

การพัฒนาโปรแกรมนำเสนองานแบบมัลติมีเดีย

PROGRAM PRESENTATION



กษณะ โอภาสนาถ
ณัฐพล ธนศรีสถิตย์
อิสรา ชุ่มเชื้อ

เลขหมู่.....
เลขทะเบียน 42999
วัน, เดือน, ปี 26 ส.ย. 2545

| |
|---------|
| .b..... |
| .i..... |

ปัญหานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM PRESENTATION






**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2001**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมนำเสนองานแบบมัลติมีเดีย
 ชื่อนักศึกษา นายกษณะ โอภาสธนากร 41056002
 นายณัฐพล ธนศรีสถิตย์ 41056025
 นายอิสรา ชุ่มเชื้อ 41056145
 ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
 สาขาวิชา วิทยาการคอมพิวเตอร์
 อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2544

| คณะกรรมการสอบ | | ลายมือชื่อ |
|----------------------------|--|---|
| ประธานกรรมการ | ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์ |  |
| กรรมการ | อาจารย์วิสันต์ ตั้งวงษ์เจริญ |  |
| กรรมการและอาจารย์ที่ปรึกษา | ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล |  |



(ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|------------------|---|----------|
| หัวข้อปัญหาพิเศษ | โปรแกรมนำเสนองานแบบมัลติมีเดีย | |
| ชื่อนักศึกษา | นายกษณะ โอภาสธนากร | 41056002 |
| | นายณัฐพล ธนศรีสถิตย์ | 41056025 |
| | นายอิสรา ชุ่มเชื้อ | 41056145 |
| ปริญญา | วิทยาศาสตรบัณฑิต | |
| ภาควิชา | คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ | |
| สาขาวิชา | วิทยาการคอมพิวเตอร์ | |
| ปีการศึกษา | 2544 | |
| อาจารย์ที่ปรึกษา | ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล | |

บทคัดย่อ

ในปัจจุบันเป็นที่ยอมรับกันอย่างกว้างขวางว่าระบบคอมพิวเตอร์มีความสำคัญและมีความจำเป็นต่อการพัฒนาเพื่อเพิ่มประสิทธิภาพในการทำงานของทุกระบบงาน ไม่ว่าจะเป็นงานประจำหรืองานส่งเสริมงานบริการของหน่วยงานนั้น ๆ และในหน่วยงานจะประสบความสำเร็จในการดำเนินงานนั้น ๆ ต้องมีการประชุมและการวางแผนที่ดี การนำระบบสารสนเทศเข้ามาใช้ในหน่วยงานนั้น ต้องมีการแสดงระบบสารสนเทศนั้นอย่างมีรูปแบบ ทำให้เป็นที่เข้าใจทั่วกันได้ จึงจะทำให้หน่วยงานนั้น ๆ ประสบความสำเร็จในการดำเนินกิจการ

ดังนั้น โปรแกรมนำเสนองานแบบมัลติมีเดีย (Program Presentation) จึงได้ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกในการแสดงข้อมูล และสารสนเทศต่าง ๆ แสดงขั้นตอนการทำงานของหน่วยงานได้ เพื่อประโยชน์ในการประชุมร่วมกัน ทำให้สามารถทำความเข้าใจและตกลงร่วมกันได้ เพื่อให้เกิดประโยชน์กับหน่วยงานให้ได้มากที่สุด

| | | |
|--------------------------------|--|----------|
| Special Project Title | Program Presentation | |
| Student | Mr.Kasana Opasthanakul | 41056002 |
| | Mr.Nuttapol Thanasrisatit | 41056025 |
| | Mr.Isara Chumchua | 41056145 |
| Degree | Bachelor's Degree of Science | |
| Department | Mathematics and Computer Science, Faculty of Science | |
| Programme | Computer Science | |
| Academic Year | 2001 | |
| Special Project Advisor | Assistant Professor Teerawat Prakobpon | |

ABSTRACT

It is widely accepted in the present day that computer now plays an important role in enhancing the performance of all works. In an attempt to archive the successfulness the organization needs to have a meeting and well planning. To implement the information technology to the organization needs complete understanding in the field to be implemented.

Therefore the multimedia-styled presentation program is developed to help the organization present data, information and its workflow so that people in that organization will have complete understanding and will be able to reach the final agreement as for the benefit of the organization.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษ เรื่อง โปรแกรมนำเสนองานแบบมัลติมีเดีย สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล อาจารย์ผู้รับผิดชอบ ปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ ตลอดระยะเวลาในการทำวิจัยในครั้งนี้และขอขอบพระคุณอาจารย์ทุกท่านที่มอบความเมตตาและให้คำปรึกษาด้วยดีตลอดมา

นอกจากนี้คณะผู้จัดทำต้องขอกราบขอบพระคุณบุพการี บิดา มารดา ผู้อุปการะที่ได้เลี้ยงดู อุ้มชู ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนได้พบกับวันแห่งความสำเร็จเช่นวันนี้ ขอขอบพระคุณอาจารย์ทุกท่านที่มอบความเมตตา และให้คำปรึกษาด้วยดีตลอดมาจนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อนๆ และน้องๆ ทุกคนที่ให้กำลังใจและความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษ ขอขอบพระคุณทุกคนอีกครั้งที่ทำให้งานวิจัยนี้สำเร็จไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2545

สารบัญ

| | หน้า |
|---|------|
| บทคัดย่อภาษาไทย..... | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | X |
| สารบัญภาพ..... | XI |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 1 |
| 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา..... | 1 |
| 1.3 สมมติฐานของการศึกษา..... | 1 |
| 1.4 ขอบเขตของการศึกษา..... | 1 |
| 1.5 ขอบเขตของปัญหา..... | 1 |
| 1.5.1 ศึกษาเกี่ยวกับระบบการสร้างโปรแกรม..... | 1 |
| 1.5.2 ศึกษาโปรแกรมและเครื่องมือต่างๆ..... | 2 |
| 1.5.3 เก็บรวบรวมเอกสาร..... | 2 |
| 1.5.4 วิเคราะห์และออกแบบระบบงาน..... | 2 |
| 1.5.5 พัฒนาโปรแกรม..... | 2 |
| 1.5.6 การทดสอบโปรแกรมและการปรับปรุงโปรแกรม..... | 2 |
| 1.6 ข้อตกลงเบื้องต้น..... | 2 |
| 1.7 ข้อจำกัดของการศึกษา..... | 2 |
| บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง..... | 3 |
| 2.1 วัตถุประสงค์และคุณสมบัติ..... | 3 |
| 2.2 IDE และส่วนประกอบต่างๆ ของ IDE..... | 4 |
| 2.2.1 Toolbar..... | 5 |
| 2.2.2 Toolbox..... | 6 |
| 2.2.3 วินโดว์ Form..... | 7 |
| 2.2.4 วินโดว์ Project Explorer..... | 7 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | |
|--|----|
| 2.2.5 วินโดว์ Properties..... | 8 |
| 2.2.6 วินโดว์ Form Layout..... | 9 |
| 2.2.7 วินโดว์ Code editor..... | 9 |
| 2.3 การบันทึกและการเปิดโปรเจค..... | 10 |
| 2.3.1 การบันทึกโปรเจค..... | 10 |
| 2.3.2 การเปลี่ยนชื่อไฟล์..... | 11 |
| 2.3.3 การเปิด โปรเจค..... | 11 |
| 2.4 การคอมไพล์..... | 12 |
| 2.5 ความแตกต่างระหว่างการบันทึกและการคอมไพล์..... | 13 |
| 2.6 คอมไพล์เลอร์และอินเตอร์พรีเตอร์..... | 13 |
| 2.7 ขั้นตอนการเขียนโปรแกรมด้วย Visual Basic..... | 14 |
| 2.7.1 ออกแบบหน้าต่างของโปรแกรมที่เราต้องการ..... | 14 |
| 2.7.2 การลบบอบเจค..... | 15 |
| 2.8 ชนิดของตัวแปรใน Visual Basic..... | 15 |
| 2.9 การประกาศตัวแปรและกฎการตั้งชื่อตัวแปร..... | 16 |
| 2.10 จุดประสงค์ของการประกาศตัวแปรและ คำสั่ง Option Explicit..... | 16 |
| 2.11 ตัวแปรแบบค่าคงที่..... | 18 |
| 2.12 โปรแกรมย่อย..... | 18 |
| 2.13 ฟังก์ชัน..... | 18 |
| 2.14 Event Procedure..... | 19 |
| 2.15 ขอบเขตของตัวแปร..... | 19 |
| 2.15.1 ตัวแปร Public..... | 19 |
| 2.15.2 ตัวแปร Private..... | 19 |
| 2.15.3 ตัวแปร Local..... | 19 |
| 2.16 ตัวแปรแบบ static..... | 19 |
| 2.17 ตัวดำเนินการและนิพจน์..... | 20 |
| 2.17.1 ตัวดำเนินการทางคณิตศาสตร์..... | 20 |
| 2.17.2 ตัวดำเนินการเปรียบเทียบ..... | 20 |
| 2.17.3 ตัวดำเนินการทางตรรกะ..... | 20 |
| 2.18 ลำดับการประมวลผลของตัวดำเนินการ..... | 22 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | | |
|--------|--|----|
| 2.19 | ประโยคคำสั่ง..... | 23 |
| 2.19.1 | ประโยคกำหนดค่า..... | 23 |
| 2.19.2 | ประโยคเงื่อนไข..... | 23 |
| 2.19.3 | ประโยคแบบวนซ้ำ..... | 23 |
| 2.20 | ประโยคแบบเงื่อนไข..... | 23 |
| 2.21 | คอนโทรลหลัก..... | 26 |
| 2.21.1 | Textbox..... | 26 |
| 2.21.2 | Command Button..... | 27 |
| 2.21.3 | Option Button และ Checkbox..... | 28 |
| 2.21.4 | Listbox และ Combobox..... | 28 |
| 2.21.5 | Scrollbar..... | 30 |
| 2.21.6 | Label..... | 30 |
| 2.21.7 | Frame..... | 31 |
| 2.22 | คำสั่ง MsgBox..... | 32 |
| 2.22.1 | MsgBox ที่เป็นฟังก์ชัน..... | 33 |
| 2.23 | Microsoft Common Dialog Control..... | 34 |
| 2.23.1 | เมธอด ShowOpen..... | 35 |
| 2.23.2 | เมธอด ShowSave..... | 35 |
| 2.24 | การนำ Common Dialog Control มาใช้ในโปรเจค..... | 35 |
| 2.25 | การเคลื่อนย้ายวัตถุด้วยเมธอด Move..... | 36 |
| 2.26 | การ Drag และ Drop..... | 36 |
| 2.27 | พรอพเพอร์ตี้ DragMode..... | 36 |
| 2.28 | พรอพเพอร์ตี้ DragIcon..... | 36 |
| 2.29 | เหตุการณ์ DragDrop..... | 36 |
| 2.30 | เหตุการณ์ DragOver..... | 37 |
| 2.31 | เมธอด Drag..... | 37 |
| 2.32 | การออกแบบเมนู..... | 38 |
| 2.33 | การเรียกใช้และใช้งาน Menu Editor..... | 38 |
| 2.34 | การสร้างเมนูย่อย..... | 40 |
| 2.35 | การควบคุมเมนูขณะโปรแกรมทำงานอยู่..... | 41 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | | |
|---------|---|----|
| 2.36 | เมนูพอป๊อป..... | 41 |
| 2.37 | การทำงานกับฟอร์ม..... | 42 |
| 2.38 | เมธอด Show,Hide และ คำสั่ง Unload..... | 43 |
| 2.39 | การเปิดฟอร์มแบบ Modal และ Modeless..... | 43 |
| 2.40 | การอ้างตัวแปรหรือออบเจ็คระหว่างฟอร์ม..... | 44 |
| 2.41 | ออบเจ็คที่เกี่ยวข้องกับกราฟฟิก..... | 44 |
| 2.41.1 | Picture Box..... | 44 |
| 2.41.2 | Image..... | 44 |
| 2.41.3 | Line..... | 45 |
| 2.41.4 | Shape..... | 45 |
| 2.42 | การกำหนดภาพ..... | 46 |
| 2.43 | การทำงานเกี่ยวกับออบเจ็คของภาพ..... | 47 |
| บทที่ 3 | การออกแบบและการพัฒนาโปรแกรม..... | 49 |
| 3.1 | ขั้นตอนการออกแบบ..... | 49 |
| 3.1.1 | การออกแบบระบบ..... | 49 |
| 3.1.1.1 | การออกแบบทั่วไป..... | 49 |
| 3.1.1.2 | การออกแบบเมนูและการรับข้อมูล..... | 50 |
| 3.1.2 | การออกแบบอินเทอร์เฟซติดต่อกับผู้ใช้..... | 51 |
| 3.1.2.1 | บริเวณพื้นที่การทำงาน..... | 51 |
| 3.1.2.2 | แถบเมนูคำสั่ง..... | 52 |
| 3.1.2.3 | แถบของปุ่มคำสั่ง..... | 52 |
| 3.1.2.4 | ส่วนที่แสดงสถานะของหน้า..... | 53 |
| 3.1.3 | การออกแบบการทำงานในส่วนต่างๆของโปรแกรม..... | 53 |
| 3.1.3.1 | ส่วนของการวางภาพ..... | 53 |
| 3.1.3.2 | ส่วนของการวางข้อความ..... | 53 |
| 3.1.3.3 | ขั้นตอนการบันทึกโปรแกรม..... | 54 |
| 3.1.3.4 | รูปแบบการเคลื่อนที่ของรูปภาพ..... | 56 |
| 3.1.3.5 | รูปแบบการเคลื่อนที่ของข้อความ..... | 56 |

สารบัญ(ต่อ)

| | |
|---|-----------|
| 3.1.3.6 ลำดับการแสดงผล..... | 57 |
| 3.1.3.7 เพลง..... | 57 |
| 3.1.3.4 รูปแบบการเคลื่อนที่ของรูปภาพ..... | 57 |
| 3.1.3.5 รูปแบบการเคลื่อนที่ของข้อความ..... | 58 |
| 3.1.3.10 ลำดับการแสดง..... | 58 |
| 3.1.3.11 การลบรูปภาพ..... | 58 |
| 3.1.3.12 การลบข้อความ..... | 58 |
| 3.1.3.13 การเปิดไฟล์..... | 58 |
| 3.2 ขั้นตอนการเขียนโปรแกรม..... | 59 |
| บทที่ 4 การใช้โปรแกรม..... | 69 |
| 4.1 การใส่พื้นหลัง..... | 69 |
| 4.1.1 พื้นหลังแบบมีสี..... | 69 |
| 4.1.2 พื้นหลังแบบมีรูปภาพ..... | 69 |
| 4.2 การใส่รูป..... | 70 |
| 4.3 การใส่ข้อความ..... | 71 |
| 4.4 การใส่เพลง..... | 71 |
| 4.5 การกำหนดการเคลื่อนที่ให้กับรูปภาพ..... | 72 |
| 4.6 การกำหนดการเคลื่อนที่ให้กับข้อความ..... | 72 |
| 4.7 การเปลี่ยนแปลงรูปภาพ..... | 73 |
| 4.8 การเปลี่ยนแปลงข้อความ..... | 73 |
| 4.9 การเปลี่ยนแปลงขนาดรูปภาพ..... | 73 |
| 4.10 การลบรูป..... | 74 |
| 4.11 การลบข้อความ..... | 74 |
| 4.12 การเปลี่ยนแปลงลำดับการแสดง..... | 74 |
| 4.13 การเพิ่มหน้า..... | 74 |
| 4.14 การนำเสนองาน..... | 74 |
| 4.15 การบันทึกงาน..... | 75 |
| 4.16 การเปิดไฟล์..... | 75 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | |
|---|-----------|
| 4.17 การออกจากโปรแกรม..... | 75 |
| บทที่ 5 สรุปและข้อเสนอแนะ..... | 76 |
| 5.1 บทสรุป..... | 76 |
| 5.2 ข้อจำกัดในการใช้งานโปรแกรม..... | 77 |
| 5.2.1 คุณสมบัติที่ใช้ในการทดสอบ..... | 77 |
| 5.2.2 ข้อจำกัดของโปรแกรม..... | 77 |
| 5.2.2.1 ข้อจำกัดเรื่องหน่วยความจำ..... | 77 |
| 5.2.2.2 การเคลื่อนที่ของรูปภาพ..... | 77 |
| 5.2.2.3 จำนวนรูปภาพและข้อความ..... | 78 |
| 5.2.2.4 จำนวนหน้าของการนำเสนองาน..... | 78 |
| 5.2.2.5 ไฟล์เพลงที่ใช้ในการนำเสนองาน..... | 78 |
| 5.3 บทการวิจารณ์และแนวทางการพัฒนา..... | 78 |
| 5.3.1 ข้อจำกัดในการแสดงรูปภาพและข้อความ..... | 78 |
| 5.3.2 ข้อจำกัดของจำนวนหน้า..... | 78 |
| 5.3.3 การกระพริบของรูปภาพในการเคลื่อนที่..... | 79 |
| ภาคผนวก ก..... | 81 |
| ภาคผนวก ข..... | 84 |
| บรรณานุกรม..... | 94 |

สารบัญตาราง

ตารางที่

หน้า

2.1 ค่าคงที่รับกลับมาจากฟังก์ชันMsgBox.....34



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

| รูปที่ | หน้า |
|--|------|
| 2.1 ตัวอย่างออบเจ็ค..... | 3 |
| 2.2 ส่วนประกอบต่างๆของ IDE..... | 4 |
| 2.3 Toolbar..... | 5 |
| 2.4 Toolbox..... | 6 |
| 2.5 วินโดว์ Form..... | 7 |
| 2.6 วินโดว์ Project Explorer..... | 8 |
| 2.7 วินโดว์ Properties..... | 9 |
| 2.8 วินโดว์ Form Layout..... | 9 |
| 2.9 วินโดว์ Code Editor..... | 10 |
| 2.10 การบันทึกโปรเจค..... | 10 |
| 2.11 การเปิดโปรเจควิธีที่ 1..... | 11 |
| 2.12 การเปิดโปรเจควิธีที่ 2..... | 12 |
| 2.13 การคอมไพล์..... | 12 |
| 2.14 การเลือกและวางออบเจ็ค..... | 14 |
| 2.15 การจัดตำแหน่งและปรับขนาดออบเจ็ค..... | 14 |
| 2.16 การลบออบเจ็ค..... | 15 |
| 2.17 การใช้คำสั่ง Option Explicit..... | 17 |
| 2.18 แสดงผลการเลือกใช้คำสั่ง Option Explicit..... | 17 |
| 2.19 ตัวอย่างการใช้ Textbox..... | 26 |
| 2.20 ตัวอย่างการใช้ Button..... | 27 |
| 2.21 ตัวอย่างการใช้ Option Button และ Check Box..... | 28 |
| 2.22 ตัวอย่างการใช้ Listbox และ ComboBox..... | 29 |
| 2.23 ตัวอย่างการใช้ Scrollbar..... | 30 |
| 2.24 ลักษณะการใช้งาน Label..... | 31 |
| 2.25 การใช้ Frame..... | 32 |
| 2.26 ไดอะล็อกบ็อกซ์ Open..... | 35 |
| 2.27 ไดอะล็อกบ็อกซ์ Save..... | 35 |
| 2.28 Menu Editor..... | 38 |

