อที่ป้อนพร้อมด้วยไอคอน (หากโหลดให้แก่ imglTree แล้ว) ในกรอบ TreeView

2. ในขณะรัน การใช้งาน TreeView เมื่อไม่ได้ให้แสดงข้อมูลในระบบของวินโดวส์ (ดังเช่นของโปรแกรม Explour) ควรเป็นรายการที่ให้ผู้เปลี่ยนแปลงได้ดังในรูป 3 ซึ่งอาจให้ผู้ป้อนเป็นเท็กซ์ไฟล์ ตามที่กล่าวในหัวข้อ “กรอบ Outline” แต่อาจไม่สะดวกแก่ผู้ใช้

จึงกำหนดให้ป้อนดังในลิสตดังที่ 2 โดยมี

- ชื่อเดือน เป็นหัวข้อใหญ่ ให้ป้อนขีดซ้าย

- ชื่อวัน เป็นหัวข้อรอง ให้กดคีย์ space bar หนึ่งครั้งแล้วป้อน

- ชื่ออาหาร ซึ่งควรเป็นหัวข้อรองของวัน คือกดคีย์ space bar สองครั้งแล้วป้อนบรรทัดละ 1 ชื่อ แต่เห็นว่าไม่สะดวกจึงจัดให้เป็นข้อมูล โดยให้กดคีย์ Tab หนึ่งครั้งแล้วป้อน รวมทั้งเมื่อขึ้นบรรทัดใหม่ด้วยเช่นของวันอาทิตย์ในการป้อนชื่ออาหารให้แยกจากกันด้วยเครื่องหมาย , และวรรค

ขอให้ผู้ป้อนข้อความตามลิสตดัง 2 เก็บไว้ในไฟล์foodmenu.txt ในไดเรกทอรีเดียวกับโปรแกรม
ลิสตดังที่2

มกราคม	
จันทร์	ต้มยำกุ้ง , คะน่าน้ำมันหอย , ปลาทอด
อังคาร	แกงเขียวหวานไก่ , กุ้งชุบแป้งทอด , ผัดบุ้งไฟแดง
พุธ	ไข่ยัดไส้ , แกงจืดเต้าหู้ , ผัดพริกหยวก
พฤหัสบดี	ผัดเนื้อน้ำมันหอย , แกงเลียง , ยำปลาทูกระป๋อง
ศุกร์	ปลาช่อนแปะชะ , แกงป่า , ผัดถั่วงอก
เสาร์	แกงแดงเนื้อ , ไข่ลูกเขย , ผัดพักรวมมิตร
อาทิตย์	ต้มมะระขี้คัส , กระเจตหมูกรอบ , ปูผัดผงกระหรี่ , ข้าวเหนียว , กล้วย , ส้มตำ , น้ำตก
กุมภาพันธ์	
มีนาคม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิสต์เรียงที่ 3

```
const chTab = #9; chspce = #32; cdCmma = ',';
function GetToken ( var txt : String ) : String;
var ic : integer;
begin
    ic := Pos ( chTab , txt );
    if ic = 0 then ic := Pos ( chCmma . txt );
    if ic > 0 then Result := Copy ( txt , 1 , ic - 1 )
else
    begin
        if Lenth ( txt ) > 0 then
            begin Result := txt ; txt := '' end
        else Result := ''
    end;
    if txt <> '' then
        txt := Copy ( txt , ic + 1 , Length ( txt ) );
    end;
    procedure TForm1.button1click ( Sender : TObject );
    var oList : TStringList ; itm , ind : integer;
        txt , adTxt : String; oNde , oSub : TTreeNode;
    begin
        TreeView1.ItemS.Clear; oList := TStringList.Create;
        oList.LoadFromFile ( 'foodmenu.txt' );
        itm := 0; oMnuList.Clear;
        while itm < oList.Count do
            begin
                txt := oList [ itm ]; inc ( itm );
                if txt [ 1 ] > chSpec then
                    begin
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ผ่านการ
ไม่วารณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

oNde := TreeView1.ItemS.Add(TreeView1.Selected, txt);
oNde.ImageIndex := 0 ; oNde.SelectedIndex := 0 ;
end
else if txt[1] = chSpce Then
begin
oSub.data := Pointer ( oMnuList.count);
oSub.ImageIndex := 2 ; oSub.SelectedIndex := 3 ;
end
else
begin
txt[1] := chSpce ; ind := oMnuList.Count-1 ;
oMnuList.Delete(ind) ; oMnuList.add( adtxt);
end;
end;
oList.Free ; Button1.Enabled := False ;
end;

```

จะให้ค่าแก่ TreeView 1 ได้ตั้งโปรแกรมของกิจกรรม Button 1 Click ในลิสต์ดิ่งที่ 3 คือเมื่อผู้ใช้เลือกปุ่ม Create คือปุ่ม Button1 ซึ่งได้ให้ดำเนินการดังนี้คือ

- 2.1 ให้ oList โหลดข้อมูลจากไฟล์ foodmenu.txt มาเก็บไว้เป็นรายการสตริง สตริงละ 1 บรรทัด
- 2.2 ในวงรอบ while ได้ให้อ่านสตริงจาก oList มาครั้งละ 1 สตริง
- 2.3 ซึ่งถ้าอยู่ต้นบรรทัด คือชื่อเดือน จะให้เพิ่มโหนดเป็นหัวข้อใหญ่ด้วย Add โดยที่ Add จะ

- ให้กำเนิดโหนด (โหนดเป็นอปเจ็กต์) ดังที่ให้ค่าแก่ oNde
- และผนวกต่อท้ายโหนดในระดับเดียวกัน ตามค่าโหนดที่ผ่านไปให้ตรง Selected
- ซึ่งในครั้งแรกนี้ Selected จะเป็น nil จึงผนวกเป็นโหนดหัวข้อใหญ่ หัวข้อแรกคือ

“มกราคม”

- ในครั้งต่อไป Selected จะเป็นโหนดของเดือน “มกราคม” จึงเป็นการบวกโหนดของเดือน “กุมภาพันธ์” ต่อท้ายเดือน “มกราคม”

เอกสารนี้เป็นสิ่งที่ท่านได้จัดทำขึ้นเพื่อใช้ประกอบการเรียนการสอน เมื่อผู้ดูแลเนื้อหาเว็บไซต์นโยบายด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใ้ค่า ImageIndex และ Selected Index เป็นศูนย์ (วงกลมสีน้ำเงิน) ทั้งคู่ เพื่อจะบอกผู้ใช้งานว่าเป็นหัวข้อที่ไม่มีข้อมูลภายใน

2.4 แต่ถ้าอักขระแรกเป็นวรรค คือชื่อวัน จะให้ AddChild ผนวกให้เป็นโหนดลูก เช่น “วันจันทร์” ของโหนดแม่ เช่น ของเดือน “มกราคม” (ตามตัวอย่างนี้มีแค่เดือนนี้) โดย

- ให้ GetTokm ตัดคำเฉพาะที่เป็นชื่อวันมาให้เป็นค่า Text
- ค่าที่เหลือใน txt จะเป็นรายชื่ออาหาร ใ้เก็บไว้เป็นรายการสตริงใน oMmuList
- และใ้ค่าอินเด็กซ์บวทหนึ่ง ของรายการนี้แก่ Data ซึ่งใ้เราเก็บค่าใดก็ได้ตามต้องการแต่ต้องเก็บเป็น Pointer (ที่ใ้เก็บเป็นอินเด็กซ์บวทหนึ่ง เพราะไม่ต้องการใ้มีค่าศูนย์ ซึ่งจะตรงกับ mil)

2.5 ถ้าอักขระแรกเป็น Tab เช่น กรณีของวันอาทิตย์ ใ้เก็บต่อท้ายสตริงเดิม

การใช้ ListView

มีลักษณะการใช้ใกล้เคียงกับ TreeView แต่จะต้องใช้ ImageList สองตัวใ้กำหนดคอมโปเนนต์เป็น ListView1 และใ้ค่า

- LargeImage เป็น imgLarge

- SmallImages เป็น imgSmall

กับกำหนดคอมโปเนนต์เป็น RadioGroup1 โดยใ้ชื่อปุ่มวิทยูเป็น Large, Small, List และ Report

จะใ้ค่าแก่ ListView1 เพื่อใ้แสดงรายการตั้งในรูปที่ 3 กรอบขวา ใ้ตั้งโปรแกรมของกิจกรรม TreeView1MouseDown ในลิสตติง 4 ซึ่งจะมาทำงานที่กิจกรรมนี้เมื่อผู้ใ้เลือกรายการใน TreeView ดังที่ใ้ดำเนินการตั้งนี้

ลิสตติงที่ 4

```
procedure TForm1.TreeView1MouseDown(Sender : TObject ; Button :
TMouseButton ;
    Shift : TShiftState ; X,Y : Integer);
var HT : Thittests ; oNde : TTreeNode ; txt ,adTxt : String ;
    oItem : TListItem ; cnt : Integer ;
begin
    HT := Treeview1.GetHitTestInfoAt (X,Y) ;
    if not (htOnItem in HT) then Exit ;
    Label1.caption := omnuList [Integer(oNde.Data) - 1];
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การดูแลของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำข้อมูลไปเผยแพร่โดยไม่ได้รับอนุญาต

```

txt := oMnuList [Integer(oNde,data) - 1];
ListView1.ItemS.Clear;
While (txt <> '') do
begin
adtxt := GetToken (txt) ;
oitem := ListView1.ItemS.add ;
oem.subItems.add ('10 ');
oitem.subItems.add ('20 ');
oItem.subItems.add ('30');
end;
end;
procedure TForm1.RadioGroup1Click (Sender : TObject);
begin
listview.viewStyle := Tviewsty(RadioGroup1 > Itemindex);
end;
end.

```

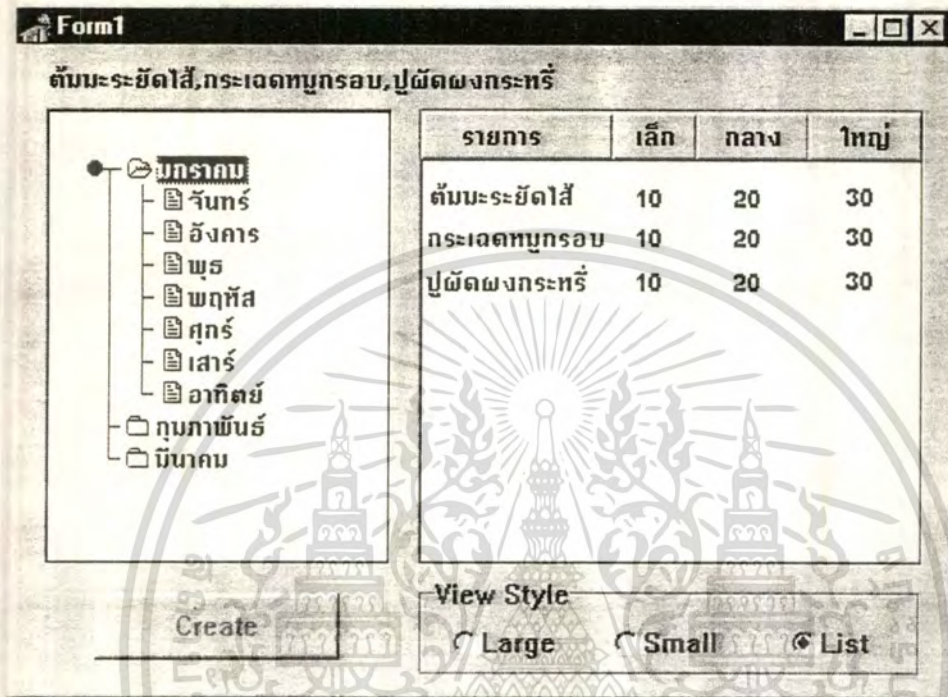
1. ให้ตรวจสอบว่าถ้าผู้ใช้ไม่ได้เลือกตรงไอคอนหรือชื่อ (htOnItem) ไม่ดำเนินการต่อ
2. หาโหนด ณ ตำแหน่งที่ผู้ใช้คลิกเมาส์ให้แก่ oNde
3. ถ้าเป็นโหนดที่ไม่ได้ให้ค่าข้อมูลไว้แก่ Data ไม่ดำเนินการต่อ
4. นำค่าใน Data มาทำเป็นอินเด็กซ์โดยลบด้วยหนึ่ง
5. แล้วอ่านสตริงของรายการนี้ มาให้เป็นค่า Caption ของกรอบ Label 1 เพื่อเป็นตัวอย่าง เมื่อไม่ต้องการใช้ ListView และให้ค่าแก่ txt
6. นำค่าใน txt มาทำเป็นโหนดของ ListView โดย
 - ให้ GetToken แยกเป็นชื่ออาหารให้ครั้งละ 1 ชื่อให้แก่ adTxt
 - ให้ Add ให้กำเนิดโหนดและผนวก เป็นรายการหลัก (เท่านั้น)
 - ให้ชื่ออาหารเป็นค่า Caption เพื่อนำออกแสดงและให้หมายเลขไอคอนแก่ ImageIndex (จะได้ผลดังในรูปที่ 3)
7. โดยที่แต่ละโหนดหลักของ ListView จะมีโหนดลูกได้ 1 ชั้น เพื่อแสดงผลเป็นตารางดังในรูปที่ 4-4 ซึ่งจะต้อง
 - ให้กำเนิดโหนดลูกและให้ค่าด้วย SubItemS.Add ดังที่ได้แสดงการให้ค่าไว้พอเป็นตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ให้ชื่อคอลัมน์แก่ Columns ในขณะเตรียมโปรแกรม ซึ่งตามรูป 4 ให้ไว้เป็นรายการเล็ก กลาง ใหญ่ รวม 4 คอลัมน์ (ถ้าไม่ให้จะไม่แสดงภาพ)

- เลือกรูปแบบของการแสดงคือ ViewStyle เป็น vsReport

สามารถเลือกรูปแบบของการแสดงได้ 4 แบบ ตามที่กล่าวมา ดังที่ได้จัดกรอบ RadioGroup 1 ไว้ และมีโปรแกรมอยู่ในกิจกรรม RadioGroup 1 Click



รูป 4 การแสดงผลในกรอบ ListView ในรูปของตาราง เมื่อเลือกปุ่ม Report

RichEdit

RichEdit เป็นเอดิเตอร์เช่นเดียวกับ Memo แต่มีขีดความสามารถสูงกว่าคือ ให้ผู้ใช้สามารถกำหนดรายละเอียดของฟอนต์ให้แก่อักษรเป็นตัว ๆ ได้ เช่น ทำเป็นหัวข้อ และรายละเอียดเกี่ยวกับการย่อหน้า ดังจะนำโปรแกรมเก่ามาเปลี่ยนเป็นใช้ RichEdit กล่าวคือ

ให้สร้างไดเรกทอรีใหม่แล้วลอกไฟล์ทุกไฟล์ของโปรแกรม TxtEdit มาไว้ในไดเรกทอรีใหม่นี้ (อาจเว้นไฟล์ที่เป็น backup) แล้วดำเนินการดังนี้คือ

1. ยลบกรอบ Memo 1 (เลือกแล้วกดคีย์ Del)
2. กำหนดคอมโปเนนต์เป็น RichEdit 1 แล้วให้ค่าพรอพเพอร์ตี้เช่นเดียวกับของ Memo

1 คือ

- Align ให้เป็น alClient
- HideSelection ให้เป็น False
- ScrollBars ให้เป็น sBoth

3. เปลี่ยนค่าจาก Memo 1 เป็น RichEdit 1 ทุกแห่ง (ใช้ Search/Replace)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ป้อนแก้โปรแกรมของกิจกรรม SrchMemo ให้เป็นดังในลิสติง 5 ทั้งนี้เพราะใน RichEdit มีกิจกรรม FindText ค้นหาคำอยู่แล้ว และต้องเปลี่ยนการให้ค่าแฟล็กใน Options ของ FindDialog 1 ใหม่คือ frDisableMatchCase และ frDisableWholeWord ให้เป็น False ทั้งคู่ เพื่อให้ผู้ใช้เลือกได้

ลิสติงที่ 5 RichEdit1.Paragraph.

```
function Tfrom1.SrchMemo( const srchTxtS : OpenString ) : Boolean ;
var findAt , srchStt , srchLen : Integer ; opt ; Tsearchtypes ;
begin
    srchStt := RichEdit1.SelStart + Lengthz( serchTexts);
    rchLen := RichEdit1.SelLength ;
    if SerchLen <= Length ( SerchTexts ) Then
        SerchLen := RichEdit .GetTextLend ;
    Opt := [ ] ;
with FinDislog1 do
begin if FrWholeword in Option Then
    opt := Opt + [stWholeWord] ;
if frmathCase in Options Then
    Opt := Opt + [ stMatchCase] end ;
findAt := RichEdit1.FindTexts( srchstt , srchLen , opt );
If findAt >= 0 Then
begin RichEdit1.SetFosucs;
    Ricshedit .SelSart := findAt ;
    RichEdit1.SelLength := Llength ( srchTextS);
    lblTopMag.Caption := ' search Ffond ' ;
    lblTopMag.Font.Color := clBlue ;
    SrchMemo := True end
else
begin lblTopMsg.captio := 'Search "' + SrchTtxts + '" NOT FOND ' ;
    lblTopmsg.Font.Colo := clRed ;
    srchMemo := False end
end ;
```

เอกสารนี้เป็นเอกสาร (NEW IN DELPHI 2.0) นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  if FontDialog1.Execute Then
  With RichEdit1.SelAttributes do
    begin
      Color := FontDialog1.Font.Color;
      Hight := FontDialog1.Font.Hight;
      Name := FontDialog1.Font.Name;
      Pitch := FontDialog1.Font.Pitch;
      Size := FontDialog1.Font.Size;
      Style := FontDialog1.Font.Style end;
end;

Procedure TForm1.SpeedButton2Click( Sender : TObject );
begin
  RichEdit1.SelAttributes.Color := clRed ;
  RichEdit1.SelAttributes.Name := 'Times New Roman ' ;
  RichEdit1.SelAttributes.Hight := 16 ;
  RichEdit1.SelAttributes.Style := [ fsBold, fsItalic ] ;
  RichEdit1.Paragraph.Alignment := { TaCenter } taLeftJustify ;
  RichEdit1.Paragraph.FirstIndent := 120 ;
  RichEdit1.Paragraph.LeftIndent := 100 ;
  { RichEdit1.Paragraph.RightIndent := 100 ;}
end ;

```

5. กำหนดคอมโปเนนต์เป็น SpeedButon 1 และ SpeedBulton 2 แล้วป้อนโปรแกรมตามลิสต์ดัง ก-7 เพื่อให้ผู้ใช้เลือกข้อความแล้วเลือกปุ่มทั้งสองนี้ดังนี้

5.1 ปุ่ม SpeedBulton 1 จะแสดงฟอนต์ไดอะล็อกบ็อกซ์ แล้วนำค่าที่ผู้ใช้เลือกใน FontDialog 1.Font มาให้แก่วัดข้อความที่ผู้ใช้เลือกคือ SelAttributes

5.2 ปุ่ม SpeedBulton 2 จะให้ค่าที่เตรียมไว้แก่วัดข้อความที่ผู้ใช้เลือก ทั้งนี้เพียงเพื่อเป็นตัวอย่าง ไม่ควรใช้ฟอนต์ไดอะล็อกบ็อกซ์ ควรจัดคอนโทรลให้ในแต่ละเรื่อง ดูตัวอย่างได้จาก โปรแกรม WordPad (อยู่ใน Accessories) เช่น

- ฟอนต์ ใช้กรอบรายการฟอนต์
- สี ใช้กรอบรายการสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโปเนนต์อื่น

คอมโปเนนต์ที่เหลือในหน้า Win 95 เป็นคอมโปเนนต์ที่ปรับปรุงจากของเดิม ในด้านความสวยงามบ้าง เพิ่มขีดความสามารถบ้าง มีวิธีการใช้เช่นเดียวกับหรือใกล้เคียงกับของเดิมได้แก่

1. HeaderComponent ใช้แทน Header ของเดิม
2. StatusBar ปรับปรุงมาจาก Panel เพื่อใช้เป็นกรอบสเตตัส ทางด้านล่างของฟอร์ม
3. TrackBar มีลักษณะการใช้เช่นเดียวกับ ScrollBar แต่ดูสวยงามกว่า
4. ProgressBar ของเดิมคือ Gauge เป็นคอมโปเนนต์ตัวอย่างอยู่ในหน้า Samples
5. UpDown ของเดิมมี SpinButton กับ SpinEdit สำหรับ UpDown หากใช้ตัวเดียวคือ SpinButton หากใช้ร่วม (associate) กับกรอบ Edit คือ SpinEdit
6. HotKey เป็นคอนโทรลใหม่ เพื่อให้ผู้ใช้กำหนดคีย์ด่วน (hot key) ให้แก่คอนโทรล

3.6 Object Repository

ในเคลฟ 2.0 ใช้ Object Repository แทน Gallery ในรุ่น 1.0 ทำหน้าที่เก็บต้นแบบของโปรเจกต์ฟอร์ม ฯลฯ เพื่อใช้งานร่วมกันทั้งในโปรเจกต์เดียวกันและต่างโปรเจกต์ มีต้นแบบที่สร้างไว้ก่อนแล้วและที่เราจะสร้างเพิ่มตามที่จะกล่าวในหัวข้อ “การสร้างฟอร์มต้นแบบ”

ให้นำมาใช้ได้โดยเลือกรายการ File/New จะแสดงไดอะล็อกบ็อกซ์ New Item ดังในรูปที่ ค-5 แสดงต้นแบบที่เก็บอยู่ใน ObjectRepository แบ่งเป็น 6 หน้า ภายในแต่ละหน้าจะมีต้นแบบต่าง ๆ ให้เลือกใช้



รูป 5 ไดอะล็อกบ็อกซ์ New Items ซึ่งแบ่งเป็น 6 หน้า ในหน้า Forms จะแสดงไอคอนของฟอร์มต้นแบบต่าง ๆ ตามรูปนี้ ให้เก็บเมื่อได้ สร้างฟอร์มต้นแบบเป็น My Origin Form แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่จํากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การสืบสกุลของหน้าต่าง

เมื่อเปิดฟอร์มใหม่ เช่น เมื่อเปิดโปรเจกต์ใหม่ ได้ฟอร์มเป็น Form 1 จะเท่ากับเป็นการสร้างสกุล Tform 1 ให้สืบสกุลจาก Tform ซึ่งเป็นหน้าต่าง เป็นการสืบสกุลของหน้าต่าง

แต่ถ้าต้องการจะสร้างสกุลใหม่สืบสกุลจาก Tform 1 ซึ่งเป็นการสืบสกุลอีกทอดหนึ่งในเดลไฟ 1.0 ทำไม่ได้ มาทำได้ในเดลไฟ 2.0 ดังที่จะกล่าวในหัวข้อนี้

จุดประสงค์

จุดประสงค์ของการสร้างสกุลคือ เพื่อใช้เก็บไว้ใช้ และเพื่อการสืบสกุล

ส่วนจุดประสงค์ของการสืบสกุลคือ เพื่อใช้เปลี่ยนแปลงและต่อเติมโปรแกรมเดิมในโปรแกรมใหม่

บรรพบุรุษและการสืบสกุล

เมื่อสร้างสกุล AA สืบสกุลจากสกุล A จะได้ AA เป็นผู้สืบสกุล (descendent) ของสกุล A และสกุล A เป็นบรรพบุรุษ (ancestor) ของสกุล AA และอาจสร้างสกุล AB, AC..สืบสกุลจากสกุล A จึงได้สกุล A เป็นสกุลต้นตระกูลของสกุลเหล่านี้ เช่นกันอาจได้สกุล AA เป็นต้นตระกูลของสกุล AAA, AAB...

ฟอร์มต้นแบบ

เราอาจสร้างสกุลธรรมดาโดยทั่วไปรวมทั้งที่ให้สืบสกุลจากสกุลที่เป็นคอมโปเนนต์ไว้ในยูนิิต(โดยไม่ทำสกุลใหม่นี้เป็นคอมโปเนนต์) แล้วนำยูนิิตไปใช้โดยสร้างสกุลใหม่สืบสกุลจากสกุลในยูนิิตได้โดยตรงไม่ต้องพึ่งเดลไฟ

แต่ถ้าเป็นสกุลของฟอร์ม เช่น เมื่อเปิดโปรเจกต์ใหม่จะได้ Form 1 เป็นออกเจกต์ของ Tform 1 สืบสกุลจาก Tform ถ้าเราต้องการเก็บสกุล Tform 1 ไว้เป็นสกุลต้นตระกูล เช่น เพื่อสร้างผู้สืบสกุลเป็น TFormX มีฟอร์มเป็น FormX

เนื่องจากอาจต้องกำหนดคอมโปเนนต์เพิ่มในฟอร์ม FormX ในเดลไฟจึงให้เราเก็บสกุลแต่ความหมายจะหนักไปทางฟอร์มคือ TFormX ซึ่งจะขอเรียกเป็นฟอร์มต้นแบบไว้ใน Object Repository ตามที่จะกล่าวในหัวข้อ “การสร้างฟอร์มต้นแบบ” และให้นำมาใช้คือนำมาสืบสกุลได้ดังที่จะกล่าวในหัวข้อ “การใช้ฟอร์มต้นแบบ”

3.8 ฟอร์มต้นแบบ

ไคเรคทอรีของฟอร์มต้นแบบ

โดยที่ในการสร้างฟอร์มต้นแบบ จะต้องเก็บไฟล์ของฟอร์มต้นแบบ (ไฟล์ .pas .dfmและ .dcu) ไว้ในไคเรคทอรีประจำ เพื่อเก็บไฟล์ของฟอร์มต้นแบบทั้งหมดคือที่จะสร้างต่อไปในอนาคต เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ด้วยจึงขอให้สร้างไคเรคทอรีใหม่ เช่น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C: mydelphi \ myobjrep

และต่อไปจะขออ้างถึงไคเรคทอรีนี้ว่า myobjrep

การสร้างฟอร์มต้นแบบ

เพียงเพื่อเป็นตัวอย่างในการสืบสกุลเท่านั้น จะสร้างฟอร์มต้นแบบให้ประกอบด้วยกรอบ Edit 1 กรอบ Label 1 ปุ่ม Bulton 1 และ Bulton 2 โดยที่เมื่อผู้ใช้ป้อนข้อความในกรอบ Edit 1 แล้วเลือกปุ่ม Bulton 1 จะนำข้อความในกรอบ Edit 1 ไปแสดงในกรอบ Label 1 และถ้าเลือกปุ่ม Bulton 2 จะแสดงเมสเสจบ็อกซ์ จะดำเนินการได้ดังนี้คือ

1. เปิดโปรเจกต์ใหม่ (ยังไม่ต้องเก็บลงไฟล์)
2. เปลี่ยนชื่อ Form 1 เป็น OrgForm โดยให้ค่าแก่ Name ทั้งนี้เพื่อไม่ให้ขัดต่อการตั้งชื่อของเคิลไฟ และลบข้อความที่ Caption เพื่อให้เป็นฟอร์มที่ไม่มีชื่อที่ไคเดิลบาร์
3. กำหนดคอมโปเนนต์เป็น Edit1, Label 1, Bulton 1 และ Bulton 2 ณ ตำแหน่งที่เห็นเหมาะสมแล้วป้อนโปรแกรมตามลิสติงที่ 6 เสร็จแล้วให้ทดลองรันลิสติงที่ 6

```
procedure TOrgForm.Button1Click ( Sender : Tobject );  
begin  
    Label1.Caption := Edit1.Text ;  
end ;  
procerurd TorgForm.Button2Click ( Sender : Tobject );  
begin  
    ShowMessage ( 'Here , I Am ' );  
end ;
```

4. เก็บไว้ใน Object Repository โดย

4.1 เลือกรายการ File/ Save As เพื่อเก็บไฟล์ของฟอร์มต้นแบบไว้ในไคเรคทอรี myobjrep โดยให้ชื่อไฟล์เป็น orgformO.pas

4.2 เลือกในฟอร์มในแอกทิฟแล้วคลิกเมาส์.ขวาเพื่อเปิดเมนูลอย เลือกรายการ Add To Repository หรือเลือกรายการ Project/Add To Reposibory ที่เมนูบาร์ จะแสดงไดอะล็อกบ็อกซ์ ดังในรูปที่ 6 ให้ป้อนค่าต่าง ๆ ดังนี้

- Title ป้อนชื่อที่จะให้แสดงเป็นชื่อต้นแบบ เช่น My Original Form

- Description ป้อนคำอธิบาย เช่น Exanple Form

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเชิงนี้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Page ป้อนชื่อหน้าคือ Forms
- Author ป้อนชื่อผู้เขียน โปรแกรม
- หากได้เตรียมภาพไว้ไฟล์ .ico ให้โหลดด้วยปุ่ม Browse
- เลือกปุ่ม OK

รูปที่ 6 ใคจะเลือกบ็อกซ์ที่ใช้ในการเก็บฟอร์มต้นแบบไว้ใน Object Repository

5. ปิดโปรเจกต์โดยเลือกรายการ File/Close All หรือเปิดโปรเจกต์ใหม่ และให้ปฏิเสธคำถามที่จะให้เก็บลงไฟล์ เนื่องจากจะไม่เก็บโปรเจกต์ไว้

สำหรับการรันทดสอบ ในบางกรณีอาจต้องสร้างฟอร์มต้นแบบที่ไม่สมบูรณ์ในตัวเองรันไม่ได้ จึงเพียงแต่ให้ผ่านการคอมไพล์ และโดยทั่วไปโปรแกรมประเภทนี้จะยาว จึงควรอย่างยิ่งที่จะเก็บลงไฟล์ไว้เป็นระยะ ๆ ตามปกติ เมื่อรันทดสอบหรือคอมไพล์เรียบร้อยแล้ว จึงดำเนินการตามข้อ 4 นั่นคือสามารถนำโปรแกรมที่เขียนไว้แล้วมาทำเป็นฟอร์มต้นแบบได้ทุกโปรแกรม และถ้ามีหน้าต่างหลายบานให้เลือกบานที่ต้องการ

การยกเลิกฟอร์มต้นแบบ

ฟอร์มต้นแบบที่ไม่ใช่แล้ว ให้ยกเลิกได้โดยเลือกรายการ Tools/Repository จะแสดงไดอะล็อกบ็อกซ์ ให้

- เลือกหน้าคือ Forms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกชื่อฟอร์ม เช่น My Original Form
- เลือกปุ่ม Delete Object

การใช้ฟอร์มต้นแบบ

การใช้ฟอร์มต้นแบบเป็นไปตามที่กล่าวมาคือ เพื่อใช้ เปลี่ยนแปลง และต่อเติม อย่างใดอย่างหนึ่งหรือร่วมกัน ซึ่งอาจจำแนกการนำมาใช้ได้เป็น 2 เรื่อง คือ

1. วิธีการนำมาใช้ ได้แก่

- Copy เป็นการลอกฟอร์มมาใช้เช่นเดียวกับที่ใช้อยู่ในเคลฟ 1.0
- Inherit เป็นการนำมาใช้ด้วยการสืบสกุล เป็นวิธีการที่ถูกต้องตามหลักของการเขียนโปรแกรมในแบบ โอโอพี ที่ใช้อยู่ในเคลฟ
- Use เป็นการนำฟอร์มต้นแบบมาใช้โดยตรง การเปลี่ยนแปลงใด ๆ จึงมีผลต่อฟอร์มต้นแบบโดยตรง เป็นวิธีการที่สวนทางกับหลักการของ โอโอพี แต่ก็มีประโยชน์ เช่น เมื่อโปรแกรมที่จะเตรียมให้แก่ฟอร์มต้นแบบ ไปพร้อม ๆ กัน

2. ใช้ในฐานะ คือจะนำมาใช้เป็นหน้าต่างหลัก หรือหน้าต่างรอง

ดังจะแสดงตัวอย่างการนำมาใช้ด้วยการสืบสกุลในฐานะหน้าต่างหลักในหัวข้อ ๖.๒ นี้

การสืบสกุลจากฟอร์มต้นแบบ

ตามที่กล่าวว่าการสืบสกุลจากฟอร์มต้นแบบเพื่อ ใช้เปลี่ยนแปลงและต่อเติมนั้นความจริงแล้วเป็นการยากที่จะแยกเรื่องทั้งสามนี้ออกจากกัน ดังที่ในตัวอย่างนี้พยายามจะแยกเป็น

- การใช้ จะใช้ปุ่ม Bulon2 ตามที่ออกแบบไว้คือ ให้แสดงเมสเสจบ็อกซ์
 - การเปลี่ยนแปลงจะเปลี่ยนแปลงการทำงานของปุ่ม Bulon 1 ให้นำข้อความที่ผู้ใช้ป้อนในกรอบ Edit 1 ไปเก็บไว้ในกรอบรายการ ตามที่จะกำหนดเพิ่มเป็น ListBox 1
 - การต่อเติม จะเพิ่มปุ่ม Bulon 3 เพื่อลบข้อความในกรอบ LisBox 1
- จะเตรียม โปรแกรมด้วยการสืบสกุลจากฟอร์มต้นแบบ ได้ดังนี้

1. เปิด โปรเจกต์ใหม่ แต่ยังไม่ต้องเก็บลงไฟล์ จะได้ฟอร์ม Form1 ตามปกติ
2. เลือกรายการ File/New ซึ่งจะแสดงไดอะล็อกบ็อกซ์(คล้ายกับในรูป ค-5) ให้
 - เลือกหน้า Forms
 - เลือกฟอร์มต้นแบบเป็น My Origin Form
 - เลือกปุ่ม Inherit (จะเห็นว่ามิปุ่มให้เลือก 3 ปุ่มคือ Copy, Inherit และ Use ตามที่

กล่าวมา)

- เลือกปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จะได้ฟอร์ม (หน้าต่าง) อีก 1 ฟอร์มมีชื่อเป็น OrgForm 2 ซึ่งอาจใช้ชื่อนี้ แต่ควรเปลี่ยนใหม่โดยให้ค่าแก่ Name เช่น ให้เป็น NewForm1

สำหรับทางด้านของโปรแกรมในหน้าต่างยูนิท จะเห็นว่าเคลฟได้สร้างสกุล TNewForm สืบสกุลจาก TOrgForm (แทนเรา) เช่นเดียวกับที่เคยสร้างสกุล TForm1 สืบสกุล TForm และเป็น การสืบสกุลโดยสมบูรณ์แล้ว และจะใช้ฟอร์มใหม่นี้ได้ดังนี้

3.1 ใช้เป็นหน้าต่างรอง ได้เช่นเดียวกับที่เปิดด้วย File/ New Form

3.2 ใช้เป็นหน้าต่างหลัก คือตามตัวอย่างนี้ซึ่งจะต้องยุบฟอร์ม Form1 โดยเลือกรายการ View/Project Manager จะแสดงหน้าต่าง Project Manager ให้

- เลือกฟอร์ม Form 1

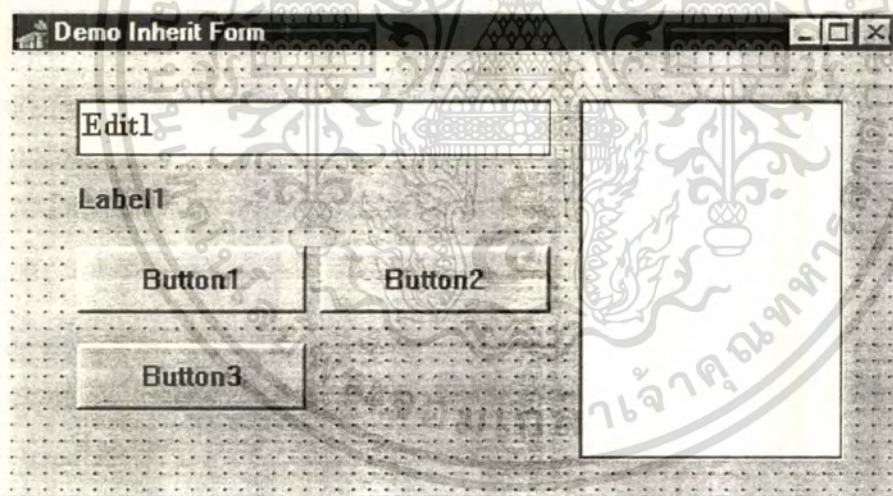
- เลือกปุ่ม Remove Unit คือปุ่มที่มีเครื่องหมายลบ (ให้ปฏิเสธการเก็บลงไฟล์)

- ฟอร์ม Form1 จะหายไปเหลือแต่ NewForm1 เป็นหน้าต่างหลัก และควรที่จะได้เก็บลงไฟล์ไว้ครั้งหนึ่งก่อน

4. ปรับปรุงโปรแกรมโดย

- กำหนดคอมโปเนนต์เป็น ListBox 1 และ Button3 จะได้ผลดังในรูป 7

- ป้อนโปรแกรมตามลิสต์ที่ 7 ซึ่งเมื่อรันจะได้ผลตามที่กล่าวมา



รูป 7 ฟอร์มของผู้สืบสกุลหลังจากกำหนดคอมโปเนนต์เพิ่มแล้ว

ลิสต์ที่ 7

```
procedure TnewForm1.Button1Click (sender : TObject) ;
begin
    inherited;
    ListBox1.Items.Add ( Edit1.Text);
end ;

procedure TnewForm1.Button3Click ( Sender := TObject );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    inherited;
    ListBox1.Clear ;
end ;
```

คำอธิบายเสริม

1. ตามที่กล่าวมานี้ถึงแม้จะยกตัวอย่างเป็น โปรแกรมขนาดเล็กมาก แต่ก็ครอบคลุมเนื้อหาของ การสร้างและการใช้ฟอร์มต้นแบบทั้งหมด

2. การสร้างฟอร์มต้นแบบในงานจริงจะพบในโปรแกรมขนาดใหญ่ ที่ใช้หน้าต่างหลาย บาน คือหลายฟอร์ม และในหน้าต่างเหล่านี้มีส่วนที่เหมือนกัน จึงจะนำส่วนที่เหลือกันไปทำฟอร์ม ต้นแบบ

3. ในโปรแกรมนี้ให้เปลี่ยนแปลงการทำงานของปุ่ม Bulon 1 โดยเพิ่มกรอบรายการ ListBox1 ซึ่งถ้าต้องการตัดกรอบ Label 1 ออกจะทำไม่ได้ เพราะเคลไฟจะไม่ให้ยุบคอมโปเนนต์ ในฟอร์มต้นแบบ วิธีการคือ

- ให้กดแสดงกรอบ Label1 โดยให้ค่า Visible เป็น False ข้อเสียคือทำให้เสียรีซอร์ส ไปจำนวนหนึ่งโดยไม่ได้ใช้

- หรือ เลือกวิธีการนำมาใช้เป็น Copy แต่ก็มีข้อเสียเช่นกันก็จะมีโปรแกรมเดียวกัน หลายชุด

ในการสร้างฟอร์มต้นแบบจึงควรตรวจสอบที่จะไม่ให้เกิดกรณีเช่นนี้

4. เมื่อนำฟอร์มมาใช้ด้วย Use สามารถปรับปรุง โปรแกรมเดิมได้ทั้งหมด แต่จะต้องไม่ เปลี่ยนค่าใน Name ของฟอร์ม

3.8 Finalization

ในเคลไฟ 1.0 เราสามารถกำหนดส่วน initialization ไว้คอนท้ายของยูนิท เพื่อให้รัน โปรแกรมในส่วนนี้ตามลำดับของยูนิทที่บ่งใช้ใน uses ก่อนแสดงหน้าต่างหลัก

ในเคลไฟ 2.0 ได้เพิ่มส่วน finalization อีก 1 ส่วน โดยกำหนดว่าจะต้องอยู่หลังส่วน initialization เช่น

```
initialization
    ShowMessage ('Start ');
finalization
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แบ่งไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MessageBeep (0)

end.

จะแสดงเมสเสจบ็อกซ์ก่อนเปิดหน้าต่างและส่งเสียงบีบหลังหน้าต่างแล้ว

3.8 เปิดโปรเจกต์ใหม่

ให้เริ่มการโปรแกรมทุกครั้งด้วยการเปิดโปรเจกต์ใหม่ โดยเลือกรายการ File/New Project หรือใช้โปรเจกต์ที่เคลฟเปิดให้เมื่อเริ่มต้นดังเช่นในรูปที่ 8 เป็นโปรเจกต์ใหม่เช่นกัน

เมื่อเปิดโปรเจกต์ใหม่เคลฟจะเตรียมหน้าต่างให้ 2 บาน คือหน้าต่างฟอร์มคือ Form 1 เพื่อให้กำหนดคอมโปเนนต์ และหน้าต่างยูนิตคือ Unit1 เพื่อใช้ป้อนโปรแกรม

หมายเหตุ

การเปิดโปรเจกต์ใหม่ตามที่กล่าวมานี้ หมายถึง ได้ติดตั้งเคลฟโดยไม่ได้เซตค่าใด หากได้เซตปุ่ม Options/Environment/Gallery/Uses on New Project ไว้ จะแสดงรูปแบบของฟอร์ม ให้เลือกเป็น Blank project

3.9 รันโปรแกรมทดสอบและยุติการรัน

สามารถรันโปรแกรมทดสอบได้ทุกโอกาสที่ต้องการ แม้แต่เมื่อเพิ่งเปิดโปรเจกต์ใหม่ โดยเลือกรายการ Run/Run หรือกดคีย์ F9 หรือเลือกปุ่ม Run ที่สปีคบาร์

เมื่อรันโปรแกรมทดสอบแล้ว สามารถยุติการรันได้โดยเลือกรายการ Close จาก System menu (คือปุ่มที่มุมบนซ้ายของหน้าต่าง หรือปุ่ม X ของวินโดวส์ 95)

ก่อนที่จะทดลองรันโปรแกรม หากเป็นโปรแกรมขนาดใหญ่ ควรได้คอมไพล์ก่อน โดยเลือกรายการ Compile/Compile แล้วเก็บลงไฟล์ตามที่กล่าวต่อไป

เมื่อรันโปรแกรมทดสอบ เคลฟจะคอมไพล์ให้ก่อนแล้วจึงรัน

โปรแกรม MyNull

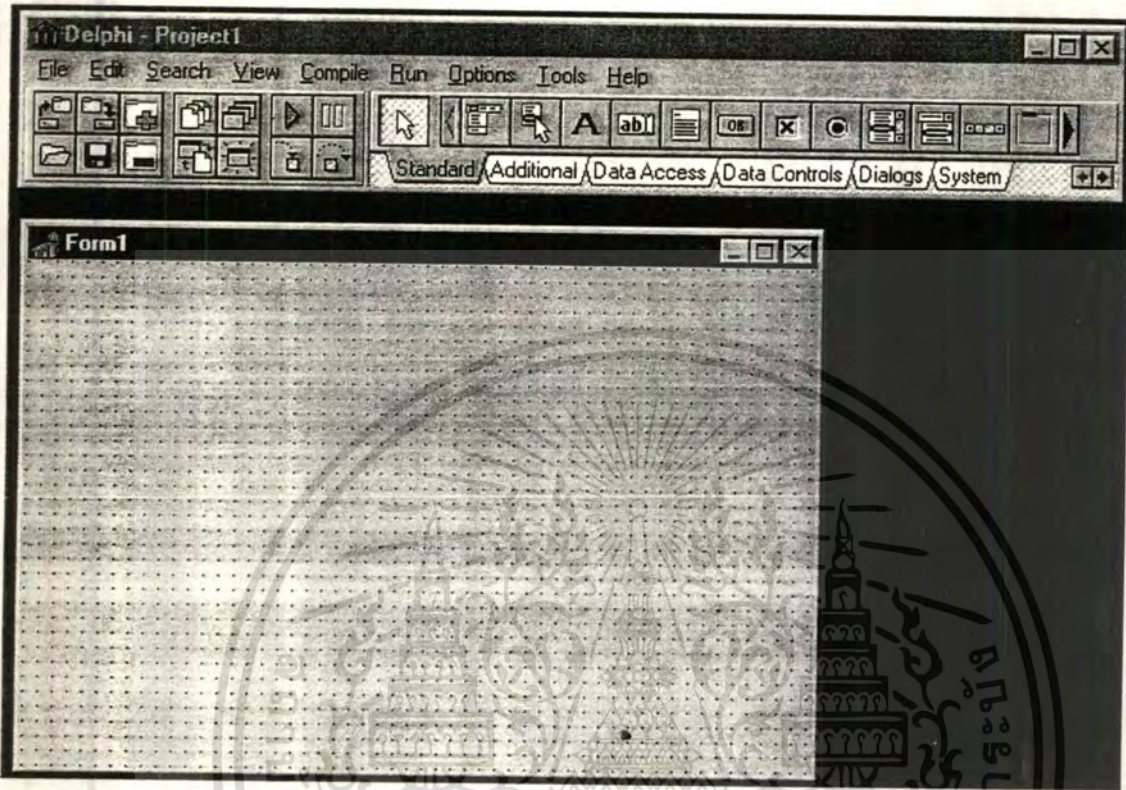
เพื่อแสดงถึงการโปรแกรมด้วยเคลฟโดยที่ยังไม่ต้องเตรียมโปรแกรมเลย ซึ่งจะได้เป็นโปรแกรมเปล่า ขอให้ดำเนินการดังนี้คือ

1. เปิดโปรเจกต์ใหม่
2. รันโปรแกรมทดสอบ

จะได้ผลบนจอภาพดังรูปที่ ก-1 แสดงหน้าต่างของโปรแกรม MyNull มีชื่อหน้าต่างว่า

Form1 นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นโปรแกรมเปล่าที่ผู้ใช้ใช้งานไม่ได้ แต่ก็ เป็น Windows application คือเป็นโปรแกรมบนวินโดวส์ที่สมบูรณ์ สามารถย่อ ขยาย ย้ายหน้าต่างได้ มินิไมส์และแมกซิไมส์ได้ และยุติการรันได้โดยเลือกรายการ Close



รูป 8 แสดงหน้าต่าง Form1 ของโปรแกรม Mynull ซึ่งเป็นโปรแกรมเปล่าในขณะรันทดสอบ

โปรแกรม MyBeep

เพื่อแสดงถึงงานที่จะให้ผู้ใช้ทำได้ คือ เมื่อรันโปรแกรมแล้วหากผู้ใช้คลิกเมาส์ในหน้าต่างจะให้ส่งเสียงดัง 1 ครั้ง จะเตรียมโปรแกรมได้ดังนี้

1. เปิดโปรเจกต์ใหม่ (ขณะนี้โปรเจกต์ที่อ้างถึงเป็นโปรแกรม MyNull ยังคงเป็นโปรเจกต์ใหม่เพราะยังไม่ได้ใช้เตรียมโปรแกรม)

2. ป้อนโปรแกรม โดย

2.1 เลือกหน้า Events ในหน้าต่าง Object Inspector เลือกอีเวนต์ เป็น OnClick แล้วดับเบิลคลิกในกรอบทางขวา (ของหัวข้อนี้) จะแสดงโครงโปรแกรมในหน้าต่าง Unit 1 ดังในรูปที่ 9

2.2 ให้ป้อนโปรแกรมคือคำสั่ง MessageBeep ให้เป็นดังลวดสีที่ 8

3. รันโปรแกรมทดสอบ จะได้ผลการรันตามที่กล่าวมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UNIT1.PAS
procedure TForm1.Button1Click(Sender: TObject);
begin
|
end;

end.

```

28: 1 Modified Inset

Unit1

รูป 9 แสดงหน้าต่างยูนิท Unit1 หลังจากดับเบิลคลิกที่กรอบทางขวาของหัวข้อ OnClick ในหน้า Events

ลิสต์ที่ 8

```

procedure TForm1.FormClick ( Sender : TObject );
begin
    MessageBeep ( 0 );
end ;

```

3.10 การเก็บลงไฟล์

เมื่อคอม파일หรือรันแล้ว หากเป็นไปตามที่กล่าวมาจะได้ไฟล์ project1.exe อยู่ในไดเรกทอรี C:\delphi ซึ่งถ้าเป็นโปรแกรมที่สมบูรณ์ ไฟล์ .exe คือไฟล์ที่จะนำไปใช้งาน

การเก็บลงไฟล์ หมายถึง การเก็บไฟล์ในโปรเจกต์ทั้งหมด (หลายไฟล์) ไว้ในดิสก์ เพื่อเก็บไว้เป็นหลักฐาน ตัวอย่าง หรือเพื่อการปรับปรุงต่อไป

ซึ่งถ้าถือตามนี้ก็ควรจะเก็บลงไฟล์เมื่อได้ ป้อน แก้ ตรวจสอบโปรแกรมเป็นที่พอใจแล้ว แต่ในหนังสือนี้จะถือว่าและขอแนะนำให้เก็บลงไฟล์เมื่อเปิดโปรเจกต์ใหม่ 1 ครั้ง และเป็นระยะๆ ตามที่เห็นควร ทั้งนี้เพื่อ

1. ให้ได้ชื่อไฟล์ .exe
2. รวบรวมไฟล์ทุกไฟล์ของโปรแกรมหนึ่งๆ คือของโปรเจกต์หนึ่งๆ ไว้ในไดเรกทอรีเดียวกัน
3. ถึงแม้ในเคลฟจะป้อนกันไว้แน่นหนา แต่โอกาสที่จะสูญเสียไฟล์ในหน่วยความจำยังมีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการเก็บลงไฟล์ เช่น ถ้าต้องการเก็บของโปรแกรม MyBeep ลงไฟล์ จะดำเนินการได้ดังนี้

1. เตรียมไดเรกทอรีเช่น

D:\bkondphi\02start\1mybeep

2. เตรียมชื่อไฟล์ .dpr หนึ่งชื่อ เช่น ให้เป็น mybeep.dpr ซึ่งเมื่อคอมไพล์แล้วจะได้เป็นไฟล์ mybeep.exe

3. เตรียมชื่อไฟล์ .pas หนึ่งชื่อ ซึ่งในหนังสือนี้จะนำชื่อไฟล์ .dpr มาต่อท้ายด้วยเลข 1 เช่น เป็น mybeep 1.pas (ชื่อไฟล์ทั้งสองนี้จะต้องไม่ซ้ำกัน)

4. เลือกรายการ File/Save Project As จะแสดง ไดอะล็อกบ็อกซ์ให้ป้อนชื่อไฟล์ ให้ดำเนินการดังนี้

4.1 เลือกไดเรกทอรีตามที่เตรียมไว้ในข้อ 1

4.2 ป้อนชื่อไฟล์เป็น mybeep1.pas เลือกปุ่ม OK

4.3 จะแสดงไดอะล็อกบ็อกซ์ใหม่ ให้ป้อนชื่อไฟล์เป็น mybeep.dpr (คือจะให้ป้อนชื่อไฟล์ .pas ก่อน .dpr)

หมายเหตุ

1. ที่ให้ชื่อไฟล์ .pas ตามที่ชื่อไฟล์ .dpr ต่อด้วยหมายเลข เช่น mybeep1.pas เพราะต่อไปอาจมีการเปิดฟอร์มหรือยูนิคเพิ่ม จะได้ให้เป็นหมายเลข 2 และต่อๆไป

2. ทำให้ตั้งชื่อไฟล์ .dpr ได้ไม่เกิน 7 อักขร ซึ่งชื่อนี้จะเป็นชื่อไฟล์ .exe เป็นชื่อโปรแกรมจริงแต่โดยที่ต้องการให้ชื่อไฟล์ที่มีนัยชื่อหัวชื่อเช่น “โปรแกรม MyBeep” อื่นได้ความพอสมควร ชื่อทั้งสองนี้อาจไม่ตรงกัน เช่น ButtonBeep กับ btibeep

3.11 การกำหนดคอมโปเนนต์

หลักใหญ่ของการโปรแกรมด้วยเคลฟ คือ การนำคอมโปเนนต์มาใช้ โดยกำหนดในฟอร์มให้คำพรอพเพอร์ตี้ส์เพื่อปรับแต่งให้เป็นตามที่ต้องการ ในหัวข้อนี้จะกล่าวถึงการกำหนดคอมโปเนนต์กับเรื่องที่เกี่ยวข้องกัน คือ

กำหนดคอมโปเนนต์

เมื่อต้องการกำหนดคอมโปเนนต์ ดังเช่นเมื่อต้องการกำหนดคอมโปเนนต์เป็นปุ่มควบคุมคือ Button ดังแสดงในรูปที่ 10 จะกำหนดได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

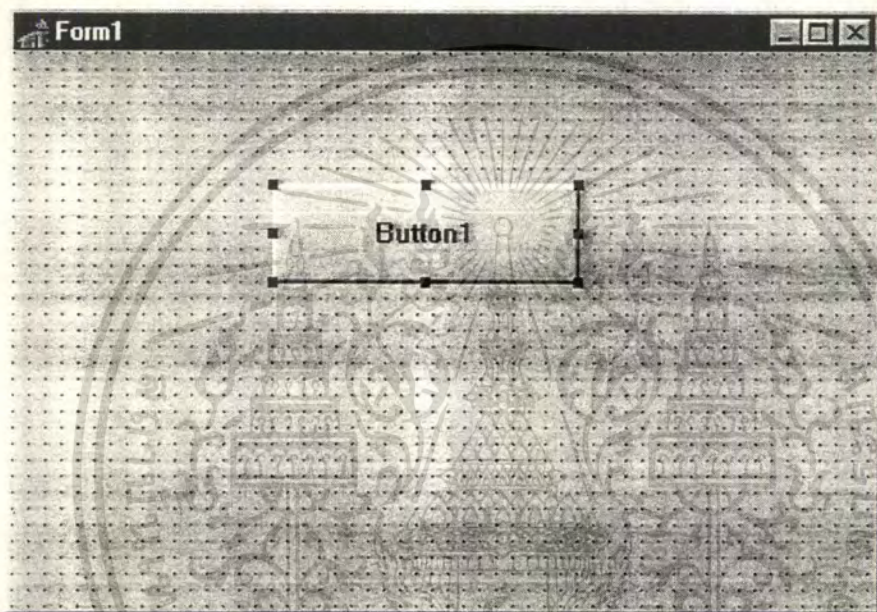
1. เลือกคอมโปเนนต์เป็น Button ในกลุ่มคอมโปเนนต์หน้า Standard (ปุ่มที่ 6 จากซ้าย ไม่นับปุ่มหัวลูกศร)

2. นำไปกำหนดในฟอร์มโดย

2.1 คลิกเมาส์ ณ ตำแหน่ง (จุดมุมบนซ้าย) ที่ต้องการ จะได้ขนาดมาตรฐาน หรือ

2.2 ดึงกรอบเพื่อให้ได้ตำแหน่งและขนาดที่ต้องการ

จะได้คอมโปเนนต์ตามที่ต้องการเช่นดังรูปที่ ก-3 คือปุ่ม Button สำหรับการกำหนดขนาดและตำแหน่งขอให้ประมาณจากรูปประกอบ และขอให้ถือว่าเป็นเพียงตัวอย่างแนะนำเท่านั้น อาจกำหนดได้ตามที่เห็นควร



รูป 10 แสดงผลของการกำหนดคอมโปเนนต์เป็น ปุ่ม Button1 และอยู่ในสถานะเลือก

สถานะเลือก

เมื่อกำหนดหรือเบี่ยงคอมโปเนนต์ใด คอมโปเนนต์นั้นจะอยู่ในสถานะเลือก มีปุ่มเล็ก ๆ 8 ปุ่ม ซึ่งจะขอเรียกว่า “ปุ่มเลือก” ล้อมรอบ

ซึ่งถ้ากำหนดคอมโปเนนต์ใหม่ หรือเลือกคอมโปเนนต์อื่น หรือเลือกฟอร์ม คือ คลิกเมาส์ ในฟอร์ม (ตรงที่ว่าง) สถานะเลือกนี้จะหมดไป ไปอยู่คอมโปเนนต์ตัวใหม่หรือฟอร์ม แต่ฟอร์มไม่แสดงปุ่มเลือก

เมื่อคอมโปเนนต์หรือฟอร์มอยู่ในสถานะเลือก จะแสดงชื่อออปเจกต์ (ตามที่จะกล่าวต่อไป) เช่นเป็น Button 1 หรือ Form 1 ที่กรอบอิดิตของกรอบคอมโป ที่ใต้ไคเดิลบาร์ของหน้าต่าง Object Inspector จึงเป็นอีกแห่งหนึ่งที่จะดูว่าคอมโปเนนต์ใด หรือโดยเฉพาะอย่างยิ่งคือ ฟอร์มอยู่ในสถานะเลือกหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

ที่ชื่อออปเจ็กต์ตามที่กล่าวมานี้ จะแสดงชื่อสกุลกำกับด้วย เช่น TButton:Button1 ตามที่จะกล่าวต่อไป

การย้าย ย่อ และขยายขนาด

เมื่อต้องการย้ายคอมโปเนนต์ ให้คลิกเลื่อนไปไว้ ณ ตำแหน่งที่ต้องการ และถ้าต้องการย่อหรือขยายขนาดให้เลือกแล้วคลิกเมาส์ ณ ปุ่มเลือกที่อยู่ในทิศทางที่ต้องการ แล้วเลื่อนเมาส์เพื่อย่อหรือขยายขนาดตามความต้องการ แล้วปล่อยมือ

การยุบ

เมื่อกำหนดคอมโปเนนต์แล้ว หากกำหนดผิดหรือไม่ต้องการใช้ให้ยุบได้โดยเลือกคอมโปเนนต์นั้นแล้วกดคีย์ Del

หมายเหตุ

1. ขอให้ทดลองกำหนดคอมโปเนนต์โดยเฉพาะในหน้า Standard เพื่อทำความเข้าใจกับคอมโปเนนต์เหล่านี้แล้ว ย่อ ขยาย รวมทั้งยุบด้วย แต่คงให้มีคอมโปเนนต์จำนวนหนึ่งในฟอร์มโปรแกรมทดสอบแล้วทดลองเลือกคอมโปเนนต์เหล่านั้นด้วยเมาส์และคีย์บอร์ด (ใช้คีย์ Tab กับ Enter)

2. เมื่อกำหนดคอมโปเนนต์ ขอให้สังเกตชื่อคอมโปเนนต์ดังเช่น Button 1 ด้วย

ชื่อคอมโปเนนต์

ในเคลฟได้สร้างสกุลไว้เป็นจำนวนมาก โดยตั้งชื่อสกุลขึ้นต้นด้วยอักษร T เช่น Tbutton แล้วนำสกุลจำนวนหนึ่งไปทำเป็นคอมโปเนนต์ ใช้ชื่อชนิดของคอมโปเนนต์ตามชื่อสกุล โดยตัดอักษร T เช่นเป็น Button

เมื่อเรากำหนดคอมโปเนนต์ในฟอร์ม จะได้ชื่อคอมโปเนนต์ตามชื่อชนิดต่อท้ายด้วยหมายเลข เช่น Button1 ซึ่งถ้ากำหนดคอมโปเนนต์ชนิดนี้อีก จะได้เป็น Button2 และต่อ ๆ ไป

โดยที่คอมโปเนนต์จะมีสถานะอยู่ 2 สถานะ คือ

1. เป็นคอนโทรลในสายตาของผู้ใช้ คือเห็นเป็นปุ่มและกรอบต่าง ๆ มีชื่อตามที่เห็น เช่น เป็นปุ่ม Button 1 ในรูปที่ 10 ซึ่งในเคลฟจะให้ค่านี้ไว้เป็นค่า Caption หรือ Text แล้วแต่ชนิดของคอมโปเนนต์ทำให้ไปปรากฏเป็นชื่อคอนโทรล (ต่อไปเมื่อกล่าวถึง Caption ขอให้หมายถึง Text ด้วยโดยอันโลม)

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เป็นออปเจ็กต์ในสายตาของผู้เขียนโปรแกรมคือ เพื่อใช้ในการเขียนโปรแกรม โดยที่
ในเดลไฟจะให้ค่าไว้แก่ Name เช่น เป็น Button1 เช่นกัน เป็นชื่อออปเจ็กต์

ค่า Caption ซึ่งใช้เป็นชื่อคอนโทรลนี้ ปกติจะให้ค่าใหม่เพื่อสื่อความหมายกับผู้ใช้ เช่น
เป็นปุ่ม OK และ Cancel เป็นต้น

สำหรับค่า Name ถ้ามีคอมโปเนนต์จำนวนน้อย อาจไม่ต้องให้ค่าใหม่ได้ แต่ส่วนมากแล้ว
ต้องให้ค่าใหม่ ซึ่งถือเป็นการตั้งชื่อเช่นกัน เพื่อให้จำง่าย อ่านโปรแกรมง่าย เช่น ถ้าให้ชื่อ
คอนโทรลเป็น Load, Save และ Clear (รวม 3 ปุ่ม) ก็นิยมกันและในหนังสือนี้ จะให้เป็น bttLde,
bttSve และ bttClr ตามลำดับ เพื่อให้ทราบว่า เป็นคอมโปเนนต์ชนิดใด แยกไว้ว่าชื่อใดเป็นชื่อ
คอนโทรล ชื่อใดเป็นชื่อออปเจ็กต์ แต่อ้างอิงถึงกัน

หมายเหตุ

1. คอมโปเนนต์ทุกชนิดเป็นออปเจ็กต์แต่ไม่ทุกชนิดที่เป็นคอนโทรล
 2. ตามที่กล่าวมานี้ จะเห็นว่า มีค่าถึง 3 ค่าที่อ้างถึงสิ่งเดียวกัน คือ คอมโปเนนต์ออปเจ็กต์
และคอนโทรลซึ่งสามารถใช้แทนกันได้ โดยที่ในหนังสือนี้จะใช้แนวทางนี้คือ
- คอมโปเนนต์ เป็นคำรวม และโดยปกติจะใช้ในการกำเนิดและการเลือก
 - ออปเจ็กต์ในด้านการเขียนโปรแกรม
 - คอนโทรล เพื่อเน้นว่าเป็นคอนโทรลและในคำอธิบายโปรแกรม

ให้ค่าพรอพเพอร์ตี้

เมื่อกำหนดคอมโปเนนต์ เดลไฟจะให้ค่าพรอพเพอร์ตี้ที่ต้องให้ทั้งหมด เป็นค่าดีฟอลต์
(default) ให้ผู้เขียนโปรแกรมให้ค่าใหม่ตามต้องการ

ค่าที่ต้องให้เป็นประจำคือค่า Caption คึงจะนำมาเป็นตัวอย่างของการให้ค่าพรอพเพอร์
ตี้ดังเช่น ถ้าต้องการเปลี่ยนชื่อปุ่ม Button 1 ในรูป 10 เป็น Beep จะให้ค่า Caption ใหม่ได้ดังนี้

1. เลือกคอมโปเนนต์ คือ ปุ่ม Button 1
2. เลือกหน้าต่าง Properties แล้วเลือกหัวข้อพรอพเพอร์ตี้ที่ต้องการคือ Caption จะแสดง

กรอบอิดิตทางด้านขวา

(อาจเลือกหัวข้อพรอพเพอร์ตี้ก่อนเลือกคอมโปเนนต์ได้)

3. ให้ป้อนค่าที่ต้องการคือ Beep ในกรอบอิดิต (โดยลบข้อความเดิมออก)

เมื่อให้ค่าพรอพเพอร์ตี้แล้วจะเกิดผลแตกต่างกันดังนี้

- ปรากฏผลทันทีทุกคีย์ที่ป้อน คึงเช่นค่า Caption ข้อความที่เป็นชื่อปุ่มจะเปลี่ยนตามคีย์ที่

ป้อน

- เกิดผลเมื่อเปลี่ยนไปทำงานอื่น เช่น เลือกฟอร์ม โดยทั่วไปจะเป็นเช่นนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เกิดผลเมื่อกดคีย์ Enter คือค่า Name (คั่น#น เมื่อให้ค่า Nameแล้วจะต้องกดคีย์ Enter ทุกครั้ง)
- เกิดผลในขณะรัน เป็นจำนวนน้อย เช่นค่า Enabled

หมายเหตุ

1. การให้ค่าพรอพเพอร์ตี้สังขมิวธีอื่นอีก เช่นให้เลือกรจากกรอบคอมโบตามทีกล่าวการในหัวข้อ “การให้ค่าเฉพาะ” ในบทที่ 2
2. การให้ค่าพรอพเพอร์ตี้ในบางกรณีจะเป็นแต่เพียงให้ลบค่าที่มีอยู่เดิม

โปรแกรม ButtonBeep

เพื่อให้มีโปรแกรมไว้อ้างอิง โดยสมมติว่าได้ออกแบบโปรแกรมไว้ว่าจะแสดงหน้าต่างมีปุ่ม Beep เมื่อผู้เลือกใช้ปุ่มนี้จะส่งเสียงดังหนึ่งครั้ง

ดังจะแสดงตัวอย่างการเตรียมโปรแกรม โดยเน้นในเรื่องของการป้อนโปรแกรมตามที่จะกล่าวต่อไป

การป้อนโปรแกรม

สามารถแยกการป้อนโปรแกรมได้เป็น 2 ลักษณะคือ

จากการออกแบบ

ดังโปรแกรม ButtonBeep โดยขอสมมติว่าผู้อ่านเป็นผู้ออกแบบ จะเตรียมโปรแกรมได้ดังนี้

การเขียนโปรแกรมบนวินโดวส์จะเขียนจากเหตุ (เท่านั้น) ตามตัวอย่างนี้คือ “เมื่อผู้ใช้เลือกปุ่ม Beep” ดังนั้นจึงต้อง

1. กำหนดคอมโปเนนต์เป็นปุ่ม Button1 แล้วเปลี่ยนชื่อปุ่ม (ให้ค่าแก่ Caption) เป็น Beepซึ่งโดยทั่วไปจะถือว่าเป็นต้นเหตุด้วย (ส่วนชื่อออปเจ็คต์คือ Name ยังคงเป็น Button1)

2. จะต้องทราบ (เช่นจากหนังสือนี้) ว่าอีเวนต์ของเหตุ “อันเกิดจากผู้เลือก” คือ OnClick และเป็นเหตุอันเกิดกับปุ่ม Button1 ดังนั้นกิจกรรมที่ต้องการคือ Button1 Click ซึ่งในเดลไฟให้เรากำหนดกิจกรรมนี้โดย

- เลือกปุ่มนี้ เลือกหน้าต่าง Events เลือกอีเวนต์เป็น OnClick แล้วดับเบิลคลิกในกรอบอิดิตทางขวา ดังที่เห็นเป็นกรอบพื้นขาวในรูป 11

- จะได้โครงโปรแกรมของกิจกรรม Button1 Click ในหน้าต่าง VNIT.PAS ดังในลิสต์ที่ 9 ท่อนบน และขอให้อ่านรูปที่ 9 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่วารณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ป้อนโปรแกรม จากการออกแบบต้องการให้ส่งเสียง 1 ครั้ง จึงให้ป้อนโปรแกรมเป็น MessageBeep (0) ดังในลิสต์ดิ่งที่ 9 ท่อนล่าง



รูป 11 แสดงหน้าต่าง Object Inspector

ลิสต์ดิ่งที่ 9

```

procedure TForm1.Button1Click ( Sender : Tobject );
begin
end ;

-----

procedure Tform1.ButtonClick ( Sender : Tobject ) ;
begin
    MessageBeep ( 0 ) ;
end ;