สารบัญภาพ(ต่อ)

| | |
|--|----|
| 2.29 การสร้างเมนูย่อย..... | 40 |
| 2.30 เมนูแบบพ็อปอัพ..... | 41 |
| 2.31 การเพิ่มฟอร์มเข้าสู่โปรเจ็ค..... | 42 |
| 2.32 การเลือกฟอร์มเปล่า..... | 42 |
| 2.33 ฟอร์มใหม่ที่เพิ่มเข้ามาในโปรเจค..... | 43 |
| 2.34 Image..... | 45 |
| 2.35 Line..... | 45 |
| 3.1 รูปภาพ ข้อความ ปุ่มบังคับลำดับการทำงาน..... | 50 |
| 3.2 หน้าจอแสดงการนำเสนอ..... | 50 |
| 3.3 อินเทอร์เฟซที่ติดต่อผู้ใช้..... | 51 |
| 3.4 แถบเมนูคำสั่ง..... | 52 |
| 3.5 แถบของปุ่มคำสั่ง..... | 53 |
| 3.6 ส่วนที่แสดงสถานะของหน้า..... | 53 |
| 3.7 แผนภาพการทำงานของโปรแกรม..... | 64 |
| 3.8 แผนภาพแสดงการรับ Input..... | 66 |
| 3.9 แผนภาพแสดงขั้นตอนการ Play..... | 67 |
| 3.10 แผนภาพแสดงขั้นตอนการ Play..... | 68 |
| 4.1 การใส่พื้นหลังแบบมีสี..... | 69 |
| 4.2 การใส่พื้นหลังแบบรูปภาพ..... | 70 |
| 4.3 การใส่รูป..... | 70 |
| 4.4 การใส่ข้อความ..... | 71 |
| 4.5 การใส่เพลง..... | 71 |
| 4.6 การกำหนดรูปแบบการแสดงของรูปภาพ..... | 72 |
| 4.7 การกำหนดรูปแบบการเคลื่อนที่ของข้อความ..... | 72 |
| 4.8 การเปลี่ยนแปลงรูปภาพ..... | 73 |
| ก.1 ไลอะต็อกบ็อกซ์เพื่อทำการติดตั้งโปรแกรม..... | 81 |
| ก.2 การเลือก Directory เพื่อทำการติดตั้งโปรแกรม..... | 81 |
| ก.3 การเลือก Program Group..... | 82 |
| ก.4 เริ่มการติดตั้งโปรแกรม..... | 82 |
| ก.5 โปรแกรมติดตั้งเสร็จสมบูรณ์..... | 83 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

| | |
|--|----|
| ข.1 การเปิดไฟล์..... | 84 |
| ข.2 การบันทึกไฟล์..... | 85 |
| ข.3 การบันทึกไฟล์เป็น..... | 85 |
| ข.4 การปิดไฟล์..... | 86 |
| ข.5 การเลือก Background แบบที่ 1..... | 86 |
| ข.6 การเลือก Background แบบที่ 2..... | 87 |
| ข.7 การเลือกรูปภาพ..... | 87 |
| ข.8 การเปลี่ยนแปลงรูปภาพ..... | 88 |
| ข.9 การเปลี่ยนแปลงขนาดรูปภาพ..... | 88 |
| ข.10 การลบรูป..... | 89 |
| ข.11 การใส่ข้อความ..... | 89 |
| ข.12 การเปลี่ยนแปลงรูปแบบข้อความ..... | 90 |
| ข.13 การลบข้อความ..... | 90 |
| ข.14 การเลือกเพลง..... | 91 |
| ข.15 การกำหนดรูปแบบการแสดงข้อความ..... | 92 |
| ข.16 การกำหนดรูปแบบการแสดงรูปภาพ..... | 92 |
| ข.17 การกำหนดลำดับการแสดงผล..... | 93 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในทุกวันนี้ เมื่อมีการทำงานอย่างใดอย่างหนึ่งออกมา ก็ต้องมีการนำงานนั้นมานำเสนอให้คนภายนอกได้รับรู้ ได้ทราบรายละเอียดของงานนั้น ๆ ซึ่งวิธีการนำเสนองานนี้ก็มิได้หลายแบบ เช่น อาจจะเป็นแบบข้อความ หรือเป็นรูปภาพหนึ่ง ๆ อาจจะมีมัลติมีเดียต่าง ๆ ด้วย เช่น เสียง ภาพ เคลื่อนไหวในลักษณะต่าง ๆ ซึ่งโปรแกรมในลักษณะนี้ ในต่างประเทศอาจมีการพัฒนาเป็นจำนวนมากแล้ว แต่ในประเทศไทย การพัฒนาโปรแกรมนี้อย่างไรยังไม่ได้รับความนิยมเท่าที่ควร เพราะฉะนั้น โปรแกรมนี้จะ เป็นโปรแกรมที่ทำการพัฒนาขึ้นมาเป็นภาษาไทย เพื่อให้คนไทยได้ใช้โปรแกรมนี้ อย่างเต็มที่ ซึ่งถ้าเป็นโปรแกรมที่นำมาจากต่างประเทศ อาจจะมีคำสั่งเป็นภาษาของประเทศนั้น ๆ ทำให้เราไม่สามารถใช้คำสั่งการทำงานได้เต็มที่ และหวังว่าจะเป็นประโยชน์เพื่อนำไปศึกษาต่อ เพื่อเป็นแนวทางเบื้องต้นสำหรับผู้สนใจที่จะพัฒนาโปรแกรมและยังเป็นแรงผลักดันให้มีการส่งเสริม การเกิดอุตสาหกรรมการผลิต โปรแกรมในประเทศไทย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อศึกษาระบบและวิธีการสร้างโปรแกรม
- 1.2.2 เพื่อสร้างโปรแกรมที่สามารถนำเสนองานที่เราได้เตรียมไว้แล้ว
- 1.2.3 เพื่อศึกษาความสามารถทางด้านกราฟฟิกของ Visual Basic 6

1.3 สมมติฐานของการศึกษา

- 1.3.1 โปรแกรมสามารถที่จะเสนองานได้หลายรูปแบบ โดยผู้ใช้สามารถกำหนดรายละเอียดได้
- 1.3.2 เป็นประโยชน์ต่อผู้ที่นำไปศึกษาเป็นแนวทางต่อไป
- 1.3.3 สามารถเขียนโปรแกรมเสนองาน โดยสามารถแสดงวัตถุเคลื่อนที่ได้ตามลักษณะงาน

1.4 ขอบเขตของการศึกษา

การพัฒนาโปรแกรมทำได้โดยการใช้โปรแกรม Microsoft Visual Basic 6.0

1.5 ขอบเขตของปัญหา

1.5.1 ศึกษาเกี่ยวกับระบบการสร้างโปรแกรม

เป็นขั้นตอนการศึกษาถึงวิธีการและหลักการที่ใช้ในการสร้างโปรแกรม รวมทั้งเทคนิคต่างๆที่จะนำมาใช้ในการเขียนและพัฒนาระบบ

1.5.2 ศึกษาโปรแกรมและเครื่องมือต่างๆ

เป็นการศึกษาโปรแกรมและเครื่องมือ ที่จะนำมาใช้ในและพัฒนาในการเขียนโปรแกรม โดยโปรแกรมและเครื่องมือที่นำมาใช้ในการพัฒนาจะเน้นโปรแกรม Microsoft Visual Basic 6.0 และโปรแกรมทางด้านกราฟฟิกเพื่อนำมาใช้ในการสร้างภาพ โดยจะเน้นที่โปรแกรม Adobe PhotoShop

1.5.3 เก็บรวบรวมเอกสารและข้อมูลต่างๆ

เป็นการนำเอาเอกสารและข้อมูลที่เกี่ยวข้องมารวบรวม และใช้ประกอบการทำงาน โดยส่วนมากจะเป็นการรวบรวมจากหนังสือ ตำรา และ Web site ที่เกี่ยวข้อง รวมถึงข้อมูลที่ได้จากผู้ที่มีความรู้ในแต่ละด้าน

1.5.4 วิเคราะห์และออกแบบระบบงาน

วิเคราะห์และออกแบบระบบงานโดยแบ่งงานออกเป็นส่วนๆ เช่น ส่วนการรับข้อมูล ส่วนประมวลผล ส่วนการแสดงผล และออกแบบระบบเพื่อให้ทำงานได้เกิดประสิทธิภาพมากที่สุด

1.5.5 พัฒนาโปรแกรม

เป็นขั้นตอนการเขียนโปรแกรมตามขั้นตอนที่ได้ออกแบบไว้ในขั้นตอนของการวิเคราะห์และขั้นตอนของการออกแบบ

1.5.6 การทดสอบโปรแกรมและการปรับปรุงโปรแกรม

เป็นขั้นตอนของการทดสอบโปรแกรม และบอกถึงความสามารถทั้งหมดที่เป็นไปได้ของโปรแกรม

1.6 ข้อตกลงเบื้องต้น

1.6.1 โปรแกรมจะสามารถรันได้บนระบบปฏิบัติการ Windows ใน Version 98

1.6.2 ต้องการคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางตั้งแต่ระดับ Pentium 2 ความเร็ว 266 MHz ขึ้นไปเป็นอย่างน้อย

1.6.3 มีหน่วยความจำอย่างน้อย 64 Mbytes

1.6.4 ต้องมีฮาร์ดแวร์อินพุทประเภท Mouse และ Keyboard

1.7 ข้อกำหนดของการศึกษา

1.7.1 ระยะเวลาที่ทำการศึกษา

1.7.2 ความรู้และความเข้าใจที่ได้จากการศึกษา

1.7.3 ทฤษฎีที่กำหนดขึ้นมาแล้วในการศึกษาปัญหา

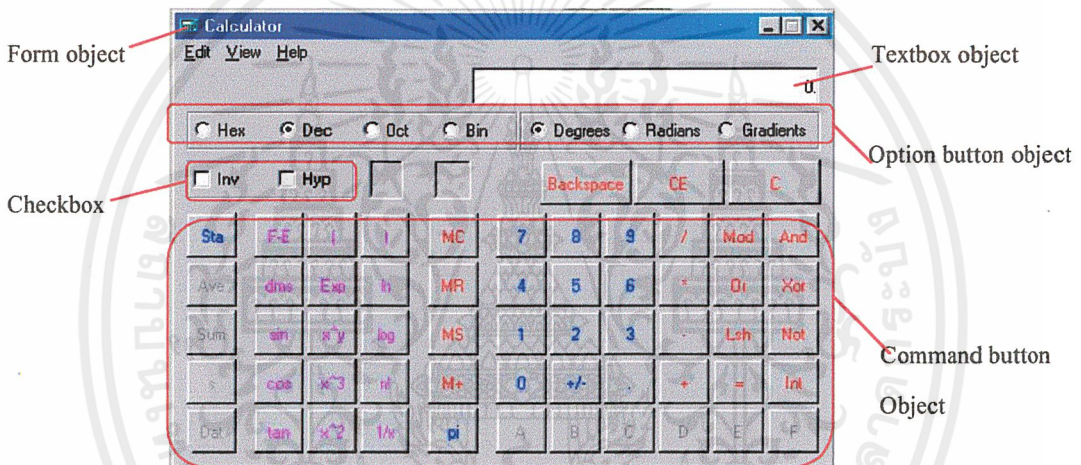
1.7.4 เทคโนโลยีการศึกษาปัญหาพิเศษ

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 วัตถุและคุณสมบัติ

คำว่า วัตถุ (Object) หรือวัตถุอาจหมายถึงสิ่งต่างๆ ได้มากมาย ถ้าเราพูดถึงวัตถุโดยทั่วไปจะหมายถึงสิ่งต่างๆ ที่อยู่รอบตัวเรา เช่น ก้อนหิน แก้วน้ำ แต่สำหรับการโปรแกรมของ Visual Basic นั้น คำว่า “วัตถุ” จะหมายถึงส่วนประกอบต่างๆ ที่ประกอบกันขึ้นมาเป็นโปรแกรมที่เราใช้งาน ลองพิจารณาภาพต่อไปนี้ซึ่งเป็น โปรแกรมเครื่องคิดเลข เราจะพบว่า โปรแกรมประกอบขึ้นด้วยวัตถุต่างๆ เช่น ฟอรัม, ปุ่มกด, ช่องใส่ข้อความและปุ่ม Option

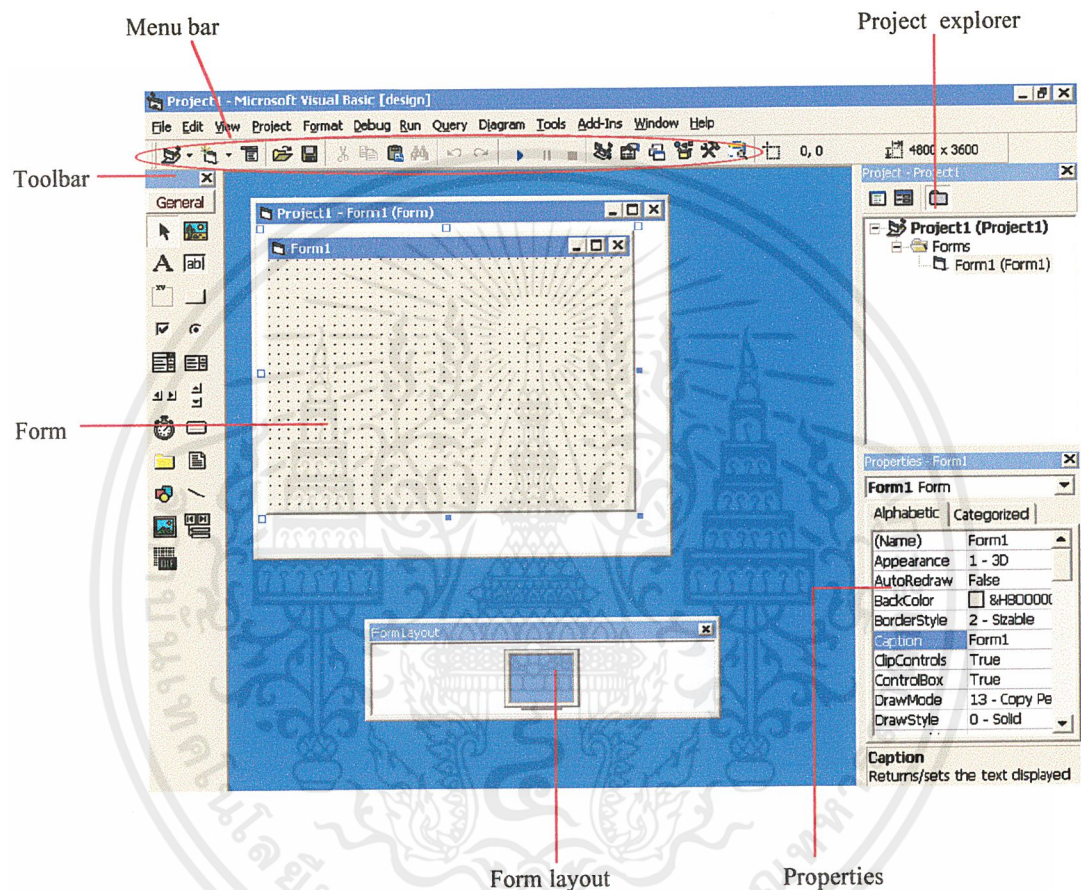


รูปที่ 2.1 ตัวอย่างวัตถุ

คำว่า พรอพเพอร์ตี้ (Property) หรือคุณสมบัติ นั้น ถ้าพิจารณาวัตถุที่เป็นแก้วน้ำคุณสมบัติของแก้วน้ำคือ ความกว้าง ความสูง ถ้าพิจารณาที่วัตถุย่อยของโปรแกรม เช่น Command Button ที่เป็นปุ่มกด ก็จะมีพรอพเพอร์ตี้ต่างๆ ได้แก่ ความกว้าง ความสูงของปุ่ม เป็นต้น

2.2 IDE และส่วนประกอบต่างๆ ของ IDE

คำว่า IDE (Integrated Development Environment) หมายถึงสภาพแวดล้อมการทำงาน ในการพัฒนาโปรแกรมโดย Visual Basic หรือจะแปลอีกอย่างคือ อุปกรณ์เครื่องมือต่างๆแบบเปรียบพร้อมที่ไมโครซอฟเตรียมมาให้ใช้ในการพัฒนาโปรแกรมด้วย Visual Basic นั้นเอง โปรแกรมจะปรากฏหน้าจอ IDE ดังนี้



รูปที่ 2.2 ส่วนประกอบต่างๆของ IDE

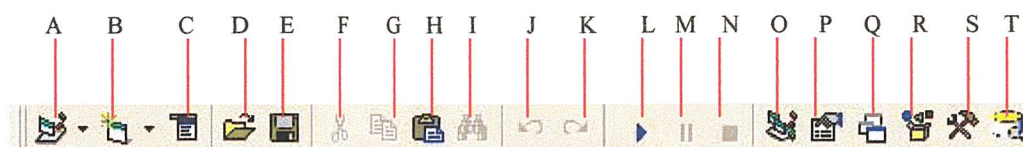
ส่วนประกอบหลังของหน้าจอ IDE มีดังนี้

- ◆ Toolbar
- ◆ Toolbox
- ◆ วินโดว์ Form
- ◆ วินโดว์ Project Explorer
- ◆ วินโดว์ Properties
- ◆ วินโดว์ Form Layout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 Toolbar

จากหน้าจอ IDE จะเห็นปุ่มต่างๆ ที่วางเรียงเป็นแถวแนวนอน ช่วยให้สามารถเรียกใช้งานคำสั่งได้อย่างสะดวกรวดเร็ว โดยเพียงแค่คลิกเมาส์ที่ปุ่มเท่านั้น รายละเอียดมีดังนี้

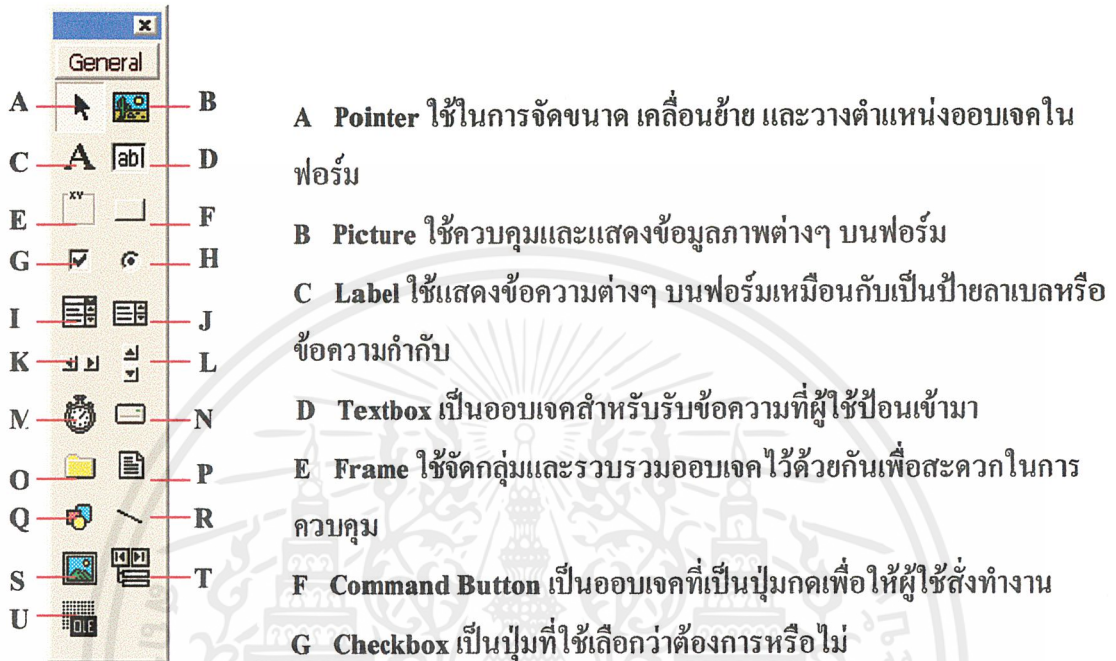


รูปที่ 2.3 Toolbar

- A เปิดโปรเจกต์ขึ้นมาใหม่
- B เพิ่มฟอร์ม โมดูลหรือออบเจกต์ประเภทต่างๆ เข้าไปในโปรเจกต์ที่เรากำลังพัฒนาอยู่
- C เปิดวินโดว์ Menu Editor
- D เปิดโปรเจกต์
- E บันทึกไฟล์โปรเจกต์
- F ตัด
- G ก๊อปปี้
- H วาง
- I ค้นหา
- J ยกเลิกการทำงานหรือการพิมพ์
- K เรียกคืนกลับสิ่งที่ undo ไป
- L สั่งให้โปรแกรมทำงาน
- M ให้โปรแกรมหยุดการทำงานชั่วคราว
- M ให้โปรแกรมหยุดทำงาน
- O เปิดวินโดว์ Project Explorer ที่แสดงฟอร์ม โมดูล และส่วนประกอบต่างๆ ของโปรเจกต์
- P เปิดวินโดว์ Properties เพื่อดูและกำหนดคุณสมบัติต่างๆ ของออบเจกต์
- Q เปิดวินโดว์ Form Layout เพื่อจัดตำแหน่งวินโดว์ของ โปรแกรมบนจอภาพเมื่อโปรแกรมทำงาน
- R Object Browser ช่วยค้นหารายละเอียดต่างๆ
- S Toolbox เป็นที่รวมของออบเจกต์ต่างๆ ที่นำมาประกอบใน โปรแกรม
- T เปิดวินโดว์ Data View เพื่อดูการติดต่อกับฐานข้อมูลต่างๆ

2.2.2 Toolbox

เป็นที่รวมออบเจกต์ต่างๆ ที่จะนำมาประกอบกันเป็นโปรแกรมหรือแอปพลิเคชัน เมื่อใช้ออบเจกต์เหล่านี้ประกอบกันจะได้เป็นหน้าตาของโปรแกรม จึงอาจเรียกได้ว่า Control object ซึ่งมีออบเจกต์หลักดังนี้



รูปที่ 2.4 Toolbox

H **Option Button หรือ Radio Button** ใช้สำหรับเลือกค่าใดค่าหนึ่งจากหลายๆค่าเลือกได้ครั้งละ 1 ปุ่มเท่านั้น

I **Combobox** สามารถเลือกตัวเลือกได้จากการกดปุ่ม Drop down เพื่อแสดงทางเลือกต่างๆ ขึ้นมา

J **Listbox** ใช้แสดงตัวเลือกต่างๆ ในลักษณะของบรรทัดรายการ โดยสามารถเลือกรายการใดรายการหนึ่งหรือหลายๆ รายการ

K **Horizontal Scroll bar** เป็นแถบเลื่อนทางแนวนอน ใช้เลื่อนปรับค่าโดยค่าจะเปลี่ยนไปตามตำแหน่งที่อยู่ของแถบเลื่อน