```

จากตัวอย่าง

แต่ถ้าเป็นการป้อน โปรแกรมจากตัวอย่างเช่นในหนังสือนี้ จะ แสดงโปรแกรมดังเช่นในลิสต์ดิ่งที่ 9 ท่อนล่างเช่นกัน จะต้องดำเนินการคล้ายกับย้อนกลับดังนี้คือ นำชื่อกิจกรรมอีเวนต์ คือ

```

Button1Click

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาแยกออกเป็น Button1 คือชื่อออปเจ็กต์ต้นเหตุ และ Click ได้เป็น อีเวนต์ OnClick นำชื่อออปเจ็กต์ไปเทียบหาชื่อคอนโทรลจะได้เป็นปุ่ม Beep นำไปป้อนโปรแกรมตามที่กล่าวมาข้างต้น

โดยที่การโปรแกรมตามหนังสือนี้ จะเป็นการป้อนโปรแกรมตามตัวอย่างทั้งสิ้น จึงขอให้นักทบทวนด้วยว่าเป็นอีเวนต์ใด เกิดกับคอนโทรลใด คือเป็นอีเวนต์ของคอนโทรลใด

สำหรับคำว่า TForm1 ที่กำกับชื่อกิจกรรมอีเวนต์ หากมองในด้านโอโอพี จะแสดงว่ากิจกรรมนี้เป็นกิจกรรมในสกุล TForm1 หากมองทางด้านคอมโปเนนต์ จะแสดงว่า Button1 เป็นคอมโปเนนต์ในฟอร์ม Form1 (เพราะอาจเปิดฟอร์มหลายฟอร์ม)

กิจกรรมของฟอร์ม

ชื่อกิจกรรมอีเวนต์ของฟอร์มแตกต่างกับของคอมโปเนนต์ เช่น ในโปรแกรม MyBeep ได้ชื่อกิจกรรมเป็น FormClick แทนที่จะเป็น Form1Click ทั้งที่มีชื่อออปเจ็กต์ คือค่า Name เป็น Form1

การกำหนดคอมโปเนนต์เป็นกลุ่ม

ในโปรแกรมส่วนใหญ่ผู้มักจะต้องกำหนดคอมโปเนนต์ชนิดเดียวกันหลายตัว หรือให้ค่าคอมโปเนนต์เป็นกลุ่ม ซึ่งจะมีเรื่องเกี่ยวพันกันดังนี้

กำหนดคอมโปเนนต์เป็นกลุ่ม

เมื่อต้องการกำหนดคอมโปเนนต์ชนิดเดียวกันหลายตัว สามารถเลือกกำหนดได้ดังนี้

1. ด้วยวิธีการตามที่กล่าวมา ซึ่งเป็นการกำหนดคอมโปเนนต์เดี่ยวทีละตัวจนได้จำนวนที่ต้องการหรือ

2. กดคีย์ Shift พร้อมทั้งคลิกเมาส์เลือกชนิดคอมโปเนนต์ในกล่องคอมโปเนนต์ นำไปกำหนดคอมโปเนนต์ให้ได้ตามจำนวนที่ต้องการแล้วเลือกคีย์ลูกศรทางซ้ายของกล่องคอมโปเนนต์เป็นการยกเลิกการเลือกซ้ำ หรือ

3. ด้วยวิธีการลอกปะ โดยกำหนดคอมโปเนนต์แรก ซึ่งจะได้เป็นคอมโปเนนต์หลักด้วยเลือกแล้วกดคีย์ Ctrl+C เป็นการลอก แล้วกดคีย์ Ctrl+V เป็นการปะหลายครั้งเพื่อให้ได้จำนวนคอมโปเนนต์ที่ต้องการ จะได้คอมโปเนนต์ซ้อนเหลื่อมทับกัน ให้เลื่อนไปไว้ ณ ตำแหน่งที่ต้องการ

การลอกปะ จะได้คอมโปเนนต์ชนิดเดียวกันกับตัวหลัก มีค่าพรอปเพอร์ตี้ส์เหมือนกัน เว้นค่าตำแหน่งคือ Left กับ Top และถ้าได้ป้อนโปรแกรมกิจกรรมอีเวนต์ให้แก่คอมโปเนนต์หลักแล้วจะใช้กิจกรรมนี้ร่วมกัน และสามารถจะใช้ลอกคอมโปเนนต์เป็นกลุ่มคือ ลอกแล้วปะเป็นกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจับกลุ่ม

สามารถจับกลุ่มคอมโปเนนต์ชนิดเดียวกันหรือต่างชนิดกัน เพื่อการดำเนินการเป็นกลุ่ม เลือกได้ดังนี้คือ

1. ดึงรอบล้อมรอบคอมโปเนนต์ที่ต้องการ หรือ
2. เลือกคอมโปเนนต์แรก ซึ่งจะได้เป็นตัวหลักด้วย กคคีย์ Shift พร้อมกับเลือกคอมโป

เนนต์ตัวต่อไป

ยกเลิกการจับกลุ่มโดยคลิกเมาส์ในฟอร์ม หรือเลือกคอมโปเนนต์เดี่ยว

การดำเนินการเป็นกลุ่ม

เมื่อจับกลุ่มคอมโปเนนต์แล้ว สามารถดำเนินการเป็นกลุ่มได้ดังนี้

1. ย้ายตำแหน่งด้วยการคลิกเลื่อน ซึ่งต้องคลิกเมาส์ในกลุ่ม คือที่คอมโปเนนต์ตัวใดตัว

หนึ่ง

2. จัดแนว โดยเลือกรายการ Align ใน Edit หรือเลือกจากเมนูลอย (เปิดเมนูลอยได้โดยคลิกเมาส์ขวาในกลุ่ม) จะแสดงไดอะล็อกบ็อกซ์ ให้เซตค่าเพื่อการจัดแนว

3. ปรับขนาด เลือกรายการ Size ใน Edit หรือเลือกจากเมนูลอย จะแสดงไดอะล็อกบ็อกซ์ ให้เวตค่า โดยให้ป้อนขนาดใหม่ หรือเลือกให้เท่ากับคอมโปเนนต์ใหญ่สุด หรือเล็กที่สุด

การให้ค่าเป็นกลุ่ม

เมื่อต้องการให้ค่าพารามิเตอร์เป็นกลุ่ม คือให้ใช้ค่าเดียวกันหรือให้ใช้กิจกรรมอีเวนต์ร่วมกัน ให้เลือกดำเนินการได้ดังนี้

1. จับกลุ่ม แล้วให้ค่ากับป้อนโปรแกรมของกิจกรรมอีเวนต์ (อย่างใดอย่างหนึ่งหรือทั้งคู่) ถ้าเป็นการป้อนโปรแกรมจะได้ชื่อกิจกรรมอีเวนต์เป็นของตัวหลัก แต่จะเป็นกิจกรรมอีเวนต์ของทุกคนในกลุ่ม หรือ

2. กำหนดคอมโปเนนต์ตัวแรก แล้วให้ค่ากับป้อนโปรแกรมของกิจกรรมอีเวนต์ แล้วใช้วิธีลอกปะ

ร่วมใช้กิจกรรมอีเวนต์

เมื่อได้ป้อนกิจกรรมอีเวนต์ของคอมโปเนนต์หนึ่งหรือเป็นกลุ่มไปแล้ว หากต้องการให้คอมโปเนนต์อื่นร่วมใช้กิจกรรมด้วย ให้เลือกคอมโปเนนต์ใหม่นั้น เลือกอีเวนต์แล้วเปิดกรอบคอมโป เลือกกิจกรรมที่ต้องการ

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อให้ค่าที่เป็นกลุ่มแล้ว หากต้องการให้ค่าอื่นหรือให้ใช้ค่าที่แตกต่างกัน รวมทั้งกิจกรรมอื่นด้วย ทำได้โดยให้ค่าโดยตรงแก่คอมโปเนนต์นั้น

3.11 การใช้งาน DELPHI

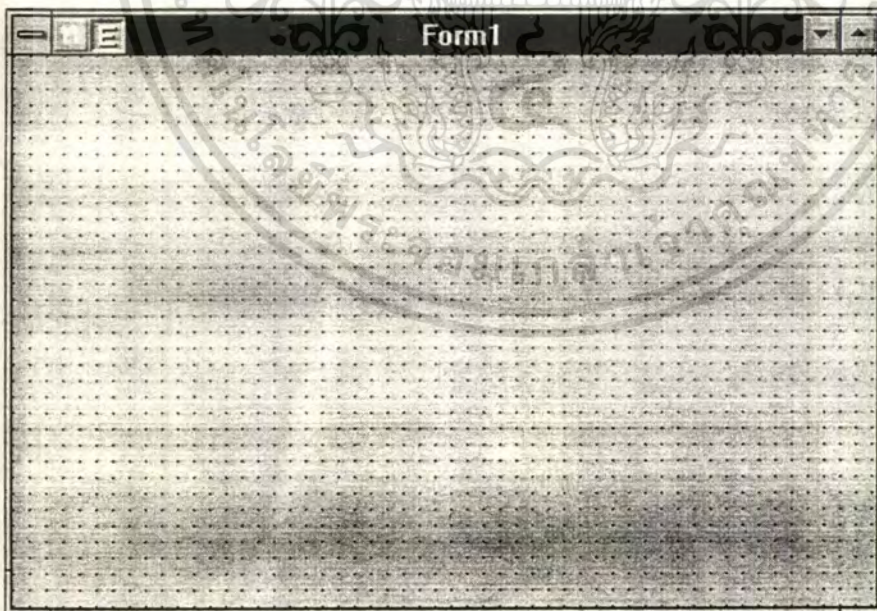
Delphi เป็นโปรแกรมที่นำมาใช้ในการพัฒนาและสร้าง Application ที่มีความสามารถในการจัดการกับงาน DataBase ได้อย่างมีประสิทธิภาพ และ Delphi ยังมีเครื่องมือที่เอื้ออำนวยความสะดวกในการสร้าง Application ต่าง ๆ ซึ่งทำให้ผู้ที่ต้องการสร้างงาน Application สามารถสร้างโปรแกรมได้ง่าย

ส่วนประกอบของ Delphi

ในการใช้งาน Delphi สร้าง Application นั้นเราจะทำการศึกษาส่วนต่างๆ ที่ต้องใช้งาน ซึ่งส่วนหลัก ๆ ในการสร้างนั้นประกอบด้วย

1. Form Designer ของ Delphi

คือรูปแบบที่จะใช้ในการออกแบบหน้าตาของ Application ซึ่งเราสามารถออกแบบส่วนต่าง ๆ ที่ต้องการสำหรับติดต่อกับ User ซึ่งในครั้งแรกนั้น Delphi จะให้ชื่อว่า Form1



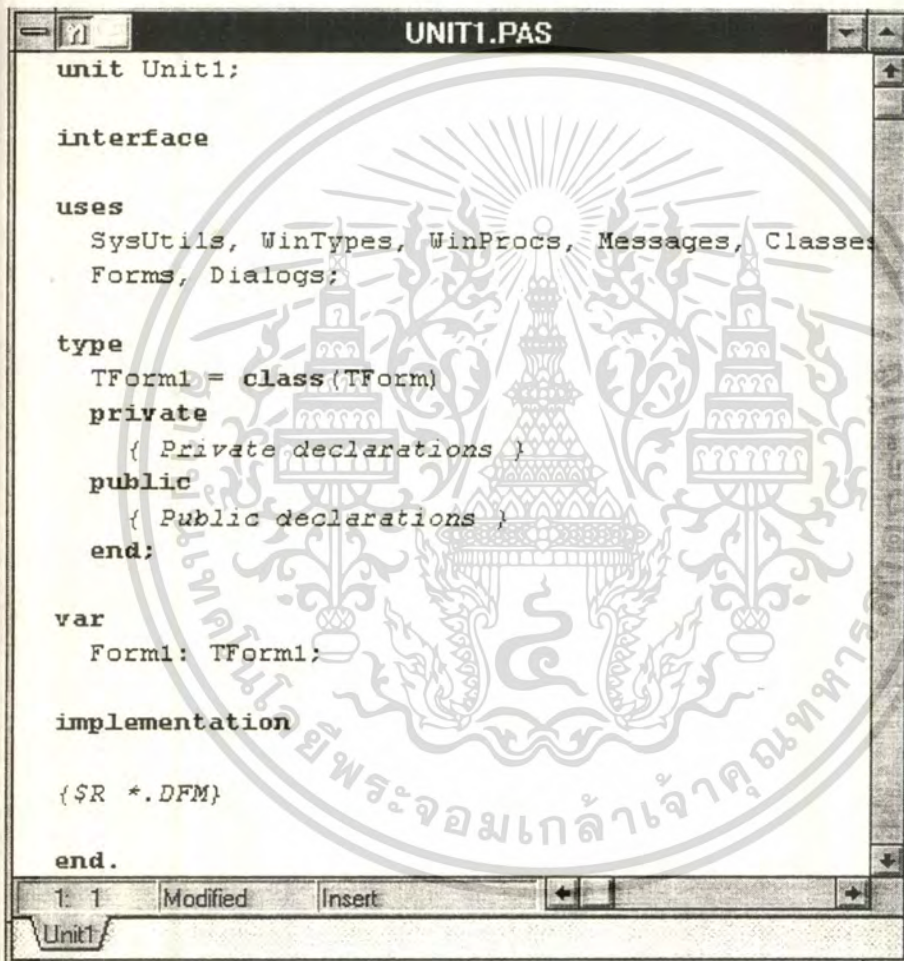
รูป 11 Form Designer

Form : คอมโพเนนต์ใหญ่ที่ทำหน้าที่รวบรวมคอมโพเนนต์อื่นๆ ไว้ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Editor Window

คือส่วนที่ Delphi ใช้ในการเขียน Code ในการควบคุมเหตุการณ์ต่าง ๆ ของ Object ที่ Programmer ต้องการให้เกิดขึ้น ในส่วนนี้ Delphi ได้กำหนด รูปแบบในการเขียน Code ซึ่งเป็นรูปแบบการเขียน Code แบบ Pascal ซึ่งเราสามารถกำหนดและควบคุมโปรแกรมได้ในส่วนนี้ได้เหมือนกัน โดยมีโค้ดต่างๆ แบ่งเป็น Unit , Interface , User ,Type ,Var และ Imprementation แต่โดยทั่วไปโค้ด ส่วนใหญ่ เบลโฟจะทำการจัดการให้เราเอง



```
UNIT1.PAS

unit Unit1;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes,
  Forms, Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