L **Verical Scroll bar** เป็นแถบเลื่อนในแนวตั้ง ใช้เลื่อนปรับค่าโดยค่าจะเปลี่ยนไปตามตำแหน่งที่อยู่ของ แถบเลื่อน

M **Timer** ใช้ในการควบคุมเวลา และการทำงานของโปรแกรมเมื่อมีเวลาเข้ามาเกี่ยวข้อง

N-P **Drive List box, Directory List Box , File List Box** ใช้ในการควบคุมการติดต่อกับระบบแฟ้มข้อมูลของเครื่อง

Q **Shape** ใช้สร้างภาพรูปทรงต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R Line ใช้วาดเส้นต่างๆในฟอร์ม

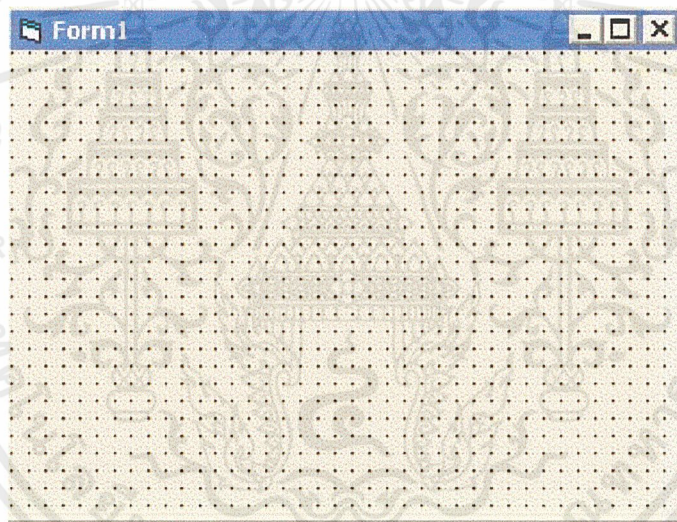
S Image เป็นคอนโทรลใช้ควบคุมข้อมูลภาพเหมือนกับ Picture

T Data Control ใช้ในการติดต่อกับฐานข้อมูล

U OLE (Object Linked and Embedded) เป็นคอนโทรลที่นำเอาโปรแกรมสำเร็จรูปต่างๆ ที่มีความสามารถ OLE มาใช้เป็นออบเจกต์ในโปรเจกต์

2.2.3 วินโดว์ Form

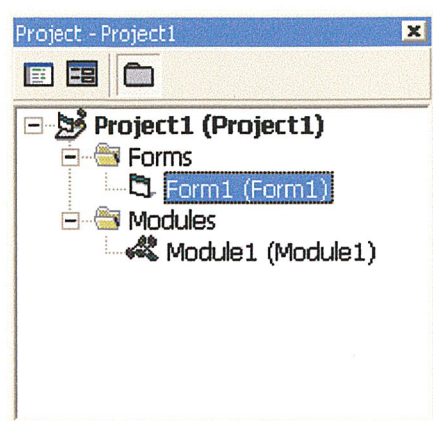
เป็นวินโดว์เปล่าๆ หรือตัวฟอร์มเปล่าๆ สำหรับสร้างองค์ประกอบของแอปพลิเคชัน โดยการนำออบเจกต์ต่างๆ มาใส่ในฟอร์ม หรือพูดอีกนัยหนึ่งคือ เป็นหน้าจอของโปรแกรมที่ผู้ใช้จะเห็นเมื่อเรียกใช้งาน โปรแกรมนั่นเอง เมื่อเริ่มเข้าสู่ Visual basic ฟอร์มเปล่าๆจะมีให้เสมอ การเรียกดูฟอร์มสามารถใช้คีย์ Shift+F7 หรือจากเมนู View > Object ก็ได้



รูปที่ 2.5 วินโดว์ Form

2.2.4 วินโดว์ Project Explorer

Project Explorer จะใช้ควบคุมส่วนประกอบและเพิ่มข้อมูลต่างๆที่อยู่ในโปรเจกต์ เพื่อความสะดวกในการควบคุมและเปลี่ยนการทำงานระหว่างส่วนประกอบต่างๆ



รูปที่ 2.6 วินโดว์ Project Explorer

โดยแต่ละโปรเจกต์จะประกอบด้วยแฟ้มข้อมูลหลักๆ ได้แก่

Project File เก็บข้อมูลต่างๆ ของโปรเจกต์ รวมทั้งรายชื่อแฟ้มที่ประกอบขึ้นมาเป็นโปรเจกต์ .

Form File เก็บฟอร์มที่ได้ออกแบบไว้ โดยไฟล์นี้จะรวมคำสั่งต่างๆ ที่เขียนโปรแกรมไว้ให้กับแต่ละออบเจกต์ที่อยู่ในฟอร์มด้วย

Standard module เก็บโปรแกรมย่อยและตัวแปรต่างๆ ที่แยกออกจากฟอร์มเพื่อให้ฟอร์มอื่นสามารถเรียกใช้ได้

Object Control ออบเจกต์ที่เพิ่มเข้าไปในโปรเจกต์นอกเหนือจากคอนโทรลพื้นฐาน

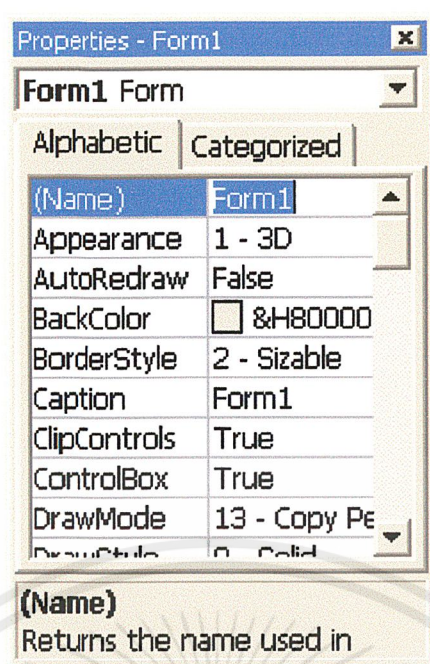
ActiveX เหมือนฟอร์มเพียงแต่ต้องดูผ่านโปรแกรมเว็บเบราว์เซอร์

Class Module เก็บออบเจกต์ต่างๆ ที่สร้างขึ้นเมื่อมีการเรียกใช้ Class Module โปรแกรมก็จะสร้างออบเจกต์นั้นขึ้นมาใหม่แทนที่จะใช้จากโมดูลนั้นโดยตรง

Resource File เก็บภาพ Bitmap, ข้อความหรือข้อมูลใดๆ ที่สามารถแก้ไขได้โดยไม่ต้องไปยุ่งกับโปรแกรมในโมดูลหรือฟอร์มต่างๆ ในโปรเจกต์

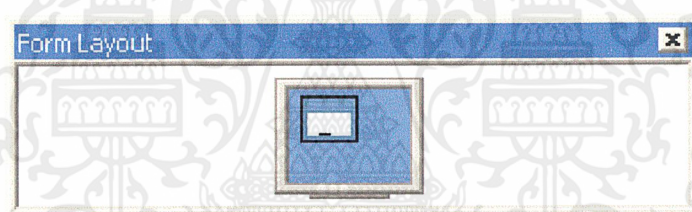
2.2.5 วินโดว์ Properties

จะแสดงคุณสมบัติทั้งหมดของออบเจกต์ที่ถูกเลือกอยู่ การเลือกที่ออบเจกต์ใดในฟอร์มจะทำให้คุณสมบัติที่แสดงในวินโดว์ Properties เปลี่ยนไปตามออบเจกต์ที่เลือก การแก้ไขหรือตั้งค่าคุณสมบัติต่างๆทำได้โดยตรงที่คุณสมบัติแต่ละค่าโดย Alphabetic จะแสดงคุณสมบัติเรียกตามชื่อตัวอักษรแท้แบบ Categorized แสดงตามการใช้งาน การเรียกวินโดว์ Properties เรียกได้จาก View > Properties window หรือกด F4



รูปที่ 2.7 วินโดว์ Properties

2.2.6 วินโดว์ Form Layout



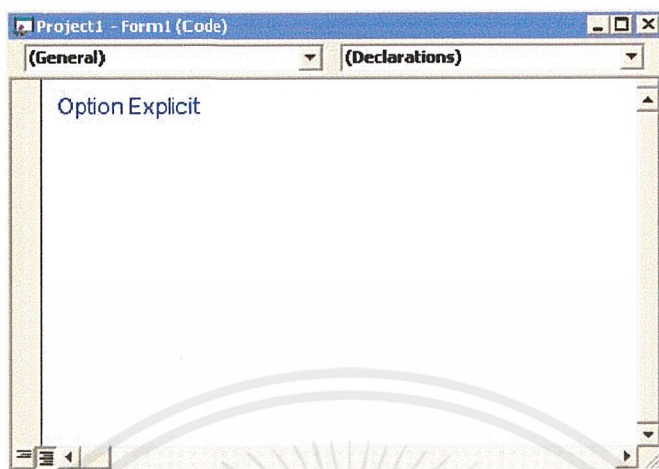
รูปที่ 2.8 วินโดว์ Form Layout

จะแสดงตำแหน่งฟอร์มของโปรแกรมที่กำลังสร้างให้ดูบนจอภาพ เพื่อกำหนดตำแหน่งสำหรับตอนที่โปรแกรมทำงานจริงๆ การย้ายตำแหน่งทำได้โดยใช้เมาส์ลาก (Drag) รูปฟอร์มตรงกลางจอภาพไปยังตำแหน่งที่ต้องการ ซึ่งสามารถทดลองได้โดยเลื่อนตำแหน่งแล้วกด F5 เพื่อรันโปรแกรม จะเห็นว่าตำแหน่งโปรแกรมถูกเคลื่อนตามไปด้วย

2.2.7 วินโดว์ Code Editor

Code Editor เป็นเนื้อที่สำหรับเขียนโปรแกรม เรียกขึ้นมาแสดงได้โดยใช้เมนู View > Code หรือดับเบิลคลิกที่ออบเจกต์ใดๆ ในฟอร์ม ซึ่งวินโดว์ Code Editor จะแสดงขึ้นมาพร้อมสำหรับการป้อนโปรแกรมให้กับเหตุการณ์หลักของออบเจกต์นั้น ส่วนที่สำคัญ คือ Combobox ทั้งสองช่องที่อยู่ตรงส่วนบนของวินโดว์ ซึ่งจะเป็นตัวควบคุมการเลือกออบเจกต์และเหตุการณ์ที่จะเกิดขึ้นกับออบ

เจคนั้น โดยโค้ดที่ปรากฏจะเป็นโปรแกรมหรือคำสั่งที่จะถูกเรียกใช้งานเมื่อมีเหตุการณ์นั้นเกิดขึ้นกับออบเจก



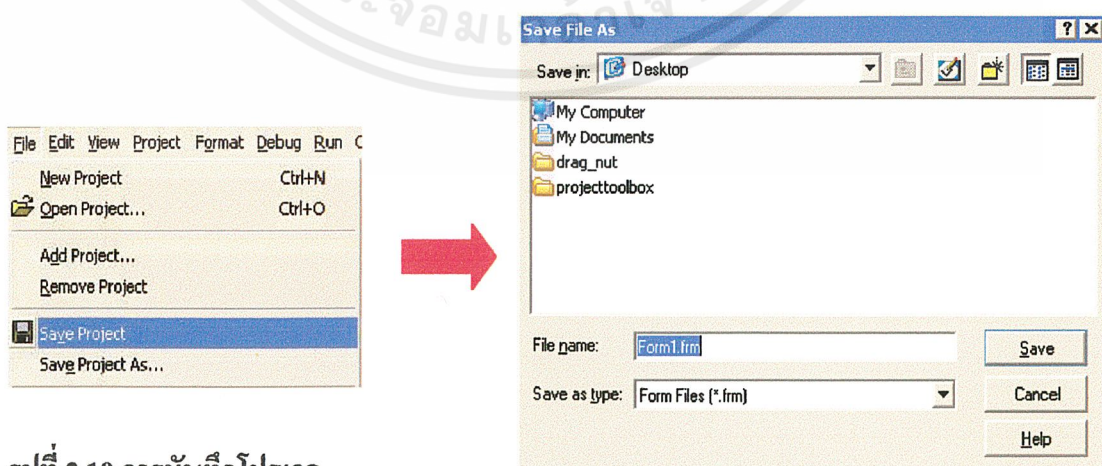
รูปที่ 2.9 วินโดว์ Code Editor

Object list box จะแสดงชื่อออบเจกต์ว่าส่วนของโปรแกรมที่กำลังแสดงอยู่ใน Code editor เป็นของออบเจกต์ใด ถ้าคลิกที่ครอปดาวนั้นจะปรากฏรายการของออบเจกต์ต่างๆที่อยู่ในฟอร์ม เราสามารถทราบออบเจกต์และเหตุการณ์ได้จากชื่อของ Procedure เช่น Command1_click จะหมายถึงส่วนของโปรแกรมที่จะทำงาน เมื่อออบเจกต์ชื่อ Command1 ถูกคลิกเป็นต้น

2.3 การบันทึกและเปิดโปรเจก

2.3.1 การบันทึกโปรเจก

การบันทึกโปรเจกทำได้โดยเลือกเมนู File > Save Project จากนั้น Visual Basic จะถามว่าต้องการเก็บฟอร์มนี้ไว้ในไฟล์ชื่ออะไร พร้อมกับเลือกตำแหน่งที่อยู่ file ใน harddisk ด้วย



รูปที่ 2.10 การบันทึกโปรเจก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

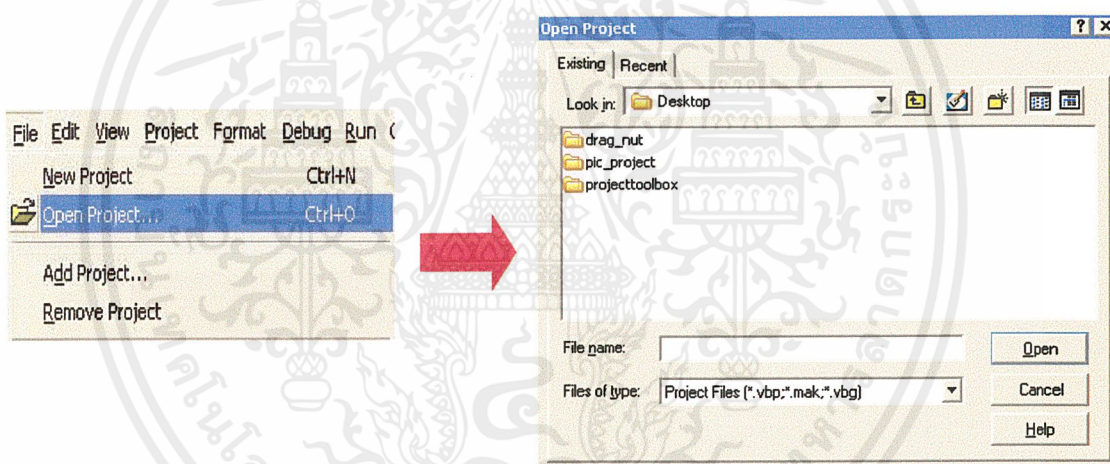
เมื่อจัดเก็บฟอร์มเรียบร้อยแล้ว Visual Basic จะแสดงไดอะล็อกบ็อกซ์ Save อีกครั้ง ซึ่งคราวนี้จะเก็บเป็น file .vbp ซึ่งเป็นข้อมูลต่างๆ ของโปรเจกต์เรา ให้เลือกตำแหน่งที่จัดเก็บและตั้งชื่อ file ในทำนองเดียวกัน ก็เป็นอันเสร็จสิ้นการบันทึกโปรเจกต์

2.3.2 การเปลี่ยนชื่อไฟล์

การบันทึก File เป็นชื่อใหม่ทำได้โดยใช้เมนู file > Save Project As ซึ่งจะใช้ในกรณีที่เราต้องการเปลี่ยนชื่อไฟล์หรือย้ายตำแหน่งของโปรเจกต์ไฟล์ อย่างไรก็ตามถ้าต้องการย้ายตำแหน่งจะต้องจัดเก็บฟอร์มโดยใช้เมนู File > Save Form As เพื่อเก็บฟอร์มไปไว้ยังตำแหน่งใหม่ด้วยจึงจะครบถ้วน

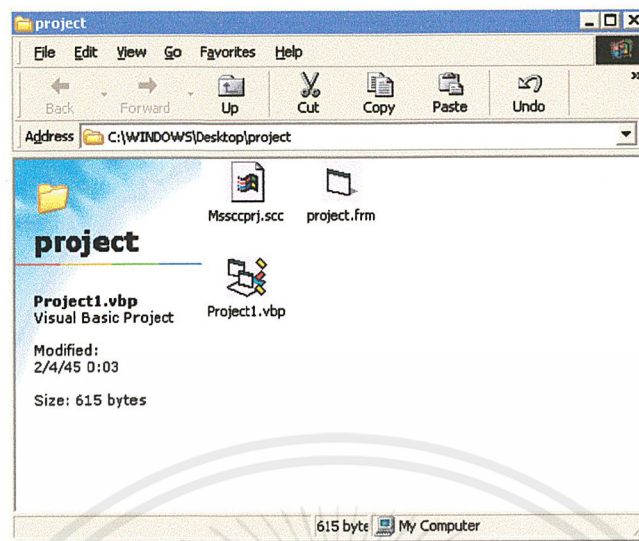
2.3.3 การเปิดโปรเจกต์

ทำได้โดยเลือก File > Open Project จากนั้นเลือกชื่อโปรเจกต์ไฟล์ที่ลงท้ายด้วย .vbp ที่ต้องการแล้วคลิกปุ่ม Open



รูปที่ 2.11 การเปิดโปรเจกต์วิธีที่ 1

หรือต้องการเปิดจากโฟลเดอร์ใน Windows ก็สามารถดับเบิลคลิกที่ไอคอนของโปรเจกต์ได้ทันที

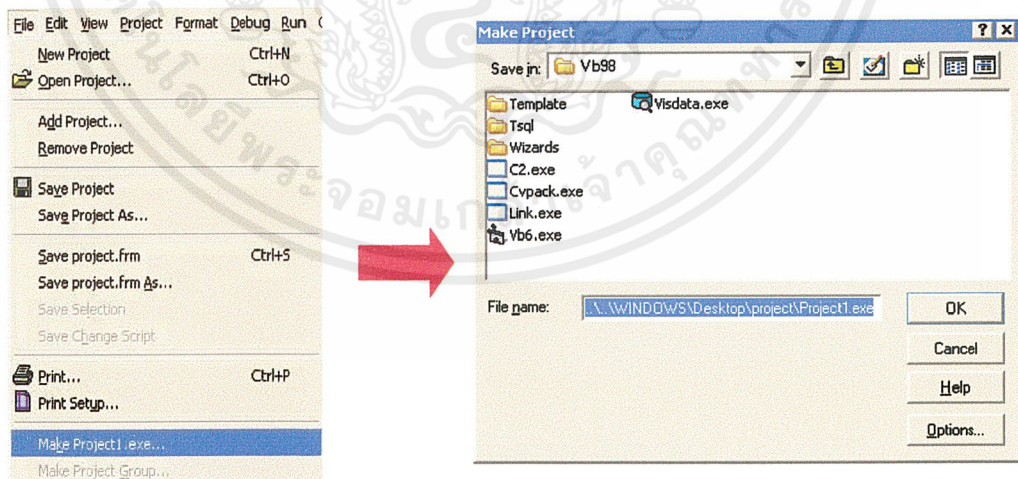


รูปที่ 2.12 การเปิดโปรเจกต์วิธีที่ 2

2.4 การคอมไพล์

จะเป็นการรวบรวมไฟล์ .vbp และไฟล์ส่วนประกอบต่างๆ ของโปรเจกต์ เมื่อเราคอมไพล์ Visual Basic จะแปลงและรวบรวมส่วนประกอบต่างๆ ให้ออกมาเป็นไฟล์ที่สามารถใช้งานได้โดยไม่ต้องผ่านโปรแกรม Visual Basic อีกซึ่งไฟล์ดังกล่าวเรียกว่า Executable File หรือไฟล์ .exe การคอมไพล์ใน Visual Basic โดยเลือกเมนู

File > Make Project.exe



รูปที่ 2.13 การคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคอมไพล์เสร็จเรียบร้อยแล้วก็จะได้ไฟล์ .exe ที่สามารถเรียกใช้งานได้ทันทีโดยไม่ต้องเรียก Visual Basic

2.5 ความแตกต่างระหว่างการบันทึกต่างกับการคอมไพล์

2.5.1 การบันทึก

ทำเพื่อจัดเก็บโปรเจกของเราไว้สำหรับการนำมาแก้ไขได้ในภายหลัง ซึ่งเป็นการจัดเก็บไฟล์โปรเจก .vbp ฟอรัม .frm และประกอบต่างๆ ซึ่งเรียกรวมกันว่าเป็น Source Code หรือโปรแกรมต้นฉบับ