end.
```

รูป 12 The Editor Window

Unit : หน้าต่างที่คุณจะสั่งด้วยโค้ดให้เบลโฟปฏิบัติกรใดๆ

3. Component Palette

คือเครื่องมือที่ใช้อำนวยความสะดวกในการสร้าง Application ซึ่งเป็น Icon ที่เราสามารถ Click ที่ Icon ที่เราต้องการใช้งานแล้วนำมาวางลงบน Form เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Designer เท่านั้น ซึ่ง Delphi จะสั่งให้มันทำงานในขณะที่ Run time ซึ่ง Component Palette นั้นยังแบ่งย่อยออกเป็นส่วนต่าง ๆ อีกหลายส่วน

ทุลบาร์ที่อยู่ด้านซ้ายของหน้าต่างหลัก แผงปุ่มลัด (Speed Button) ซึ่ง จะรวบรวมคำสั่งสำคัญไว้ เช่นการปิดเปิดโปรแกรม การจัดเก็บ การรัน การดีบัก โดยทุกปุ่มจะมีป้ายบอกตัวเอง (Tool Tips) ส่วนที่อยู่ทางซ้ายของหน้าต่างหลัก คือปุ่มที่สร้างรูปอินเทอร์เฟซบนฟอร์ม



รูป 13 Component Palette

Main Menu : ส่วนที่รวบรวมคำสั่งและคอมโพเนนต์ทั้งหลายไว้

4. Object Inspector

คือส่วนที่ใช้ในการกำหนดคุณสมบัติของ Component ที่วางอยู่บน Form Designer



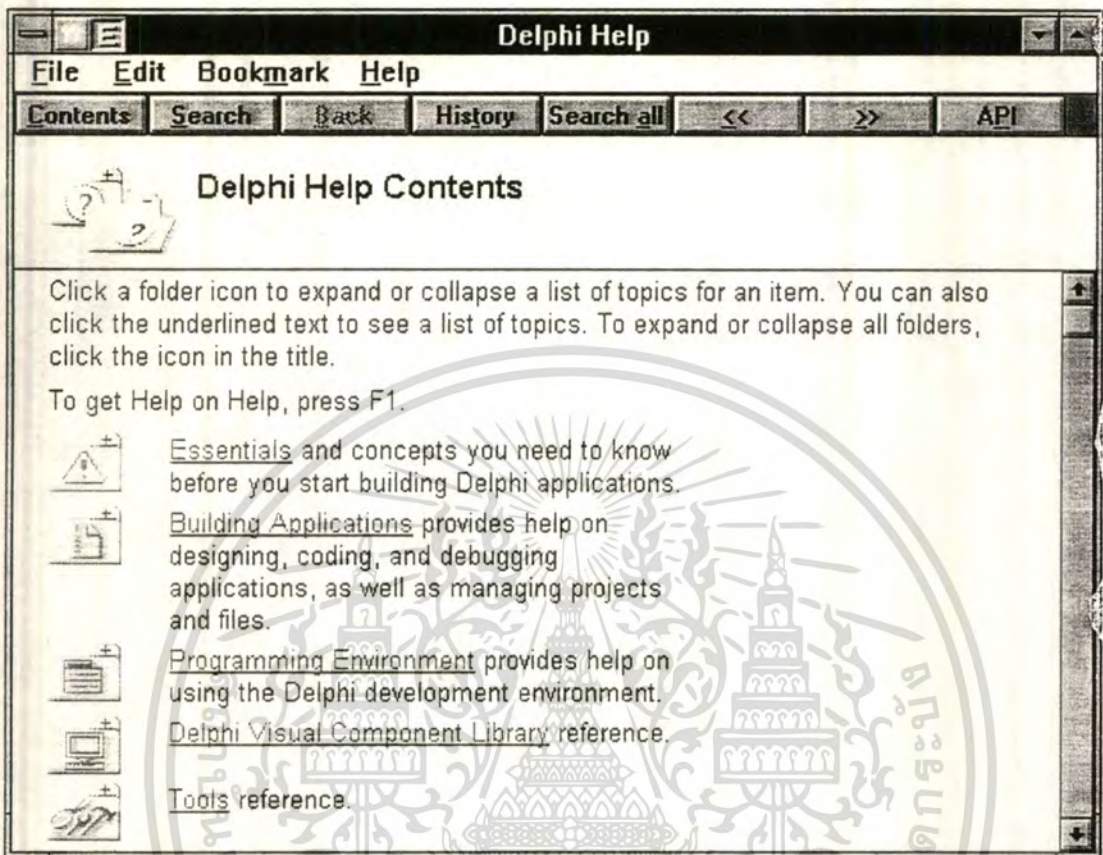
รูป 14 the Object inspector หน้าต่างที่จะนำคุณเข้าไปออกแบบการปรากฏ

และการทำงานของออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Online help

คือส่วนที่ช่วยในการอธิบายรายละเอียดและวิธีการใช้งานของ Delphi



รูป 15 Online help

Delphi กับงาน DataBase

DataBase

คือแหล่งที่ใช้เก็บข้อมูลซึ่งข้อมูลเหล่านั้นมีความสัมพันธ์กันในทาง Logically ซึ่งอาจเก็บไว้ในรูปแบบความสัมพันธ์ในลักษณะต่าง ๆ ซึ่งผู้ใช้อาจเรียกดูข้อมูลใน Table โดยผ่าน View ได้ ตามขอบเขตที่กำหนด

ในทาง Physical อาจจะเก็บ Table ไว้ใน Drive เดียวกัน หรือต่างกัน และสามารถกระจายข้อมูลไปอย่างสม่ำเสมอ เมื่ออ่านข้อมูลจะอ่านมาแบบขนาน ซึ่งทำให้สามารถอ่านข้อมูลได้รวดเร็ว แต่เมื่อเวลาเรียกมาใช้ก็จะเห็นเป็น Table เดียวกัน

Database ส่วนใหญ่จะใช้การมองข้อมูลแบบ Logical หากมองในทาง Physical จะเป็นระดับที่ซับซ้อนมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12 ODBC (Open DataBase Connectivity)

คือวิธีการติดต่อและเข้าถึง Application ผู้ระบบการจัดการฐานข้อมูล โดยใช้ภาษา SQL เป็นมาตรฐานการเข้าถึงข้อมูล ความสามารถในการเชื่อมต่อแบบนี้ทำให้ Application สามารถเข้าถึงฐานข้อมูล ได้หลายรูปแบบซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนาโปรแกรมไปได้โดยไม่ ต้องทำการระบุชนิดของระบบการจัดการฐานข้อมูล

แต่เดิมนั้นการพัฒนาโปรแกรมประยุกต์ที่ใช้งานเกี่ยวกับฐานข้อมูล การเข้าใช้ฐานข้อมูลโปรแกรมเหล่านี้จะทำการเรียกใช้ Embedded SQL ซึ่งในขณะนั้นวิธิต่างแบบนี้ก็ดูจะเป็นไปได้ทีเดียว เพราะว่าตัวโปรแกรมสามารถ ทำการเปลี่ยนรูปแบบของระบบไม่ว่าจะเป็นทางด้านฮาร์ดแวร์ หรือ ซอฟต์แวร์ ได้หลายรูปแบบ รวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มี การย้ายระบบ)

อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่มีความแตกต่างกัน เช่น การเรียกใช้ข้อมูลของ Oracle จาก Microsoft Excel วิธีการเข้าถึงข้อมูลแบบเดิมนั้น จะต้องทำการคอมไพล์โค้ดของ Excel และ Oracle โดยการใช้ IBM Precompiler และ Oracle Precompiler ตามลำดับ ซึ่งจะเห็นว่าเป็นการยุ่งยากมากทีเดียว

สำหรับวิธีการต่อเชื่อมแบบ ODBC จะให้ความสะดวกใน การติดต่อข้อมูลมากกว่าวิธีการดั้งเดิม โดยการกำหนดมาตรฐานการต่อเชื่อมของข้อมูล (Data Protocol,DBMS Capability) และแนวทางนี้ได้ทำให้เกิดความคิดที่จะสร้าง Driver การติดต่อกับการของงานข้อมูลขึ้นมา (DLL)

ข้อดีของการติดต่อโดยใช้ ODBC

- ฟังก์ชันของ ODBC อนุญาตให้ Application ติดต่อกับระบบจัดการฐานข้อมูล ได้โดยสะดวก (การทำคำสั่ง SQL และการรับผลลัพธ์)
- ใช้ภาษา SQL ตามมาตรฐาน SQL CAE,X/OPEN และ SQL ACCESS GROUP (SAG)
- มีการกำหนดการส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตรฐานเดียวกัน
- เป็นวิธีการมาตรฐานในการติดต่อกับระบบจัดการฐานข้อมูล
- มีการกำหนดชนิดของข้อมูล (Data Type) เป็นมาตรฐาน
- ชุดคำสั่ง SQL สามารถกำหนดได้แม้ในขณะที่ Run
- สามารถเขียน โปรแกรมชุดเดียวแต่สามารถเข้าใช้ระบบจัดการฐานข้อมูลได้หลาย

ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวโปรแกรมไม่ต้องรับผิดชอบการดูแลการติดต่อข้อมูลกับระบบจัดการฐานข้อมูล
- ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

องค์ประกอบของ ODBC

สถาปัตยกรรมของ ODBC ประกอบด้วย 4 ส่วนสำคัญ

1. Application ทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของ ODBC ตามคำสั่งภาษา SQL พร้อมทั้งทำการรับผลลัพธ์ด้วย
2. ตัวจัดการ Driver Manager ทำหน้าที่โหลด Driver และเชื่อมต่อกับแหล่งข้อมูล
3. Driver ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของ ODBC ส่งคำสั่ง SQL ไปสู่แหล่งข้อมูลที่ต้องการและทำการส่งผลลัพธ์กลับให้ Application และในบางครั้ง Driver จะทำหน้าที่แปลงคำสั่งที่ส่งมาให้อยู่ในรูปแบบที่สนับสนุนโดยระบบจัดการฐานข้อมูลแต่ละชนิดอีกด้วย
4. Data Source เป็นแหล่งข้อมูลที่ใช้ต้องการเข้าถึง



รูป 16 องค์ประกอบของ ODBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะเรียกใช้การเชื่อม ODBC ในการทำงานต่อไปนี้

1. ร้องขอการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำสั่ง SQL สู่มแหล่งข้อมูล
3. กำหนดพื้นที่การจัดเก็บและรูปแบบของข้อมูล ที่เป็นผลลัพธ์จาก SQL

Request

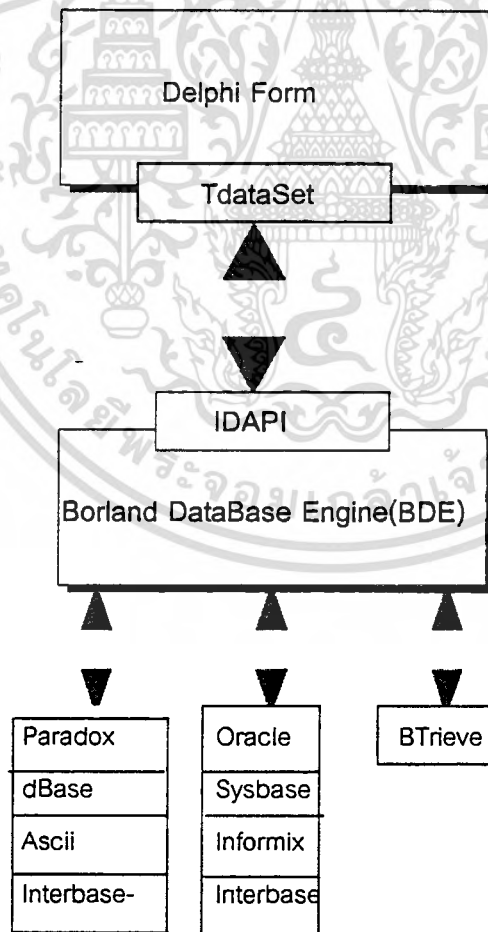
4. ร้องขอผลลัพธ์
5. ประมวลผลและจัดการกับข้อผิดพลาด
6. รายงานผลให้กับผู้ใช้ (ถ้าจำเป็น)
7. ร้องขอการ Commit หรือ Roll back สำหรับควบคุมการประมวลผลTransaction

ผลTransaction

8. ยกเลิกการติดต่อกับแหล่งข้อมูล

Architecture of Delphi Database Applications

ซึ่งประกอบด้วย 3 Layers ดังรูปที่แสดง



รูป 17 ลำดับขั้นฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Layer แรก เป็น การเข้าถึง (Accessing) และจัดการกับ database สามารถทำได้โดยใช้ Standard Database Components ซึ่ง Delphi ได้จัดเตรียมไว้ให้

Layer ที่สอง เป็นตัวกลางในการจัดการกับ database จะใช้ BDE (Borland Database Engine) ทำการติดต่อ Interface กับ Database ซึ่งมี Subroutines และ Services สำหรับใช้งาน โดย BDE Application Programming Interface

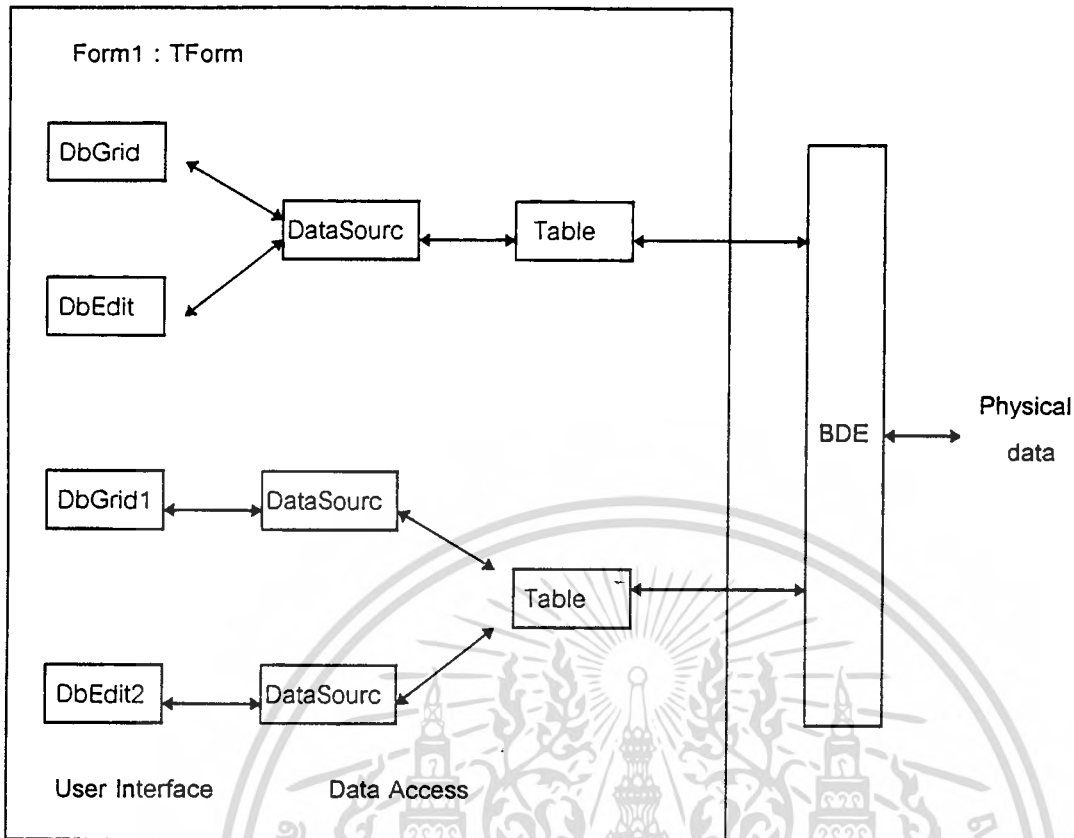
Layer ที่สาม เป็นการเข้าถึง Data ทาง Physical ซึ่งเก็บอยู่ใน Database Tables ซึ่งจะมี DBMS เป็นตัวจัดการกับ Database เช่น อาจมีรูปแบบเป็น Paradox หรือ Dbase Tables และ DBMS คือ Oracle, Sybase, Informix, or Interbase Server ซึ่งมีวิธีการจัดการแตกต่างกัน

ความสามารถและคุณลักษณะพิเศษทางด้านงานฐานข้อมูลของ Delphi

การสร้าง Application งาน DataBase โดย Delphi จะอาศัย

- เครื่องมือพัฒนางานด้านงานฐานข้อมูลของ Delphi (delphi database development tool)
- Delphi data access component คุณสามารถใช้มันเข้าไปกำหนดค่าของ Fields ใน records และสามารถจัดการข้อมูลได้ เช่น กำหนด table, list table ภายใน Database ซึ่งเครื่องมือเหล่านี้จะอยู่ใน Data Access page ของ Delphi's Component Palette ซึ่งประกอบด้วย Ttable, TDataSource, TQuery และอื่น ๆ ซึ่งในการผ่านเข้าไปใน Low Level จะผ่าน BDE

Delphi data aware GUI component เป็น Visual Component ที่ใช้แสดงข้อมูลของ Data Field ของ Table ได้ เช่น DbGrid, DbLabel, DbListBox และอื่น ๆ ซึ่งจะเกี่ยวข้องกับ Database แต่ Tool เหล่านี้จะต้องใช้ร่วมกับ DataSource, Data Access Component โดยใช้คอม component ในการติดต่อกับบอร์เครื่องข่ายท้องถิ่น คาค้าเบส เอนจิน หรือบีดีอี (Borland Database Engine, BDE) ซึ่งจะทำหน้าที่ติดต่อกับฐานข้อมูลอีกที ภาพต่อไปนี้จะแสดงความสัมพันธ์ ของเครื่องมือต่าง ๆ และ Application งาน Database กับ BDE และแหล่ง ข้อมูล (DataSource)



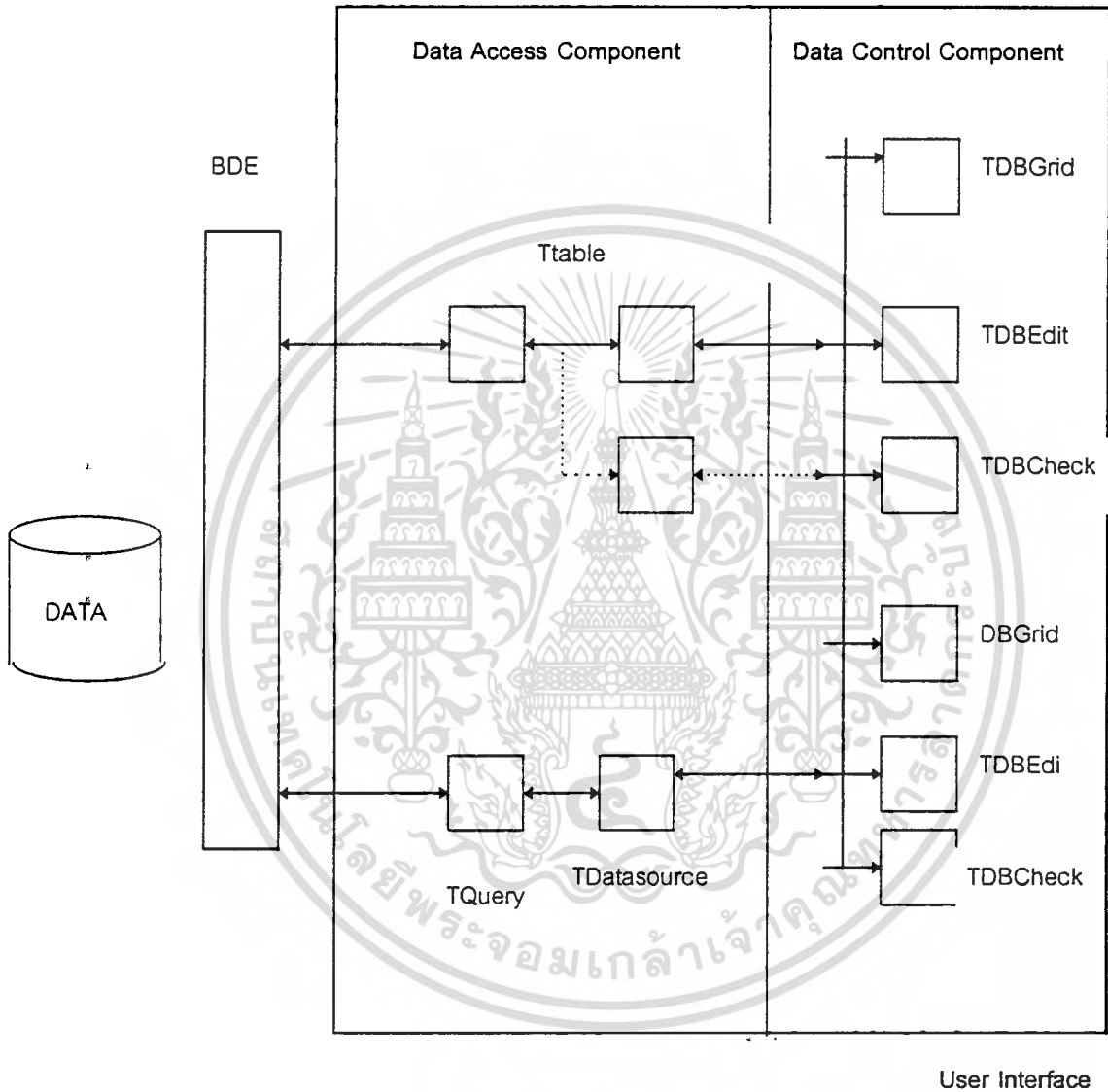
รูปที่ 18 สถาปัตยกรรมฐานข้อมูล

Delphi ใช้ออปเจกโอเรียนเต็ด คอมโพเนนท์ (Object-Oriented Component) ในการสร้างแอปพลิเคชันฐานข้อมูลเช่นเดียวกับแอปพลิเคชันอื่นๆที่มี ซี อพพลีเคชันฐานข้อมูล คาด้าเบสคอมโพเนนท์ (Database Component) จะเหมือนกับ คอมโพเนนท์มาตรฐานทั่วไป ที่มีเอททริบิวท์ หรือเรียกว่า พรอพเพอร์ตี้ (Property) ซึ่งจะถูกตั้งค่าโดยโปรแกรมเมอร์ในขณะที่ทำการออกแบบ และอาจจะตั้งค่าใน ขณะที่ กำลังรันโปรแกรมก็ได้

- คาด้าเบสคอมโพเนนท์จะมีค่าที่ถูกตั้งไว้ให้แล้ว (Default) เพื่อให้มันสามารถ ทำงาน ได้โดยไม่ต้องเขียน โปรแกรมเพิ่มหรือเพิ่มเพียงเล็กน้อย
- Delphi ประกอบด้วย 2 คอมโพเนนท์พาเลท (Component Palette) ที่มีคาด้าเบสคอมโพเนนท์ คือ
- หน้าสำหรับคอมโพเนนท์ที่เข้าถึงข้อมูล (Data-Access Page) จะมีออปเจกต์ที่ทำให้การเข้าถึงข้อมูลง่ายขึ้น โดยการเอนแคปซูลเลท (Encapsulate) แหล่งข้อมูล เช่น ฐานข้อมูลที่ติดต่อดารางในฐานข้อมูล หรือฟิลด์ (Field) ของข้อมูลที่ต้องการตัวอย่างของออปเจกต์เหล่านี้ได้แก่ Ttable Tquery ,Tdata-source และ Treport

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลของมันผ่านบีดีอีคอมโพเนนท์ ควบคุมข้อมูลจะติดต่อกับ คาด้าเซท คอมโพเนนท์ โดยผ่านทาง Tdatasource คอมโพเนนท์ เพื่อให้การติดต่อกับข้อมูล แบบมองเห็นได้จากภาพแสดงให้เห็นว่า คอมโพเนนท์เข้าถึงข้อมูล กับ คอมโพเนนท์ควบคุมข้อมูลสัมพันธ์กับข้อมูล สัมพันธ์ซึ่งกันและกัน และสัมพันธ์ กับส่วนติดต่อกับผู้ใช้แอปพลิเคชันอย่างไร



รูป 19 การติดต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

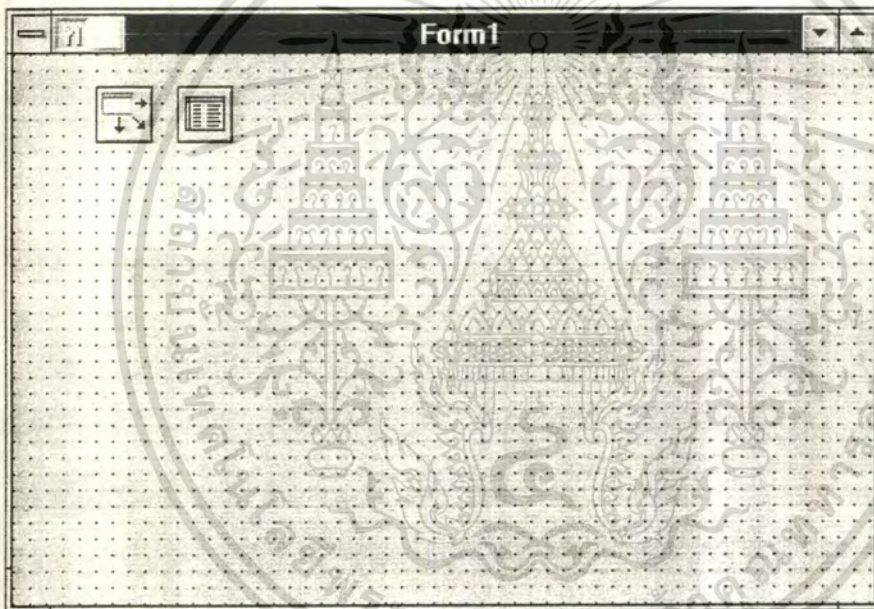
Component Based Database Development Patterns

Table Browser เป็นการเรียกดู Database Table ซึ่งมีอยู่แล้ว เราสามารถใช้ Tool เพียงไม่กี่ตัว คือ TTable, TDataSource และ TDbGrid

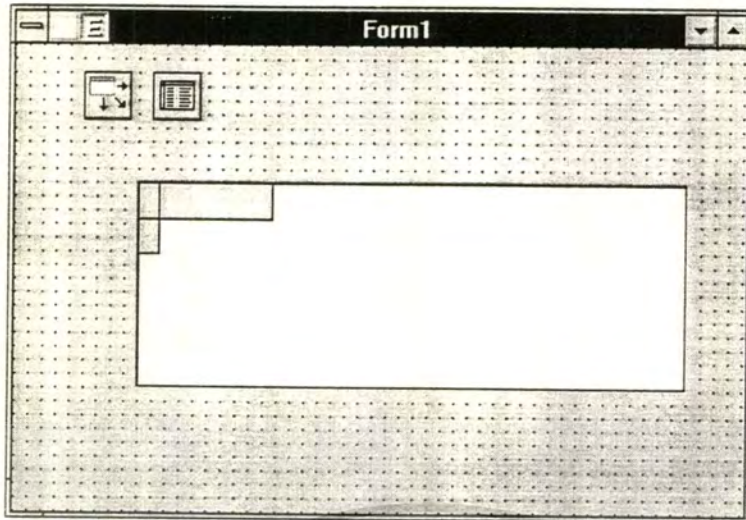
1. สร้าง Project ใหม่ โดยการเลือก File| New ที่ Main Menu ของ Delphi ตั้งชื่อ DBROWSER.DPR จะปรากฏ Form Designer ที่หน้าจอ

2. เลือก Table Component จาก Data Access Page และวางลงบน Form Designer จากนั้น Delphi จะตั้งชื่อให้ว่า Table1 และกำหนด Type Ttable ไว้ใน Class ของ TForm

3. เลือก DataSource Component จาก Data Access Page และวางลงที่ Form Designer ข้าง ๆ Table Component ดังรูป ซึ่ง Delphi จะตั้งชื่อ ให้ว่า DataSource1



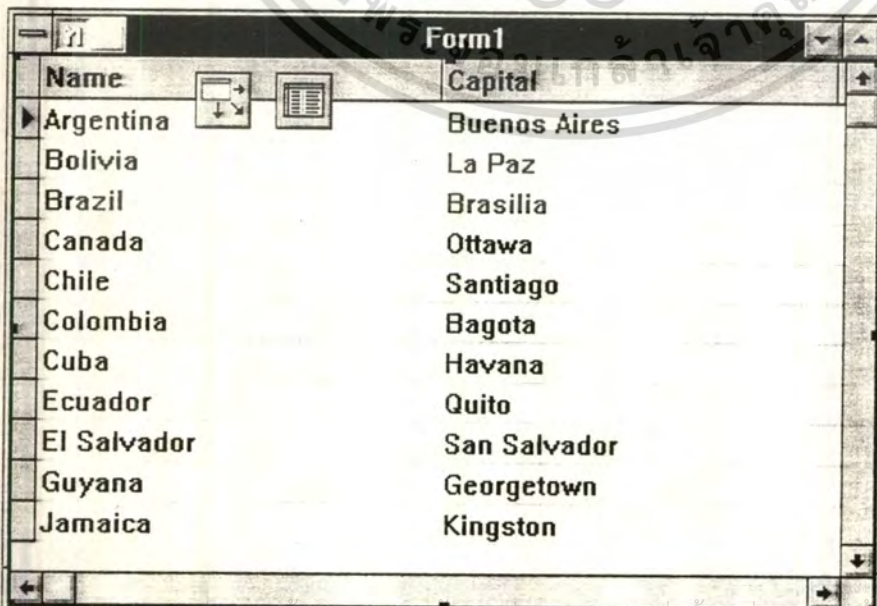
4. เลือกไปที่ Icon DbGrid จาก Data Control Page และวางลงที่ Form Designer จากนั้น Delphi จะตั้งชื่อให้ว่า DbGrid1 ซึ่งตอนนี้จะยังไม่แสดงข้อมูลออกมา เพราะยังไม่ได้กำหนด Property ของ Component ต่าง ๆ



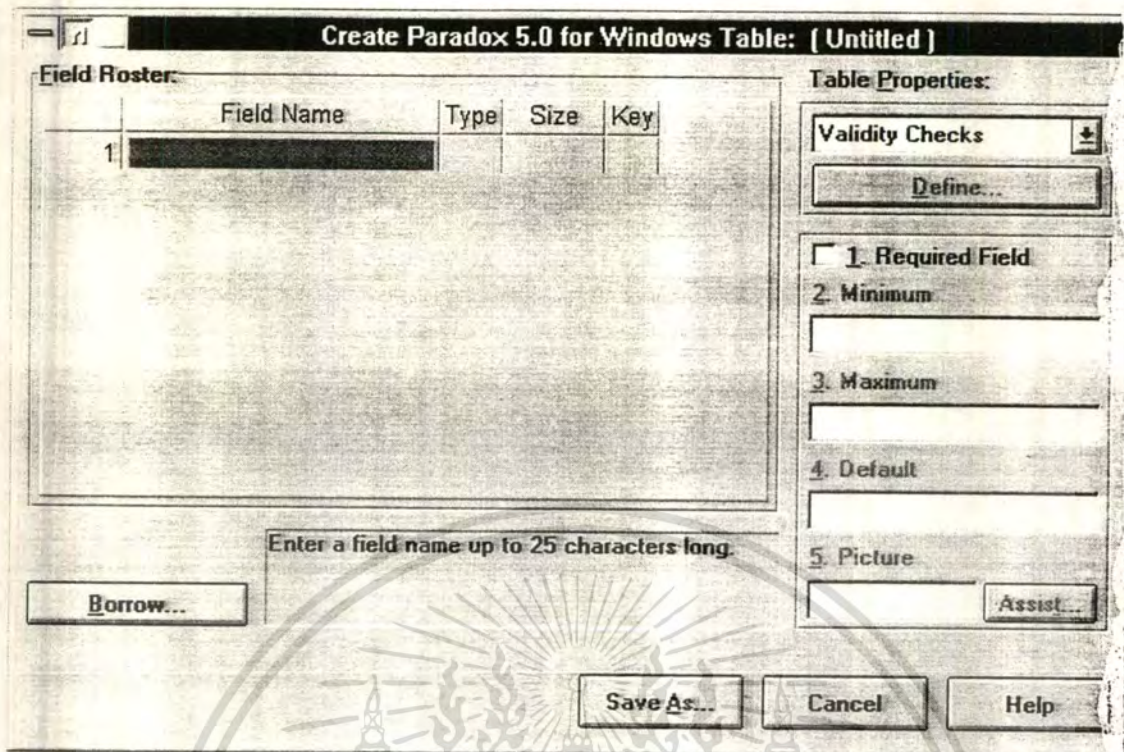
5. ทำการเปลี่ยน Property ตามส่วนของข้างล่างนี้ซึ่งเป็นของแต่ละ Component
ดังนี้

TDataSource.DataSet = Table1
 TDBGrid.Align = alClient
 TDBGrid.DataSource = DataSource1
 TTable.DatabaseName = DBDEMOS
 TTable.TableName = COUNTRY.DB
 TTable.Active = True

เมื่อทำการ Set Property ตามที่กำหนดนี้แล้ว Form Designer ก็จะแสดงข้อมูล
ออกมาตามรูปข้างล่างนี้



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ชื่อที่คุณกำหนดในรูปแบบ Paradox Field จะต้องเป็นไปตามกฎ

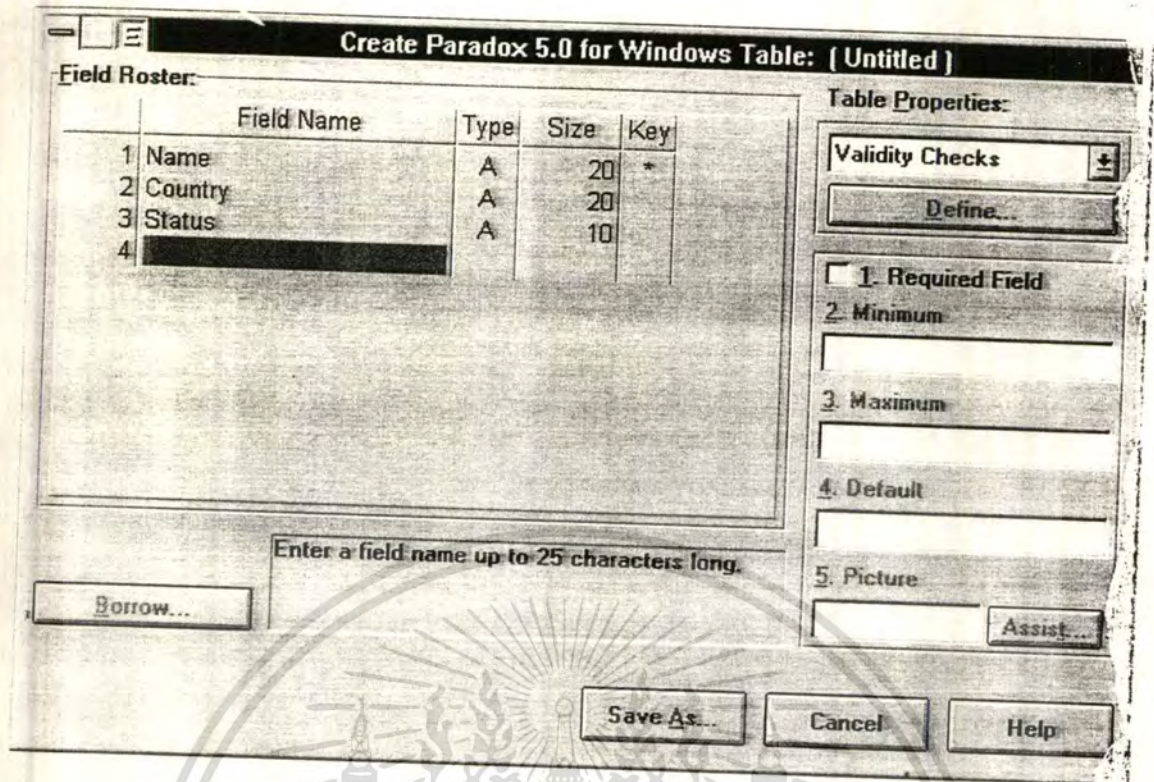
- Names ต้องประกอบไม่เกิน 25 ตัวอักษร

- Names ต้องไม่ขึ้นต้นด้วยช่องว่าง แต่ภายในมีได้ แต่ถ้าต้องการที่จะใช้งานร่วมกับ format อื่น ควรจะมีไม่เกิน 9 ตัวอักษร และไม่ควรมีช่องว่างภายในชื่อ

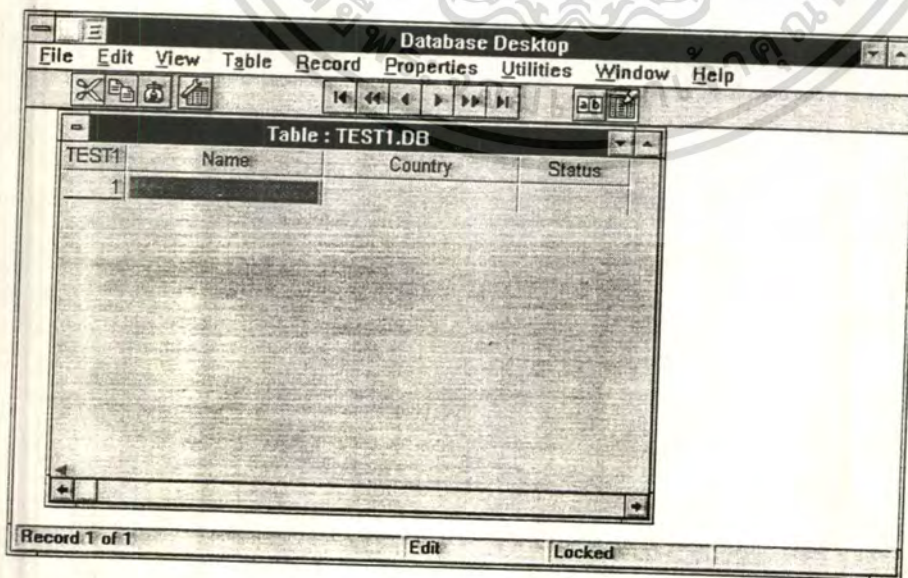
- และต้องไม่มี [], { หรือ () สัญลักษณ์พิเศษต่าง ๆ การกำหนด Type

ใช้ click ด้านขวาของ mouse แล้วเลือกตามชนิดที่ต้องการหรือจะพิมพ์เองก็ได้ ซึ่งเป็นสัญลักษณ์ของชนิดที่มีใน field type ในส่วนของการกำหนด primary index กำหนดได้โดยใส่เครื่องหมาย * ที่ key field ในรูปแบบ Paradox จะกำหนดให้ field แรกของ record จะถูกกำหนดให้เป็น Primary Key จากตัวอย่างดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คุณอาจสร้าง a secondary index, validity check or referential integrity โดยเลือก Utility/Restructure จาก DBD main menu
 หลังจากที่คุณสร้าง Table เสร็จแล้วคุณสามารถเข้าไปบันทึกข้อมูล ด้วย โดยการเลือก File/Open/Table ชื่อ Table ของคุณจากนั้นเลือก Restructure จากนั้น Click ที่ icon Edit ก็จะแสดงให้คุณดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้า ต้องการแก้ไขก็สามารถเข้าไปแก้ไข เพิ่มเติมได้โดยใช้ Table|Table Restructure

3.17 การสร้าง Tables ด้วย SQL

ในการสร้าง Tables ด้วย SQL จะมีรายละเอียดและ Syntax ในการสร้างแตกต่างกันไปตาม Server ที่คุณใช้ แต่พื้นฐาน basic statement ส่วนใหญ่จะสามารถใช้กับ Server ทั่ว ๆ ไป

การสร้างขั้นแรกให้ drop a Tquery object ลงที่ form กำหนด DatabaseName Property to a valid alias และพิมพ์

```
create table address (Firstname char(20),Lastname char(20))
```

ใน SQL property ดังรูปถัดไป



จากนั้น Run Program โดยเรียก Query.Open โดยใส่ที่ procedure TForm.create แล้ว server จะสร้าง table ให้

3.18 หลักการพัฒนาแอปพลิเคชันงานฐานข้อมูล

1. รูปแบบของการพัฒนา

เนื่องจากการออกแบบแอปพลิเคชัน จะต้องขึ้นกับโครงสร้างฐานข้อมูลที่จะเข้า

ถึงโดยฐานข้อมูลจะต้องถูกกำหนดล่วงหน้าก่อนที่จะทำการพัฒนาแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบในการพัฒนาแอปพลิเคชันงานฐานข้อมูลของ Delphi เป็นไปได้ 4 รูปแบบคือ

- ไม่มีฐานข้อมูลอยู่ก่อน หรือจำเป็นต้องทำการกำหนดใหม่หมดสามารถทำได้โดย

- ใช้ คาค้าเบส เดสก์ทอป ยูทิลิตี้ (Database Desktop Utility) เพื่อกำหนดตารางแบบพาราคอกซ์และดีเบส

- สำหรับ เอสคิวแอลเซิร์ฟเวอร์ สามารถใช้เครื่องมือที่เป็นของเซิร์ฟเวอร์ หรือของคาค้าเบส เดสก์ทอปก็ได้ ตัวอย่างเช่น สำหรับโลกอลอินเตอร์เบสเซิร์ฟเวอร์ สามารถใช้ วินโดว์ ไอเอสคิวแอล (Windows ISQL) ในการกำหนดฐานข้อมูล

- มีฐานข้อมูลปรากฏอยู่บนแหล่งข้อมูล ที่เป็นเดสก์ทอป หรือ เป็นเครือข่าย ท้องถิ่น (LAN) และฐานข้อมูลเข้าถึงข้อมูล ณ ที่แห่งนั้น ถ้าบีคิอิ และแหล่งข้อมูลอยู่บนเครื่องเดียวกันกับแอปพลิเคชันแล้ว แอปพลิเคชันนั้น จะเป็นแบบเสตนค้อโลน (Stand-Alone)

- ฐานข้อมูลปรากฏบนแหล่งข้อมูลแบบเดสก์ทอป และจะถูกอัพไซส์ (Upsize) เป็นเอสคิวแอลเซิร์ฟเวอร์แบบนี้จะเรียกว่าเป็นโลกอลเอสคิวแอลเซิร์ฟเวอร์

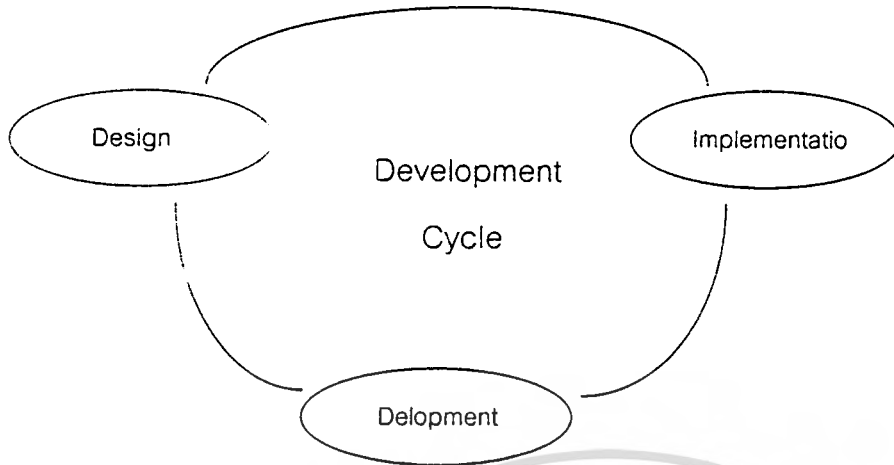
- ฐานข้อมูลอยู่บนเอสคิวแอลเซิร์ฟเวอร์และแอปพลิเคชันจะเข้าถึงข้อมูลที่อยู่ทีเซิร์ฟเวอร์ แบบนี้จะเรียกว่าเป็นไคลเอ็นท์เซิร์ฟเวอร์แบบมาตรฐาน

2. วงจรในการพัฒนาแอปพลิเคชันงานฐานข้อมูล การพัฒนา แอปพลิเคชันงานฐานข้อมูลมี 3 ขั้นตอนหลักคือ

- การออกแบบ และ ทำต้นแบบ (Prototyping)

- การสร้าง (Implement)

- การส่งมอบระบบใช้งานจริง (Deployment) และการบำรุงรักษา (maintenance)



วงจรในการพัฒนาสำหรับแอปพลิเคชันแบบไคลเอนท์เซิร์ฟเวอร์ งานของ ฐาน

ข้อมูลและงานของแอปพลิเคชันจะแยกกันเด่นชัดเนื่องจากจะรันบนแพลตฟอร์มที่ต่างกันและระบบปฏิบัติการก็จะต่างกันด้วย เช่น ยูนิกซ์เซิร์ฟเวอร์ และวินโดวส์ 3.1 ไคลเอนท์ดังนั้น ในการพัฒนา ความรับผิดชอบในการแบ่งว่างานใดควรทำโดยคาด้าเบสเซิร์ฟเวอร์ งานใดควรทำโดยไคลเอนท์แอปพลิเคชันจะอยู่ในขั้นตอนการออกแบบ โดยทั่วไปแล้วจะมีการแบ่งงาน กันอย่างชัดเจนระหว่าง 2 ฝ่าย แต่กรรมวิธีทางฐาน ข้อมูลเช่น สตอร์โพรซีเคอร์บางอย่างก็สามารถทำได้โดยไคลเอนท์แอปพลิเคชัน ดังนั้นจึงขึ้นกับลักษณะของการนำระบบไปใช้จริงที่คาดหวังไว้,ความต้องการของแอปพลิเคชัน และข้อควรพิจารณาอื่นๆ ดังนั้นในการออกแบบสามารถจะ กำหนด ไว้การ ทำงานจะอยู่ฝั่ง ไคลเอนท์ หรือฝั่งเซิร์ฟเวอร์

3.19 การส่งมอบแอปพลิเคชันเพื่อนำไปใช้จริง (Deploying an application)

หมายถึงการนำแอปพลิเคชันไปให้ผู้ใช้งานใช้ และจัดหาซอฟต์แวร์ที่จำเป็นที่ ผู้ใช้จะต้องมีไว้ใช้ในการรันแอปพลิเคชัน ในสถานะแวดล้อมของผู้ใช้ สำหรับ แอปพลิเคชันที่ไม่ใช่ฐานข้อมูลของ Delphi จะใช้เพียงไฟล์ .EXE เท่านั้น เนื่องแอปพลิเคชันของ Delphi ไม่ต้องการตัวแปลภาษาขณะทำงาน (run-time interpreter) หรือ ไดนามิก ลิงค์ ไลบรารี (DLL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ที่ต้องการในการส่งมอบแอปพลิเคชันเพื่อโน้ตบุ๊กไปใช้จริง (แบบสแตนด์อโลน)

โดยทั่วไปในการส่งมอบแอปพลิเคชันงานฐานข้อมูลเพื่อนำไปใช้จริง จะต้องสร้างแพคเกจ (Package) ที่ประกอบด้วยไฟล์ทั้งหมดที่ผู้ใช้ต้องการในการรัน แอปพลิเคชันและเข้าถึงแหล่งข้อมูล

ไฟล์เหล่านั้นได้แก่

- ไฟล์ .EXE และ ไฟล์ .DLL (ถ้ามี) ของแอปพลิเคชัน
- ไฟล์ที่เป็นส่วนประกอบต่าง ๆ เช่น ไฟล์ .README . ไฟล์ .HLP

หรือไฟล์สำหรับ ออนไลน์เฮลป์ (OnLine Help)

- บีคียู ซีพอร์ท สำหรับการเข้าถึงฐานข้อมูล (เคสคัทอป / เซิร์ฟเวอร์)
- รีพอร์ทสมิธ รันไทม์ (ReportSmith Runtime) สำหรับการพิมพ์

รายงาน

- ถ้าแอปพลิเคชันใช้ วิบีเอ็กซ์คอนโทรล (VBX Control) จะต้องใช้

ไฟล์ BIVBX11.DLL



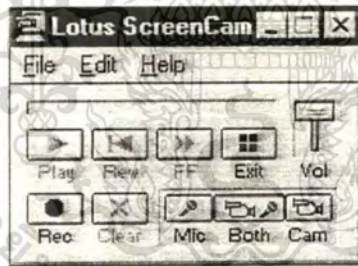
บทที่ 4

SCREENCAM 97









4.1 คำนำ

ScreenCam เป็นแอปพลิเคชันที่สามารถบันทึกภาพเหตุการณ์ต่างๆ ที่เกิดขึ้นบนเดสก์ทอปของวินโดวส์ไม่ว่าจะเป็นการเลื่อนพอยท์เตอร์ การคลิก และ ดับเบิ้ลคลิก ภาพการเปิดโปรแกรม กล่าวคือ ทุกอย่างที่เกิดขึ้นบนจอภาพ และสามารถบันทึกเสียงบรรยายต่างๆ เพิ่มเข้าไปได้ด้วย ซึ่งจะเก็บอยู่ในรูปของไฟล์ .scm เพื่อสามารถที่จะนำกลับมาแสดงใหม่ในภายหลังได้ด้วยโปรแกรม ScreenCam Player ScreenCam ใช้เป็นแอปพลิเคชัน ที่อยู่ในชุดของโปรแกรมของ Lotus Smartsuit 97 ในการเก็บภาพเหตุการณ์บนเดสก์ทอปของวินโดวส์ 95 จำเป็นที่จะต้องใช้ โปรแกรม ScreenCam 97 หรือใช้เวอร์ชันที่ใหม่กว่า เพราะโปรแกรม ScreenCam เวอร์ชันก่อนหน้านี้ยังไม่สามารถทำงานบนวินโดวส์ 95 ได้แต่สามารถทำงานบนวินโดวส์ 3.1 ได้

4.2 การทำงานของโปรแกรม



รูป 24 อินเทอร์เฟซของ Screencam 97

- | | |
|---|----------------------------|
|  | เริ่มต้นการแสดงผล |
|  | ย้อนกลับไปยังจุดเริ่มต้น |
|  | การเร่งการแสดงผล |
|  | ออกจากโปรแกรม |
|  | เริ่มต้นการบันทึกรายการ |
|  | ยกเลิกการบันทึกที่ผ่านมา |
|  | เลือกบันทึกเฉพาะเสียง |
|  | เลือกบันทึกทั้งภาพและเสียง |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เลือกบันทึกเฉพาะภาพ



Vol

ควบคุมระดับเสียง



รูป 25 อินเทอร์เน็ตของ ScreenCam Player



Play

เริ่มต้นการแสดงผล



Rew

ย้อนกลับจุดเริ่มต้น



Exit

ออกจากโปรแกรม



Open


เปิดไฟล์สำหรับการแสดง



Vol

ควบคุมระดับเสียง

4.3 การบันทึกภาพ

1. เริ่มต้นด้วยการเรียกโปรแกรม ScreenCam กดปุ่ม  เพื่อทำการเริ่มบันทึก หรือกดปุ่ม F12 (ในกรณีที่กำหนดให้เป็นคีย์เริ่มต้น)
2. เมื่อทำการบันทึกภาพเสร็จสิ้นให้กดปุ่ม Esc (ในกรณีที่กำหนดเป็นคีย์หยุด)
3. ต้องมีเนื้อที่ว่างอย่างน้อย 5 MB มิฉะนั้น โปรแกรมจะ ไม่ยอมให้บันทึก


4.4 การเก็บข้อมูลลงไฟล์

เมื่อทำการบันทึกภาพแล้วให้เลือก คำสั่ง Save ในเมนู File เพื่อทำการบันทึกโดยการใส่ชื่อที่ต้องการแล้วกดปุ่ม OK การบันทึกสามารถเลือกคำสั่ง Save As เพื่อเลือกการบันทึกแบบนามสกุลเป็นแบบ .EXE ซึ่งสามารถนำกลับมาแสดงใหม่ได้โดยไม่ต้องเรียกใช้โปรแกรม ScreenCam Player

4.5 การแสดงผลโดยใช้โปรแกรม ScreenCam player

การแสดงผลโดยใช้โปรแกรม Screencam Player สามารถทำได้ 2 วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เรียกโปรแกรม ScreenCam Player ขึ้นมาแล้วใช้คำสั่ง Open ในเมนู File เลือก File ที่จะแสดง แล้วกดปุ่ม 

2. แสดงการใช้คอมมานด์ Scmply.exe < ตัวเลือก > (ชื่อไฟล์ .scm) ซึ่ง โปรแกรมมีตัวเลือกดังนี้

ตัวเลือก

- /S ซ่อน Screencam Logo
- /P รอให้ผู้ใช้กด Play ก่อนแสดงภาพ
- /R แสดงภาพซ้ำ
- /C ปิดโปรแกรมแสดงเมื่อแสดงภาพจบ
- /H ซ่อนชุดควบคุมขณะแสดงภาพ
- /F หยุดภาพหลังการแสดงผล
- /Z จำนวนครั้งในการแสดง

4.6 การใช้งาน ScreenCam ร่วมกับ Multimedia

การเรียกใช้โปรแกรม Screencam เพื่อแสดงเหตุการณ์ สามารถทำได้โดยใช้คำสั่งคอมมานด์ไลน์ เช่น Scplyer.exe /sc Screencam เป็นการสั่งให้แสดงไฟล์ชื่อ Screen.scm พารามิเตอร์ sc เป็นการสั่งให้ซ่อน LOGO และเปิดโปรแกรมเมื่อการแสดงผลสิ้นสุด

การเรียกใช้ Screencam บน MTB ทำได้โดยใช้คำสั่ง Run เช่น Run “ scplayer.exe /sc Screencam “ แต่ในการใช้งานจริงเมื่อทำคำสั่ง Run แล้ว MTB จะทำคำสั่งถัดไปโดยไม่มีการหยุดรอโปรแกรม ที่ถูกสั่ง Run เพื่อให้ MTB หยุด รอการทำงาน จึงต้องเพิ่มสคริปต์คำสั่งควบคุมดังนี้