2.5.2 การคอมไพล์

ทำเพื่อแปลงโปรเจกโดยรวมรวมส่วนประกอบต่างๆ ออกมาเป็นไฟล์ที่สามารถทำงานได้โดยโดยไม่ต้องเรียกจาก Visual Basic ซึ่งเรียกว่า Executable File ที่พร้อมสำหรับจัดส่งให้กับผู้ใช้งานโปรแกรมของเรา ผู้ที่ได้ Executable File นั้นจะไม่สามารถทำการแก้ไขเพิ่มเติมโปรแกรมของเราได้ แต่จะสามารถใช้งานโปรแกรมได้โดยไม่ต้องมี Visual Basic

2.6 Compiler และ Interpreter

โปรแกรมภาษาจะสามารถแบ่งได้เป็นสองประเภทคือ Interpreter และ Compiler ภาษาที่เป็น Compiler ตัวโปรแกรมภาษาจะตรวจสอบไวยากรณ์และแปลงโปรแกรมที่ผู้ใช้เขียนให้เป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจหรือที่เรียกว่า “Native Code” โดย Compiler จะแปลงโปรแกรมทั้งหมดเป็น Native Code ในขณะที่ผู้ใช้ส่งคอมไพล์ จากนั้นเราจึงจะสามารถนำ Native Code นี้ไปใช้งานได้ แต่สำหรับโปรแกรมภาษาที่เป็น Interpreter โดยทั่วไป เช่น ภาษา Basic โปรแกรมจะมีการตรวจสอบไวยากรณ์ทีละบรรทัดในขณะที่ผู้ใช้เขียน โดยเวลาที่สั่ง Run โปรแกรมจึงจะมีการตีความคำสั่งโดยไม่ต้องแปลงโปรแกรมให้เป็นภาษาเครื่อง การตีความและประมวลผลจะเข้าไปทีละบรรทัด ข้อดีของ Interpreter คือโปรแกรมเมอร์สามารถแก้ไขโปรแกรมและรันโปรแกรมต่อไปได้ทันที แต่ก็มีข้อเสียคือ การตีความและทำงาน ไปทีละบรรทัดจะทำให้การทำงานช้าลง

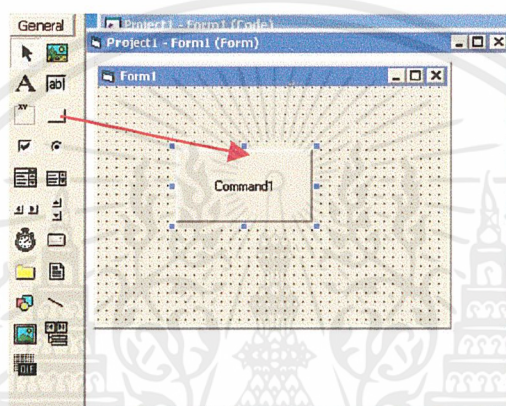
สำหรับภาษา Visual Basic นั้นเรียกว่าเป็น Pseudocompiled Language กล่าวคือทั้ง Interpreter และ Compiler โดยตัว Visual Basic จะมีการตรวจสอบไวยากรณ์โดยตลอด ในขณะที่เขียนโปรแกรมและไม่ได้แปลงเป็น Native Code ที่คอมพิวเตอร์เข้าใจได้เต็มที่เหมือนกับคอมไพเลอร์แท้ๆ แต่เรายังคงต้องคอมไพล์โปรแกรมที่เขียนเสร็จแล้วให้เป็น P-Code ซึ่งเป็น code ที่มีการแปลงมาขั้นหนึ่งแล้ว เพื่อให้สามารถทำงานได้เร็วขึ้น สำหรับใน Visual Basic6 สามารถกำหนดได้เลยว่าต้องการคอมไพล์ให้เป็น Native Code เพื่อให้ได้โปรแกรมทำงานได้เร็วที่สุด วิธีการก็คือเลือกเมนู Project > Project Properties แล้วเลือกออกป๊อปอัพ Compile to native Code จาก แท็บ

Compile อย่างไรก็ตามการเรียกใช้ไฟล์ Native Code ของ Visual Basic ที่คอมไพล์แล้วนี้จะไม่
สามารถเรียกใช้ได้เลยทีเดียว แต่ยังต้องการเพิ่มข้อมูลที่เป็น Library บางเพิ่มอยู่

2.7 ขั้นตอนการเขียนโปรแกรมด้วย Visual Basic

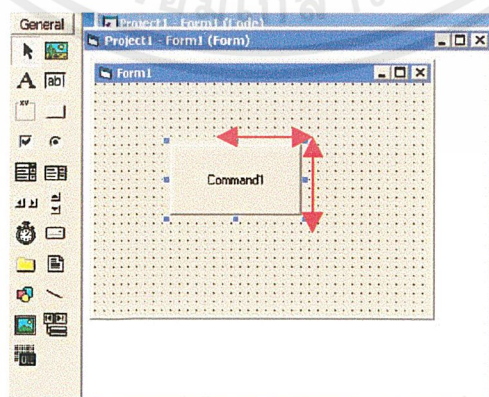
2.7.1 ออกแบบหน้าต่างของโปรแกรมที่เราต้องการ โดยนำ Control Object ต่างๆ ที่อยู่ใน
Toolbox มาปะไว้ที่ฟอร์ม ซึ่งมีวิธีควบคุมออบเจกต์ต่างๆ

เลือกและวางออบเจกต์ คลิกเมาส์ที่ออบเจกต์ที่ต้องการใน Toolbox แล้วนำเมาส์มายังตำแหน่งที่
ต้องการวางออบเจกต์นั้นบนฟอร์ม คลิกเมาส์ซ้ายค้างไว้และลากเมาส์จนได้ขนาดที่ต้องการ



รูปที่ 2.14 การเลือกและวางออบเจกต์

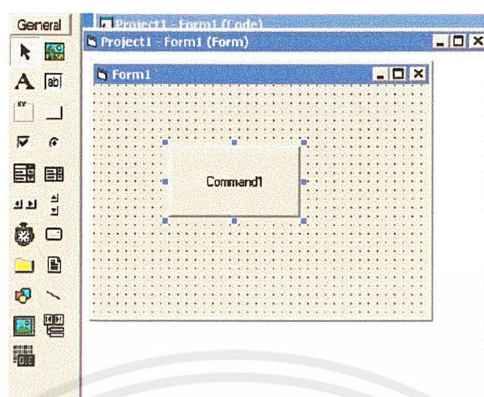
การจัดตำแหน่งและปรับขนาดออบเจกต์ ทำได้โดยเปลี่ยนเครื่องมือเลือกให้เป็น Pointer รูปลูก
ศรแล้วจึงนำเมาส์ไปคลิกและลากที่ออบเจกต์ที่เราเลื่อนเพื่อจัดให้อยู่ในตำแหน่งที่ต้องการ สำหรับ
การปรับขนาดจะทำโดยคลิกเลือกที่ออบเจกต์ จากนั้นนำเมาส์ไปที่ขอบของออบเจกต์ เราจะสามารถ
คลิกและลากเพื่อเปลี่ยนขนาด ได้ทั้งสี่ด้านของออบเจกต์



รูปที่ 2.15 การจัดตำแหน่งและปรับขนาดออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 การลบออบเจกต์ ออกจากฟอร์ม ทำโดยคลิกเลือกออบเจกต์ที่ต้องการลบ จากนั้นกดปุ่ม Delete หรือคลิกเมาส์ขวาเลือกเมนู Delete



รูปที่ 2.16 การลบออบเจกต์

2.8 ชนิดของตัวแปรใน Visual Basic

ตัวแปรในภาษาทางคอมพิวเตอร์นั้นมีอยู่มากมายหลายชนิดตามลักษณะการใช้งาน โดยตัวแปรแต่ละชนิดก็จะมีช่วงการเก็บค่าหรือประเภทของค่าที่เก็บ รวมทั้งเนื้อที่ที่ใช้ในหน่วยความจำในการเก็บค่าแตกต่างกันไป

| ชนิด | รายละเอียด | หน่วยความจำ |
|----------|---|--------------------------|
| Byte | มีค่าตั้งแต่ 0 ถึง 255 | 1 |
| Integer | ใช้เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ -32768 ถึง 32767 | 2 |
| Long | ใช้เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ -2147483648 ถึง 2147483647 | 4 |
| Single | ใช้เก็บเลขจำนวนจริงมีจุดทศนิยม | 4 |
| Double | ใช้เก็บเลขจำนวนจริงมีจุดทศนิยม | 8 |
| Currency | ใช้เก็บเลขทศนิยมตำแหน่งจำกัดมีค่าตั้งแต่ -922337203658477.5808 ถึง 922337203658477.5808 | 8 |
| Boolean | ใช้เก็บค่าทางตรรกะ คือ True และ False | 2 |
| String | ใช้เก็บกลุ่มของตัวอักษรหรือชุดของข้อความ | ขึ้นกับความยาวของข้อความ |
| Date | ใช้เก็บตัวแปรที่เป็นเวลาหรือวันที่ | 8 |
| Object | ใช้เก็บตัวแปรที่อ้างถึงออบเจกต์ใดๆ | 4 |
| Variant | เป็นตัวแปรพิเศษของ Visual Basic สามารถใช้เก็บตัวแปรได้ทุกแบบตามข้างต้น | 22 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าตัวแปร Variant สามารถจัดเก็บตัวแปรได้ทุกประเภท แล้วทำไมต้องมีตัวแปรชนิดอื่นให้วุ่นวายด้วยแต่ถ้าสังเกตดีๆ จะพบว่าตัวแปรแบบ Variant นั้นต้องการเนื้อที่ต่อหนึ่งตัวถึง 22 byte ซึ่ง Integer ใช้ 2 byte นอกจากนี้ การตั้งตัวแปรไม่ตรงชนิดยังมีผลต่อความเร็วของการประมวลผลที่ลดลงอีกด้วย

2.9 การประกาศตัวแปรและกฎการตั้งชื่อตัวแปร

การประกาศตัวแปร คือ การที่เราบอกเครื่องคอมพิวเตอร์ให้เตรียมเนื้อที่ในหน่วยความจำสำหรับเก็บตัวแปรที่เราจะใช้ในการประมวลผลในโปรแกรม โดยการประกาศตัวแปรจะใช้คำสั่งว่า Dim (Dimension) ซึ่งมีไวยากรณ์ดังนี้

รูปแบบ Dim ชื่อตัวแปร [As ชนิดของตัวแปร]

ค่าที่อยู่ใน [] จะเป็น Optional คือใส่หรือไม่ใส่ก็ได้ ถ้าไม่ได้ใส่จะเป็นค่าดีฟอลต์ที่กำหนดไว้คือ Variant

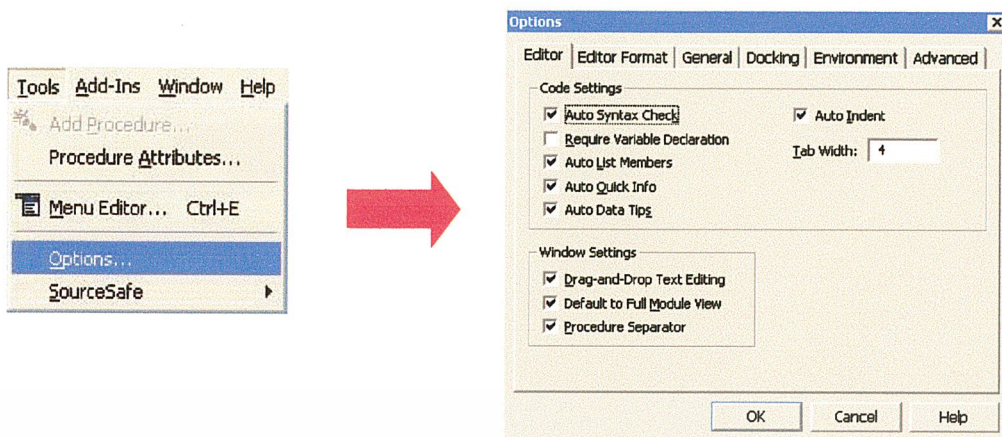
ส่วนการตั้งชื่อตัวแปรมีกฎเกณฑ์ดังนี้

- ชื่อตัวแปรต้องขึ้นต้นด้วยตัวอักษร
- ชื่อตัวแปรห้ามยาวเกิน 255 ตัวอักษร
- สามารถใช้ตัวอักษรตัวเลขหรือเครื่องหมาย _ ผสมเป็นชื่อแต่ห้ามเป็น . - หรือเครื่องหมายอื่นๆในชื่อ
- ห้ามซ้ำกับคำที่สงวนไว้ในโปรแกรม เช่น Dim เป็นต้น
- ชื่อตัวแปรห้ามซ้ำกันถ้าอยู่ในขอบเขตเดียวกัน

2.10 จุดประสงค์ของการประกาศตัวแปรและคำสั่ง Option Explicit

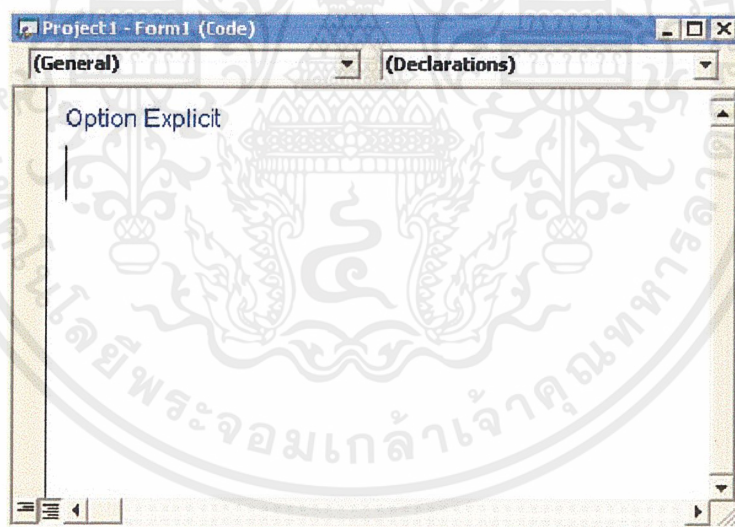
การประกาศตัวแปรเป็นการบอกให้เครื่องทราบ เพื่อเตรียมเนื้อที่ในหน่วยความจำสำหรับจัดเก็บข้อมูลต่างๆ ที่เราต้องการในการคำนวณประมวลผล ส่วนคำว่า Dim ที่เราใช้ประกาศตัวแปรนั้นมาจากคำว่า Dimension ซึ่งคิดค้นมาจากในอดีตที่การประกาศตัวแปรเราต้องกำหนดขนาดของตัวแปรให้เครื่องทราบ ไม่ได้หมายความว่าตัวแปรมีค่านับแต่อย่างใด

การใช้คำสั่ง Option explicit เพื่อบอกว่าตัวแปรต่างๆที่ใช้ในโปรแกรมของเรานั้นต้องมีการประกาศก่อนถึงใช้งานได้โดยเราสามารถกำหนดให้ Visual Basic ใส่คำสั่งนี้ให้เราโดยอัตโนมัติด้วยการเลือกเมนู Tools > Options ดังนี้



รูปที่ 2.17 การใช้คำสั่ง Option Explicit

เมื่อเลือก Variable Declaration แล้วออกจากโปรแกรม Visual Basic และกลับเข้ามาอีกครั้งให้ลองเข้าไปที่วินโดว์ Code editor จากนั้นเลือกเมนู View > Code ก็จะพบคำสั่ง Option Explicit ประกาศไว้อัตโนมัติ



รูปที่ 2.18 แสดงผลการเลือกใช้คำสั่ง Option Explicit

คำสั่ง Option Explicit จะเป็นการบอกว่าการใช้ตัวแปรใดในโปรแกรมจะต้องมีการประกาศอย่างชัดเจนด้วยคำสั่ง Dim

2.11 ตัวแปรแบบค่าคงที่

สมมติว่าเขียนโปรแกรมบัญชี โดยมีโปรแกรมเป็นพันๆ บรรทัด และมีการคำนวณค่าที่ใช้ค่าภาษีมูลค่าเพิ่มซึ่งปัจจุบันมีค่า 7% ในการคำนวณประมาณ 200 แห่งในจุดต่างๆ ของโปรแกรม ในอนาคตรัฐบาลอาจจะประกาศออกมาใหม่ว่าจะปรับอัตรา VAT เป็น 10% จำนวน 200 กว่าจุดในโปรแกรมของเราอีกจากนี้ถ้าเราตามหาไม่ครบก็หมายถึงข้อผิดพลาดต่างๆ ในการคำนวณที่จะตามมาในภายหลัง จากปัญหาดังกล่าวนักคอมพิวเตอร์จึงคิดชนิดของตัวแปรแบบคงที่ขึ้นมา โดยมีรูปแบบการประกาศค่าตัวแปรดังนี้

รูปแบบ Constant ชื่อตัวแปร = ค่าที่ต้องการ

ในกรณีค่าภาษีมูลค่าเพิ่มเราเพียงกำหนดว่า Constant vat = 7%

จากนั้นเราก็นำตัวแปรชื่อ Vat ไปใช้ในจุดต่างๆ ของโปรแกรมที่มีค่าภาษีมูลค่าเพิ่มเข้าไปเกี่ยวข้อง ทีนี้เมื่อเราต้องการเปลี่ยนอัตราภาษีมูลค่าเพิ่มก็เพียงเปลี่ยนค่าบรรทัดแรกจากค่า 7% เป็น 10% เท่านั้นไม่ต้องเข้าไปแก้ไขส่วนอื่นๆ ของโปรแกรมเลย

2.12 โปรแกรมย่อย

ในการเขียนโปรแกรมนั้น มักมีการแบ่งโปรแกรมออกเป็นส่วนย่อยๆ เพื่อสะดวกต่อการควบคุมการทำงานและจัดการ โปรแกรมให้ง่ายขึ้น โปรแกรมย่อยๆ เหล่านี้เรียกว่า Sub Procedure หรือ โปรซีเจอร์ย่อย สมมติว่ามีโค้ดที่เหมือนกัน 3 จุด ทำให้เปลืองเนื้อที่ และถ้าเกิดมีการแก้ไขโปรแกรมในส่วนนี้เราต้องแก้ 3 ที่ด้วยกัน วิธีการที่ดีกว่านั้นคือแยกโปรแกรมที่ซ้ำๆ กันนี้ออกมาเป็นโปรซีเจอร์ย่อยจะเห็นว่าโปรแกรมของเราจะสั้นลงมาก นอกจากนั้นยังสะดวกต่อการจัดการ การแก้ไข และดูแลโปรแกรมเข้าใจได้ง่ายกว่าเดิม

2.13 ฟังก์ชัน

มีลักษณะเดียวกับโปรซีเจอร์ แต่จะมีข้อแตกต่างกันตรงที่ฟังก์ชันมีการคืนค่ากลับไปให้โปรแกรมที่เรียกใช้ด้วย วิธีใช้ก็เพียงแต่ใช้คำสั่ง Function แทนคำว่า Sub และมีการกำหนดชนิดของค่าที่คืนกลับมาต่อท้ายฟังก์ชันเหมือนกันกับตัวแปร เช่น

Function CalculateInterest () As Currency

ในฟังก์ชันจะส่งค่ากลับไปโดยการกำหนดค่านั้นให้กับชื่อฟังก์ชันเสมือนกับว่าชื่อฟังก์ชันเป็นตัวแปรตัวหนึ่ง และในการเรียกใช้ฟังก์ชันก็สามารถใช้ในสูตรคำนวณหรือนำไปกำหนดค่าให้กับตัวแปรอื่นๆ ก็ได้

2.14 Event Procedure

Event Procedure คือ โปรแกรมย่อยชนิดหนึ่ง ที่ทำงานก็ต่อเมื่อมีเหตุการณ์เกิดขึ้นบนจอภาพ โดยใน Event Procedure จะเก็บโค้ดของแต่ละเหตุการณ์สำหรับแต่ละออบเจกต์ตามหลักการเขียนโปรแกรมแบบ Event-driven เช่น โพรซีเจอร์ย่อยชื่อ Command1_Click() เป็น Event Procedure ซึ่งเก็บโค้ดที่ทำงานก็ต่อเมื่อมีการกดปุ่ม Command1 หรือโพรซีเจอร์ ชื่อ Form_Load () จะเป็น Event Procedure ซึ่งเก็บโค้ดที่ทำงานก็ต่อเมื่อมีการเปิดฟอร์มครั้งแรก เป็นต้น

การเข้าถึง Event Procedure นั้นทำได้โดยการเลือก Object list box และ Event list box ของ Code Editor เมื่อเราเลือกออบเจกต์และอีเวนต์ตามต้องการแล้ว หน้าจอของ Code Editor ก็จะเปลี่ยนไปเป็น Sub Procedure ของเหตุการณ์และออบเจกต์นั้น

2.15 ขอบเขตของตัวแปร (Scope of variable)

ในการตั้งตัวแปรนั้นตัวแปรแต่ละตัวจะมีขอบเขตในการมองเห็นและใช้งานแตกต่างกันขึ้นอยู่กับว่าเราต้องการให้โปรแกรมส่วนใดมองเห็นตัวแปรนั้นได้บ้าง

ตัวแปร **Public** มองเห็นและเรียกใช้ได้ในทุกฟอร์มโมดูล หรือทุกโพรซีเจอร์ การประกาศทำได้โดยใช้ คำว่า Public แทนคำว่า Dim ในบรรทัดต่อจาก Option Explicit นอกโพรซีเจอร์ใดๆ

2.15.1 ตัวแปร **Private** มองเห็นและเรียกใช้ได้เฉพาะทุกโพรซีเจอร์ใน โมดูลหรือฟอร์มที่ประกาศเท่านั้น อาจเรียกได้ว่าเป็นตัวแปร Module Level Variable การประกาศทำได้โดยใช้คำว่า Private แทนคำว่า Dim ใน บรรทัดต่อจาก Option explicit

2.15.2 ตัวแปร **Local** มองเห็นและเรียกใช้ได้เฉพาะ โพรซีเจอร์ที่ประกาศเท่านั้น อาจเรียกได้ว่าเป็น Local Variable ดังนั้นตัวแปรชื่อเดียวกันที่อยู่ในคนละ โพรซีเจอร์จะเป็นตัวแปรคนละตัวกัน ตัวแปรเหล่านี้จะ ถูกสร้างขึ้นเฉพาะเมื่อเข้ามาทำงานใน โพรซีเจอร์นี้เท่านั้น และเมื่อออกไปจาก โพรซีเจอร์นี้ก็จะถูกทำลายทิ้งไป

2.16 ตัวแปรแบบ Static

ในบางครั้งหากเราต้องการให้ตัวแปรของโพรซีเจอร์มีค่าคงอยู่แม้ว่าโปรแกรมจะทำงานไปจนจบโปรแกรมย่อยนั้นแล้วสามารถทำได้โดยประกาศตัวแปรนั้นให้เป็นตัวแปรแบบ Static ซึ่งตัวแปรแบบนี้จะต่างจากตัวแปรทั่วไปคือ มันจะเก็บค่าที่อยู่ในตัวแปรไว้ตราบใดที่โปรแกรมยังทำงานอยู่ โดยไม่ทำลายทิ้งหรือสร้างขึ้นใหม่ทุกครั้งที่จบโพรซีเจอร์หรือฟังก์ชัน แต่ยังคงใช้ได้เฉพาะในของเขตโพรซีเจอร์หรือฟังก์ชันนี้เท่านั้น กล่าวอีกนัยหนึ่งคือ ตัวแปรแบบ Static ก็คือตัวแปรแบบ

local ในโปรแกรมเมอร์แต่ทำงานเหมือนกับตัวแปร Private ที่อยู่ภายนอกซึ่งจะไม่ลืมค่าเมื่อถึงคำสั่ง End Sub นั้นเองและยังคงใช้ได้เฉพาะ โน โปรแกรมเมอร์ประกาศเท่านั้น

2.17 ตัวดำเนินการและนิพจน์

ตัวดำเนินการ (Operator) หมายถึงการคำนวณหรือการทำฟังก์ชันทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร โดยเรียกประโยคที่มีการนำตัวดำเนินการมาใช้กับตัวแปรหรือตัวเลขว่าเป็น "นิพจน์" หรือ Expression

สำหรับทางคอมพิวเตอร์จะแบ่งตัวดำเนินการออกเป็น 3 กลุ่ม คือ ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic), ตัวดำเนินการเปรียบเทียบ (Relational) และตัวดำเนินการตรรก (Logical) ซึ่งมีรายละเอียดดังนี้

2.17.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator) เป็นตัวดำเนินการที่ใช้คำนวณทางคณิตศาสตร์ เช่น + - * / และยกกำลัง

2.17.2 ตัวดำเนินการเปรียบเทียบ (Relational Operator) หรือเรียกอีกอย่างว่า Comparison Operator เป็นตัวดำเนินการที่ใช้ประเมินค่าข้อมูล ผลลัพธ์ที่ได้จะมีค่าเป็นจริงหรือเท็จเท่านั้น

| ตัวดำเนินการ | ความหมาย | ตัวอย่างนิพจน์ |
|--------------|---------------------|----------------|
| = | เท่ากับ | A = B |
| > | มากกว่า | A > B |
| < | น้อยกว่า | A < B |
| <> | ไม่เท่ากับ | A <> B |
| >= | มากกว่าหรือเท่ากับ | A >= B |
| <= | น้อยกว่าหรือเท่ากับ | A <= B |

2.17.3 ตัวดำเนินการทางตรรก (Logical Operator) จะเป็นตัวดำเนินการที่ให้ค่าเป็นจริงหรือเท็จ โดยจะมีค่าการกระทำดังนี้

And เทียบได้กับคำว่า "และ"

| นิพจน์1 | นิพจน์2 | นิพจน์1 and นิพจน์2 |
|---------|---------|---------------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Or เทียบได้กับคำว่า "หรือ"

| นิพจน์1 | นิพจน์2 | นิพจน์1 or นิพจน์2 |
|---------|---------|--------------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Xor หรือ Exclusive OR จะมีค่าเป็นจริงต่อเมื่อประโยคทั้งสองมีค่าตรรกะต่างกัน ถ้าเหมือนกันจะเป็นเท็จ

| นิพจน์1 | นิพจน์2 | นิพจน์1 xor นิพจน์2 |
|---------|---------|---------------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Not เป็นการสลับค่าตรรกะของตัวแปรหรือนิพจน์ จากจริงเป็นเท็จหรือจากเท็จเป็นจริง

| นิพจน์1 | not นิพจน์ |
|---------|------------|
| 0 | 1 |
| 1 | 0 |

Eqv มาจาก Equivalence จะเป็นจริงต่อเมื่อค่าตัวแปรที่นำมา Eqv กันมีค่าตรรกะเหมือนกัน

| นิพจน์1 | นิพจน์2 | นิพจน์1 eqv นิพจน์2 |
|---------|---------|---------------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Imp มาจากคำว่า Implication หรือถ้า...แล้ว ค่าผลลัพธ์จะเป็นจริงเมื่อ นิพจน์1 จริง และนิพจน์2 เป็นเท็จเท่านั้น กรณีอื่นจะเป็นจริงหมด

| นิพจน์1 | นิพจน์2 | นิพจน์1 imp นิพจน์2 |
|---------|---------|---------------------|
|---------|---------|---------------------|

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

เราอาจนำตัวดำเนินการต่างๆ มาใช้ในการสร้างเงื่อนไข เพื่อตรวจสอบค้นหาข้อมูลหรือประมวลผล

2.18 ลำดับการประมวลผลของตัวดำเนินการ

นิพจน์หนึ่งเขียนว่า $3 + 4 * 12$ ถามว่าคำตอบของนิพจน์นี้คืออะไร ถ้าคุณก่อนก็จะได้คำตอบคือ 51 แต่ถ้าบวกก่อนจะได้คำตอบเป็น 84 ถ้าคอมพิวเตอร์ไม่กำหนดเป็นมาตรฐานที่แน่นอนไว้ว่าจะประมวลผลตัวดำเนินการใดก่อนหลังจากปัญหาดังกล่าวจึงมีการกำหนดว่า ถ้าในนิพจน์หนึ่งมีการคำนวณหลายตัวการคำนวณจะมีลำดับการประมวลผลดังนี้

| ลำดับ | ก่อน | | | | | |
|-------|----------------------|--------------------------|---------|--|--|--|
| | Arithmetic | Relational | Logical | | | |
| | ยกกำลัง(^) | เท่ากับ (=) | not | | | |
| | ทำเป็นลบ(-) | ไม่เท่ากับ (<>) | and | | | |
| | คูณและหาร (*, /) | น้อยกว่า (<) | or | | | |
| | หารจำนวนเต็ม (\) | มากกว่า (>) | xor | | | |
| | หาเศษการหาร (mod) | น้อยกว่าหรือเท่ากับ (<=) | eqv | | | |
| | บวกและลบ (+, -) | มากกว่าหรือเท่ากับ (>=) | imp | | | |
| | เชื่อมตัวข้อความ (&) | Like, Is | | | | |
| | หลัง | | | | | |

จะเห็นว่าถ้านิพจน์มีตัวดำเนินการหลายแบบปนกัน จะประมวลผลจากซ้ายไปขวาโดยเริ่มจากตัวดำเนินการ Arithmetic ก่อนต่อมาจึงทำ Relational และจากนั้นจึงทำ Logical เป็นอันดับสุดท้าย แต่ถ้าจะเปลี่ยนลำดับการประมวลผลก็ให้ใส่เครื่องหมาย() โดยจะทำการประมวลผลนิพจน์ที่อยู่ในวงเล็บในสุดก่อนแล้วไล่ออกมาตามลำดับ

2.19 ประโยคคำสั่ง

Statement หมายถึงประโยคคำสั่งในโปรแกรมที่เราเขียนขึ้นใช้ควบคุมการทำงานของโปรแกรม โดยแบ่งได้ 3 ประเภท

2.19.1 ประโยคกำหนดค่า (Assignment Statement) คือ การเขียนคำสั่งที่ป้อนค่าหรือใส่ค่าให้กับตัวแปรด้วยนิพจน์ต่างๆ ตัวดำเนินการหลักที่ใช้ในประโยคกำหนดค่าของ Visual Basic คือ เครื่องหมาย =

2.19.2 ประโยคแบบเงื่อนไข (Condition Statement) เป็นประโยคคำสั่งที่ใช้ในการตรวจสอบค่า ตัดสินใจ

เปรียบเทียบ และเลือกประมวลผลส่วนของโปรแกรมที่เราต้องการ คำสั่งประเภทนี้ได้แก่ IF-THEN IF-THEN-ELSE IF-THEN-ELSEIF และ SELECT-CASE

2.19.3 ประโยคแบบวนซ้ำ (Iteration Statement) เป็นประโยคคำสั่งที่ทำให้คอมพิวเตอร์ทำส่วนของโปรแกรมซ้ำๆ กันหลายหน คำสั่งประเภทนี้ได้แก่ คำสั่ง FOR-NEXT DO-WHILE LOOP DO-UNTIL LOOP

2.20 ประโยคแบบเงื่อนไข (Condition Statement)

จะเป็นประโยคคำสั่งที่ใช้การสร้างเงื่อนไขในโปรแกรมเพื่อให้โปรแกรมไปดำเนินการที่ส่วนของโปรแกรมที่เราต้องการเมื่อเกิดเงื่อนไขที่เรากำหนดไว้ ได้แก่ IF-THEN IF-THEN-ELSE IF-THEN-ELSEIF และ SELECT-CASE

IF-THEN

คำสั่ง IF-THEN เป็นคำสั่งที่ใช้เมื่อต้องการให้คอมพิวเตอร์ตัดสินใจว่าจะไปทำงานในโปรแกรมส่วนใดจากทางเลือกและข้อกำหนดที่เราได้เตรียมไว้ให้ในโปรแกรมรูปแบบจะมีดังนี้

IF (Condition1) THEN Statement1

การทำงานจะตรวจสอบ Condition1 ว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงจะทำงานที่ Statement1 ถ้าเป็นเท็จจะข้าม Statement1 และวิ่งไปทำงานบรรทัดถัดไปแทน หรือ IF (Condition1) THEN Statement1 ELSE Statement2 เพื่อให้ทำงาน Statement2 ในกรณีที่ Condition1 เป็นเท็จด้วยก็ได้

IF-THEN-ELSE-END IF

จะเหมือนกับ IF - THEN - ELSE เพียงแต่สามารถมี Statement ที่จะทำงานมากกว่าหนึ่งบรรทัดได้

(IF-THEN มี Statement ได้เพียงหนึ่งคำสั่งหลัง THEN และหนึ่งคำสั่งหลัง ELSE และจะต้องอยู่ในบรรทัดเดียวกัน) รูปแบบของ IF-THEN-ELSE IF จะเป็นดังนี้

- ถ้า Condition1 เป็นจริงจะวิ่งไปทำงานส่วนของโปรแกรมในบล็อก A จาก Statement1 จนถึงบรรทัด StatementN แล้วจึงทำงานต่อที่คำสั่งหลังบรรทัด END IF

- ถ้า Condition1 เป็นเท็จจะวิ่งข้ามบล็อก A ไปทำงานส่วนของโปรแกรมในบล็อก B จาก StatementA จนถึงบรรทัด StatementX แล้วจึงทำงานต่อหลังบรรทัด END IF คำสั่งนี้จะใช้โดยไม่มีส่วนของ ELSE เพื่อให้ IF-THEN สามารถทำงานหลายๆ คำสั่งได้โดยจะต้องยังคงมี END IF

IF-THEN-ELSEIF

จะมีรูปแบบการใช้งานเหมือนกับ IF-THEN-ELSE-END IF แต่จะสามารถใส่ condition ซ้อนเข้าไปได้อีก ถ้า Condition1 เป็นจริงโปรแกรมจะทำงานที่ Block1 หลังจากทำ Block1 เสร็จแล้วจะไปทำงานบรรทัดหลัง END IF เลย ไม่มีการตรวจสอบ Condition2 อีก ถ้า Condition1 เป็นเท็จโปรแกรมจึงจะไปตรวจสอบ Condition2 ต่อไป

SELECT-CASE

คำสั่ง SELECT-CASE เป็นคำสั่งในการเลือกทำ ซึ่งจะทำงานคล้ายกับ IF-THEN-ELSEIF แต่ SELECT-CASE จะเหมาะสำหรับการเปรียบเทียบค่าตัวแปรเดียว แล้วกระโดดไปทำงานที่ส่วนอื่นของโปรแกรมตามเงื่อนไขของตัวแปรที่ตรวจสอบ

เงื่อนไขที่ใช้ใน Case ถ้าเป็นการเปรียบเทียบตัวแปรที่ Select ว่าเท่ากับเท่าใด ก็สามารถใช้ค่าคงที่หรือตัวเลขได้ในทันที แต่ถ้าต้องการเปรียบเทียบซับซ้อนขึ้นให้ใช้คำว่า IS แทนตัวแปรที่ Select และใช้เงื่อนไขการเปรียบเทียบใดๆ เช่นเดียวกับ IF-THEN-ELSE IF

DO-WHILE-LOOP และ DO-LOOP-WHILE

รูปแบบคำสั่ง DO-WHILE-LOOP มีดังนี้

DO WHILE Condition1

 ชุดคำสั่ง

[EXIT DO]

LOOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งภายในประโยค DO-WHILE จะทำงานซ้ำไปเรื่อยๆ ตราบใดที่ Condition1 เป็นจริง เมื่อใดที่ Condition1 เป็นเท็จจะหลุดออกจากลูปไปทำงานโปรแกรมในบรรทัดถัดจากคำสั่ง LOOP

รูปแบบคำสั่ง DO-LOOP-WHILE

```
DO
    ชุดคำสั่ง
[EXIT DO]
LOOP WHILE Condition1
```

ชุดคำสั่ง DO-LOOP จะทำงานซ้ำไปเรื่อยๆ ตราบใดที่ Condition1 เป็นจริง เมื่อใดที่ Condition1 เป็นเท็จจะหลุดออกจาก LOOP WHILE ไปทำงานโปรแกรมในบรรทัดถัดจากคำสั่ง LOOP WHILE จะเห็นว่าเหมือนกับ DO-WHILE-LOOP แต่ต่างกันที่การทำชุดคำสั่งในครั้งแรกเท่านั้น คือ DO-WHILE-LOOP จะมีการตรวจสอบเงื่อนไข Condition1 ก่อนแล้วจึงทำชุดคำสั่ง แต่ DO-LOOP-WHILE จะทำชุดคำสั่งก่อนเลยแล้วค่อยตรวจสอบเงื่อนไขว่าจะวนลูปหรือไม่

DO-UNTIL-LOOP และ DO-LOOP-UNTIL

คำสั่ง DO-UNTIL จะคล้ายกับ DO-WHILE เพียงแต่จะมีตรรกะในการตัดสินใจการวนลูปกลับกัน โดย DO-UNTIL-LOOP มีรูปแบบดังนี้

```
DO UNTIL Condition1
    ชุดคำสั่ง
[EXIT DO]
LOOP
DO
    ชุดคำสั่ง
[EXIT DO]
LOOP UNTIL
```

สำหรับ DO-UNTIL ชุดคำสั่งจะทำงานซ้ำต่อเมื่อ Condition1 เป็นเท็จ กล่าวคือจะถูกทำงานกระทั่ง Condition1 เป็นจริงแล้วจะเลิกทำ สำหรับ DO-LOOP-UNTIL ก็เหมือนกับ DO-UNTIL-LOOP แต่จะต่างกันที่การตรวจสอบเงื่อนไขในครั้งแรกเท่านั้น

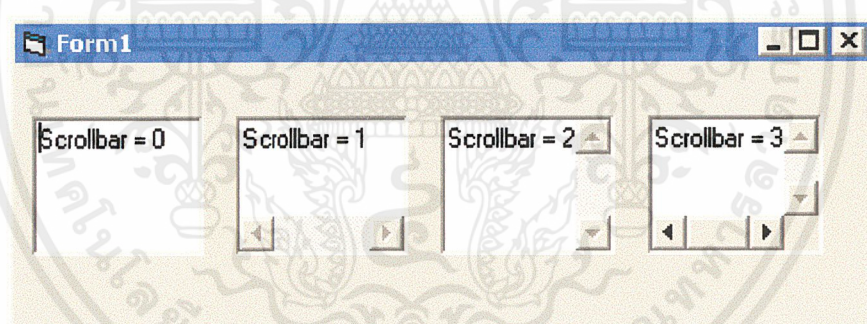
2.21 คอนโทรลหลัก



2.1.1 Text Box

เป็นคอนโทรลที่ใช้สำหรับรับข้อความ ซึ่งข้อความที่รับนี้อาจยาวเกินกว่าจะแสดงในกรอบหรือ Box ให้เห็นพร้อมกันหมดก็ได้ โดยปกติเราจะใช้ properties เหมือนกับคอนโทรลทั่วไป ตามที่ได้กล่าวไปในหัวข้อที่แล้ว แต่ที่สำคัญเพิ่มเติมขึ้นมาคือ Text ซึ่งเป็นที่เก็บข้อความที่ปรากฏอยู่ในกรอบ เมื่อจะเอาข้อความไปใช้ก็จะอ้างถึง Text ซึ่ง Text Box เป็นออบเจกต์พื้นฐานที่สุดในการรับข้อความ

Properties scrollbar ใช้กำหนดว่าจะให้มี scrollbar ของ Text Box ด้วยหรือไม่ โดยกำหนดให้มีได้ทั้งแนวนอน,แนวตั้ง หรือทั้งสองอย่าง ด้วยค่า 1,2,3 ตามลำดับ ส่วน MultiLine ระบุว่ายอมให้มีการพิมพ์ข้อความได้หลายบรรทัดหรือไม่ ปกติแล้วมักจะให้มี scrollbar ในแนวตั้งด้วยเมื่อกำหนดให้ MultiLine เป็นจริง การจะให้เห็นแสดงเส้นกรอบหรือไม่นั้นกำหนดได้ที่ BorderStyle และกำหนดความยาวของข้อความที่จะรับได้ใน Maxlength ซึ่งปกติจะเป็นค่า 0 หมายถึง ให้เก็บได้ยาวที่สุดคือ 32000 ตัวอักษร



รูปที่ 2.19 ตัวอย่างการใช้ Textbox

ถ้าต้องการรับรหัสผ่านหรือ Password ก็ทำได้ง่ายๆ เพียงกำหนดตัวอักษรที่จะแสดงเมื่อพิมพ์ข้อความไว้ในคุณสมบัติ PasswordChar ซึ่งจะทำให้ทุกตัวอักษรที่พิมพ์เข้าไปถูกแสดงใน TextBox ด้วยสัญลักษณ์ที่กำหนดนี้แทนข้อมูลที่เราก็คีย์เข้าไปจริง อย่างไรก็ตามการเก็บข้อความยังคงเป็นข้อความปกติที่เราพิมพ์ หมายถึงเรายังนำคุณสมบัติ Text ไปใช้ได้เหมือนเดิม หากจะยกเลิกก็เพียงให้ PasswordChar เป็น Chr\$(0) หรือเป็นสตริงว่าง (“ ”)

เหตุการณ์ keydown,keyup,keypress จะเกิดขึ้นเมื่อมีการกด,ปล่อยหรือพิมพ์ตัวอักษรด้วยคีย์บอร์ด ซึ่งเราสามารถนำมาใช้จัดการกับการกดตัวอักษรที่สนใจได้ หรือหากต้องการตรวจสอบ

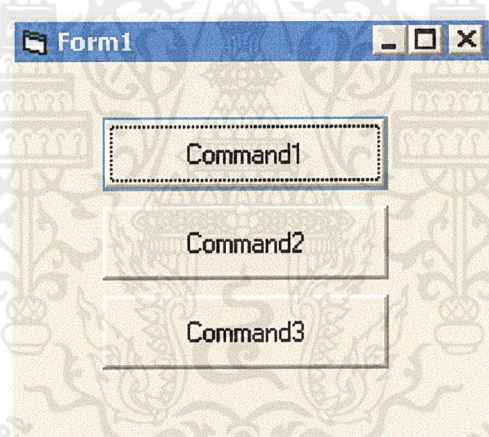
ข้อความใน TextBox อย่างเช่น ต้องการรับค่าในช่วง 1-100 เท่านั้น อาจใช้เหตุการณ์ LostFocus ซึ่งเกิดขึ้นเมื่อเคอร์เซอร์ออกจาก TextBox นี้