```
linkDll"user"  
word findWindow ( Dword,string )  
end linkDll  
linkDll. " TB30win.dll"  
int Yieldapp( )  
end linkDll  
run " Scplyer.exe /sc Screencam.scm "  
do
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

get Findwindow ( 0 ,” Screen.scn “) { คำสั่งหาหมายเลขหน้าต่างที่ แอปพลิเคชัน}
until it = 0 { ชื่อ Screen.Scsm ถ้าหาหน้าต่างไม่พบจะส่ง}
UnlinkDll “ User “ { ค่ากลับเป็น ศูนย์ซึ่งจะเก็บไว้ที่ตัวแปร it }
unlinkDll “ TB30win.Dll “

```

4.7 การใช้งานในScreenCam ที่มีพีเจอรี่ใหม่

ScreenCam 97 ได้รับการแต่งตั้งพีเจอรี่ใหม่ๆเข้ามามากมาย พร้อมกันนี้ในบางพีเจอรี่ยังได้รับการเพิ่มขีดความสามารถเข้าไปด้วย ซึ่งเราจะต้องรู้จัก ScreenCam 97 มากขึ้น เราจะต้องทดลองสร้างงานที่ต้องใช้พีเจอรี่ใหม่ ไปพร้อมๆกัน โดย

4.7.1 การทดสอบพีเจอรี่ใหม่

1. คลิก Start/Programs/Lotus ScreenCam 97 Edition/Lotus ScreenCam 97
2. เลือกคำสั่ง New ในเมนู File
3. เลือกลักษณะการบันทึกว่าจะบันทึกเสียงอย่างเดียว(รูปไมโคร โฟน) บันทึกทั้งภาพและเสียง (รูปกล้องวิดีโอกับ ไมโคร โฟน) หรือ บันทึกภาพเพียงอย่างเดียว (รูปกล้องวิดีโอ)
4. เริ่มต้นบันทึก คลิกปุ่ม Record (ปุ่มที่มีรูปวงกลมสีแดงอยู่ภายใน)
5. สังเกตจะเห็นแผงควบคุมการทำงานที่ประกอบด้วยปุ่มหยุดการบันทึกชั่วคราว (puase)กับปุ่มหยุดบันทึก (Stop)ซึ่งขณะนี้ไม่ว่าจะทำอะไรบนเคสก์ทอป เช่น เปิด/ปิดโปรแกรม,คลิกบนไอคอน,เปิดเมนู จะถูกบันทึกทันที และในกรณีที่มีซาวด์การ์ดและไมโครโฟน สามารถเลือกบันทึกเสียงเข้าไปพร้อมด้วยกันเลขก็ได้
6. เมื่อบันทึกแล้ว ให้เลื่อนพอยท์เตอร์ไปคลิกปุ่มหยุดการบันทึก
7. ทดลองสั่งให้ ScreenCam เพลย์แบ็คสิ่งที่ได้บันทึกไปแล้ว โดยคลิกปุ่มเพลย์แบ็ค
8. ถ้าพอใจกับผลลัพธ์ที่ได้ เลือกคำสั่ง Save ในเมนู File

4.7.2 การแบ่งเป็นตอนๆ (มากที่สุด 20 ตอน)

พีเจอรี่แรกที่ทำกรทดลองดูนั้นจะเป็นการแบ่งการบันทึกออกเป็นตอนย่อยๆซึ่งทำให้ไม่ต้องสร้างสรรค์เสร็จในคราวเดียวกัน เพราะสามารถนำตอนย่อยๆเหล่านี้มารวมกันได้ภายหลัง โดยทำตามขั้นตอนต่อไปนี้

1. เปิดScreenCam ขึ้นทำงาน พร้อมทั้งเลือกคำสั่ง NewในเมนูFile
2. กดปุ่ม F12 (Hotkey ในเมนู Edit)เพื่อเริ่มบันทึก
3. วาดลวดลายบนเคสก์ทอปตามต้องการ (ควรเขียน Scrip ก่อนลงมือทำจริง)
4. กดปุ่ม F12 เพื่อสั่งหยุดชั่วคราว สังเกตที่แผงควบคุมขณะบันทึกจะกระพริบ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการค้าโดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กดปุ่ม F12 อีกครั้งหนึ่ง เพื่อบันทึกตอน (segment) ที่สองต่อไป

* 6. วาดลวดลายบนเคสท์ทอปจนพอใจ เสร็จแล้วกดปุ่ม Esc (ฮอตคีย์สำหรับหยุดบันทึก)

7. ลองเลือกคำสั่ง segment.s.. ในเมนู Edit หน้าจอ โปรแกรม ScreenCam จะเปลี่ยนไป

8. สังเกตที่เป็น Sequence ทางขวามือจะมีสองตอนย่อยๆ ดังรูปที่ 4

9. ให้กำหนดชื่อตอนในแท็บบ็อกซ์ Menu Label สามารถกำหนดชื่อให้แต่ละตอนได้ โดยเลือกคลิกแต่ละตอนในช่อง Sequence ก่อน ซึ่งชื่อที่พิมพ์เข้าไปให้กับแต่ละตอนนี้ จะปรากฏให้เลือกเวลาคลิกบนปุ่ม segment.s ขณะเพลย์แบคดังรูปที่ 5

10. เสร็จแล้วจัดเก็บไฟล์ (File/Save)

หมายเหตุ: ถ้าเก็บแต่ละตอนแยกเป็นไฟล์เมื่อต้องการรวมทุกตอนเข้ากันขณะเปิดไฟล์ให้เลือกอปชั่น Append onto existing movie

4.7.3 จัดเก็บเป็น .avi

พีเจอร์นี่ทำให้สามารถส่งภาพยนตร์ไปให้กับยูสเซอร์ที่ไม่มี ScreenCam ดูได้ ซึ่งภายหลังที่บันทึกเสร็จแล้ว และกำลังจะจัดเก็บไฟล์ ให้เลือกคำสั่ง save As ... ในเมนูไฟล์ โคอะลอกบ็อกซ์ SaveAs จะปรากฏขึ้นที่ครอปควาน์ลิสต์บ็อกซ์ Save As type เลือก Video for windows file (*.avi) คลิกปุ่ม Options...คลิกแท็บ Video ดังรูปที่ 6 สามารถเลือกกำหนดจำนวนเฟรมต่อวินาที (1,2,5,10,15,20 หรือ 30) ของภาพยนตร์ที่จะจัดเก็บเป็นไฟล์ AVI ได้ โดยทั่วไปจะใช้ 15 เฟรมต่อวินาทีจะให้คุณภาพของวิดีโอที่ดีพอสมควร (ไม่ควรเลือกต่ำกว่า 8 เฟรมต่อวินาที)

หมายเหตุ : ไฟล์ที่จัดเก็บเป็น .avi จะมีขนาดใหญ่กว่าไฟล์ .scm

4.7.4 ปรับแต่งขนาดได้

* ในกรณีที่นำไฟล์ซึ่งบันทึกที่ความละเอียดต่ำ ไปเล่นที่สภาพแวดล้อมที่ความละเอียดสูงกว่า เช่นคุณอาจบันทึกที่ความละเอียด 640 X 480 แล้วไปเล่นบนความละเอียด 800 X 600 เวลาเล่นเราจะเห็นหน้าจอมีขนาดเล็กกว่าเคสท์ทอป ซึ่งในกรณีนี้เราสามารถปรับขนาดได้ โดยที่แผงควบคุมโปรแกรม ScreenCam เลือกคำสั่ง Preference ในเมนู Edit จากนั้นคลิกแท็บ Setting ในพื้น General เลือกเช็คบ็อกซ์ Allow Resizing of the Mview View During Playback คลิกปุ่ม OK เพื่อกลับไปแผงควบคุมการทำงานเหมือนเดิม คราวนี้เมื่อเราสั่งเล่นไฟล์ที่บันทึกที่ความละเอียดต่ำกว่า เราก็จะสามารถปรับขนาดของการแสดงผลได้แล้วดังรูปที่ 7

4.7.5 ใส่คำอธิบาย

เราสามารถใส่คำอธิบาย (caption) เข้าไปในงานของเราได้ เพราะบางที่ผู้ที่นำไฟล์ของคุณไปเล่น อาจจะไม่มีชาวคาร์ท์คี่ได้ ซึ่ง ScreenCam 97 สามารถให้คำตอบของการใส่คำอธิบายที่ง่ายมาก ลองทำตามขั้นตอนต่อไปนี้ดูนะครับ

1. เรียกโปรแกรม ScreenCam เลือกคำสั่ง New ในเมนู File
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการค้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกคำสั่ง Captions ในเมนู Edit

3. เลือกแท็บ Background เพื่อเลือกกรอบคำอธิบาย สมมติเลือกแบบ sticky.bmp ดังรูปที่ 8

4. เมื่อเลือกกรอบได้แล้ว ที่แท็บ Font ให้เราเลือกกำหนดฟอนต์ที่ต้องการ (ใช้ฟอนต์ภาษาไทยได้)

5. ที่แท็บ Position เราสามารถกำหนดตำแหน่งที่คำอธิบายจะปรากฏ ซึ่งมีสองออปชันให้คุณเลือก คือตรงไหนก็ได้บนเดสก์ทอปตามที่คุณต้องการ (Leave at Freehand Position) หรือไว้ตรงกลาง (Center On Screen) ดังรูปที่ 9

6. ไปที่แท็บ Sequence เพื่อแก้ไขข้อความที่จะใส่เข้าไปโดยคลิกปุ่ม Edit Text พิมพ์คำอธิบายที่ต้องการเข้าไป คลิกปุ่ม View Text เพื่อตรวจสอบว่าจัดวางดีแล้วหรือยัง จากนั้นใส่คำอธิบายอันต่อไปซึ่งถ้าเป็นอันที่อยู่หลังจากอันนี้ ก็ให้คลิกปุ่ม Add After แต่ถ้าต้องการแทรกคำอธิบายไว้ก่อนหน้านี้ให้คลิกปุ่ม Insert Before ดังรูปที่ 10

7. หลังจากใส่คำอธิบายเข้าไปหมดแล้วให้เริ่มบันทึกได้ซึ่งเมื่อถึงตำแหน่งที่เราต้องการใส่คำอธิบายอันแรก ให้กดปุ่ม F2 (ชอคคีย์เรียกคำอธิบายไปกด F3 เมื่อต้องการใส่คำอธิบายถัดไปก็กด F2 อีกที เห็นไหมครับว่า คุณจะต้องเตรียมสคริปท์ให้ดีก่อนบันทึก

4.7.6 ปิดกล่อง

พีเจอร์ที่ช่วยให้สมรรถนะการทำงานของ ScreenCam ดีขึ้น (ซึ่งที่ดีฟอลต์โปรแกรม ScreenCam จะกำหนดค่าที่เหมาะสมมาให้แล้ว)

บทที่ 5

เริ่มต้นกับ HTML

5.1 ความรู้ทั่วไปของภาษา HTML

HTML ย่อมาจากคำว่า Hypertext Text Markup Language เป็นรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบนอินเทอร์เน็ต เหมือนกับที่เราใช้ระบบปฏิบัติการ DOS ซึ่งถูกตัดแยกออกมาจากโปรแกรมระบบปฏิบัติการ UNIX เช่นเดียวกับ HTML ซึ่งเป็นภาษาหลักสำหรับการสร้างโฮมเพจ เพิ่มเอกสาร HTML ที่สร้างขึ้นจะนำไปแสดงผลได้ด้วยโปรแกรมเว็บเบราว์เซอร์ เช่น โปรแกรม Netscape Navigator หรือ Mosaic

HTML เป็นภาษาที่ง่ายต่อการเรียนรู้และการเขียน ซึ่งจัดได้ว่าง่ายที่สุด ง่ายกว่าภาษาคอมพิวเตอร์ทุกตัวที่เคยมีมาในโลกนี้ แต่ก่อให้เกิดคุณประโยชน์มากมายจนทำให้เราลืมไปว่า นี่เป็นเพียงส่วนหนึ่งของภาษาใหญ่ที่มีขีดความสามารถสูงกว่านี้

5.2 HTML ทำงานอย่างไร

การให้บริการบนอินเทอร์เน็ต ไม่ว่าจะเป็น E-Mail, FTP, Gopher, Telnet หรือบริการอื่นๆ ต้องใช้อุปกรณ์เชื่อมต่ออินเทอร์เน็ตของฮาร์ดแวร์ที่สามารถทำงานได้ด้วยโปรแกรมเฉพาะที่ทำงานบนอินเทอร์เน็ตเท่านั้น

WWW แบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็น Client และส่วนที่เป็น Server เหมือนกับที่มีแบ่งกันในระบบเครือข่ายทั่วไป ทั้งสองส่วนจะถูกเชื่อมโยงต่อกันผ่านทางอินเทอร์เน็ต โดยมี HTML เป็นส่วนฐานข้อมูลสำคัญ เมื่อเว็บเบราว์เซอร์ส่งข้อความร้องขอข้อมูลที่อยู่ในรูปแบบของไฟล์ HTML จากเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ผ่านโมเด็มหรืออุปกรณ์สื่อสารข้อมูลอื่นไปยังศูนย์บริการอินเทอร์เน็ต (ISP) ตามโพรโตคอล (Protocol) ที่กำหนดไว้ผ่านทาง URLs (Uniform Resource Locations) และเมื่อข้อมูลเดินทางมาถึงเว็บเบราว์เซอร์ ศูนย์บริการปลายทางที่ผู้ใช้ต้องการ ณ ที่นี้เครื่องเว็บเบราว์เซอร์ของศูนย์ฯ จะทำการอ่านข้อมูลที่ถูกส่งเข้ามาและจะทำงานตามคำสั่งที่กำหนด โดยอาจมีการเชื่อมโยงไปยังเว็บเบราว์เซอร์อื่นอีก หลังจากจบสิ้นกระบวนการแล้วจะทำการจัดส่งข้อมูลคำตอบย้อนกลับมายังเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ โปรแกรมเว็บเบราว์เซอร์ที่เครื่องคอมพิวเตอร์ของเรา ก็จะแปลงสัญญาณคำสั่งและแสดงผล เป็นข้อความ รูปภาพ เสียง ที่เราใช้งานกันต่อไป

HTML นอกจากใช้ในการสร้างฐานข้อมูลบนอินเทอร์เน็ตแล้ว ยังมีความสามารถทางด้าน การเชื่อมโยงข้อมูลจากแหล่งข้อมูลอื่นบนอินเทอร์เน็ต ไม่ว่าจะเป็น E-Mail, FTP, Gopher, ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

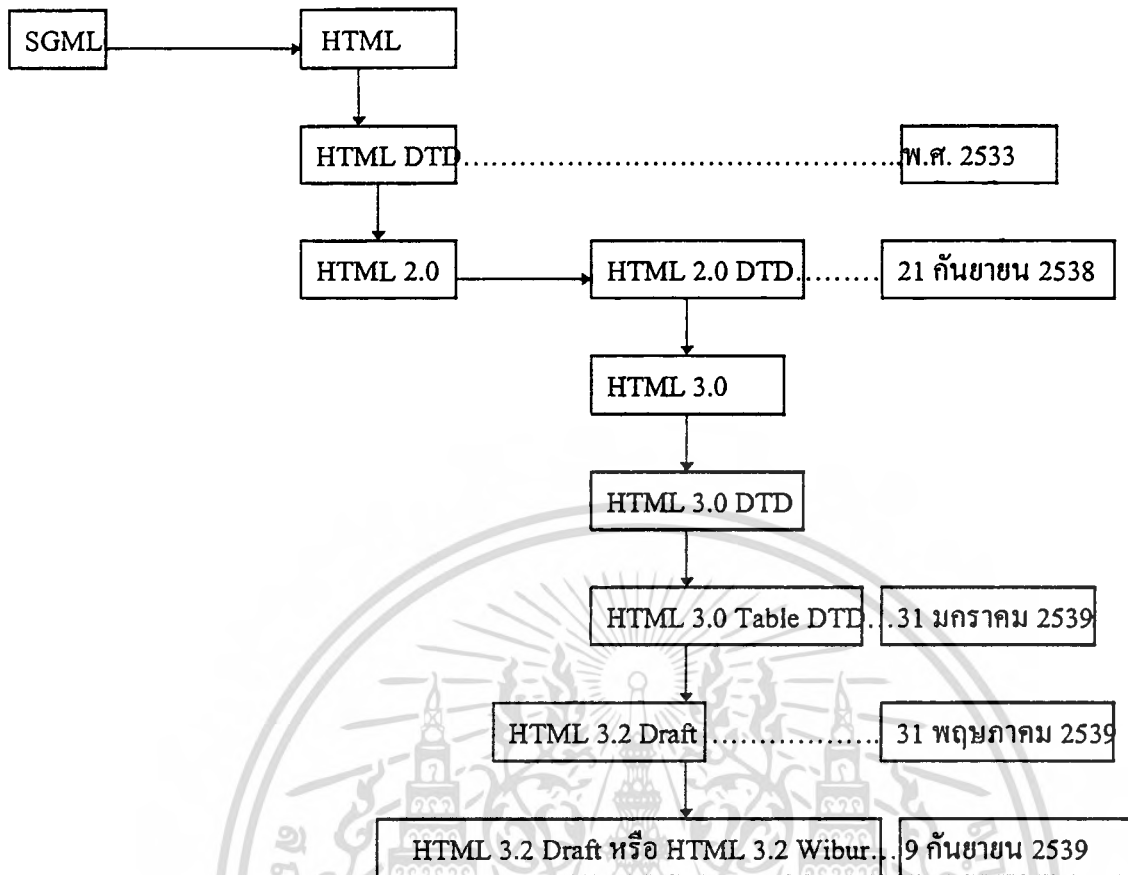
Telnet หรือ News (ขึ้นอยู่กับชนิดของเว็บเบราว์เซอร์แต่ละชนิดว่ามีความสามารถหรือไม่) ทำให้เราสามารถเรียกใช้บริการเหล่านี้ได้โดยทันที ต่างจากเมื่อขณะเริ่มแรก จะใช้บริการใดก็ต้องไปหาโปรแกรมที่ทำงานเฉพาะมาทำงาน แต่เนื่องด้วยความสามารถของ HTML ข้อมูลต่างๆ จะถูกนำมาแสดงตรงหน้าผู้ใช้โดยเครื่องคอมพิวเตอร์ทำหน้าที่ประมวลผลข้อมูลผ่านทาง โพรโทคอล HTTP เป็น โพรโทคอลหลักที่ทำให้เราติดต่อสื่อสารบนอินเทอร์เน็ตได้

5.3 พัฒนาการของภาษา HTML

ภาษา HTML ได้รับการดัดแปลงมาจากภาษา SGML เพื่อใช้เป็นไฟล์เอกสารในระบบ เวิลด์ไวด์เว็บ เมื่อปี พ.ศ. 2533 ซึ่งไฟล์เอกสารดังกล่าวนับเป็นภาษา HTML รุ่นแรก ต่อมาได้รับการพัฒนาให้เป็น HTML 2.0 และ HTML 2.0 DTD ได้รับการยอมรับตามข้อตกลงเมื่อวันที่ 21 กันยายน 2538 ภาษา HTML ได้รับการพัฒนาโดยผู้ผลิตเวิลด์ไวด์เว็บเบราว์เซอร์ เพื่อเพิ่มความสามารถและประสิทธิภาพในการแสดงข้อมูลบนจอภาพของ โปรแกรมเวิลด์ไวด์เว็บเบราว์เซอร์นั้น ให้มากที่สุด โดยได้พัฒนาภาษา HTML 3.0 DTD ตามข้อตกลงเมื่อวันที่ 4 มีนาคม 2538 และได้รับการเพิ่มความสามารถในการแสดงตารางบน โปรแกรมเวิลด์ไวด์เว็บเบราว์เซอร์จนกลายเป็น HTML 3.0 Table DTD เมื่อวันที่ 23 มกราคม 2539 ดังนั้นภาษา HTML ที่ใช้กันอย่างกว้างขวางจึงเป็น HTML 3.0

โปรแกรมเวิลด์ไวด์เว็บเบราว์เซอร์มีความสามารถแสดงผลของไฟล์เอกสาร HTML ได้ต่างกัน เนื่องจากผู้พัฒนาโปรแกรมเวิลด์ไวด์เว็บเบราว์เซอร์ได้พัฒนา โปรแกรมให้มีความสามารถ และเข้าใจคำสั่งใหม่ได้ตลอดเวลา ทำให้ภาษา HTML มีการพัฒนาและแตกต่างไปจากกรอบของ ภาษาที่วางไว้ แม้ในปัจจุบันภาษา HTML ได้รับการพัฒนาโดยบริษัทผู้เขียนเวิลด์ไวด์เว็บเบราว์เซอร์ตามแบบฉบับและความคิดของตนเองทำให้ภาษา HTML ขาดความเป็นมาตรฐาน ด้วยเหตุนี้กลุ่มบริษัทผู้เขียน โปรแกรมเวิลด์ไวด์เว็บเบราว์เซอร์และองค์กรผู้มีส่วนรับผิดชอบในระบบเวิลด์ไวด์เว็บจึงได้ก่อตั้งองค์กรร่วม ชื่อ “เวิลด์ไวด์เว็บ-คอนเซอร์เทียม” หรือ WWWC (World Wide Web Consortium) ประกอบด้วยบริษัท Sun Microsystem, บริษัท IBM, บริษัท Microsoft, บริษัท Netscape Corporation, บริษัท Novell , บริษัท SoftQuad, บริษัท Spyglass จึงได้ตกลงร่วมกัยในการสร้างภาษา HTML ให้เป็นมาตรฐานเดียวกัน โดยเสนอให้มีการร่างภาษา HTML ฉบับที่มีชื่อว่า HTML 3.2 Draft ตามข้อตกลงเมื่อวันที่ 31 พฤษภาคม 2539 ซึ่งในที่สุดได้รับการพัฒนาจนเป็นภาษา HTML ฉบับที่มีชื่อว่า HTML 3.2 DTD Draft หรือที่เรียกว่า HTML 3.2 รุ่น Wibur เมื่อวันที่ 21 สิงหาคม 2539 สำหรับภาษา HTML ฉบับต่อไปนั้นเป็นฉบับที่เรียกว่า HTML 3.2 รุ่น Cougar การพัฒนา HTML สามารถสรุปเป็นแผนผังดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแผนผังแสดงการพัฒนาของภาษาเอชทีเอ็มแอล

5.4 การใช้ Notepad

การเขียน HTML นั้นเราสามารถ ใช้ Text Editor อะไรก็ได้ในการการสร้าง ไฟล์ HTML แต่ในที่นี้จะแนะนำการใช้ Notepad ซึ่งเป็น Editor ที่มีอยู่แล้วใน Microsoft windows ทั่วไป
ขั้นตอน:

- 1.เปิดNotepad โดยคลิกบน ปุ่มSTART ที่มุมขวาล่างของจอ
เลือก Program ตามด้วย Accessories แล้วเลือกที่ Icon ของ Notepad
- 2.Notepad นั้นหน้าต่างจะเป็นหน้าต่างเอกสารเปล่า ซึ่งสามารถพิมพ์ได้ทันที
แต่ถ้าต้องการเปิดไฟล์ที่เขียนไว้แล้ว ก็เลือก Open จาก File menu
- 3.ปกติ Notepad จะเปิดแต่ไฟล์ที่มีจุดเป็น .txt แต่ไฟล์ HTML นั้นจะเป็น
.htm หรือ .html ซึ่งสามารถเปลี่ยนได้ในช่องของ Files of type เป็น All files
- 4.เราสามารถใส่เพิ่ม แก้ไข ไฟล์ HTML ของเราได้เหมือนกับ Word procesor ทั่วไป
- 5.การบันทึก ไฟล์ก็เลือก Save หรือ Save as จาก File Menu
แล้ว Save ไฟล์ ให้นามสกุลเป็น .htm หรือ .html

เพราะถ้าใช้นามสกุลอื่น WEB จะไม่รู้จักรว่าเป็นไฟล์ HTML
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อการศึกษาเท่านั้น. เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การใช้ Markup Tag

การใช้ Markup Tag เป็นการเพิ่มเติมส่วนที่เป็นการจัดรูปแบบของ Web Page ที่จะออกมาสามารถที่จะเชื่อมต่อกับไฟล์อื่น เสียง ภาพ ทำให้เกิดสีสันต่างๆของ Webpage

รูปแบบการใช้ Markup Tag คือ <...>ตามด้วยเนื้อหา</...>เช่น

การให้ข้อความเป็นตัวหนาพิมพ์ That is my Web page

2.3 การเขียน HTML อย่างง่าย

โดยทั่วไปมี 2 ส่วน คือ ส่วน Head และ Body