1.2 Command Button

คอนโทรลนี้ถือเป็นคอนโทรลพื้นฐานที่ใช้บ่อยที่สุดตัวหนึ่ง คือใช้เป็นปุ่มในการสั่งงานเพราะอย่างน้อยที่สุดเกือบทุกฟอร์มจะมีปุ่ม OK และ Cancel อยู่ด้วย คุณสมบัติที่ใช้บ่อยๆ ได้ถูกกล่าวถึงแล้วในหัวข้อแรกของบทนี้ จึงขอข้ามไปถึงเหตุการณ์ที่เป็นเรื่องเฉพาะของ Command Button เลย

เหตุการณ์หลักของคอนโทรลตัวนี้คือเหตุการณ์ Click ซึ่งเกิดขึ้นเมื่อมีการกดปุ่มนี้ ปกติเราจะเขียนโปรแกรมให้ทำงานเมื่อเกิดเหตุการณ์นี้ขึ้น นอกจากนี้ก็มีเหตุการณ์ KeyUP, KeyDown และ KeyPress เหมือนกัน TextBox และยังมีคุณสมบัติข้อหนึ่งซึ่งมีอยู่เฉพาะใน Command Button คือ Default ที่ระบุว่าปุ่มคีย์ฟอลต์หรือไม่ ปุ่มคีย์ฟอลต์จะเป็นปุ่มที่เมื่อผู้ใช้กด Enter จะเหมือนกับการกดปุ่มนี้ทันทีโดยมากมักจะกำหนดให้ปุ่ม OK เป็นปุ่มคีย์ฟอลต์



รูปที่ 2.20 ตัวอย่างการใช้ Button

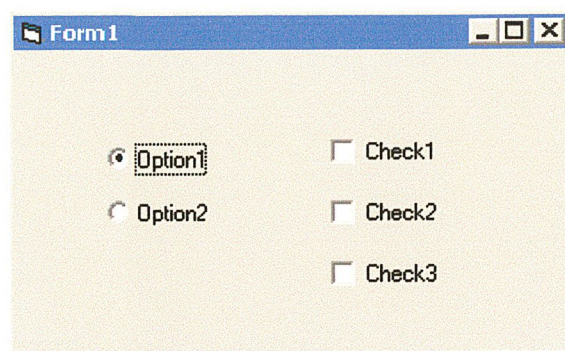


1.3 Option Button และ



Check Box

บางครั้งเรียกว่า Radio Button ซึ่ง Option Button จะใช้กับการเลือกที่ต้องเลือกตัวใดตัวหนึ่งจากกลุ่มของ Option Button ซึ่งมักจะรวมกันอยู่ใน Frame การเลือกปุ่มใดปุ่มหนึ่งจะเป็นการยกเลิกปุ่มอื่นๆ ทั้งหมดโดยอัตโนมัติ



รูปที่ 2.21 ตัวอย่างการใช้ Option Button และ Check Box

ในการตรวจสอบว่ามีการเลือกหรือไม่นั้นจะมีพรอพเพอร์ตี้ Value โดยมีค่าจริง (True) แสดงว่าถูกเลือก แต่ถ้ามีค่าเท็จ (False) ก็คือไม่ถูกเลือก โดยมากเมื่อผู้ใช้เลือกเสร็จแล้วและกดปุ่ม OK ที่เตรียมไว้ เราจึงจะนำค่า Value ของแต่ละคอนโทรลมาตรวจสอบและกำหนดการทำงานอีกต่อหนึ่ง ส่วนพรอพเพอร์ตี้ Alignment เป็นตัวระบุว่า จะแสดงช่องให้เลือกอยู่ทางซ้ายหรือขวาของข้อความที่เป็น Caption

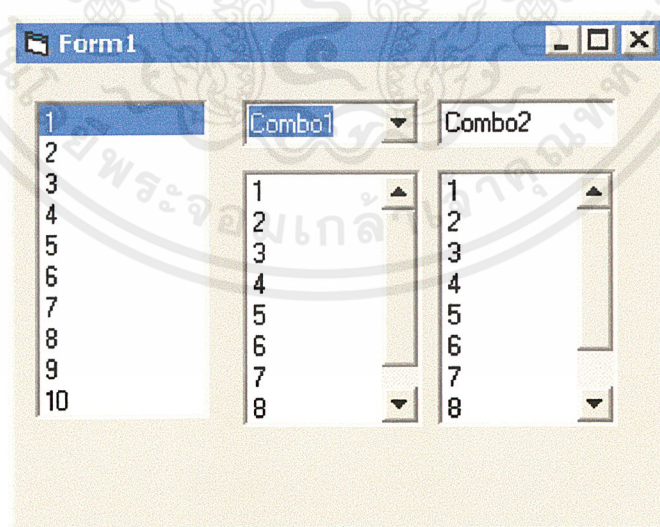


21.4 List Box และ



ComboBox

ListBost จะแสดงรายการให้เลือกแบบตรงไปตรงมา ส่วน Combo Box แสดงได้ 3 รูปแบบขึ้นอยู่กับ พรอพเพอร์ตี้ Style



รูปที่ 2.22 ตัวอย่างการใช้ Listbox และ ComboBox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะใช้ Method เหมือนกันคือ AddItem สำหรับการเพิ่มรายการดังตัวอย่าง

```
ListbBox.AddItem "List Item", 1
```

เป็นการเพิ่มรายการ "List Item" ในคอนโทรลที่ชื่อ ListBox ณ ตำแหน่งที่ 1 ซึ่งตำแหน่งนี้อาจจะไว้ก็ได้ จะถือว่าเป็นการเพิ่มรายการต่อท้าย แต่หากระบุตำแหน่งเป็น 0 จะเป็นการเพิ่มไว้ตอนต้นของรายการ สำหรับการยกเลิกรายการออกใช้ Method RemoveItem เช่น

```
ListBox.RemoveItem 1
```

เพื่อยกเลิกข้อมูลตัวที่ 1 ออกจากรายการ และหากจะยกเลิกทั้งหมดให้ใช้วิธีการ Clear ซึ่งเรียกว่าการเขียนโปรแกรมให้วนรอบยกเลิกข้อมูลทีละตัว

```
ListBox.Clear
```

พรอพเพอร์ตี้ที่สำคัญได้แก่ Text คือข้อมูลจากรายการที่ถูกเลือกอยู่ในขณะนั้น List เป็น array ของสตริงซึ่งเก็บรายการ เพราะข้อความในรายการจะเก็บเป็น array ดังนั้นเราอาจเรียกใช้ข้อมูลแต่ละตัวได้โดยระบุตำแหน่ง เช่น ListBox.List (1) จะได้ข้อมูล ณ ตำแหน่งที่ 1 ListIndex เก็บหมายเลขลำดับของรายการที่ถูกเลือก นั่นคือเมื่อมีการเลือกรายการค่าของ Text จะเท่ากับ List (ListIndex) เสมอ โดยค่า index จะเริ่มจาก 0 นั่นคือ ข้อมูล แรกมี index เป็น 0 ส่วนถ้า ListIndex มีค่า -1 หมายถึงไม่มีการเลือกข้อมูลในรายการ ListCount เก็บค่าจำนวนหัวข้อทั้งหมดในรายการ หากต้องการแสดงรายการหลายๆ คอลัมน์ให้กำหนดจำนวนคอลัมน์ในพรอพเพอร์ตี้ Columns และกำหนดลักษณะการเลือกข้อมูลในรายการด้วยพรอพเพอร์ตี้ MultiSelect ซึ่งค่า 0 จะให้เลือกข้อมูลได้ทีละ

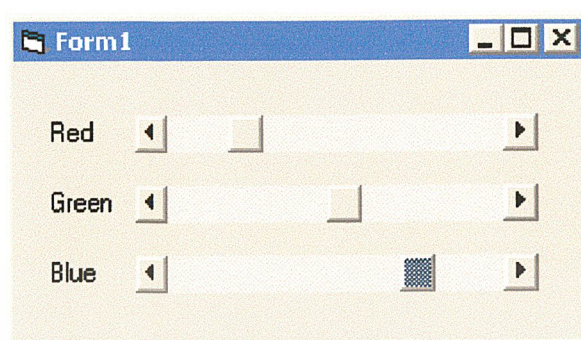
ตัว ค่า 1 เลือกกลุ่มข้อมูลที่ติดกันได้โดยกด Shift + ปุ่มลูกศร หรือการคลิกเลือกด้วยเมาส์ ค่า 2 เลือกข้อมูลได้หลายตัวทั้งที่อยู่ติดกันหรือไม่ติดกัน โดยใช้ปุ่ม Shift หรือ Ctrl ประกอบกับคีย์ลูกศรหรือประกอบกับการเลือกด้วยเมาส์



2.21.5 scrollbar

เป็นแถบเลื่อนที่มีทั้งแนวตั้งและแนวนอน โดยจะให้ค่าตามตำแหน่งของปุ่มเลื่อนหรือ Scroll box ว่าอยู่ตรงแถบไหน พรอพเพอร์ตี้พื้นฐานได้แก่ Min กำหนดค่าตำแหน่งทางซ้ายสุดหรือบนสุด Max กำหนดค่าตำแหน่งทางขวาสุดหรือล่างสุด Value เก็บค่าตำแหน่งปัจจุบันที่ scroll box อยู่ ซึ่งค่านี้จะอยู่ในช่วง Min กับ max เสมอ LargeChange กำหนดค่าตำแหน่งที่จะเปลี่ยนไปเมื่อกดคลิกใน

ช่วงที่อยู่ทางซ้าย/ขวา หรือ บน/ล่างของ scrollbar และ SmallChange กำหนดค่าตำแหน่งที่จะเปลี่ยนไปเมื่อคลิกที่รูปลูกศรซึ่งอยู่ที่ปลายทั้งสองข้าง



รูปที่ 2.23 ตัวอย่างการใช้ Scrollbar



.21.6 Label

ใช้แสดงข้อความโดยไม่ใช้ผู้ใช้แก้ไข การแก้ไขข้อความนั้นทำได้โดยเขียน โปรแกรมกำหนดข้อความใหม่ให้กับพรอพเพอร์ตี้ Caption ส่วนพรอพเพอร์ตี้ที่เกี่ยวกับการแสดงผลอันเป็นหน้าที่ของลาเบลโดยตรง ได้แก่

Alignment เป็นการจัดข้อความให้ชิดซ้าย ชิดขวา หรืออยู่ตรงกลาง โดยค่าเป็น 0, 1, 2 ตามลำดับ

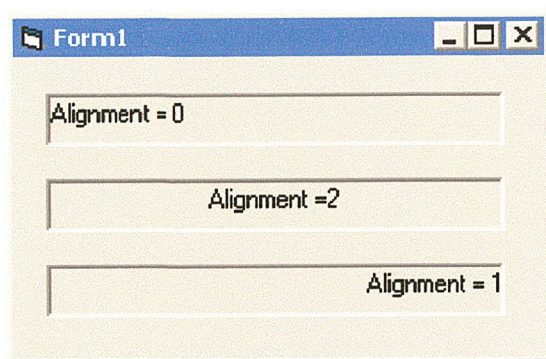
Border Style และ Back Style

- Border Style ใช้กำหนดลักษณะของเส้นกรอบของคอนโทรลและ
- Back Style ใช้กำหนดการแสดงผลแบบโปร่งใสหรือระบายทึบด้วยสี
- Back Color โดยการแสดงผลแบบโปร่งใสนั้นผู้ใช้จะเห็นวัตถุซ้อนอยู่ข้างหลัง

Autosize และ Wordwrap

- Autosize นั้นเมื่อมีค่าเป็น True จะมีการปรับขนาดของคอนโทรลให้เหมาะกับ Caption
- Wordwrap เมื่อมีค่าเป็น True ทำให้มีการตัดคำให้แสดงในบรรทัดถัดไปในกรณีข้อความ

ไม่สามารถแสดงได้หมดในบรรทัดเดียว



รูปที่ 2.24 ลักษณะการใช้งาน Label



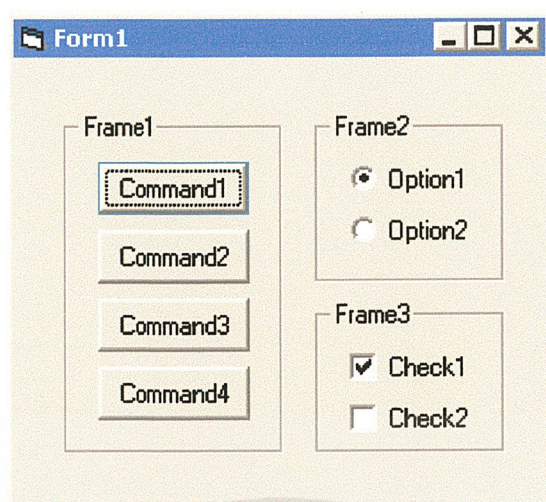
2.21.7 Frame

ในการจับกลุ่มหรือกรุปคอนโทรลต่าง ๆ ในฟอร์มเข้าด้วยกันนั้น เราจะใช้เฟรมเข้ามาช่วย โดยที่การเลื่อนเฟรมไปยังตำแหน่งอื่นในฟอร์ม จะเป็นการเลื่อนคอนโทรลทั้งหมดในเฟรมนั้นด้วย วิธีนี้จะช่วยให้เราสะดวกขึ้นมา เพราะเมื่อวางตำแหน่งคอนโทรลได้ตามต้องการแล้ว การเปลี่ยนตำแหน่งของกลุ่มจะทำได้โดยเคลื่อนย้ายเฟรมแทนที่จะต้องมาย้ายคอนโทรลทีละตัวแล้วมาจัดตำแหน่งใหม่อีกครั้งนอกจากนี้การลบเฟรมทิ้ง จะเป็นการลบคอนโทรลทั้งหมดในเฟรมทิ้งไปด้วย

การสร้างกลุ่มของคอนโทรลด้วยเฟรมนั้นจะต้องสร้างเฟรมขึ้นมาก่อน แล้วจึงสร้างคอนโทรลภายใน โดยจะต้องวางตำแหน่งคอนโทรลไว้ภายในเฟรมตั้งแต่แรก เราไม่สามารถเลื่อนคอนโทรลภายนอกมาไว้ข้างในเฟรมได้ เพราะ Visual Basic จะถือว่าเป็นการนำคอนโทรลมาซ้อนกันเท่านั้น ไม่ได้ได้อยู่ภายในเฟรมจริงๆ ดังนั้นการวางคอนโทรลในเฟรมจึงต้องคลิกเลือกที่ Toolbox ก่อนแล้วจึงนำมาลากกำหนดขนาดในเฟรม

การเลือกขอบเขตหลายๆ ตัวในเฟรมทำได้โดยกดปุ่ม Ctrl ค้างไว้แล้วจึงคลิกเลือกขอบเขตแต่ละตัวหรือจะใช้วิธีกดปุ่ม Ctrl ค้างไว้พร้อมกับลาก mouse เพื่อกำหนดพื้นที่ที่จะเลือกขอบเขตก็ได้ ที่ต้องใช้วิธีกดปุ่ม Ctrl ค้างไว้ด้วยเมื่อต้องการเลือกขอบเขตก็เนื่องมาจาก เป็นการทำให้ Visual Basic รู้ว่าเป็นการเลือก มิฉะนั้น Visual Basic จะแยกความแตกต่างไม่ออกระหว่างการเลือกกับการเลื่อนตำแหน่งของเฟรม เพราะการเลื่อนตำแหน่งใช้วิธีคลิกค้างในบริเวณเฟรม แล้วลากเมาส์ เหมือนกับการเลื่อนขอบเขตอื่นๆ ตามปกติ

เฟรมมีคุณสมบัติเช่นเดียวกับคุณสมบัติพื้นฐานต่างๆ ไปของขอบเขตอื่นๆ เช่น ความกว้าง ความสูงและมีการกำหนด Font รวมไปถึง Caption ซึ่งโดยมากจะไม่ค่อยมีการเปลี่ยนแปลงมากนัก เพราะเรามักจะสร้างไว้ใช้จัดกลุ่มของคอนโทรลเสียมากกว่า



รูปที่ 2.25 การใช้ Frame

2.22 คำสั่ง MsgBox

เป็นคำสั่งที่ช่วยให้เราสร้างกรอบข้อความโต้ตอบกับผู้ใช้ได้โดยง่าย ซึ่งมีรูปแบบของคำสั่งดังนี้

```
MsgBox(Prompt[, buttons] [, title])
```

Prompt เป็นข้อความที่ไม่ต้องการให้ปรากฏในวินโดว์ MsgBox
Buttons ใช้ควบคุมปุ่มและภาพไอคอนที่ปรากฏบนวินโดว์
Title ใช้กำหนดข้อความที่แถบด้านขอบของวินโดว์

ค่าคงที่ในการกำหนดไอคอนในวินโดว์ของ MsgBox

ค่า VbCritical

ไอคอนที่ปรากฏคือ



ค่า VbExclamation

ไอคอนที่ปรากฏคือ



ค่า VbInformation

ไอคอนที่ปรากฏคือ



ค่า VbQuestion

ไอคอนที่ปรากฏคือ



ค่าคงที่ในการกำหนดปุ่มควบคุมในวินโดว์ของ MsgBox

| | | |
|------------------------|-----------------|--|
| ค่า vbOKOnly | ปุ่มที่ปรากฏคือ |  |
| ค่า vbOkCancel | ปุ่มที่ปรากฏคือ |  |
| ค่า vbYesNo | ปุ่มที่ปรากฏคือ |  |
| ค่า vbYesNoCancel | ปุ่มที่ปรากฏคือ |  |
| ค่า vbAbortRetryIgnore | ปุ่มที่ปรากฏคือ |  |
| ค่า vbRetryCancel | ปุ่มที่ปรากฏคือ |  |

ค่าคงที่ในการกำหนดปุ่มดีฟอลต์

| | |
|-----------------------|---------------------------------------|
| ค่า vbDefaultButton 1 | เลือกปุ่มที่ 1 จากซ้ายเป็นปุ่มดีฟอลต์ |
| ค่า vbDefaultButton 2 | เลือกปุ่มที่ 2 จากซ้ายเป็นปุ่มดีฟอลต์ |
| ค่า vbDefaultButton 3 | เลือกปุ่มที่ 3 จากซ้ายเป็นปุ่มดีฟอลต์ |
| ค่า vbDefaultButton 4 | เลือกปุ่มที่ 4 จากซ้ายเป็นปุ่มดีฟอลต์ |

2.22.1 MsgBox ที่เป็นฟังก์ชัน

ในกรณีที่เราต้องทราบคำตอบรับจากผู้ใช้ว่า มีการกดปุ่ม Yes, No, OK หรือปุ่มใดๆ เราจะต้องใช้คำสั่ง MsgBox ที่เป็นฟังก์ชัน โดยที่พารามิเตอร์ (prompt, title, button) จะเหมือนเดิมทุกประการเพียงแต่ค่าพารามิเตอร์ต่างๆ จะอยู่เรียงกันภายในเครื่องหมายวงเล็บ () และต้องมีการเตรียมตัวแปรเพื่อรับค่าที่ส่งกลับมาจากฟังก์ชัน MsgBox ด้วย

โดยค่าตัวแปรที่ส่งคืนกลับมานี้สามารถตรวจสอบได้โดยใช้ค่าคงที่จากตาราง 2.1 ลองพิจารณาตัวอย่างด้านล่าง ซึ่งจะเป็นโปรแกรมที่ใช้ตรวจสอบว่าผู้ใช้กดปุ่ม Yes หรือ No โดยมีเงื่อนไขถ้ากด Yes ให้แสดงวินโดว์ MsgBox ว่ากดปุ่ม Yes แต่ถ้ากด No ก็ให้แสดงวินโดว์ MsgBox ว่ากดปุ่ม No

```
Response = MsgBox ("Are you sure?", vbQuestion + vbYesno, _
    "Please answer :)")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Response = vbYes Then      ‘ตรวจสอบว่าค่า Response ที่ส่งกลับมาจาก
                               ‘MsgBox ฟังก์ชันเท่ากับ vbYes หรือไม่
    MsgBox “You pressed ‘Yes” ‘ถ้าเท่าให้แสดงวิน โคว์ว่า You pressed “Yes”
Else
    MsgBox “You pressed ‘No” ‘ถ้าไม่เท่าให้แสดงวิน โคว์ว่า You pressed “no”
End If

```

ตาราง 2.1 ค่าคงที่รับกลับมาจากฟังก์ชัน MsgBox

| ค่าคงที่ | คำอธิบาย |
|----------|----------------------|
| vbOK | มีการกดปุ่ม “OK” |
| vbCancel | มีการกดปุ่ม “Cancel” |
| vbAbort | มีการกดปุ่ม “Abort” |
| vbRetry | มีการกดปุ่ม “Retry” |
| vbYes | มีการกดปุ่ม “Yes” |
| vbNo | มีการกดปุ่ม “No” |