```
<HTML>
:
<HEAD> 1) HEAD จะเป็นการบอกข้อมูลสำคัญ
:       ของ Home Page นั้นเช่น หัวเรื่อง (Title)
</HEAD>
<BODY> 2) BODY จะเป็นส่วนเนื้อหาจริงๆ
:
</BODY>
<HTML>
```

5.6 ทำความเข้าใจขั้นพื้นฐานของ HTML

1 ส่วนหัว(Head Section)

<HEAD>.....<HEAD> ในนี้มีอยู่ส่วนเดียวคือ TITLE จะปรากฏที่ แถบ title ของ Browser ซึ่งเป็นคำสั้นๆ ได้ใจความบอก

2 ส่วนเนื้อหา (Body Section)

จะมีส่วนที่เป็น text, graphic, hypertext link และข้อมูลอื่นๆที่จะปรากฏบน Page

3 การใช้ Heading

เป็นการเขียนส่วนหัวเรื่องในบรรทัดแรกของ Page ซึ่งมี 6 ขนาด

รูปแบบ: <HEAD> | / x คือ ตัวเลข 1....6
<Hx>.....</Hx>
<HEAD>

เช่น <H1>1 จะขนาดใหญ่ที่สุด <H2><H3>....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การใช้ Paragraph Tag

ใช้กำหนดย่อหน้าของข้อความย่อหน้าของข้อความ

<P>.....</P>

5.7 รูปแบบตัวอักษร

1 รูปแบบตัวอักษร โดยใช้ Physical Tag

เป็นรูปแบบของ Tag ที่เปลี่ยนข้อความที่ปรากฏ ตัวหนา ตัวเอียง ด้วยก ตัวห้อย ตัวพิมพ์ ตัวขีดเส้น ตัวใหญ่ ตัวเล็ก

 <I> <TT> <SUP> <SUB> <BIG> <SMALL> <U> ใช้เหมือน tag ทั่วไป เช่น This is bold text

2 รูปแบบตัวอักษร โดยใช้ Logical Markup Tag

เป็นtagอีกแบบที่ ผลของการใช้ตัวอักษรพิเศษแบบนี้จะขึ้นอยู่กับ Web Browser แต่ละตัว ซึ่งผลที่เกิดขึ้นไม่จำเป็นต้องเหมือนกัน ได้แก่ <CITE> <CODE> <KBD> <SAMP> <VAR>

8 การจัดวางตำแหน่งข้อความ(Format Paragraphs)

การจัดวางรูปแบบของข้อความเช่น ให้ชิดซ้าย ขวา หรือ จัดไว้ตรงกลาง

<P ALIGN=LEFT> right or center
<P CLEAR=LEFT>

4 การแบ่งข้อความ(Text Break)

เป็นการตัดข้อความในจุดที่ต้องการ

 <HR> <BLOCKQUOTE>

5.8 Hypertext Links

เราสามารถเขียน HTML เพื่อใช้เชื่อมต่อกับ Web Page จาก web site อื่น หรือกับส่วนต่างๆใน Web page เดียวกัน

การสร้าง Hyperlink

ต้องรู้ที่ต้ง (URL) ที่ต้องการจะเชื่อมต่อ

เช่น <http://www.xensei.com/users/zug/zugmain/zugmain.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The zug home page.
เป็น text ที่ click เพื่อเชื่อมต่อ

การใช้ ID Attribute

ใช้กำหนดส่วนหนึ่งของ Web Page ให้สามารถเรียกใช้ได้ง่าย ส่วนมากจะเป็นส่วนของ Heading หรือ Paragraph

<P ID="Myparagraph">.....</P>

โดยเวลาเรียกโยเพิ่มเครื่องหมาย #ตามด้วยชื่อ ID

การใช้ Relative Path Names

โดยปกติการที่จะเชื่อมต่อ Web Page อื่นๆ จะต้องพิมพ์รูปแบบเต็มๆ แต่ถ้าเป็นกรณีที่ Web Page นั้นอยู่ Web Server เดียวกันก็พิมพ์แค่ชื่อของ Web Page ก็พอ

เช่น

<A HERF="Football/ball97.html"

5.9 การใส่รูปภาพลงในเว็บเพจ

การใส่รูปภาพเพื่อการแสดงผลนั้น ให้ใช้คำสั่ง เป็นตัวกำหนดรูปภาพที่ต้องการ ไฟล์รูปภาพควรเป็นไฟล์ชนิด GIF หรือ JPEG เพราะโปรแกรมเว็บเบราว์เซอร์ส่วนใหญ่สนับสนุนไฟล์ทั้ง 2 ชนิดนี้

5.10 การใส่ข้อความสำหรับการแสดงผลแบบเท็กซ์

ผู้ใช้บริการเว็บเพจทั่วไปในอินเทอร์เน็ต ไม่ใช่ทุกรายที่สามารถใช้งานแบบกราฟิกได้ สามารถดูภาพสวยงามซึ่งเราบรรจุไว้ในเว็บเพจที่สร้างขึ้นมาได้ ยังมีอีกหลายท่านที่ใช้งานอินเทอร์เน็ต แบบเท็กซ์ ไม่สามารถแสดงข้อมูลออกมาเป็นรูปภาพ การที่จะสื่อสารกับผู้ใช้เหล่านั้นด้วยนั้นย่อมเป็นไปได้ แต่เพื่อไม่ให้เกิดความน้อยเนื้อต่ำใจและป้องกันความสับสนอันเกิดจากการกำหนด ให้แสดงรูปภาพแต่บนจอภาพกลับไม่มีรูปอะไรแสดงให้เห็น การสร้างเว็บเพจเพื่อให้ผู้ใช้สามารถใช้งานได้กว้างขวางนั้นจำเป็นต้องสร้างข้อมูลเพื่อไว้สำหรับผู้ที่ไม่ได้ใช้งานแบบกราฟิกด้วย โดยให้เขียนข้อความกำกับลงไปกับรูปภาพที่จะนำออกแสดง เพื่อให้ผู้ใช้ได้ทราบว่ารูปนั้นเป็นรูปอะไร ส่วนจะเห็นว่าสวยงามอย่างไรนั้นก็ต้องใช้จินตนาการตามหลักการจิตรกร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการเพิ่มแอตทริบิวต์ ALT=“ข้อความ” ต่อจาก SCR = “ไฟล์รูปภาพ” แอตทริบิวต์นี้จะเป็นตัวกำหนดคำอธิบายของรูปภาพที่นำมาแสดงบนจอภาพ ทั้งนี้สำหรับผู้ใช้อินเทอร์เน็ตแบบเท็กซ์รูปภาพที่กำหนดไว้จะปรากฏบนจอภาพแทนด้วยตัวอักษร [IMAGE] เท่านั้น เมื่อเรากำหนดให้ ALT= “ข้อความ” สิ่งที่ปรากฏบนจอภาพนอกเหนือจาก [IMAGE] แล้วยังจะมีข้อความที่เราเขียนอธิบายความหมายของภาพนั้นให้กับผู้ใช้อินเทอร์เน็ตแบบเท็กซ์ได้ทราบเพิ่มขึ้นมาอีกด้วย

5.11 การวางตำแหน่งรูปภาพ

ปกติเมื่อเรากำหนดข้อความเพื่อแสดงความหมายหรือคำกับภาพ ถ้ารูปภาพที่แสดงนั้นไม่ใหญ่เกินไป ข้อความนั้นจะปรากฏอยู่ที่ส่วนล่างของภาพทางขวามือเสมอ เราสามารถกำหนดตำแหน่งของรูปภาพที่แสดงบนจอภาพได้ด้วยการใช้แอตทริบิวต์ ALIGN ซึ่งเป็นตัวกำหนดตำแหน่งการแสดงผลว่ารูปภาพนั้นจะถูกจัดวางที่ใด HTML กำหนดตำแหน่งของการแสดงผลได้ 2 แบบดังนี้

1. แบบแนวนอน ประกอบไปด้วยตำแหน่งซ้าย (left) และ ขวา (right)
2. แบบแนวตั้ง ประกอบไปด้วยตำแหน่งต่าง ๆ 3 ระดับ ดังนี้

เสมอบน มี 2 คำสั่ง คือ top และ texttop

กึ่งกลาง มี 2 คำสั่ง คือ middle และ absmiddle

เสมอล่าง มี 3 คำสั่ง คือ baseline , bottom และ absbottom

5.12 WIDTH และ HEIGHT

รูปภาพที่นำมาแสดงผลด้วยวิธีปกติจะมีขนาดเท่ากับขนาดจริงของภาพนั้น ๆ ซึ่งบางครั้งก็เล็กหรือใหญ่เกินไป ไม่ตรงกับขนาดที่เราต้องการจะนำมาแสดงผล จึงควรมีการกำหนดขนาดของรูปภาพที่จะนำมาแสดงใหม่ให้เหมาะสม

การทำให้รูปภาพที่นำมาแสดงผลมีขนาดตามความต้องการของเรานั้น ต้องกำหนดขนาดความกว้างและความสูงของรูปภาพที่นำมาแสดงด้วยแอตทริบิวต์ WIDTH และ HEIGHT ซึ่งมีหน่วยเป็นจุดพิกเซลล์

แอตทริบิวต์ WIDTH และ HEIGHT จะวางไว้ส่วนหน้าหรือต่อท้ายจาก SRC ได้ทั้งนั้น จะเขียนเป็น ก็ได้

มีอยู่บ่อยครั้งที่เรานำรูปภาพมาลดขนาดลงจนเป็นไอคอน เพื่อความสะดวกในการแสดงผลรูปภาพ ถ้าต้องการให้รูปขยายใหญ่ขึ้นเท่าขนาดปกติเพียงเลื่อนเมาส์ไปชี้และคลิกเท่านั้น ทำได้โดยกำหนดให้จุดเชื่อมโยงเป็นไฟล์รูปภาพนั้น ๆ ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.13 การปรับขนาดรูปภาพให้ได้สัดส่วน

การกำหนดขนาดความกว้างยาวของรูปภาพนั้น เป็นการกำหนดขนาดรูปภาพที่อาจทำให้ได้รูปภาพที่มีสัดส่วนไม่ถูกต้อง ถ้าเรากำหนดความกว้างหรือความยาวไม่เหมาะสม ภาพที่ได้จะผิดเพี้ยนไป การกำหนดขนาดของรูปภาพเป็นสิ่งเลียมจตุรัสนั้นอาจจะทำให้ภาพที่ได้เกิดการผิดเพี้ยนได้ เนื่องจากขนาดความกว้างและความยาวไม่เท่ากัน ดังนั้นก่อนจะทำการกำหนดให้รูปภาพมีขนาดเท่าใด จึงควรดูขนาดของรูปภาพจริง ๆ ก่อนว่าเหมาะสมหรือไม่ มีความสมดุลเพียงใด

แต่ถ้าต้องการให้ภาพที่แสดงมีขนาดเล็กลงหรือใหญ่ขึ้นตามอัตราส่วนที่ถูกต้องแล้ว ให้กำหนดความกว้างของรูปภาพเป็นจำนวนเปอร์เซ็นต์ของรูปภาพเพียงอย่างเดียว เช่น $WIDTH = 50\%$ หรือ $WIDTH = 150\%$ เท่านั้น ภาพที่ได้ก็จะมีสัดส่วนเหมาะสมตามขนาดที่กำหนดลงไป

5.14 BORDER

รูปภาพที่แสดงบนจอภาพ โดยปกติจะ ไม่มีเส้นกรอบ ถ้าต้องการให้ภาพที่แสดงมีเส้นกรอบให้ใส่แอตทริบิวต์ BORDER เราสามารถกำหนดขนาดกรอบของรูปภาพได้ด้วยการกำหนดให้แอตทริบิวต์ $BORDER = n$ ปกติ n จะมีค่าเป็น 0 (เขียนเพียงคำว่า BORDER ไม่ต้องใส่ $=0$ ลงไป) ค่าเลข n ยิ่งมากความหนาของกรอบรูปภาพก็ยิ่งมากขึ้นตามลำดับ

5.15 BSPACE และ HSPACE

ปกติรูปภาพที่ใช้แสดงผล จะถูกวางตำแหน่งให้ชิดติดกันกับภาพ หรือข้อความที่ปรากฏอยู่ใกล้เคียง หากต้องการเพิ่มระยะห่างดังกล่าวนี้ ทำได้โดยใช้แอตทริบิวต์ VSPACE สำหรับระยะห่างด้านบน - ล่าง และ HSPACE สำหรับระยะห่างด้านซ้าย - ขวา

บางครั้งเราอาจมีความจำเป็นจะต้องใช้รูปภาพที่มีขนาดเท่ากันวางซ้อนกัน เพื่อเพิ่มความน่าสนใจให้กับการแสดงผล ภาพแรกที่เกิดขึ้นจะปรากฏเพียงชั่วขณะ จากนั้นภาพที่สองจะถูกแสดงแทนภาพแรกทันที ด้วยการกำหนดแอตทริบิวต์ LOWSRC = "รูปภาพ"

5.16 การแสดงภาพฉากหลัง

การแสดงผลบนจอภาพของเว็บเบราว์เซอร์นั้นจะแสดงด้วยตัวอักษรสีค้ำบนพื้นสีเทา ยกเว้นว่าเราจะทำการเปลี่ยนสีตัวอักษร และสีของฉากหลังไปเท่านั้น ซึ่งนั่นยังให้ผลไม่เป็นที่น่าพอใจมากนัก ในที่นี้เราจะนำรูปภาพมาใส่ลงที่ฉากหลังเพื่อเพิ่มความสวยงามให้กับเว็บเพจ เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดฉากหลังของเว็บเพจให้มีรูปลักษณะตามที่เรต้องการ โดยการเพิ่มแอตทริบิวต์ BACKGROUND = “รูปภาพ” เข้าไปในคำสั่ง <BODY> เพียงเท่านั้น การแสดงผลข้อมูลก็จะมีรูปภาพตามที่เรต้องการแสดงออกมาที่พื้นฉากหลัง

สำหรับเว็บเบราว์เซอร์บางตัวที่ไม่มีคุณสมบัติในการแสดงผลบนพื้นฉากหลัง การเขียนคำสั่งนี้ลงไป จึงไม่เกิดผลใด ๆ ขึ้นทั้งสิ้น การเพิ่มพื้นฉากหลังให้กับจอภาพได้ จึงขึ้นอยู่กับชนิดและรุ่นของเว็บเบราว์เซอร์ที่ถูกนำมาเรียกใช้งาน

การทำให้พื้นของฉากหลังมีลวดลายที่สวยงาม เป็นการเพิ่มความสนใจให้กับเว็บเพจของเราได้ดี แต่ควรระวังไม่ใช้ฉากหลังที่มีลวดลายและสีสันฉูดฉาดมากเกินไป เพราะจะทำให้พื้นของฉากหลังลดความเด่นชัดของข้อมูลที่จะนำเสนอได้

5.17 การสร้างภาพเคลื่อนไหว

ปกติรูปภาพที่แสดงบนจอภาพจะเป็นได้เพียงภาพนิ่งเท่านั้น แต่ถ้ารูปภาพที่ปรากฏบนจอภาพสามารถเคลื่อนไหวได้ ก็จะเป็นการเพิ่มความน่าสนใจให้กับเว็บเพจนั้น ๆ มากพอคู่ทีเดียว

5.18 การสร้างแบบสอบถาม

แบบสอบถามคืออะไร

แบบสอบถามหรือแบบฟอร์ม (FORM) บนอินเทอร์เน็ตก็เป็นเหมือนแบบฟอร์มทั่วไปที่เราได้พบเห็นกัน ไม่ว่าจะเป็นแบบฟอร์มที่เราต้องการกรอกกันตั้งแต่สมัครเข้าเรียนหนังสือ สมัครงาน สั่งซื้อสินค้า การตอบแบบสอบถาม สรุปก็คือเอกสารใด ๆ ที่จัดเรียงข้อมูลให้อยู่อย่างเป็นระเบียบ มีช่องให้เราเติมข้อความลงไป หรือมีตัวเลือกล้วนเรียกว่าเป็นฟอร์มได้ทั้งสิ้น การออกแบบสอบถาม

ผู้ที่ออกแบบสอบถามได้ ต้องทำความเข้าใจกับวัตถุประสงค์ของการสร้างแบบสอบถามขึ้นมาก่อนว่าต้องการที่จะทราบสิ่งใด ต้องการให้ผู้ใช้ตอบคำถามใด ต้องทำความเข้าใจในสิ่งที่ตนเองต้องการให้มากที่สุด เช่น ต้องทราบชื่อผู้สมัครรับการเลือกตั้ง หมายเลขที่ได้รับพรรค กลุ่มการเมืองที่สังกัด นโยบายที่ผู้สมัครแต่ละท่านต้องการให้ประชาชนรับทราบ สิ่งเหล่านี้เป็นส่วนสำคัญมากในการให้ข้อมูลเบื้องต้นของผู้รับสมัคร

เพศ อายุ การศึกษา อาชีพ ระดับเงินเดือน สิ่งเหล่านี้มีความสำคัญต่อการจัดจำแนกกลุ่มผู้ตอบคำถาม และหากไม่มีข้อมูลเหล่านี้ การสรุปผลจะออกมาได้อย่างถูกต้องหรือไม่

การออกแบบสอบถามจึงต้องใช้พื้นฐานของข้อมูลทางสถิติ นำไปประมวลผลได้

เมื่อทราบถึงข้อมูลพื้นฐานที่ต้องการดีแล้ว จึงนำเอาสิ่งที่มีไปสร้างแบบสอบถาม เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราเริ่มร่างแบบฟอร์มที่ต้องการลงในกระดาษเปล่า จนได้แบบที่ต้องการแล้วจึงค่อยลงมือเขียน

5.19 การสร้างแบบสอบถาม

แต่เดิม HTML จะมีลักษณะของการสื่อสารข้อมูลแบบสื่อทางเดียว (One Way) คือใช้อ่านข้อมูลได้เพียงอย่างเดียว ด้วยความสามารถของ HTML 2.0 ทำให้เราสามารถสื่อสารกับผู้อื่นได้ในลักษณะสื่อสองทาง (Two Way) นั่นคือนอกจากการอ่านข้อมูลเพียงอย่างเดียวแล้ว เรายังสามารถส่งข้อมูลกลับไปยังศูนย์บริการได้อีกด้วย

ในรุ่นต่อมา มีการพัฒนาภาษา HTML ให้มีความสามารถเพิ่มมากยิ่งขึ้น การสื่อสารระหว่างเครื่องคอมพิวเตอร์จึงมีลักษณะ 2 ทาง สามารถโต้ตอบกันได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

DIRECTOR 5.0



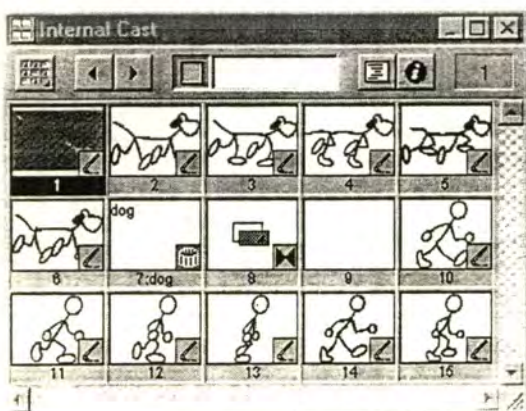
เป็นโปรแกรมที่ช่วยในการสร้างไฟล์ .avi อย่างง่ายๆ ได้ มีข้อเด่นในการแปลงไฟล์ให้เป็น .exe ได้

6.1 ส่วนประกอบสำคัญในการสร้าง movie



6.2 Cast window

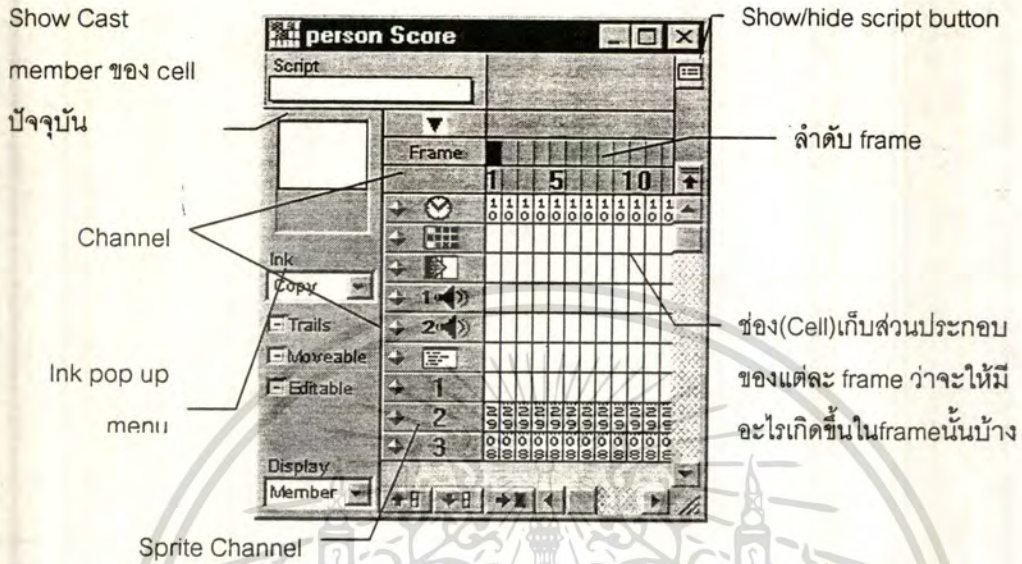
- แต่ละช่องจะเป็นmemberของmovie เช่น text, image, sound, video, film loop, script, effect(tempo, transition, palette..) สามารถimport ได้จาก file หลายประเภท



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้

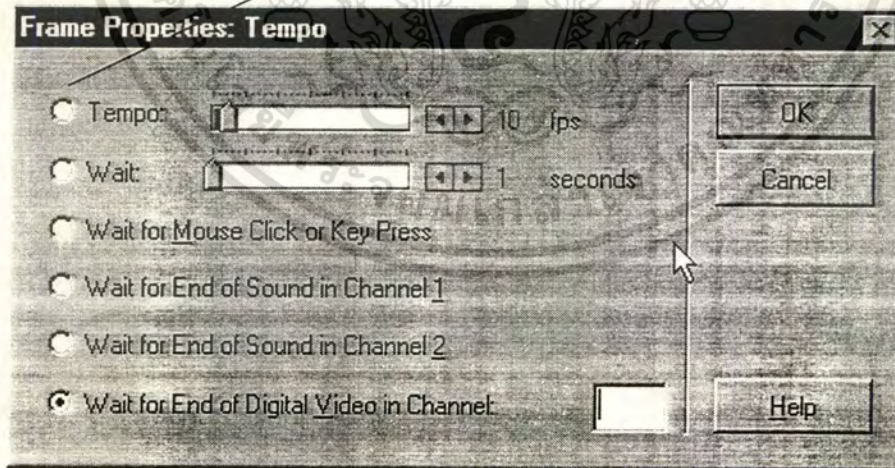
6.3 Score windows

ที่จัดการลำดับภาพและกำหนดคุณสมบัติของmemberที่อยู่ในmovie



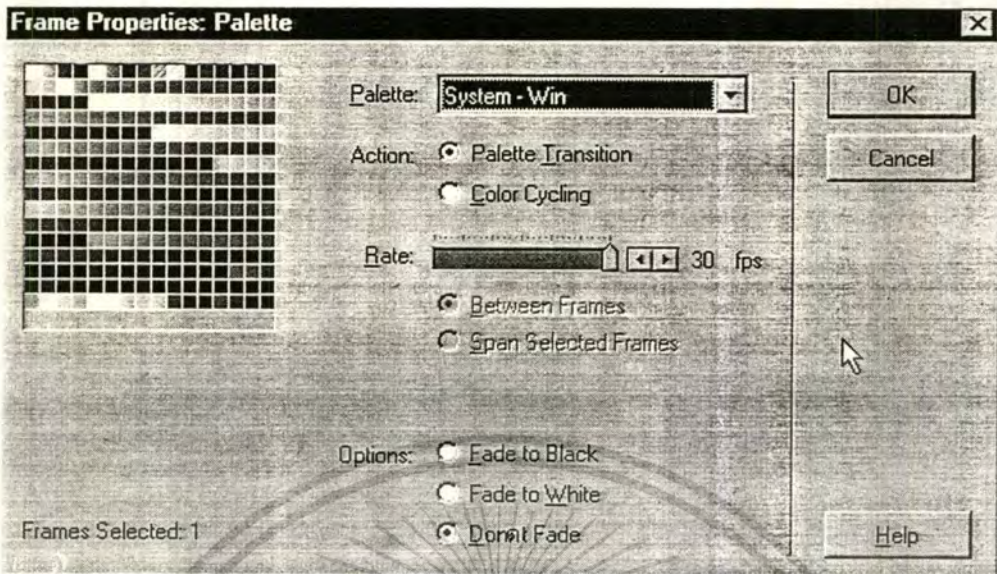
6.4 Channel

Tempo-ใช้กำหนดคุณสมบัติเรื่องเวลาของคพฟทำนั้นๆเช่น Frame per second กำหนด Frame per second



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Palette-กำหนด Palette แต่ละ Frame



Transition-รูปแบบการเปลี่ยนFrame



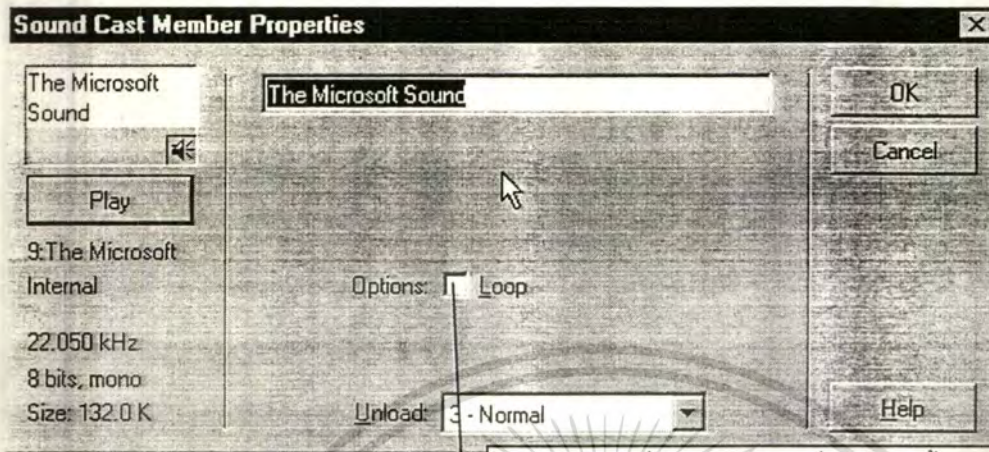
เวลาต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

→ 1 | Sound-เสียงมี2channel เล่นได้ 2 เสียงพร้อมกัน

Sound property Double Click ที่ member ที่มีSound ในcast

windows



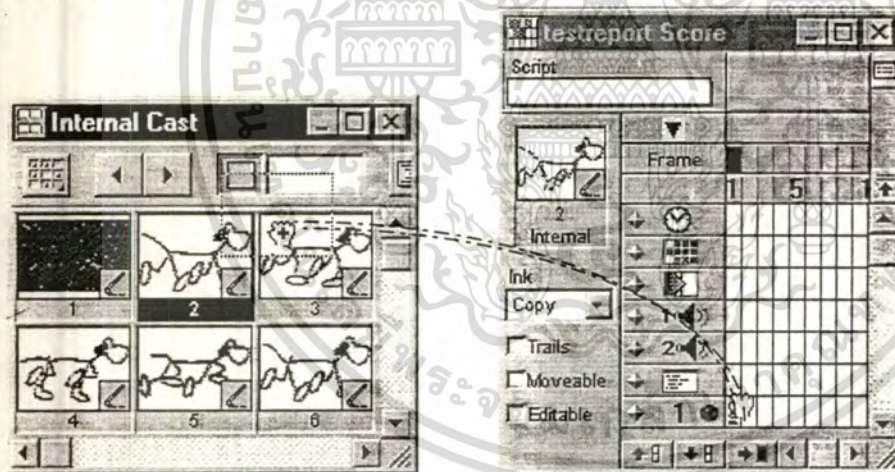
เลขวนloopไปเรื่อยๆ/เล่นเฉพาะที่อยู่Frameนั้น

→ Script-lingo(เป็นภาษาที่ใช้ควบคุม การ interactive) script in each frame

→ 1

→ 2 | Sprite Channel-ลำดับของmember ภาพในcell คือ 1- อยู่ล่างสุดเป็นพื้น 2 อยู่บน1 3อยู่

บน2Drag & Drop ลากจากCast member มาที่ Cell ใน Frame ที่ต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 INK POP UP MENU - รูปการวาง member ภาพ ลงบน frame

Annotations for the Ink popup menu:

- Transparent: วางธรรมดาเห็นพื้นขาวกรอบสี่เหลี่ยม เหมาะกับ Background
- Reverse: วางธรรมดาไม่เห็นพื้นขาวรอบนอก ซ้ำกว่าแบบอื่นเพราะใช้memoryมากกว่า
- Ghost: สีขาวที่รูปจะใสเห็นสี Background
- Not Copy: รูปจะเป็นสีที่โปร่งเห็นพื้นBackground

6.6 EXAMPLE

Example of Ink software interface and frame placement:

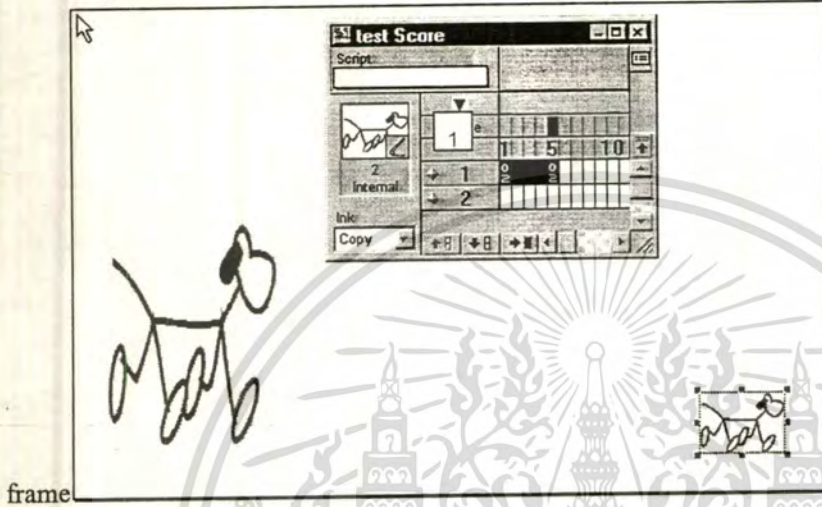
- Internal Cast**: Cast member ที่มีคย (Cast member with key)
- testreport2 Score**: Score ลำดับภาพ (Score sequence)
- Frame 1**: Shows a smiley face (4) and a stick figure (2) on a dark background.
- Frame 2**: Shows two stick figures (6 and 2) on a white background.
- Frame 3**: Shows a smiley face (4) on a dark background.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 Special

In-betweening เป็นการสร้างภาพอัตโนมัติให้เชื่อมต่อเป็นSeries ในภาพเดียวกัน
ระหว่างFrameโดยอาจเปลี่ยนตำแหน่ง มุม ขนาด

วิธี: paste between



5

select In-between (Modify Menu)

ภาพframe 1-5

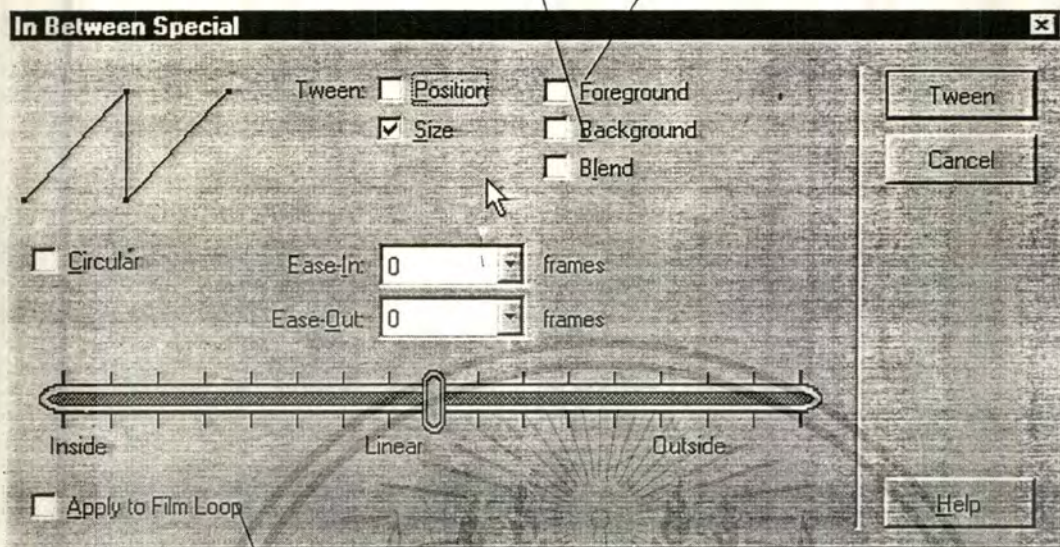


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

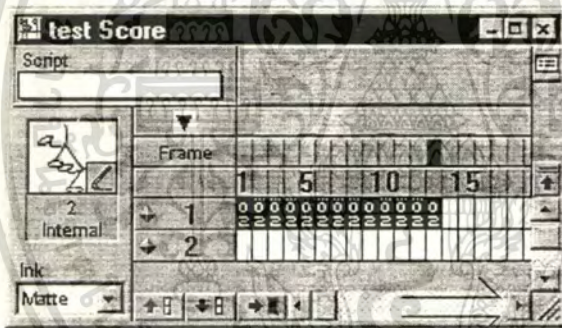
In-between Special สร้างแบบพิเศษโดยเลือก In-between Special

ใช้เมื่อเปลี่ยนขนาด

ใช้เมื่อภาพเปลี่ยนตำแหน่ง

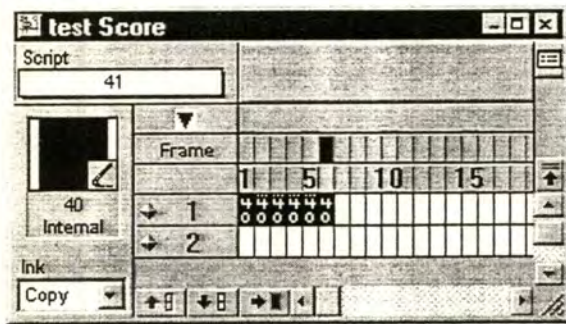


ใช้กับ film loop
**ปกติจะ Deform แต่สามารถเปลี่ยนแปลง(ขนาด,ตำแหน่ง)



** เปลี่ยนตำแหน่งแบบโค้ง โดยเพิ่ม frame ระหว่างต้น - ปลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



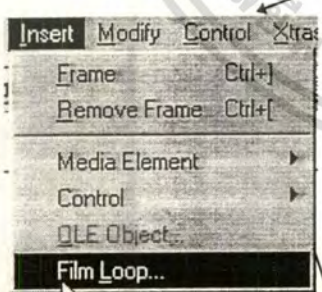
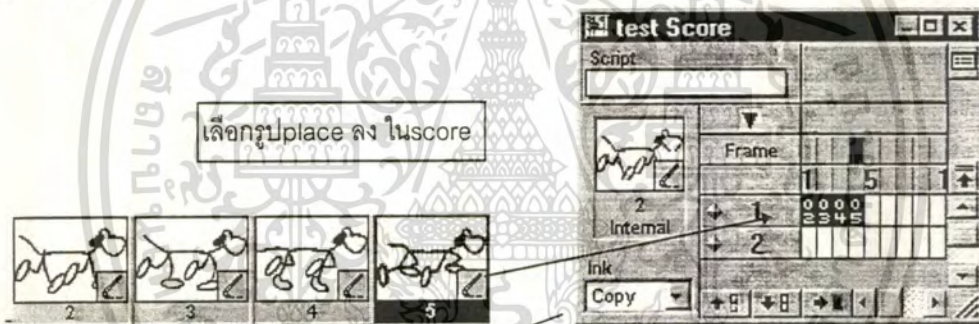
ไล่สีจาง-เข้ม โดยใช้ select

Blend ใน in Between special

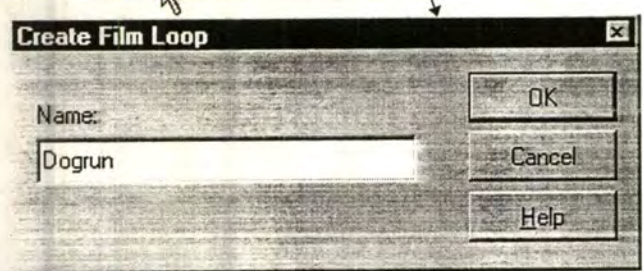
6.8 Film Loop

คือชุดภาพที่มากกว่าภาพที่เปลี่ยนไปซ้ำๆ กัน วน loop

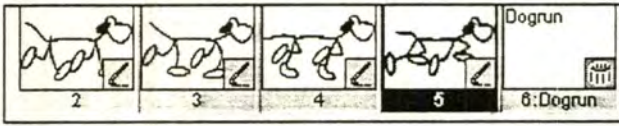
วิธี -place member ที่ต้องการให้อยู่ใน loop เรียงกันไปตามลำดับ บน cell ใน score windows ที่ว่างๆ



Select film loop (insert menu) then set name of film loop



Director creat cast member of Dog run เป็น film loop ที่เหมือน movie เลิกวนภาพซ้ำๆ ตัวอย่างนี้ เราจะได้สุนัขวิ่งอยู่กับที่ แต่ถ้าใช้ร่วมกับIn-between ก็จะได้ animation สุนัขวิ่งที่เคลื่อนที่ได้ ซึ่งใช้เหมือนภาพปกติ แต่select ที่ film loop ใน In between special



6.9 Real time Recording

คือการที่สร้างภาพเคลื่อนไหวที่โดยที่เราสามารถกำหนดเส้นทาง, ความเร็ว ได้จริงๆบน stage ที่ display

วิธี - เลือก cell ที่ต้องการ move

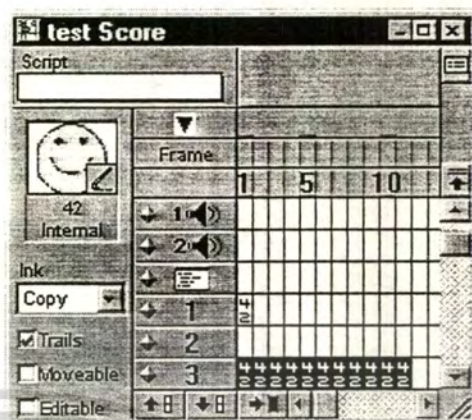
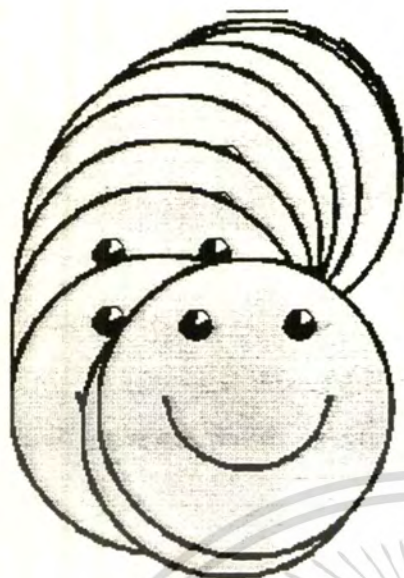


กด ctrl+space bar แล้ว pointer จะไป frame เริ่มต้นใน channel ใหม่

ขณะกด ctrl+spacebar ค้างอยู่เลื่อน mouse ไปที่จุดที่ต้องการเริ่มต้นแล้ว click ภาพจะมาอยู่ mouse ลากภาพไปตามเส้นทาง โดยที่ movie จะ play ไปด้วย เมื่อปล่อย mouse ก็เป็นอันสิ้นสุด Real time recording

Director สร้างภาพใน frame ต่อไปโดยเช็คตาม cursor ที่เลื่อนไป ความละเอียดขึ้นกับการ set Frame per second ถ้ามากภาพจะ smooth ขึ้น แต่ก็ใช้ frame มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ปัญหาที่พบคือ ภาพที่ย่อ/ขยายขนาด เมื่อfirst click ภาพจะมีขนาดเท่าเดิม

TIP:ถ้าselect Trail สามารถทำภาพที่ก่่าตั้งเขียนด้วยมือ hand writing

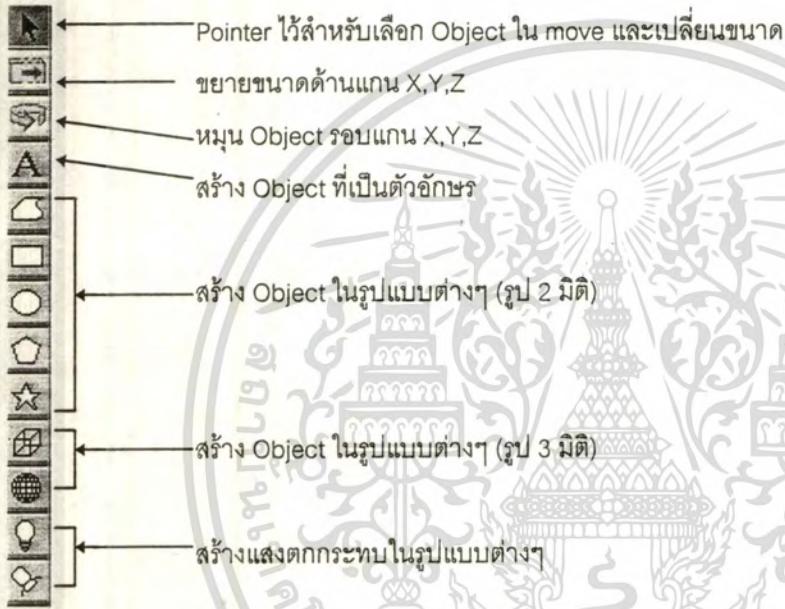


บทที่ 7

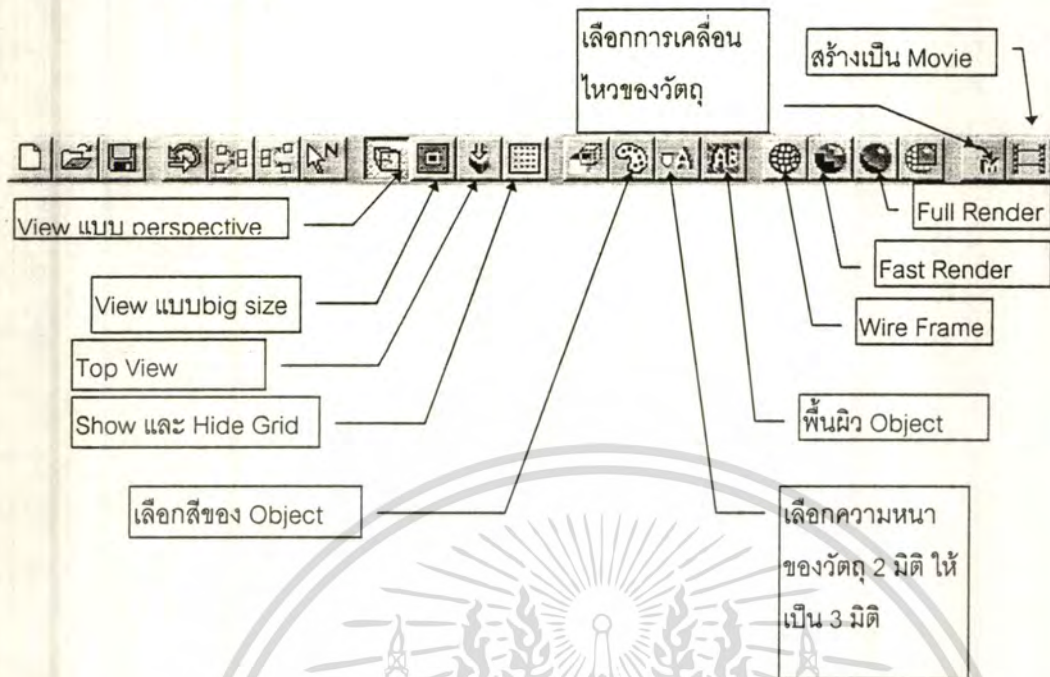
Crystal Flying Fonts

Crystal Flying Fonts เป็นโปรแกรมที่สร้าง Title ที่เคลื่อนไหว 3 มิติ โดยการโหลด Object ที่มีอยู่ หรือจะเป็นการสร้างขึ้นเองง่ายๆ ได้ โดยจะเป็นการเคลื่อนไหวสั้นๆ ไม่ซับซ้อนมากนัก ไฟล์ที่สร้างได้สามารถแปลงเป็น Movie (*.flc และ *.avi)

7.1 ส่วนประกอบในการสร้าง



7.2 Menu ที่สำคัญ

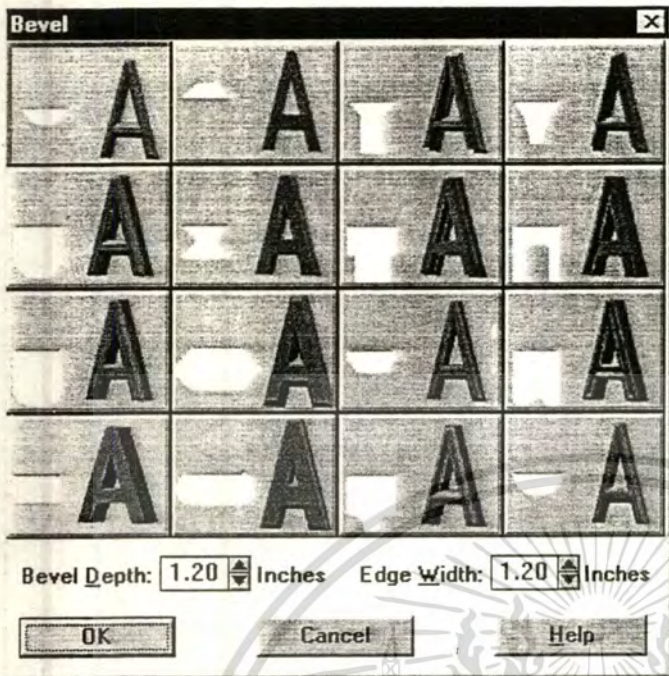


7.3 ตัวอย่าง การ render



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

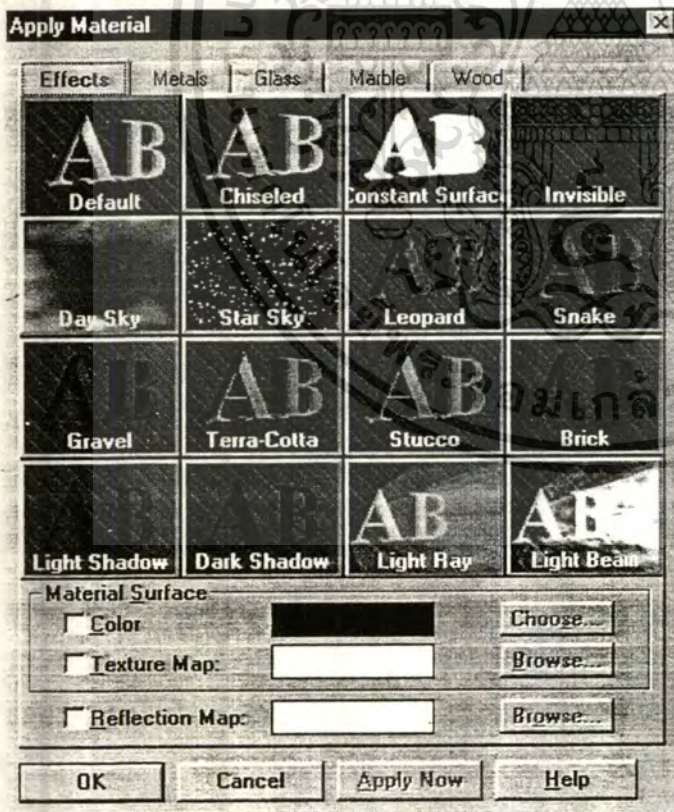
7.4 BEVEL กำหนดความลึกของวัตถุในรูปแบบต่างๆ



มักใช้กับตัวอักษรให้มีความหนาขึ้น

7.5 Apply Material

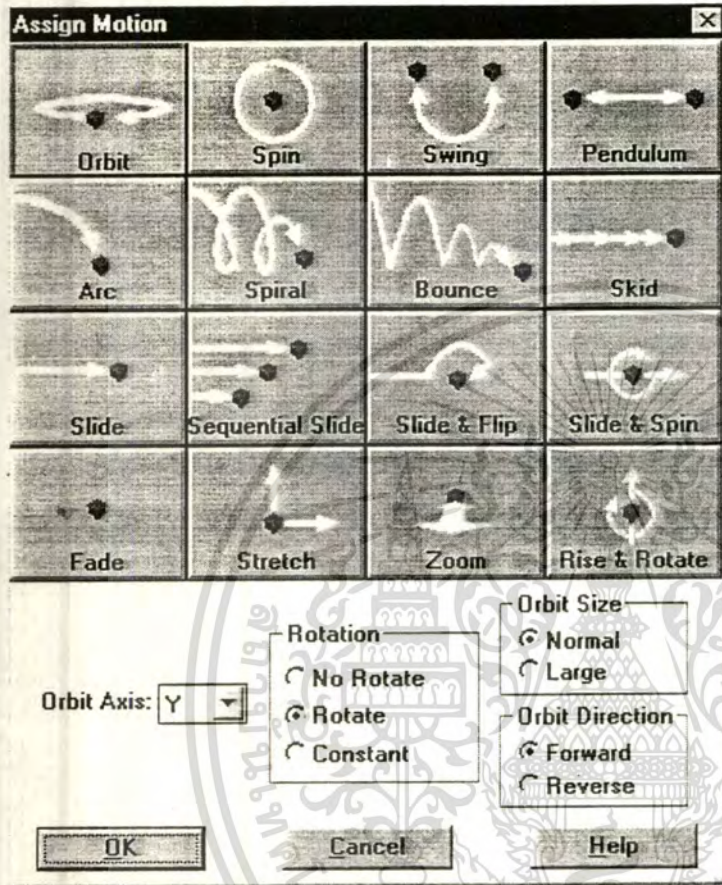
กำหนดชนิดให้แก่ Object มีทั้งแบบเป็น Gold, Chromium, Wood, Marble, Sky, Blue, Glass ฯลฯ



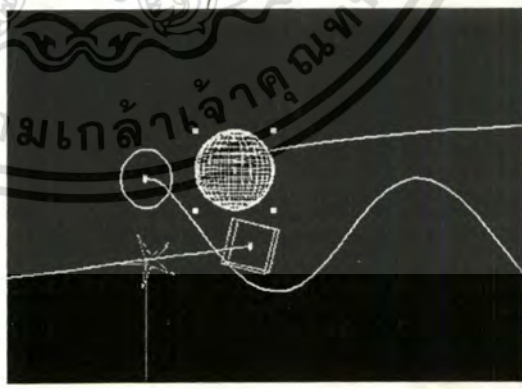
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 Motion

เราสามารถกำหนดการเคลื่อนไหวของวัตถุซึ่งจะกำหนดโดยการจัดวาง Object จะเป็นภาพสุดท้ายตอนจบ ดังนั้นเราจะกำหนดแต่ละ object เคลื่อนไหวได้เพียงแบบใดแบบหนึ่ง ซึ่งอาจไม่เหมือนกันแต่สุดท้ายจะลงเอยเป็นภาพที่เราสร้างขึ้น

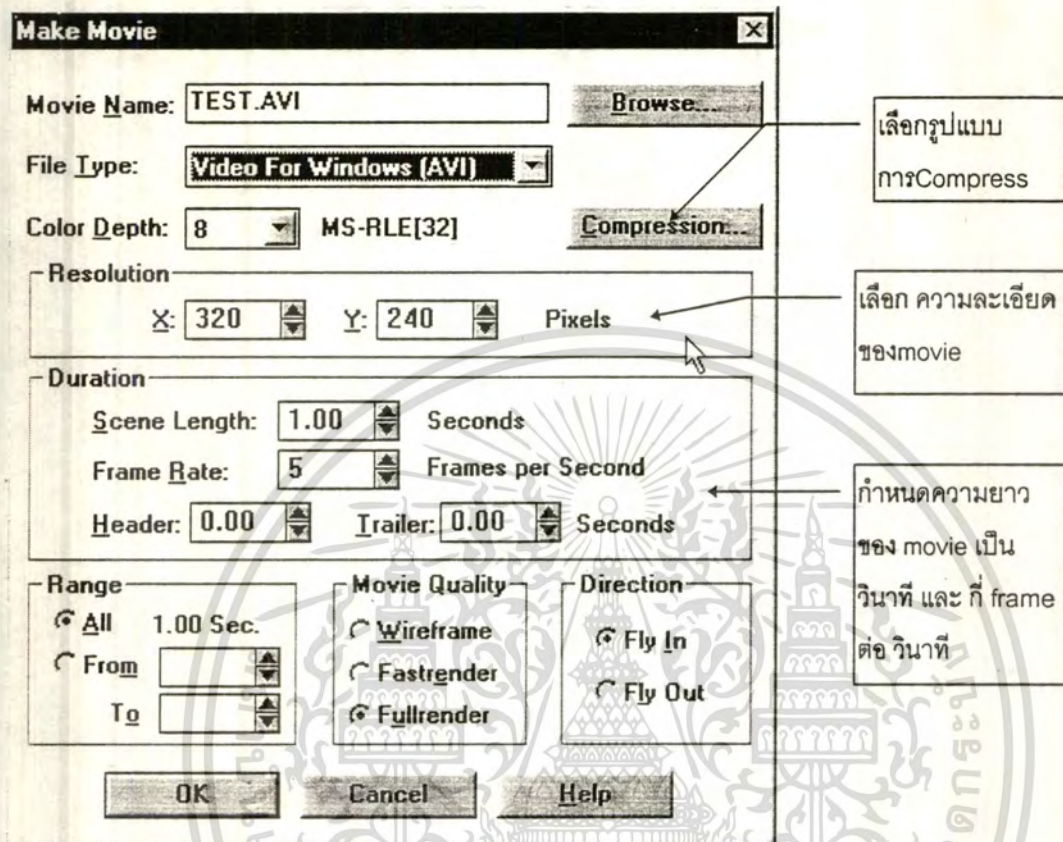


ภาพwire frame แสดงการเคลื่อนไหวมาจนถึงจุดสุดท้าย



7.7 MAKE MOVIE

เมื่อเราทำการ assign motion แก่ object แล้ว ก็ทำการ สร้างเป็น file ที่สามารถเล่นเป็นภาพเคลื่อนไหวได้ เช่น FLC หรือ AVI file



เมื่อเลือกเสร็จแล้ว โปรแกรมก็จะทำการ render ที่ละภาพๆ จนครบ ซึ่งเราสามารถสังเกต ว่าแต่ละเฟรมที่ออกมา นั้นถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การออกแบบ

8.1 หลักในการออกแบบ

จุดมุ่งหมายของโปรแกรมช่วยสอนนี้ก็เพื่อผู้ใช้งานคอมพิวเตอร์สามารถเรียนรู้การสอน การเขียนภาษา HTML โดยวิธีที่ง่ายที่สุด โดยผ่านระบบมัลติมีเดีย ทั้งภาพและเสียง ซึ่งเป็นเสมือน การเรียนโดยตรงในชั้นเรียนจริง เพราะตัวโปรแกรมได้ทำการบันทึกการทำงานของหน้าจอจริง ในขณะนั้น พร้อมกับคำบรรยายในแต่ละขั้นตอนต่างๆ อย่างชัดเจน

8.1.1 การออกแบบตัวโปรแกรม

เมื่อเราสำรวจเนื้อหาทั้งหมดของภาษาที่เราต้องการออกแบบ ในโครงการนี้ คือ ภาษา HTML เราจึงแบ่งการนำเสนอเป็น 3 ส่วนหลักเพื่อให้ง่ายต่อการเรียนรู้ของผู้ใช้เป็นดังนี้คือ

1. ส่วนที่นำเสนอในตอนแรก (Present) เป็นเสมือนหน้าปกของหนังสือที่บอกชื่อเรื่อง และรายละเอียดอย่างคร่าวๆ เช่น ชื่อโปรแกรม ชื่อผู้จัดทำ สถานที่ติดต่อ อาจารย์ที่ปรึกษา โดยในส่วนนี้ต้องมีคุณสมบัติหลักๆ คือ
 - ดึงดูดความสนใจของผู้ใช้ โดยใช้การนำเสนอภาพและเสียงที่เร้าใจ
 - ทำให้ผู้ใช้รู้คร่าวๆถึงสิ่งที่จะนำเสนอ
 - บอกที่มาของโปรแกรม (ผู้จัดทำ)
2. ส่วนที่เป็นสารบัญ (index) เป็นเสมือนสารบัญของหนังสือ ซึ่งถูกแยกบทไว้อย่างชัดเจน โดยสารบัญจะแบ่งเป็น
 - สารบัญหลัก ได้แก่ บทที่ ...
 - สารบัญย่อย ได้แก่ หัวข้อย่อยๆ ในบทนั้นๆที่จะมีตัวเลขบอกเพื่อไปดูรายละเอียดในเนื้อหาต่อไป โดยที่ส่วนนี้จะบอกรายละเอียดของเนื้อหาอย่างคร่าวๆ ด้วย
3. ส่วนที่เป็นเนื้อหาจริง เป็นเสมือนเนื้อหาทั้งหมดของหนังสือที่ถูกเรียงไว้ เป็นบทๆ และเป็นหัวข้อๆ

ซึ่งทั้ง 3 ส่วนหลักจะกล่าวถึงรายละเอียดในการออกแบบด้านหลัง ในการออกแบบของทั้ง 3 ส่วน หลัก จะพยายามใช้รูปแบบการที่ง่าย สำหรับการพัฒนาโปรแกรมขึ้นมาเพื่อที่จะสามารถทำการปรับปรุงและแก้ไขได้อย่างรวดเร็ว โดยอาศัยหลักการของ Object (ซึ่งหลักการของ Object จะทำให้เราสามารถที่จะแยกโปรแกรมแต่ละส่วนเป็นอิสระออกจากกัน แต่แต่ละ Object ก็ยังมีความสัมพันธ์กัน)

8.1.2 กำหนดคุณสมบัติของโปรแกรม

หลังจากที่เราได้คิดออกแบบหลักการทำงานของโปรแกรมแล้ว เราก็หันมาคิดว่าโปรแกรมช่วยสอน การเขียน HTML จะต้องมีความสามารถอะไรบ้าง ซึ่งเราได้วางความสามารถของโปรแกรมไว้ดังนี้คือ

1. สามารถแสดงภาพและเสียง ขณะทำการสอนได้
2. สามารถแสดงการทำงานใน Option ต่าง ๆ ได้โดยคลิกเพียงปุ่มเดียวเท่านั้น
3. สามารถอธิบายรายละเอียด ของแต่ละบทได้อย่างชัดเจน
4. สามารถเรียกใช้ ไฟล์ Execute อื่น ๆ ที่มีอยู่ใน Window ได้ รวมทั้ง ไฟล์ Execute ที่เขียนขึ้นเองด้วย
5. ใช้หน่วยความจำน้อยที่สุดแต่มีคุณภาพมากที่สุด

8.1.3 เลือกโปรแกรมที่จะนำมาใช้เขียน

จากคุณสมบัติที่กล่าวมาแล้วข้างต้นเราสามารถเลือก โปรแกรมที่จะนำมาใช้ดังนี้คือ

ชื่อ โปรแกรมที่ใช้	ส่วนที่ใช้	คุณสมบัติ	ข้อจำกัด
Crystal Flying Fonts	ส่วนนำเสนอ	1.สามารถสร้างภาพ 3 มิติที่เคลื่อนไหวได้ 2.มี object จำนวนมาก ให้เลือก 3.ใช้ง่าย โดยสามารถเลือกกำหนดทิศทาง การเคลื่อนไหวจาก โปรแกรมที่ให้มาอยู่แล้ว ไม่ต้องสร้าง เฟรมต่อเฟรม 4.สามารถกำหนดทิศทางและความเข้มแสงได้ โดยง่าย	1.ไม่สามารถสร้างเสียงประกอบได้ 2.สำหรับการใช้ภาษาไทยที่มีวรรณยุกต์จะมีการเว้นช่องว่างไว้ เช่น คำว่า “ วันว่าง ” เมื่อเขียนในโปรแกรมจะออกมา → วั น ว ำ ง 3.ไม่สามารถทำเป็น Execute file ได้
Macromedia Director 5	ส่วนนำเสนอ	1.สามารถเพิ่มเสียงประกอบได้ 2.ทำเป็น execute file ได้ 3.สามารถload file .AVI หลาย ไฟล์ มาต่อกันเป็น	1.ไม่สามารถสร้างสร้างภาพ 3 มิติเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		เรื่องเดียวได้ 4.กำหนดเวลาหยุดคอย ระหว่างเฟรมได้ เพื่อทำ ให้ภาพกับเสียงเข้ากันได้	
Delphi 2.0	ทั้งโปรแกรม	1.เขียนและความเข้าใจ ง่าย 2.เป็นที่นิยมอย่างแพร่ หลาย	
Lotus Screen cam 97	เนื้อหา	1.สามารถบันทึกการ เคลื่อนไหวต่าง ๆ แบบ real time บนหน้าจอ พร้อมเสียงประกอบ 2.มี screen cam player ซึ่งเป็น execute file และสามารถเรียกใช้ โดยDelphiได้ 3.สามารถหยุดพักใน ระหว่างอัดได้	1.ใช้เนื้อที่มากในการเก็บ 2.ในขณะที่อัดไม่ สามารถอัดเสียงจากไฟล์ เสียง นอกจากจะ อินพุต เข้ามาจากข้างนอก เช่น ไมค์ คีย์บอร์ด

8.2 ขั้นตอนในการพัฒนา

ในการพัฒนาเราเรียกใช้ delphi 2.0 เป็น โปรแกรมหลักในการพัฒนา และ โปรแกรมอื่น ๆ เป็นโปรแกรมสนับสนุนที่ Delphi สามารถเรียกใช้ได้โดยแบ่งเป็น 3 ส่วนดังนี้

8.2.1 ส่วนนำเสนอ

เป็นส่วนที่แสดงถึงชื่อโครงการ ผู้จัดทำ และอาจารย์ที่ปรึกษา โดยการพัฒนานั้นเราจะเขียนโปรแกรมโดยใช้ Delphi 2.0 ในการเรียกโปรแกรม Excecute ของ ไฟล์ที่แสดงภาพเป็นภาพเคลื่อนไหว ซึ่งได้มาจากการเขียนโปรแกรมโดยใช้โปรแกรม (Crystal Flying Font) ในการสร้างภาพ 3 มิติ และไฟล์ .AVI ที่ได้มาประกอบกันเป็น ไฟล์ Excecute ที่มีเสียง โดยการใช้โปรแกรม Director 5 ฉะนั้นจะได้ส่วนนำเสนอ ออกมาราวๆ ดังนี้

- ใช้Crystal flying fonts ทำ avi file

เอกสารนี้ ● ใช้ Director 5 ในการใส่เสียงและ รวม avi file ต่างๆ มาเป็นเรื่องเดียวกัน ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

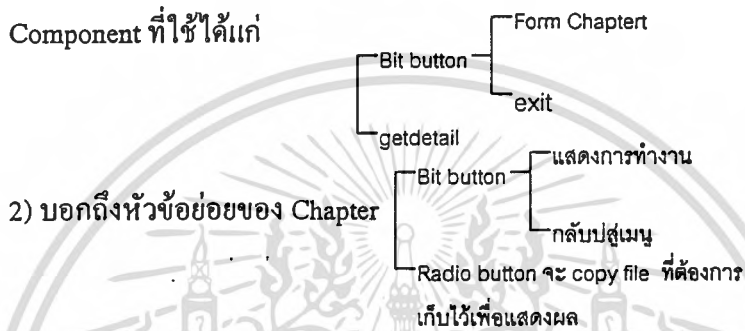
- เขียน Delphi เรียกใช้ ฟังก์ชัน Winexec ในการเรียกใช้ ไฟล์ exe

8.2.2 ส่วนสารบัญ

เป็นส่วนที่ต้องมีการจัดเรียง Component ต่าง ๆ เป็นอย่างดี ทำให้ง่ายต่อการใช้ และเป็นส่วนที่ link ระหว่าง คิวบทและเนื้อหาโดยส่วนนี้เราเขียนโดยใช้ Function ต่าง ๆ ใน Delphi และมีการเขียนโปรแกรมเพิ่มในบางจุด เพื่อทำการรับค่า input ของบทนี้แล้ว ในการเข้าสู่เนื้อหา วางแผนการสร้างโดย สร้าง form 2 form คือ

- 1) Form index บอกถึงรายละเอียดโดยการ Click และ เลื่อน mouse

Component ที่ใช้ได้แก่



- 2) บอกถึงหัวข้อย่อของ Chapter

8.2.3 ส่วนเนื้อหา

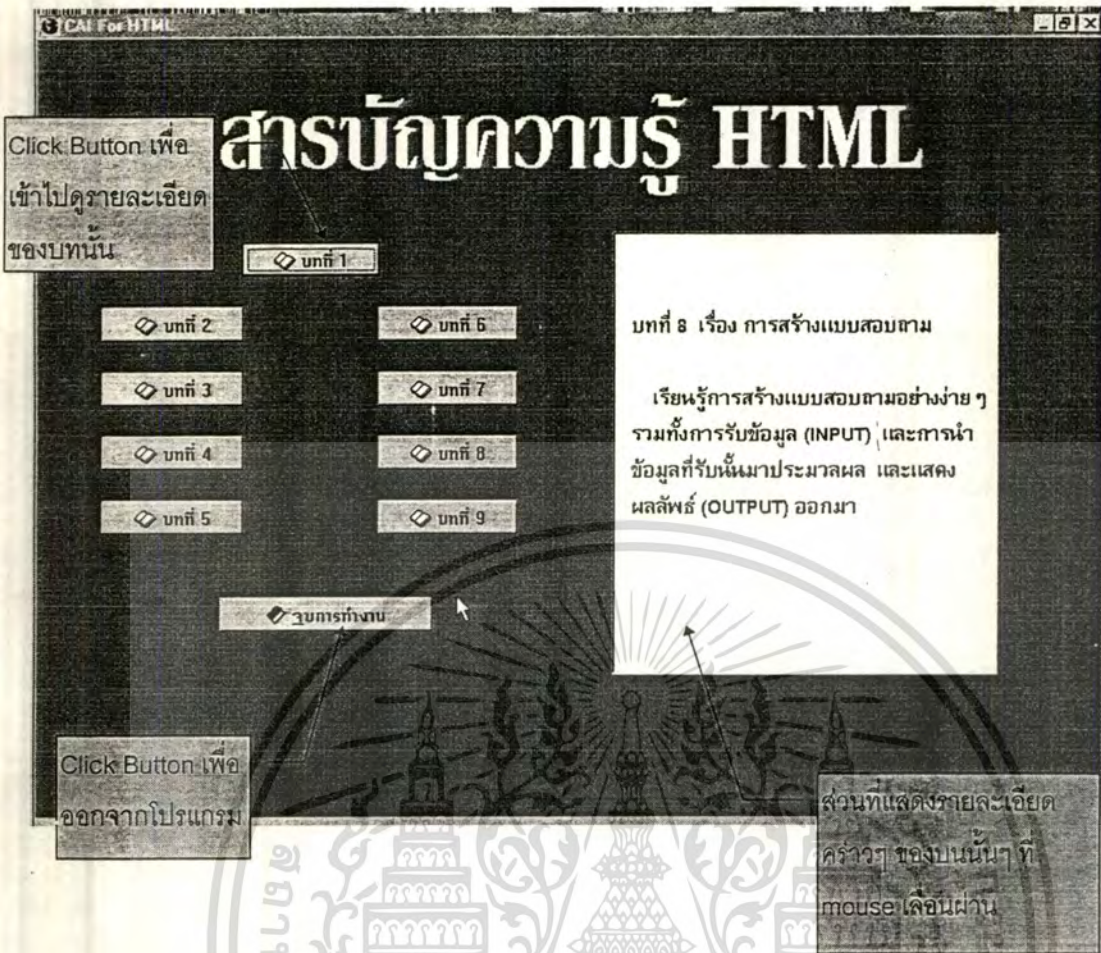
เป็นส่วนที่นำเสนอเนื้อหาเป็นรูปและคำบรรยายในแต่ละขั้นตอนอย่างชัดเจน โดยในที่นี้เราจะใช้โปรแกรม Screen Cam 97 เป็นหลักในการพัฒนาโดยจะเขียน Delphi เพื่อทำการเรียก Screen Cam Prayer. Exe เข้ามาใช้ และนำเสนอความบทที่ได้อินพุทเข้ามานั้น

8.3 อุปกรณ์และระบบที่ต้องการ

1. เครื่องคอมพิวเตอร์ สูงกว่า 486 DX2-66 หน่วยความจำอย่างต่ำ 8 Mb
2. การ์ดเสียง
3. ลำโพงและไมค์
4. Windows 95

8.4 โครงสร้างโดยรวมของโปรแกรม

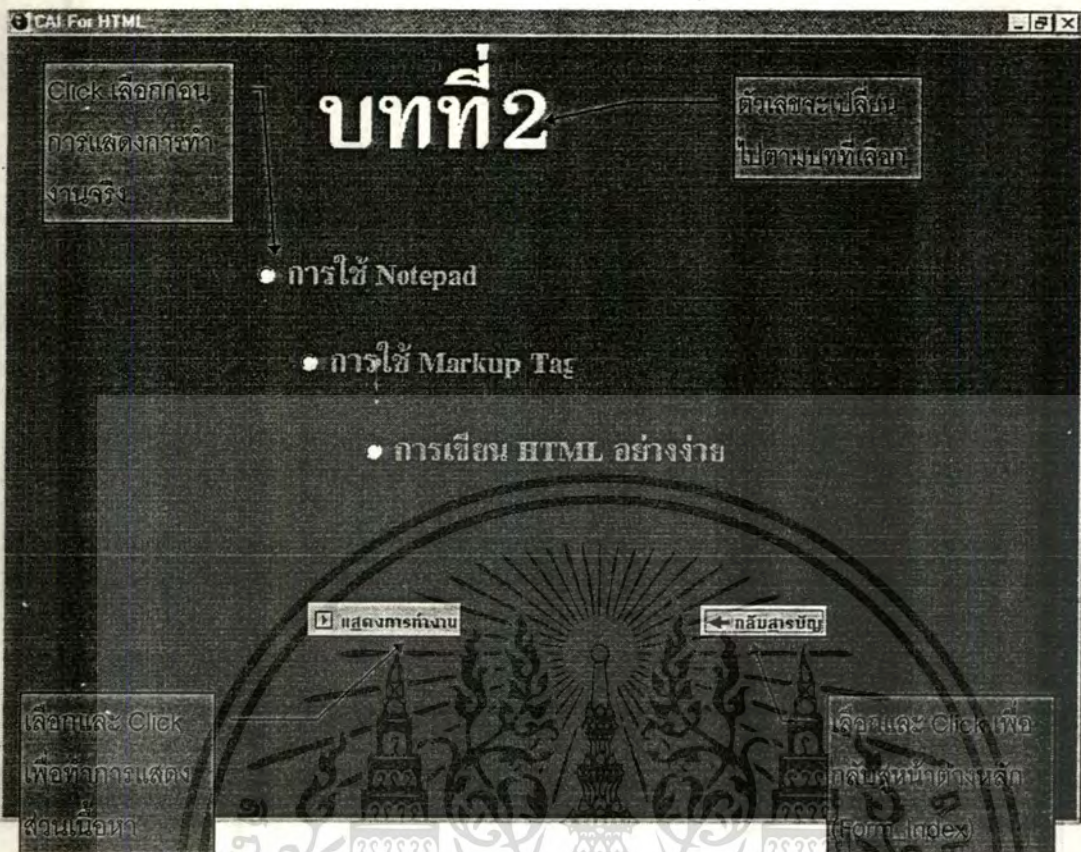
เมื่อเริ่มเปิดโปรแกรม จะปรากฏที่ส่วนนำเสนอ เป็นภาพเคลื่อนไหว หลังจากนั้นจะเข้าสู่ หน้าต่างสารบัญซึ่งหน้าต่างนั้นจะประกอบด้วย บทต่าง ๆ จำนวน 9 บท



รูปแสดงหน้าต่าง Form_Index

และเมื่อเลื่อนเมาส์ไปที่ bottom ของบทใด จะแสดง โครงร่าง ของเนื้อหาในบทนั้น ๆ อย่างคร่าว ๆ ที่ช่อง panel ทางขวามือและเมื่อคลิกที่ bottom บทใดก็จะแสดงรายละเอียดของเนื้อหาย่อใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงหน้าต่างของ Form_Chapter

เมื่อเลือกเนื้อหาข้อย่อยและคลิกที่ปุ่มแสดงการทำงานก็จะแสดงรายละเอียดของหัวข้อนั้น ๆ โดยเรียกใช้ Screen Cam player ซึ่งจะขึ้น Control ของ Screen Cam player เพื่อที่จะสามารถสั่งหยุดหรือเล่นซ้ำได้ โดยเมื่อ load screen cam player ไฟล์ที่ต้องนำเสนอจะถูกรับไว้จนกว่าจะมีการ load ไฟล์ใหม่เข้ามา

เมื่อแสดงเสร็จแล้ว จะกลับเข้าสู่หน้าต่าง Index โดยอัตโนมัติ เพื่อทำการเรียกใช้ต่อหรือ จะออกจากโปรแกรม

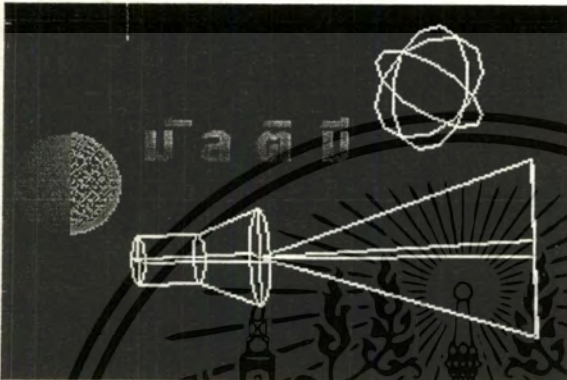
บทที่ 9

การทดลอง

ส่วนนำเสนอ

ขั้นตอน

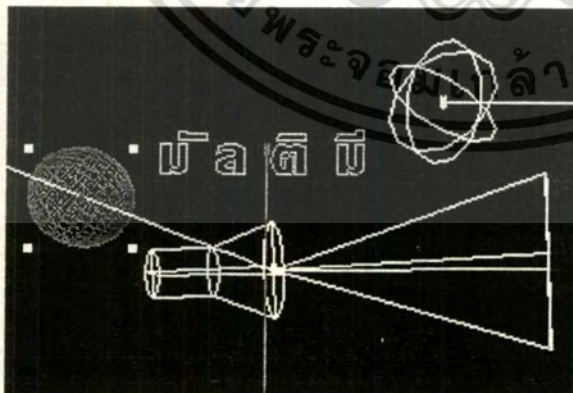
- 1) ใช้ โปรแกรม Crystal Flying font โหลด object ที่โปรแกรมมีให้แล้วมา
- 2) เลือกจัดไฟให้แสงเหมาะสม



- 3) เลือกชนิดพื้นผิว ของ object



- 4) Assign motion ของ ทุก object



5) Make movie โดยเลือกที่ 2 วินาที 15 เฟรมต่อวินาที

Resolution ที่ 800 x 600 เป็น Avi file

Make Movie

Movie Name: PRES1.AVI **Browse...**

File Type: Video For Windows (AVI)

Color Depth: 8 MS-RLE[32] **Compression...**

Resolution
X: 800 Y: 600 Pixels

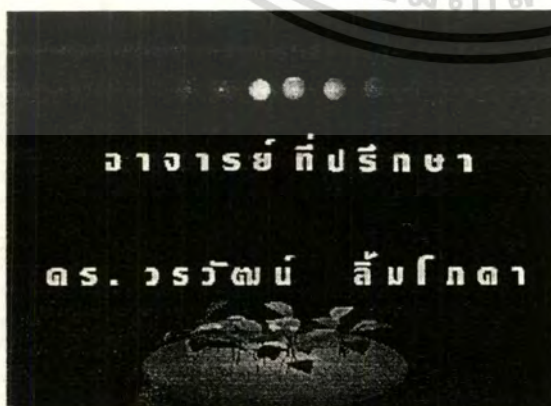
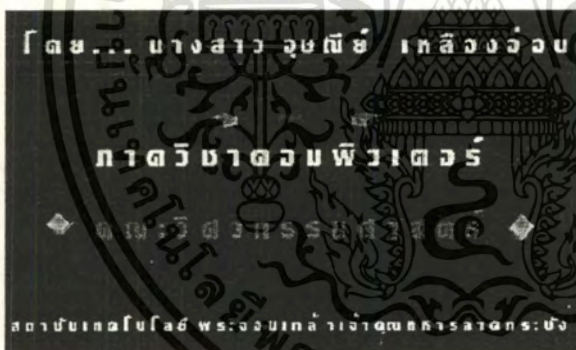
Duration
Scene Length: 2.00 Seconds
Frame Rate: 15 Frames per Second
Header: 0.00 Trailer: 0.00 Seconds

Range
 All 2.00 Sec.
 From
 To

Movie Quality
 Wireframe
 Fastrender
 Fullrender

Direction
 Fly In
 Fly Out

ทำอย่างนี้อีก 2 ตอน ก็ได้ Avi file 3 file ที่ยังแยกกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ใช้ Director 5

- load file avi ทั้ง 3 เข้ามา
- นำมาเรียงต่อกันใน score ของโปรแกรม โดยแบ่งเป็น 3 ช่วง
- ใส่เสียงโดยโหลดไฟล์เสียงมา 2 ไฟล์ โดยให้ ไฟล์ music มาก่อน แล้วพอถึงช่วงสุดท้ายก็ให้มีเสียงกราวแทรกขึ้นมา โดย การถ่วงเวลาในระหว่างช่วงให้เพียงที่สามารถดูรายละเอียดของสิ่งที่นำเสนอได้ทัน
- สร้าง execute file ได้ แต่ไม่ได้รวมเอา avi file และ wave file มาเป็นไฟล์เดียวกันทั้งหมด

ปัญหาที่เกิดขึ้นและวิธีแก้ไข

ปัญหา	วิธีแก้ไข
เมื่อจะเขียนข้อความลงใน crystal โปรแกรมจะเกิดerror จะปิด โปรแกรมทันที	ใช้ข้อความจาก Director แต่จะไม่สวยเท่า Crystal ลงโปรแกรมCrystalใหม่
ภาพที่ออกมาดูไม่ลื่นจะกระตุกเมื่อเล่นบนโปรแกรมจริงๆ ไม่เหมือนบน director	-
Director เล่นเร็วเกินไป	ใช้การถ่วงเวลาระหว่างเฟรมช่วย
Avi บน Director เป็นภาพนิ่ง	กำหนด option ใหม่โดยให้มีการ play

ปัญหาที่เกิดขึ้นและแนวคิดที่ได้จากการทำการทดลองในส่วนนี้

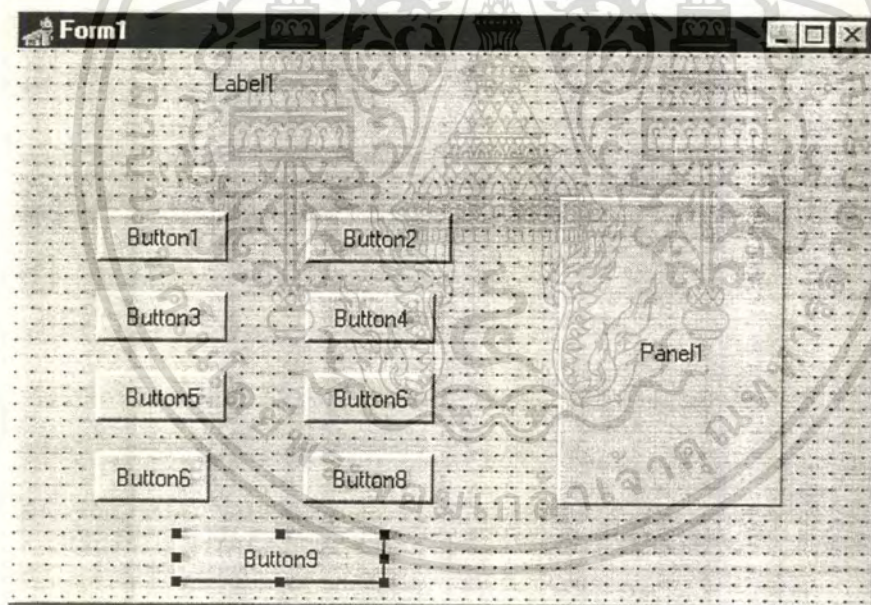
1. ปัญหาเรื่องการใช้ตัวอักษร ใน โปรแกรม Crystal Flying Fonts จะมีปัญหา 2 ส่วน คือ
 - 1.1 (พบบ่อยเสมอ) ปัญหาที่ใช้ Font ไม่ได้ ทั้งภาษาไทยและภาษาอังกฤษ เมื่อคลิกที่ Fonts จะทำให้แจ้ง Error ขึ้น จบต้อง Boot คอมพิวเตอร์ใหม่ ทางแก้ คือ การพยายามเช็คฮาร์ดดิสต์ให้พร้อมก่อนการลงโปรแกรม และควรอ่านคำสั่งระหว่างการลง โปรแกรมให้ ถ้ายังใช้ Font ไม่ได้ก็ต้องพยายามลงใหม่อีกครั้ง
 - 1.2 การที่ไม่ Support Font ภาษาไทย เช่น การที่พิมพ์คำว่า “วันว่าง” จะปรากฏที่หน้าจอเป็น “วันว่าง” ทางแก้ คือ การใช้โปรแกรมอื่นช่วยในการในการสร้าง font เช่น จาก โครงการงานครั้งนี้ ใช้ โปรแกรม Print Shop Delux และ import เข้ามา ซึ่ง Crystal Flyings Font จะมองเป็นเหมือน Object หนึ่ง

2. การจัดเก็บข้อมูล หลังจากทำการแปลงข้อมูลให้เป็น .exe แล้ว ก็ยังต้องเก็บข้อมูลคิบบ File.avi อยู่ เพราะโปรแกรมที่ทำการแปลงข้อมูลเป็น .avi แล้วต้องทำการเรียกใช้จากการทดลองที่ผ่านมา เมื่อทำการลบ ข้อมูลคิบบปรากฏว่า เมื่อเรียก ไฟล์.exe แล้วไม่สามารถเรียกใช้ได้จะแจ้ง error ว่า หาข้อมูลไม่พบ ซึ่งในโปรแกรมนี้ไฟล์ที่ต้อง Save คือ Press1.avi, Press2.avi, Press3.avi
3. เมื่อทำการทดลองการจัดเก็บข้อมูล จะพบว่า การเก็บเป็น .exe จะใช้เนื้อที่ประมาณ 5.7 MB และ .avi จะใช้เนื้อที่ประมาณ

ส่วนสารบัญ

ใช้การพัฒนาหลัก โดยการเขียน โดยใช้ Delphi 2.01 ซึ่งเขียนบนฟอร์ม 2 ฟอร์ม คือ form_index และ form_chapter ซึ่งการเขียน โปรแกรมมี Source Program ดังนี้ คือ

1. กำหนด Form และกำหนดคอม โนเนทต่างๆ ในโปรแกรม ดังรูป



รูปแสดงการกำหนดคอม โปเนทต่างๆ ในฟอร์ม Index

โปรแกรมสำหรับการเช็ค บทที่เพื่อแสดงรายละเอียดใน panel1 อยู่ในโปรแกรมภาคผนวก โดยมีอัลกอริทึม คือ

1. ใช้ฟังก์ชัน MouseMove ให้ส่งค่าที่ได้ในแต่ละ Button โดยส่งเป็นค่าแฟล็ก เริ่มต้นที่เลข 1-9 โดยกำหนดค่า Ground เป็น 0
2. นำค่า Flag ไปเปรียบเทียบ และดึงเอาข้อมูลจาก ไฟล์ detail.txt ที่เขียนไว้ใน editor โดยกำหนดแต่ละบท เป็นแต่ละ Topic เช่น [บทที่ 2

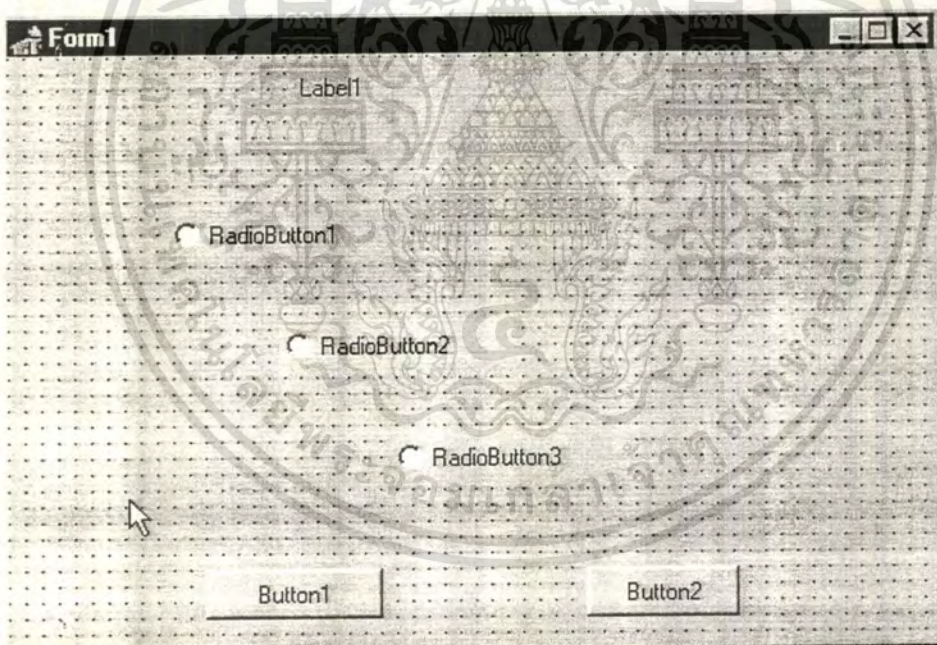
[2]

บทที่ 2 เรื่อง เริ่มต้นกับภาษา HTML

ทำความเข้าใจเกี่ยวกับการเริ่มต้นการเขียนภาษา HTML โดยทั่วไป เช่น การใช้ Notepad , Markup Tag จนสามารถเขียน โปรแกรมภาษา HTML อย่างง่ายได้

&

3. ใส่ฟังก์ชัน onclick เพื่อเชื่อมต่อกับฟอร์ม Chapter
4. กำหนดคอมโปเนนต์ต่างๆ ใน Form_Chapter เป็นดังนี้ คือ



รูปแสดงการกำหนดคอมโปเนนต์ต่างๆ ในฟอร์ม Chapter

5. ในปุ่ม Radio Button เมื่อคลิกจะเข้าไปรันใน Lotus ScreenCam โดยใช้คำสั่งฟังก์ชัน Winexec เพื่อเรียก ScreenCam Player.exe และจะทำการLoad ข้อมูลตามที่ต้องการนั้นๆ

ปัญหาที่เกิดขึ้นกับไฟล์ .scm

1. การอัดเสียงยังไม่มีคุณภาพเท่าที่ควร เพราะอุปกรณ์ต่างๆไม่ได้คุณภาพ ขาดการควบคุมเสียงที่ดี
2. ScreenCam ไม่สามารถอัดไฟล์เสียงผ่านทางคอมพิวเตอร์ได้ ต้องทำการนำ Input เข้ามาจากภายนอก จากการทดลองเราใช้ อิเล็กทรอนิกส์ต่อเข้ากับ port หนึ่ง เพื่อใช้เป็นอินพุทของคอมพิวเตอร์
3. ScreenCam ไม่สามารถอัดการเคลื่อนไหวของรูปใหญ่ๆ (เต็มหน้าจอ) ที่มีการเปลี่ยนแปลงอย่างรวดเร็วไม่ได้ เช่น การอัดภาพเคลื่อนไหว AVI

ปัญหาที่เกิดขึ้น เมื่อทำการอัดลง CD

1. ในคอนนำเสนอ ข้อมูลมีขนาดใหญ่ ทำให้เมื่อรันจริงเสียเวลา และประมวลผลข้อมูลไม่ทัน เป็นเหตุให้ข้ามบางตอนไปอย่างน่าเสียดาย
2. ก่อนอัดลง CD ต้องทำการเช็คว่ โปรแกรมส่วนใดที่ใช้ c: ต้องแก้ไข เป็น d: แทน

บทที่ 10

สรุปและวิจารณ์

จากการพัฒนาโปรแกรมมัลติมีเดียช่วยสอน การเขียน HTML นี้ทำให้ได้ข้อสรุปว่าเราสามารถที่จะนำเสนอข้อมูลที่เป็นที่น่าสนใจมากกว่าการเรียนการสอนโดยทั่วไป และมากกว่าการอ่านหนังสือ ถึงแม้จะมีข้อบกพร่องบ้างในเรื่องการอัดเสียง แต่ก็ยังถือว่าเป็นปัญหาใหญ่นัก

และจะเห็นได้ว่าจากพื้นฐานความรู้ตรงนี้ จะทำให้เราสามารถรวบรวมเอา ไฟล์ Excecute ต่างๆ มาแสดงผล บนโปรแกรมนี้ได้ เป็นสาเหตุให้ ง่ายและสะดวกในการ Present และไม่ต้องเสียเวลามากมายในการปิด - เปิดโปรแกรมเพื่อ load program ใหม่แต่สิ่งที่จะต้องคิดต่อไปคือ

1. การบีบข้อมูล เพื่อที่จะทำให้ขนาดของโปรแกรมเล็กลงอันจะเป็นประโยชน์ในการ Run Program ได้เร็วขึ้น
2. การเขียนบททดสอบเพื่อจะเป็นการตอบสนองผู้ใช้โปรแกรมว่ามีความเข้าใจดีเพียงใด
3. เพิ่มรูปแบบในการนำเสนอมากยิ่งขึ้น ในแต่ละบท
4. การนำโปรแกรมนี้เข้าไปใช้ใน Internet เพื่อสามารถติดต่อกับครูผู้สอนได้โดยตรง

ในกรณีมีเรื่องที่ไม่เข้าใจและไม่สามารถหาคำตอบในโปรแกรมได้
สุดท้ายนี้ หวังว่าโปรแกรมมัลติมีเดียช่วยสอน การเขียน HTML นี้จะเป็นแนวทางในการพัฒนาผู้อื่นให้สามารถสร้าง Application เกี่ยวกับการเรียนการสอน ในวิชาอื่น ๆ ได้เป็นอย่างดี ตลอดจนหวังว่าในอนาคตอันใกล้นี้จะสามารถพัฒนาเป็นเหมือนครูคอมพิวเตอร์ตัวจริงในวงการคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOURCE PROGRAMME

unit Index;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, ExtCtrls, StdCtrls, Buttons, Chapter, MPlayer;

type

TForm_Index = class(TForm)

BT_ch1: TBitBtn;

BT_ch2: TBitBtn;

BT_ch3: TBitBtn;

BT_ch4: TBitBtn;

BT_ch5: TBitBtn;

BT_ch6: TBitBtn;

BT_ch7: TBitBtn;

BT_ch8: TBitBtn;

BT_ch9: TBitBtn;

BT_Exit: TBitBtn;

Memo_detail: TMemo;

Label2: TLabel;

BT_Show: TBitBtn;

procedure BT_ExitClick(Sender: TObject);

procedure BT_ch1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure BT_ch1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);

procedure BT_ch2MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);

procedure BT_ch3MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure BT_ch4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch9MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure BT_ch2Click(Sender: TObject);
procedure BT_ch3Click(Sender: TObject);
procedure BT_ch4Click(Sender: TObject);
procedure BT_ch5Click(Sender: TObject);
procedure BT_ch6Click(Sender: TObject);
procedure BT_ch7Click(Sender: TObject);
procedure BT_ch8Click(Sender: TObject);
procedure BT_ch9Click(Sender: TObject);
procedure FormActivate(Sender: TObject);