2.23 Microsoft Common Dialog Control

Microsoft Common Dialog Control จะเป็น ActiveX Control ที่เราสามารถเพิ่มเข้าไปใน Project เพื่อให้โปรแกรมของเราสามารถเรียกใช้ไดอะล็อกบ็อกซ์มาตรฐานต่างๆ เช่น การจัดเก็บเพิ่มข้อมูล การเปิดเพิ่มข้อมูล การเลือกฟอนต์ การเลือกสี หรือการกำหนดสำหรับการพิมพ์ได้อย่างง่ายดาย ซึ่งวิน โคว์แบบต่างๆ Common Dialog Control ได้แก่



ไดอะล็อกบ็อกซ์ Open (เป็นวิน โคว์สำหรับเปิดเพิ่มข้อมูล)

ไดอะล็อกบ็อกซ์ Save (เป็นวิน โคว์สำหรับจัดเก็บเพิ่มข้อมูล)

ไดอะล็อกบ็อกซ์ Color (เป็นวิน โคว์สำหรับเลือกสี)

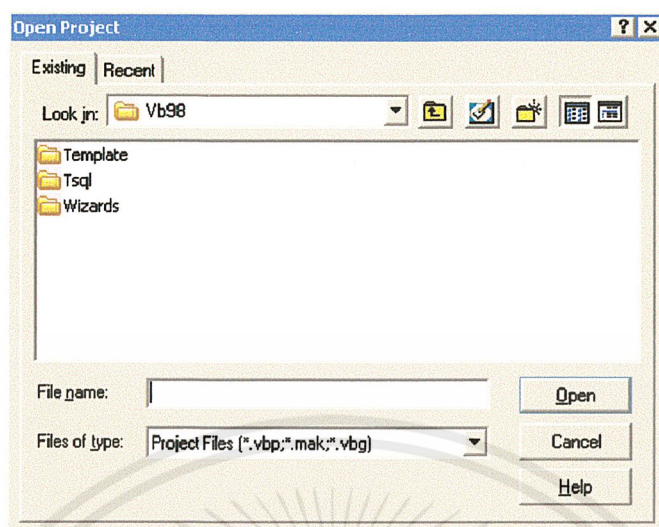
ไดอะล็อกบ็อกซ์ Font (เป็นวิน โคว์สำหรับฟอนต์)

ไดอะล็อกบ็อกซ์ Print (เป็นวิน โคว์สำหรับพิมพ์)

ไดอะล็อกบ็อกซ์ Help (เป็นวิน โคว์สำหรับเปิดไฟล์ช่วยเหลือ)

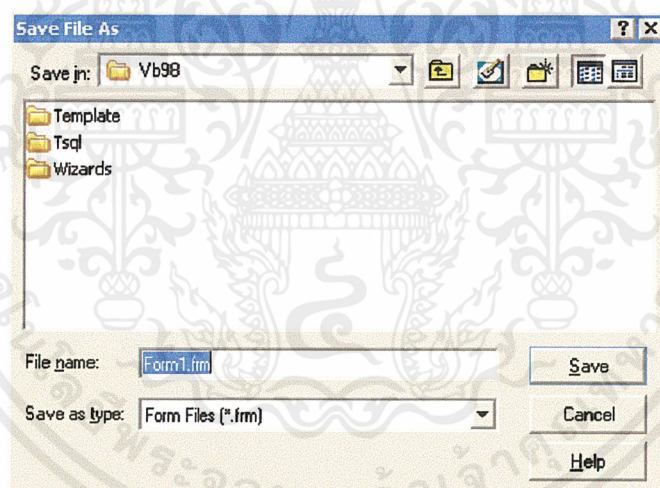
โดยตัว Common dialog Control จะมีเมธอดในการเรียกใช้งานดังต่อไปนี้

2.23.1 เมธอด ShowOpen ใช้แสดงไดอะล็อกบ็อกซ์ Open



รูปที่ 2.26 เมธอด ShowOpen

2.23.2 เมธอด ShowSave ใช้แสดงไดอะล็อกบ็อกซ์ Save



รูปที่ 2.27 เมธอด ShowSave

2.24 การนำ Common Dialog Control มาใช้งานในโปรเจ็ค

การนำ Common Dialog Control มาใช้งานใน โปรเจ็ค มีขั้นตอนดังนี้

1. คลิกเมาส์ขวาที่ Toolbox เลือกเมนู Component ซึ่งจะปรากฏวินโดว์ Component ที่สามารถเพิ่มเข้าไปในโปรเจ็ค ได้ ให้คลิกเช็คบ็อกซ์ Microsoft Common Dialog Control แล้วกดปุ่ม Ok
2. ให้นำคอนโทรลที่ได้นี้มาวางลงบนฟอร์มเหมือนกับการใช้งานคอนโทรลอื่นๆตามปกติ ก็เป็นอันเสร็จสิ้นขั้นตอนการเพิ่มคอนโทรลเข้าไปในโปรเจ็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.25 การเคลื่อนย้ายวัตถุโดยใช้เมธอด Move

ในการเคลื่อนย้ายออบเจกต์ใดๆ ไปยังตำแหน่งต่างๆ บนฟอร์ม เราสามารถทำได้โดยใช้เมธอด Move ซึ่งรูปแบบคำสั่งคือ Object.Move X,Y โดย X,Y จะเป็นจุดโคออร์ดิเนตบนฟอร์มที่ต้องการเคลื่อนวัตถุไปวาง เช่น imgSun.Move 1000,1000 จะเลื่อนภาพ imgSun ไปที่จุด 1000,1000 บนฟอร์ม โดยจะใช้จุดมุมซ้ายบนของออบเจกต์เป็นจุดอ้างอิง

2.26 การ Drag และ Drop

Drag หรือ ลาก หมายถึงการที่ผู้ใช้กดปุ่มซ้ายค้างไว้บนไอคอนหรือออบเจกต์ใดๆ พร้อมกับเลื่อนเมาส์ไปด้วย ส่วนคำว่า Drop หรือ ปล่อย จะหมายถึงการที่ผู้ใช้ปล่อยปุ่มซ้ายหลังจากการ Drag หรือ ลากมานั่นเอง

2.27 พรอปเพอร์ตี้ DragMode

กำหนดให้เกิดการ Drag โดยอัตโนมัติหรือไม่ หากให้ค่าเป็น 1 (automatic) การกดปุ่มเมาส์เมื่อตัวชี้อยู่บนออบเจกต์นั้นจะถือว่าเริ่มต้นการ Drag หากให้เป็น 0 (manual) จะต้องอาศัยเมธอด Drag ในการสั่งลาก คือ ต้องเขียนการควบคุม Drag ใน Event Procedure ขึ้นมาเอง

2.28 พรอปเพอร์ตี้ DragIcon

ใช้ระบุไอคอนที่จะแสดงในระหว่างการลากอาจใช้วิธีก๊อปปี้ไอคอนจากคุณสมบัติ DragIcon ของออบเจกต์อื่น, ก๊อปปี้ภาพ หรือโหลดภาพจากแฟ้มข้อมูล ดังตัวอย่างเหล่านี้

```
Obj1.DragIcon – Obj2.DragIcon
```

```
Obj1.DragIcon – Obj2.Picture
```

```
Obj1.DragIcon – LoadPicture (“c:\windows\desktop\DRAG.ICO”)
```

2.29 เหตุการณ์ DragDrop

เหตุการณ์นี้เกิดขึ้นกับออบเจกต์ปลายทางที่เป็นเป้าหมาย ไม่ได้เกิดขึ้นกับออบเจกต์ที่ถูกลาก เมื่อมีการลากออบเจกต์มาปล่อยลงในออบเจกต์ใด จะเกิดเหตุการณ์นี้ขึ้นกับออบเจกต์นั้น พารามิเตอร์มี 3 ตัว คือ

```
Object_DragDrop (Source As Control, X as Single, Y As Single)
```

ตัว Source นั้นถูกกำหนดให้เป็นคอนโทรล นั้นหมายถึงว่าเราสามารถอ้างอิงถึงคุณสมบัติของออบเจกต์นั้นได้เต็มที่ภายในโค้ดหรือแม้แต่จะเรียกใช้เมธอดของออบเจกต์นั้นก็ได้อีก ส่วน X และ Y นั้นเป็นตำแหน่งของตัวชี้ในออบเจกต์ที่เป็นเป้าหมายหรือออบเจกต์ที่ถูกนำออบเจกต์อื่นมาปล่อยลงบนมัน

2.30 เหตุการณ์ DragOver

เกิดขึ้นกับออบเจกต์ที่อยู่ใต้เมาส์เมื่อมีการลากออบเจกต์อื่นผ่าน ตัวอย่างเช่น ใน File Manager ขณะที่เราลากชื่อไฟล์ไปยังตำแหน่งต่างๆ จะพบว่าในบางตำแหน่งที่เราปล่อยไฟล์ลงไปไม่ได้ซึ่งแสดงไอคอนด้วยรูปเครื่องหมายห้ามโดยที่ยังไม่ได้ปล่อยปุ่มเมาส์เลย ออบเจกต์ที่สนับสนุน Drag and Drop จะใช้เหตุการณ์นี้ในการทำงาน พารามิเตอร์ของเหตุการณ์นี้จะเหมือนกับ DragDrop แต่จะเพิ่ม State ซึ่งเป็นแบบ Integer ขึ้นมาอีก 1 ตัว สำหรับบอกว่าตัวชี้ของเมาส์กำลังผ่านเข้ามาในออบเจกต์ที่รับรู้เหตุการณ์นี้ (มีค่า =0) ตัวชี้ของเมาส์เพิ่งจะออกจากหรือผ่านจากออบเจกต์นี้ (มีค่า =1) หรือออบเจกต์ที่อยู่ใต้เมาส์ คือ ตัวออบเจกต์ที่ถูกลากมาเอง (มีค่า =2)

เหตุการณ์ DragDrop และ DragOver จะเกิดขึ้นกับ ออบเจกต์ที่อยู่ใต้เมาส์และเป็นเป้าหมายที่จะปล่อยเท่านั้น ไม่ใช่ออบเจกต์ที่กำลังถูกลาก ดังนั้นออบเจกต์เป้าหมายใดที่สนับสนุน Drag and Drop ก็ต้องเขียน โปรแกรมเพื่อนทำงานตามเหตุการณ์ 2 เหตุการณ์ นี้ด้วย

2.31 เมธอด Drag

จะใช้เมื่อกำหนด DragMode เป็น Manual ซึ่งทำให้สามารถควบคุมได้ว่าจะให้ตอนไหนลาก ตอนไหนลากไม่ได้ อย่างไรก็ตามแม้ว่าจะให้ DragMode เป็นแบบอัตโนมัติก็สามารถใช้เมธอดนี้ได้ พารามิเตอร์ของเมธอดนี้สามารถมีค่าได้เป็น

- 0 หรือ vbCancel หมายถึง ยกเลิกการลาก
- 1 หรือ vbBeginDrag หมายถึง เริ่มต้นการลาก
- 2 หรือ vbEndDrag หมายถึง ปล่อยหรือ Drop

ผลของการตั้งด้วยค่า 0 และ 2 จะคล้ายกัน เว้นแต่ว่าการยกเลิก (ค่า 0) จะไม่ก่อนให้เกิดเหตุการณ์ DragDrop เหมือนการปล่อย

ในการลากและปล่อยนั้นตำแหน่งของออบเจกต์ที่ถูกลากจะไม่เปลี่ยนแปลง เพราะบางงานเราต้องการเพียงสั่งให้ทำงานเท่านั้น เช่น ลากไฟล์ไปยังไอคอนเครื่องพิมพ์เพื่อสั่งพิมพ์ ดังนั้นหากเราต้องการเปลี่ยนตำแหน่งของออบเจกต์ที่ถูกลากให้เลื่อนตามไปด้วยก็ต้องสั่งเองด้วยเมธอด Move ดังตัวอย่างนี้

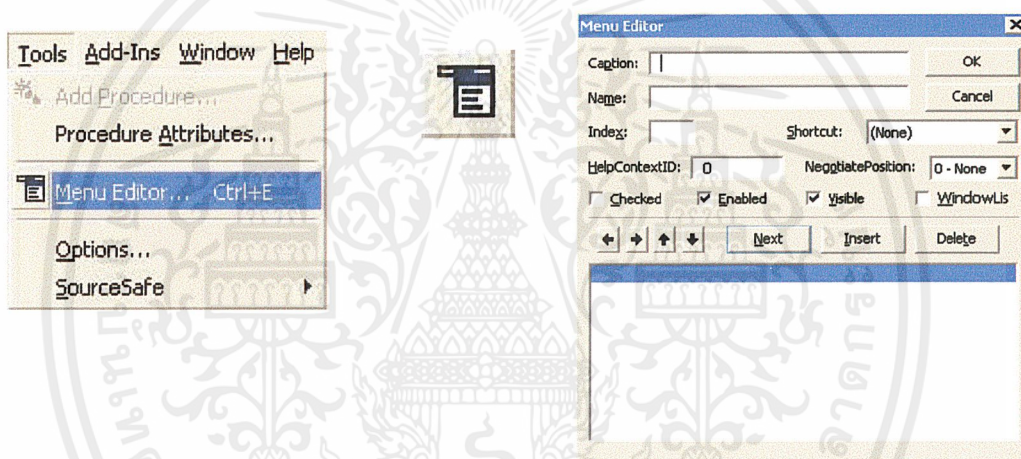
```
Sub Form_DragDrop (Source As Control, X As Single, Y As Single)
    Source.Move X, Y          'เลื่อน Source ออบเจกต์ไปยังตำแหน่ง X,Y
End Sub
```

2.32 การออกแบบเมนู

เมนู คือที่เก็บรวบรวมคำสั่งแทนทั้งหมดของแอปพลิเคชันเอาไว้เป็นกลุ่มๆ เพื่อความสะดวกในการใช้งาน ด้วยความสำคัญเช่นนี้ Visual Basic จึงเตรียมเครื่องมือในการออกแบบและสร้างเมนูให้กับผู้ใช้คือ Menu Editor ซึ่งทำให้สามารถออกแบบเมนูได้ง่ายดาย

2.33 การเรียกใช้และใช้งาน Menu Editor

การเรียกใช้ Menu Editor เพื่อสร้างเมนูใน Visual Basic ทำได้โดยไปที่เมนู Tools แล้วเลือก Menu Editor หรือคลิกที่ Menu Editor บนทูลบาร์ ซึ่งจะปรากฏวินโดว์ ดังในรูป ในวินโดว์นี้จะแบ่งออกเป็น 2 ส่วน โดยส่วนบนเป็นรายละเอียดของค่าที่กำหนดให้กับแต่ละเมนูที่มีแถบสว่าง (Highlight) อยู่ในส่วนล่าง



รูปที่ 2.28 Menu Editor

การกำหนดรายละเอียดของแต่ละหัวข้อในส่วนบน ประกอบด้วยช่องต่างๆ ที่เป็นคุณสมบัติของเมนูคุณสมบัติที่สำคัญที่สุดคือ Name และ Caption การป้อนข้อมูลในส่วนบนจะได้เป็นโครงสร้างของเมนูดังที่แสดงอยู่ในส่วนล่าง โดยมีการจัดย่อหน้าให้เห็นว่าหัวข้อใดอยู่ในเมนูใด หรือหากเราเลื่อนแถบสว่างในส่วนล่าง ก็จะมีรายละเอียดของหัวข้อนั้นปรากฏในส่วนบน นั่นคือ 2 ส่วนนี้จะสัมพันธ์กันเสมอ คุณสมบัติที่กำหนดได้ใน Menu Editor จะมีดังต่อไปนี้

Caption

จะเป็นข้อความที่ปรากฏในเมนู โดยสามารถใช้ & เพื่อสร้างชอคต์ของคีย์ได้เหมือนกับในพรอพเพอร์ตี้ Caption ของ Command button หรือออบเจค อื่นๆ เช่นถ้า Caption กำหนดเป็น &File จะสามารถใช้ Alt+F แทนการคลิกเมาส์ที่เมนูนี้ได้ นอกจากนั้นใน

| | |
|-----------------------|---|
| | กรณีที่ต้องการสร้างเส้นคั่นแบ่งเมนูเราจะใช้เครื่องหมาย - ในพรอพเพอร์ตี้ Caption นี้เพื่อนำทำให้เกิดเส้นแบ่งขึ้น |
| Name | เป็นชื่อของหัวข้อเมนูนั้นมีไว้เพื่ออ้างอิงถึงทุกหัวข้อ ซึ่งจะต้องมีชื่อที่ไม่ซ้ำกันเลย ยกเว้นแต่จะเป็นเมนูใน Control Array |
| Index | เป็นดัชนีสำหรับเมนูที่เป็น Control Array ซึ่งเป็นกลุ่มของเมนูที่ใช้ชื่อและ Event Procedure ร่วมกันตัวอย่างการใช้ Control Array ได้แก่ เมนูที่เพิ่มหรือลดตัวเลือกได้ |
| Shortcut | ส่วนนี้จะทำให้เราสามารถกำหนดคีย์ลัดให้กับเมนู ซึ่งเป็นคีย์พิเศษที่จะเรียกใช้งานหัวข้อนั้น ได้ทันทีโดยไม่ต้องเลือกที่เมนู ตัวอย่างเช่น การกด Ctrl+V จะเป็นการเลือกหัวข้อ Paste ในเมนู Edit เป็นต้นซึ่งช่องกำหนดขดคัทนี่จะเป็น List box ที่มีรายการคีย์ต่างๆให้เลือก ถ้าเป็น (none) ก็คือยังไม่ได้กำหนด |
| Window List | ใช้เมื่อมีการสร้างวินโดว์ย่อยในวินโดว์หลัก Multiple Document Interface โดยขณะใช้งานหากคลิกตรงช่องนี้ในเมนูก็จะมีรายการของวินโดว์ย่อยที่เปิดอยู่ปรากฏให้เห็น |
| Help ContextId | ใช้ระบุหมายเลขของข้อความอธิบายการใช้ ซึ่งจะสัมพันธ์กับหมายเลขหัวข้อในไฟล์ help ข้อความนี้จะถูกแสดงเมื่อเลื่อนแถบสว่างมาที่หัวข้อเมนูนี้แล้วกด F1 สำหรับการสร้างไฟล์ help นั้นจะต้องอาศัย Help Compiler ที่มีมากับชุด Professional Edition ของ Visual Basic |
| Enabled | กำหนดให้เป็นหัวข้อที่เลือกใช้ได้ ซึ่งหากไม่เลือกในช่องนี้หรือกำหนดค่าให้เป็น False เมนูนี้จะแสดงด้วยสีที่ต่างออกไปจากปกติหรือเป็นตัวจางๆ ขณะที่หัวข้อปรากฏขึ้นเพื่อแสดงว่าไม่ให้ใช้ |
| Checked | กำหนดให้มีเครื่องหมายถูกข้างหน้าหัวข้อสำหรับแสดงให้เห็นว่าขณะนี้ตัวเลือกอยู่ในสถานะถูกเลือก |
| Visible | ให้แสดงหัวข้อนี้ในเมนู หรือกล่าวอีกนัยหนึ่งคือให้ผู้ใช้งานมองเห็นหัวข้อนี้ |

โดยปกติ Visual Basic จะกำหนดให้ทุกตัวเลือกในเมนูเป็น Enabled และ Visible คือ แสดงในเมนูและเลือกได้ อย่างไรก็ตามการกำหนด 3 คุณสมบัติสุดท้ายที่กล่าวมานั้นเป็นการให้ค่าเริ่มต้นกับหัวข้อนั้นๆ ซึ่งเราสามารถเขียนโปรแกรมเพื่อเปลี่ยนคุณสมบัติเหล่านี้ได้ในขณะที่โปรแกรมทำงาน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

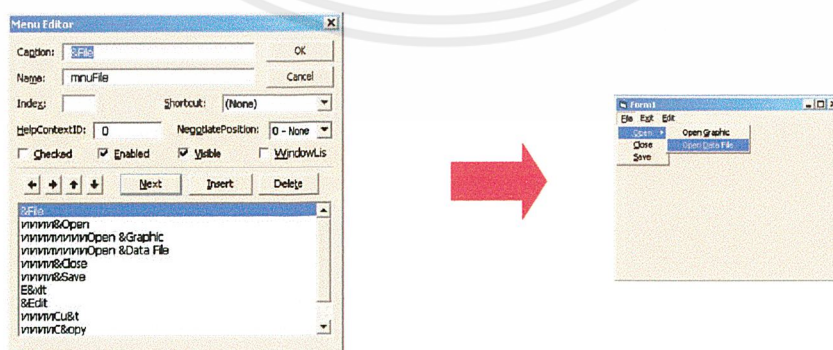
สำหรับส่วนล่างของ Menu Editor จะแสดงโครงสร้างของข้อมูลตามลำดับชั้น โดยเมนูย่อยของเมนูใดก็จะถูกย่อหน้าหรือแสดงเหลี่ยมเข้าไปจากเมนูนั้น เมนูย่อยแต่ละระดับจะถูกนำหน้าด้วยเครื่องหมาย 4 จุด (....) เพื่อแสดงให้เห็นว่าเป็นเมนูย่อย การกำหนดเมนูทำได้ถึง 6 ระดับ และผู้ใช้เพียงจัดย่อหน้าเพื่อบอกถึงระดับเท่านั้นส่วนที่เหลือเป็นหน้าที่ของ Visual Basic