```

```
private
```

```
  G_Flag : string;
```

```
  { Private declarations }
```

```
public
```

```
  dir : string;
```

```
  { Public declarations }
```

```
  procedure showdetail(ch : integer ;flag : string );
```

```
end;
```

var
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form_Index: TForm_Index;

implementation

{SR *.DFM}

Procedure TForm_index.ShowDetail(ch : integer ;Flag : string);

function getdetail(topic : string):string;

var myfile : textfile; {

data1,data2,nl : string;

begin

data1:='';

data2:='';

nl := chr(13)+chr(10);

assignfile(myfile,dir+'text\detail.txt');

reset(myfile);

readln(myfile,data1);

while data1 <> '['+topic+']' do

 readln(myfile,data1);

 while data1 <> '&' do

 begin

 readln(myfile,data1);

 data2 := data2 + data1 + nl;

 end;

 getdetail := copy(data2,1,length(data2)-3);

 closefile(myfile);

end;

begin

 if G_Flag <> flag then

 begin

 case ch of

 1 : begin

 memo_detail.text := getdetail('1');

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Flag := '1';
end;
2 : begin
    memo_detail.text := getdetail('2');
    Flag := '2';
end;
3 : begin
    memo_detail.text := getdetail('3');
    Flag := '3';
end;
4 : begin
    memo_detail.text := getdetail('4');
    Flag := '4';
end;
5 : begin
    memo_detail.text := getdetail('5');
    Flag := '5';
end;
6 : begin
    memo_detail.text := getdetail('6');
    Flag := '6';
end;
7 : begin
    memo_detail.text := getdetail('7');
    Flag := '7';
end;
8 : begin
    memo_detail.text := getdetail('8');
    Flag := '8';
end;
9 : begin
    memo_detail.text := getdetail('9');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Flag := '9';
    end;
end; { case }
end;
end;

procedure Create_ch(ch : integer);
begin
    Form_Chapter.Notebook.pageindex := ch-1;
    Form_Chapter.Show;
end;

procedure TForm_Index.BT_ExitClick(Sender: TObject);
begin
    Close;
end;

procedure TForm_Index.BT_ch1Click(Sender: TObject);
begin
    Create_ch(1);
    formt_chapter.labell.caption := 'บทที่1';
end;

procedure TForm_Index.FormCreate(Sender: TObject);
begin
    Winexec('\caihtml\chapter\demoproj.exe',sw_show);
    g_flag := "";
    Dir := paramstr(1) + '\caihtml';
end;

procedure TForm_Index.BT_ch1MouseMove(Sender: TObject; Shift: TShiftState;
X, Y: Integer);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

begin

showdetail(1,'1');

end;

procedure TForm_Index.BT_ch2MouseMove(Sender: TObject; Shift: TShiftState;

X, Y: Integer);

begin

showdetail(2,'2');

end;

procedure TForm_Index.BT_ch3MouseMove(Sender: TObject; Shift: TShiftState;

X, Y: Integer);

begin

showdetail(3,'3');

end;

procedure TForm_Index.BT_ch4MouseMove(Sender: TObject; Shift: TShiftState;

X, Y: Integer);

begin

showdetail(4,'4');

end;

procedure TForm_Index.BT_ch5MouseMove(Sender: TObject; Shift: TShiftState;

X, Y: Integer);

begin

showdetail(5,'5');

end;

procedure TForm_Index.BT_ch6MouseMove(Sender: TObject; Shift: TShiftState;

X, Y: Integer);

begin

showdetail(6,'6');

end;

```
procedure TForm_Index.BT_ch7MouseMove(Sender: TObject; Shift: TShiftState;  
  X, Y: Integer);
```

```
begin
```

```
  showdetail(7,'7');
```

```
end;
```

```
procedure TForm_Index.BT_ch8MouseMove(Sender: TObject; Shift: TShiftState;  
  X, Y: Integer);
```

```
begin
```

```
  showdetail(8,'8');
```

```
end;
```

```
procedure TForm_Index.BT_ch9MouseMove(Sender: TObject; Shift: TShiftState;  
  X, Y: Integer);
```

```
begin
```

```
  showdetail(9,'9');
```

```
end;
```

```
procedure TForm_Index.BT_ch2Click(Sender: TObject);
```

```
begin
```

```
  Create_ch(2);
```

```
  form_chapter.label1.caption := 'บทที่2';
```

```
end;
```

```
procedure TForm_Index.BT_ch3Click(Sender: TObject);
```

```
begin
```

```
  Create_ch(3);
```

```
  form_chapter.label1.caption := 'บทที่3';
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm_Index.BT_ch4Click(Sender: TObject);
begin
    Create_ch(4);
    form_chapter.label1.caption := 'บทที่4';
end;
```

```
procedure TForm_Index.BT_ch5Click(Sender: TObject);
begin
    Create_ch(5);
    form_chapter.label1.caption := 'บทที่5';
end;
```

```
procedure TForm_Index.BT_ch6Click(Sender: TObject);
begin
    Create_ch(6);
    form_chapter.label1.caption := 'บทที่6';
end;
```

```
procedure TForm_Index.BT_ch7Click(Sender: TObject);
begin
    Create_ch(7);
    form_chapter.label1.caption := 'บทที่7';
end;
```

```
procedure TForm_Index.BT_ch8Click(Sender: TObject);
begin
    Create_ch(8);
    form_chapter.label1.caption := 'บทที่8';
end;
```

```
procedure TForm_Index.BT_ch9Click(Sender: TObject);
```

begin

นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Create_ch(9);

form1_chapter.label1.caption := 'บทที่9';

end;

procedure TForm_Index.FormActivate(Sender: TObject);
begin
    WinExec('\caihtml\chapter\present.exe',sw_show);
end;

    ehd.

unit Chapter;

interface

uses
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
    Forms, Dialogs, StdCtrls, Buttons, ExtCtrls;

type
    TForm_Chapter = class(TForm)
        Bt_Backindex: TBitBtn;
        BT_Show: TBitBtn;
        Label1: TLabel;
        Notebook: TNotebook;
        RadioButton1: TRadioButton;
        RadioButton2: TRadioButton;
        RadioButton3: TRadioButton;
        RadioButton4: TRadioButton;
        RadioButton5: TRadioButton;
        RadioButton6: TRadioButton;
        RadioButton7: TRadioButton;
        RadioButton8: TRadioButton;
        RadioButton9: TRadioButton;

```

```
RadioButton10: TRadioButton;
RadioButton11: TRadioButton;
RadioButton12: TRadioButton;
RadioButton13: TRadioButton;
RadioButton14: TRadioButton;
RadioButton15: TRadioButton;
RadioButton16: TRadioButton;
RadioButton17: TRadioButton;
RadioButton18: TRadioButton;
RadioButton19: TRadioButton;
RadioButton20: TRadioButton;
RadioButton21: TRadioButton;
RadioButton23: TRadioButton;
RadioButton24: TRadioButton;
RadioButton25: TRadioButton;
RadioButton26: TRadioButton;
RadioButton22: TRadioButton;
procedure Bt_BackindexClick(Sender: TObject);
procedure RadioButton1Click(Sender: TObject);
procedure BT_ShowClick(Sender: TObject);
procedure RadioButton2Click(Sender: TObject);
procedure RadioButton3Click(Sender: TObject);
procedure RadioButton4Click(Sender: TObject);
procedure RadioButton5Click(Sender: TObject);
procedure RadioButton6Click(Sender: TObject);
procedure RadioButton7Click(Sender: TObject);
procedure RadioButton8Click(Sender: TObject);
procedure RadioButton9Click(Sender: TObject);
procedure RadioButton10Click(Sender: TObject);
procedure RadioButton11Click(Sender: TObject);
procedure RadioButton12Click(Sender: TObject);
procedure RadioButton13Click(Sender: TObject);
```

```

procedure RadioButton14Click(Sender: TObject);
procedure RadioButton15Click(Sender: TObject);
procedure RadioButton16Click(Sender: TObject);
procedure RadioButton17Click(Sender: TObject);
procedure RadioButton18Click(Sender: TObject);
procedure RadioButton19Click(Sender: TObject);
procedure RadioButton20Click(Sender: TObject);
procedure RadioButton21Click(Sender: TObject);
procedure RadioButton22Click(Sender: TObject);
procedure RadioButton23Click(Sender: TObject);
procedure RadioButton24Click(Sender: TObject);
procedure RadioButton25Click(Sender: TObject);
procedure RadioButton26Click(Sender: TObject);
private
  Filename,dir : array[0..15] of char ;
  { Private declarations }
public
  { Public declarations }
end;

var
  Form_Chapter: TForm_Chapter;

implementation

{$R *.DFM}

procedure TForm_Chapter.Bt_BackindexClick(Sender: TObject);
begin
  Close;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm_Chapter.RadioButton1Click(Sender: TObject);
begin
    strcpy(Filename,"");
    strcpy(Filename,'chap1_1.scm');
end;

```

```

procedure TForm_Chapter.BT_ShowClick(Sender: TObject);
var name : array[0..80] of char;
    i : integer;
begin

```

```

    for i := 0 to 50 do
        name[i] := ' ';

```

```

        strcpy(name,'caihtml\chapter\scplayer.exe /s/c \caihtml\chapter\');
        strcat(name,filename);
        application.processmessages;
        Winexec(name,sw_show);

```

```

end;

```

```

procedure TForm_Chapter.RadioButton2Click(Sender: TObject);

```

```

begin

```

```

    strcpy(Filename,"");
    strcpy(Filename,'chap1_2.scm');

```

```

end;

```

```

procedure TForm_Chapter.RadioButton3Click(Sender: TObject);

```

```

begin

```

```

    strcpy(Filename,"");
    strcpy(Filename,'chap1_3.scm');

```

```

end;

```

```

procedure TForm_Chapter.RadioButton4Click(Sender: TObject);

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    strcpy(Filename,");
    strcpy(Filename,'chap2_1.scm');
end;
```

```
procedure TForm_Chapter.RadioButton5Click(Sender: TObject);
begin
    strcpy(Filename,");
    strcpy(Filename,'chap2_2.scm');
end;
```

```
procedure TForm_Chapter.RadioButton6Click(Sender: TObject);
begin
    strcpy(Filename,");
    strcpy(Filename,'chap2_3.scm');
end;
```

```
procedure TForm_Chapter.RadioButton7Click(Sender: TObject);
begin
    strcpy(Filename,");
    strcpy(Filename,'chap3_1.scm');
end;
```

```
procedure TForm_Chapter.RadioButton8Click(Sender: TObject);
begin
    strcpy(Filename,");
    strcpy(Filename,'chap3_2.scm');
end;
```

```
procedure TForm_Chapter.RadioButton9Click(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcopy(Filename,'chap3_3.scm');
end;

#
procedure TForm_Chapter.RadioButton10Click(Sender: TObject);
begin
    strcopy(Filename,"");
    strcopy(Filename,'chap3_4.scm');
end;

procedure TForm_Chapter.RadioButton11Click(Sender: TObject);
begin
    strcopy(Filename,"");
    strcopy(Filename,'chap4_1.scm');
end;

procédure TForm_Chapter.RadioButton12Click(Sender: TObject);
begin
    strcopy(Filename,"");
    strcopy(Filename,'chap4_2.scm');
end;

proçedure TForm_Chapter.RadioButton13Click(Sender: TObject);
begin
    strcopy(Filename,"");
    strcopy(Filename,'chap4_3.scm');
end;

procedure TForm_Chapter.RadioButton14Click(Sender: TObject);
begin
    strcopy(Filename,"");
    strcopy(Filename,'chap4_4.scm');
end;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่มีการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm_Chapter.RadioButton15Click(Sender: TObject);
```

```
begin
```

```
    strcpy(Filename,"");
```

```
    strcpy(Filename,'chap5_1.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton16Click(Sender: TObject);
```

```
begin
```

```
    strcpy(Filename,"");
```

```
    strcpy(Filename,'chap5_2.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton17Click(Sender: TObject);
```

```
begin
```

```
    strcpy(Filename,"");
```

```
    strcpy(Filename,'chap5_3.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton18Click(Sender: TObject);
```

```
begin
```

```
    strcpy(Filename,"");
```

```
    strcpy(Filename,'chap6_1.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton19Click(Sender: TObject);
```

```
begin
```

```
    strcpy(Filename,"");
```

```
    strcpy(Filename,'chap6_2.scm');
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
procedure TForm_Chapter.RadioButton20Click(Sender: TObject);
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```
  strcopy(Filename,"");
```

```
  strcopy(Filename,'chap6_3.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton21Click(Sender: TObject);
```

```
begin
```

```
  strcopy(Filename,"");
```

```
  strcopy(Filename,'chap6_4.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton22Click(Sender: TObject);
```

```
begin
```

```
  strcopy(Filename,"");
```

```
  strcopy(Filename,'chap7_1.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton23Click(Sender: TObject);
```

```
begin
```

```
  strcopy(Filename,"");
```

```
  strcopy(Filename,'chap8_1.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton24Click(Sender: TObject);
```

```
begin
```

```
  strcopy(Filename,"");
```

```
  strcopy(Filename,'chap8_2.scm');
```

```
end;
```

```
procedure TForm_Chapter.RadioButton25Click(Sender: TObject);
```

```
begin
```

```
  strcopy(Filename,"");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
strcopy(Filename,'chap9_1.scm');  
end;  
  
procedure TForm_Chapter.RadioButton26Click(Sender: TObject);  
begin  
    strcopy(Filename,"");  
    strcopy(Filename,'chap9_2.scm');  
end;  
  
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ ดร. วรวัฒน์ ลิ้มโสภา ซึ่งเป็น อาจารย์ที่ปรึกษาที่ให้คำปรึกษาแนะนำ และให้โอกาสกับข้าพเจ้าเสมอ ตลอดจนท่านช่วยแก้ไขปัญหาที่เกิดขึ้น ตรวจสอบแก้ไขสิ่งที่บกพร่องจนทำให้โครงการพิเศษชิ้นนี้สำเร็จได้ด้วยดี

ขอขอบพระคุณ คุณ อนุชา บุญชูโต ซึ่งเป็นผู้ให้คำปรึกษาและข้อมูลต่าง ๆ เป็นอย่างดี ขอขอบคุณ คุณ วีระวัฒน์ พัฒโน และ คุณ ประชา พฤกษ์ประเสริฐ รุ่นน้องภาคคอมพิวเตอร์ ที่ให้การช่วยเหลือด้านอุปกรณ์คอมพิวเตอร์ต่างๆ ตลอดจนการเตรียมข้อมูลสำหรับการทำวิทยานิพนธ์นี้

ขอขอบคุณเพื่อนๆ และพี่น้องที่คริสตจักรความหวังลาดกระบัง ที่เป็นกำลังใจและหนุนใจในคำอธิษฐานตลอดการทำวิทยานิพนธ์นี้

ขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การศึกษาและวิทยาการความรู้ในด้านต่างๆ

ท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่และน้องสาวทั้งสองที่ให้การสนับสนุนปัจจัยต่างๆในการเรียน คำสอน พร้อมทั้งกำลังใจตลอดมา

อุษณีย์ เหลืองอ่อน

หนังสืออ้างอิง

Teach Yourself Borland Delphi in 21
สำนักพิมพ์ Sams Publishing

แต่งโดย Andrew Wozniwicz
และ Namir Shammas

Delphi Nuts & Bolts : For Experienced Programmers
สำนักพิมพ์ McGraw-Hill

แต่งโดย Gray Cornell
และ Troy Strain

Delphi Programming Problem Solvex
สำนักพิมพ์ IDG Books Worldwidge

แต่งโดย Neil Rubenking

เริ่มต้น การเขียน Delphi
สำนักพิมพ์ ซีเอ็ด ยูนิเทชั่น จำกัด

แต่งโดย บุญเลิศ เอี่ยมวัฒนา

เรียนรู้ ภาษา HTML
สำนักพิมพ์ ซีเอ็ด ยูนิเทชั่น จำกัด

แต่งโดย รศ.ดร. วิทยา เรื่องพรวิสุทธิ

How to Use HTML 3.2
สำนักพิมพ์ An imprint of Macmillan Computer Publishing

แต่งโดย Scott Arpajian
และ Robert Mullen

Windows 95 Multimedia
สำนักพิมพ์ WAITE Group Press

แต่งโดย Richard J. Simon Tony
และ David, John Eaton