วินโดว์ในส่วนล่างนี้มีไว้แสดงโครงสร้างของเมนูเท่านั้น รายละเอียดจะอยู่ทางส่วนบน ยกเว้นแต่การจัดหัวข้อจะทำผ่านทางปุ่มของส่วนล่างได้แก่ปุ่มรูปลูกศรต่างๆ เช่น ปุ่ม และ ไว้สำหรับจัดย่อหน้าของหัวข้อ ส่วนปุ่มลูกศรขึ้น และลูกศรลง สำหรับการเคลื่อนย้ายหัวข้อไปอยู่ในตำแหน่งอื่น ส่วนปุ่ม Next, ปุ่ม Insert และปุ่ม Delete มีไว้สำหรับเลื่อนแถบสว่างไปยังหัวข้อถัดไป, แทรก และลบหัวข้อตามลำดับ โปรดสังเกตว่าปุ่ม Next เป็นปุ่มที่เป็นดีฟอลต์ แสดงว่าสามารถกด Enter เพื่อไปยังรายการต่อไปได้

หลังจากกำหนดองค์ประกอบต่างๆ ของเมนูเสร็จแล้วก็ให้เลือกปุ่ม OK จะเห็นเมนูที่กำหนดไปอยู่ในฟอร์มซึ่งขณะนี้เราสามารถเลือกหัวข้อต่างๆ ในเมนูได้ แต่การเลือกหัวข้อนั้นๆ จะเป็นการ “ปะ” โปรแกรมที่จะทำงานเมื่อผู้ใช้เลือกหัวข้อนั้นในขณะที่ฟอร์มนี้กำลังทำงานอยู่จริง โดยเมื่อเลือกหัวข้อใดก็จะมีวินโดว์ Code Editor สำหรับ ป้อนโปรแกรมขึ้นมาจะเห็นว่าหลักการนี้เป็นหลักการันเดียวกันกับที่ใช้กับออบเจกต์อื่นในฟอร์ม เช่น ปุ่มกด ซึ่งในขณะที่ออกแบบถ้าคลิกที่ออบเจกต์ใดจะเป็นการเขียน โปรแกรมให้ออบเจกต์นั้น

2.34 การสร้างเมนูย่อย

การสร้างเมนูย่อยขึ้นมาอาจด้วยจุดประสงค์เพื่อจัดกลุ่มย่อยของคำสั่งในเมนู หรือเป็นเพราะเมนูบาร์มีเนื้อที่ไม่เพียงพอกับหัวข้อจำนวนมาก ในการสร้างเมนูย่อยนั้นก็มีขั้นตอนที่เพิ่มขึ้นนอกเหนือจากการกำหนดหัวข้อปกติเพียงแคใช้ปุ่ม ที่อยู่ในส่วนล่าง Menu Editor เพื่อเลื่อนหัวข้อให้เป็นหัวข้อย่อยชั้นถัดไปของเมนูที่ต้องการ เท่านั้น ดังภาพนี้



รูปที่ 2.29 การสร้างเมนูย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปขอให้สังเกตว่าได้มีการกำหนดให้คุณสมบัติ Checked ของหัวข้อ Open data File เป็นจริง จึงมีเครื่องหมายถูกแสดงอยู่ข้างหน้าหัวข้อ และการที่จะดูว่าหัวข้อใดมีเมนูย่อยลงไปอีกนั้น สังเกตง่าย ๆ ได้จากเครื่องหมายรูปสามเหลี่ยมที่อยู่ทางขวาของหัวข้อนั้น

2.35 การควบคุมเมนูขณะโปรแกรมทำงานอยู่ (Run Time)

การควบคุมเมนูขณะโปรแกรมกำลังทำงานอยู่ ก็คือการเขียนคำสั่งให้เปลี่ยนคุณสมบัติของเมนูนั่นเอง ซึ่งคำสั่งเหล่านี้อาจอยู่ในโปรซีเจอร์ของเมนูเองหรือในโปรซีเจอร์อื่นก็ได้ การสั่งให้หัวข้อนั้นๆ ถูกแสดง (Visible), สามารถเลือกได้ (Enabled) หรือแสดงเครื่องหมายถูก (checked) นั้น ใช้ประโยคคำสั่งกำหนดค่าธรรมดา โดยกำหนดค่าจริง (True หรือ -1) หรือเท็จ (False หรือ 0) ให้กับคุณสมบัติที่ต้องการ เช่น

| | |
|---------------------------------------|--------------------------------------|
| <code>mnuItem1.Visible = True</code> | ‘เป็นการกำหนดให้แสดงหัวข้อ mnuItem1 |
| <code>mnuItem2.Enabled = False</code> | ‘ทำให้ mnuItem2 เลือกไม่ได้ |
| <code>mnuItem3.Checked = True</code> | ‘แสดงเครื่องหมายถูกข้างหน้า mnuItem3 |

2.36 เมนูแบบพ็อปอัพ

นอกจากการแสดงผลเมนูที่เมนูบาร์แล้ว เราอาจให้มีการแสดงผลเมนูแบบพ็อปอัพ (pop up menu) ณ ตำแหน่งใดก็ได้ โดยใช้คำสั่ง PopupMenu ตามด้วยชื่อเมนู

`PopupMenu mnuColor`

โดยเมนู mnuColor นั้นสร้างขึ้นด้วย Menu Editor โดยเมนูนี้จะต้องมีเมนูย่อยลงไปอีกอย่างน้อย 1 เมนูย่อย ด้วยวิธีนี้ทำให้เราสามารถเขียนโปรแกรมให้มีเมนูแบบพ็อปอัพ โดยอาจผูกติดกับเหตุการณ์ใดของออบเจกต์ก็ได้

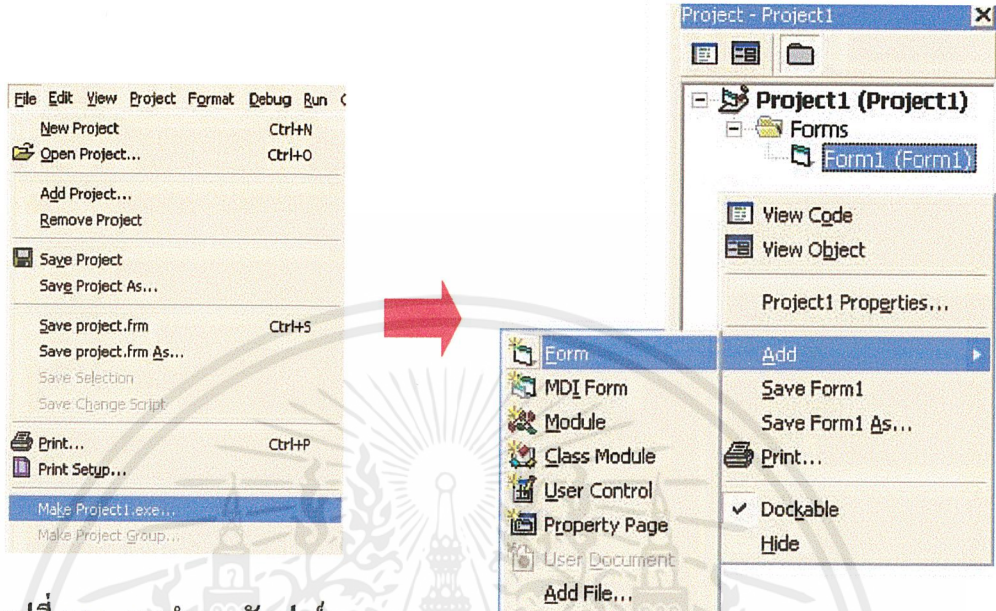


รูปที่ 2.30 เมนูแบบพ็อปอัพ

2.37 การทำงานกับฟอร์ม

การเพิ่มฟอร์มเข้าสู่โปรเจ็ค

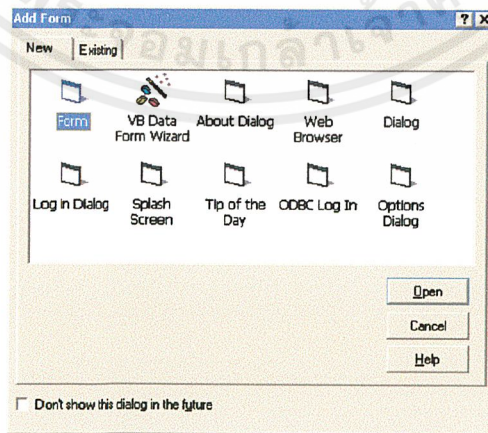
การเพิ่มฟอร์มเข้าสู่โปรเจ็คใน Visual Basic ทำได้หลายวิธีแล้วแต่ความถนัดของผู้ใช้ดังนี้



รูปที่ 2.31 การทำงานกับฟอร์ม

1. โดยใช้ Toolbar
2. โดยการคลิกเมาส์ขวาที่วินโดว์ Project Explorer เลือกเมนู Add > Form
3. โดยใช้เมนู Project > Add Form

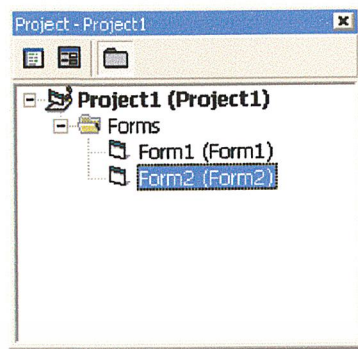
ซึ่งไม่ว่าด้วยวิธีใดก็จะปรากฏวินโดว์สำหรับเลือกรูปแบบของฟอร์มที่เราต้องการใช้งาน ซึ่ง Visual Basic เตรียมมาให้มากมายดังในภาพนี้ แต่ถ้าเราต้องการเลือกฟอร์มเปล่าให้เลือก Form จากนั้นกดปุ่ม Open



รูปที่ 2.32 การเลือกฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเพิ่มฟอร์มเรียบร้อยแล้วลองสังเกตที่วินโดว์ Project Explorer ก็จะเห็นฟอร์มใหม่ที่เราเพิ่มเข้ามาในโปรเจก



รูปที่ 2.33 ฟอร์มใหม่ที่เพิ่มเข้ามาในโปรเจก

2.38 เมธอด Show, Hide และคำสั่ง Unload

การสั่งให้ฟอร์มใดขึ้นมาทำงานทำได้โดยใช้เมธอด Show และการสั่งเลิกใช้งานฟอร์มจะใช้คำสั่ง Unload แต่ถ้าต้องการเพียงแค่ซ่อนฟอร์มไม่ให้ผู้ใช้เห็นก็ใช้เมธอด Hide ลองพิจารณาตัวอย่างโค้ด

| | |
|---------------|--|
| Form1.Show | ‘สั่งให้นำ Form1 มาแสดงบนหน้าจอ |
| Form2.Show | ‘สั่งให้นำ Form1 มาแสดงบนหน้าจอ |
| Unload.Form2 | ‘สั่งให้ปิด Form2 และนำฟอร์มออกจากหน่วยความจำ |
| Form2.Hide | ‘สั่งให้นำ Form2 ไม่ให้แสดงบนหน้าจอ |
| Unload.MyForm | ‘สั่งให้ปิด MyForm และนำฟอร์มออกจากหน่วยความจำ |

นอกจากนี้ในโค้ดของฟอร์มใดๆ เมื่อต้องการอ้างถึงฟอร์มนั้น สามารถใช้คำว่า Me แทนชื่อฟอร์มได้ เช่น ถ้าอยู่ในโค้ดเคดิตอร์ของฟอร์มชื่อ MyForm เราสามารถปิดฟอร์ม MyForm ได้ด้วยคำสั่ง Unload Me หรือซ่อนด้วยการใช้ Me.Hide

2.39 การเปิดฟอร์มแบบ Modal และ Modeless

ในการเรียกให้ฟอร์มแสดงขึ้นมาโดยใช้เมธอด Show นั้นเราสามารถใส่พารามิเตอร์ได้สองแบบคือ

1. ใช้ตัวแปร vbModal หลังเมธอด Show จะทำให้ผู้ใช้ต้องทำงานกับฟอร์มที่เปิดใหม่เท่านั้น ไม่สามารถทำงานกับฟอร์มอื่นใดได้ จนกว่าจะปิดฟอร์มนั้น

```
Form2.Show vbModal
```

โดยเราจะเรียกการเปิด Form2 แบบนี้ว่าเป็น การเปิดฟอร์มแบบ Modal

2. ใช้ตัวแปร vbModeless หลัง เมธอด Show ซึ่งวิธีนี้จะทำให้ผู้ใช้เลือกการทำงานกับฟอร์มใดก็ได้ โดยอิสระ

```
Form2.Show vbModeless
```

โดยทั่วไปถ้าเราไม่ใส่พารามิเตอร์ใดๆ จะถือว่าเป็น การเปิดฟอร์มแบบ Modeless

2.40 การอ้างตัวแปรหรือออบเจกต์ระหว่างฟอร์ม

การอ้างตัวแปรระหว่างฟอร์มสามารถทำได้โดยใช้ชื่อฟอร์ม และตามด้วยจุดพร้อมทั้งชื่อออบเจกต์ในฟอร์มนั้น แล้วตามด้วยจุดพร้อมด้วยเมธอดหรือพรอพเพอร์ตี้ที่ต้องการ เช่น

```
Form2.Text1.Text
```

‘จะเป็นการอ้างถึงพรอพเพอร์ตี้ Text ของ Text box ชื่อ Text1 ใน Form2

```
MyForm.Label1.Caption
```

‘จะเป็นการอ้างถึงพรอพเพอร์ตี้ Caption ของ Label ชื่อ Label1 ใน MyForm

```
Form1.Variable1
```

‘จะเป็นการอ้างถึงตัวแปรชื่อ Variable1 ที่ประกาศไว้เป็นตัวแปรแบบ Public ใน Form1

2.41 ออบเจกต์เกี่ยวกับกราฟฟิก

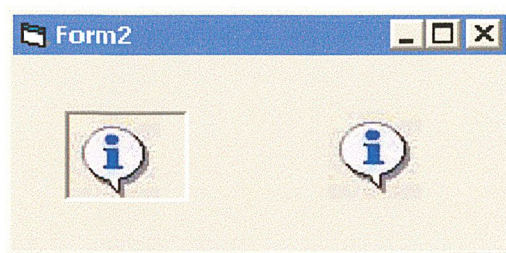
2.41.1 Picture Box

เป็นออบเจกต์ที่สามารถแสดงภาพในรูปแบบของ Bitmap, icon, metafile โดยแสดงภาพเท่าที่จะแสดงได้ในขนาดของตัวมันเท่านั้น นั่นคือหากขนาดของภาพโตกว่า PictureBox เราจะเห็นภาพไม่หมดนอกจากนี้เราอาจใช้คำสั่งด้านกราฟฟิกเพื่อวาดภาพใน PictureBox รวมไปถึงการใช้เมธอด Print ได้ด้วยหรือจะใช้แสดงข้อความก็ได้ ซึ่ง PictureBox ก็มีคุณสมบัติเกี่ยวกับ Text ไว้ให้ เช่น Font ต่างๆ, ตัวเอียง, ตัวหนา และด้วยออบเจกต์นี้ เราสามารถนำมาสร้างทูลบาร์หรือแถบเครื่องมือได้อีกด้วย นอกจากนี้ข้อแตกต่างขอ PictureBox กับ Image, Shape และ Line ก็คือ PictureBox สามารถเป็น Container หรือทำหน้าที่เก็บออบเจกต์อื่นได้เหมือนกับ Frame ซึ่งทำให้การจัดออบเจกต์เป็นกลุ่มทำได้สะดวกเพราะการย้าย Container จะเป็นการย้ายออบเจกต์ภายในไปด้วย

2.41.2 Image

ทำหน้าที่แสดงภาพเช่นเดียวกับ PictureBox ในรูปแบบของ Bitmap, icon, metafile ข้อดีของ Image คือใช้ทรัพยากรของระบบน้อยกว่า Picture Box และยังทำการวาดภาพใหม่ได้เร็วกว่าด้วย แต่ทั้งคุณสมบัติ, เมธอดและการตอบสนองเหตุการณ์จะน้อยกว่าหรือเป็นข้อจำกัดของ Picture Box นั่นเอง เพราะหน้าที่หลักของ Image คือแสดงภาพเท่านั้นแต่เพราะเหตุนี้เอง Image จึงมีคุณสมบัติหนึ่งที่ไม่มีใน Picture Box ได้แก่ พรอพเพอร์ตี้ Stretch ซึ่งเป็นคุณสมบัติหนึ่งที่กำหนดให้ขนาด

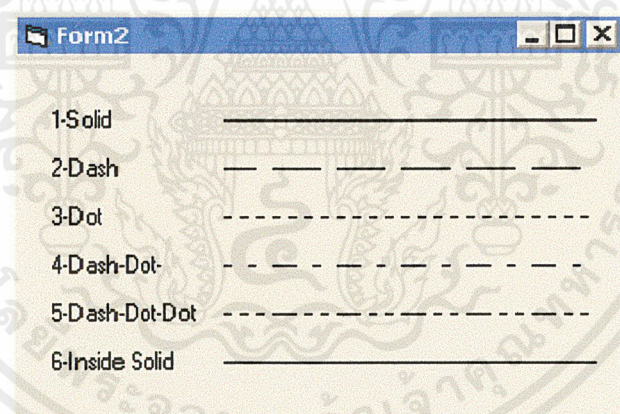
ของออบเจกต์เท่ากับภาพ หรืออีกนัยหนึ่งคือการปรับขนาดของภาพโดยย่อ/ขยายให้เท่ากับขนาดของออบเจกต์ ดังภาพนี้



รูปที่ 2.34 Image

2.41.3 Line

สำหรับการสร้างเส้นตรงในแนวตั้ง, แนวนอนหรือทะแยง อาจใช้ออบเจกต์ Line แทนการใช้คำสั่งวาดเส้นตรงได้ โดยขณะที่โปรแกรมทำงานเราสามารถเปลี่ยนคุณสมบัติของเส้นเพื่อเปลี่ยนตำแหน่ง, ความยาว, สี หรือรูปแบบของเส้นได้ ออบเจกต์นี้อาจแสดงในฟอร์ม, ใน Picture Box หรือเฟรมได้ พรอพเพอร์ตี้ `BorderWidth` ใช้กำหนดความหนาของเส้นได้ตั้งแต่ 0 ถึง 8192 จุด ส่วนพรอพเพอร์ตี้ `BorderStyle` ใช้กำหนดรูปแบบของเส้น ซึ่งมี 6 รูปดังภาพนี้



รูปที่ 2.35 line

2.41.4 Shape

ใช้แสดงรูปทรงต่างๆซึ่งมีด้วยกัน 6 แบบคือ สีเหลี่ยมผืนผ้า, สีเหลี่ยมจตุรัส, วงกลม, วงรี, สีเหลี่ยมผืนผ้าแบบมุมมน และสีเหลี่ยมจตุรัสแบบมุมมน โดยสามารถควบคุมสี, รูปแบบของเส้น, การระบายสีในรูปและขนาดได้ โดยใช้พรอพเพอร์ตี้ `BorderStyle` และ `BorderWidth` ได้เหมือนกับ Line ส่วนลักษณะของรูปทรงกำหนดโดยพรอพเพอร์ตี้ `Shape` ซึ่งมีค่าคงที่ดังตารางต่อไปนี้

| ค่าพารามิเตอร์ Shape | ค่าคงที่ | รูปทรงที่ได้ |
|----------------------|-------------------------|---------------------------|
| 0-Rectangle | vbShapeRectangle | รูปสี่เหลี่ยมผืนผ้า |
| 1-Square | vbShapeSquare | รูปสี่เหลี่ยมจัตุรัส |
| 2-Oval | vbShapeOval | รูปวงรี |
| 3-Circle | vbShapeCircle | รูปวงกลม |
| 4-Rounded Rectangle | vbShapeRoundedRectangle | รูปสี่เหลี่ยมผืนผ้ามุมมน |
| 5-Rounded Square | vbShapeRoundedSquare | รูปสี่เหลี่ยมจัตุรัสมุมมน |

2.42 การกำหนดภาพ

โดยทั่วไปการแสดงผลภาพนั้นทำได้เฉพาะในฟอร์ม , Picture Box และ Image เท่านั้น ส่วนไฟล์รูปภาพต้องอยู่ในฟอร์แมต Bitmap, Icon, MetaFile อย่างใดอย่างหนึ่ง ซึ่งสร้างได้จากโปรแกรมวาดภาพเช่น PaintBrush ที่มากับวินโดวส์หรือเลือกจากกลุ่มของไฟล์ไอคอนที่ Visual Basic เรียกไว้ให้เป็นจำนวนมาก โดยจัดกลุ่มไว้ใน Directory ย่อยของ Common\Graphics ในขณะที่ออกแบบเราสามารถเพิ่มรูปให้กับฟอร์มได้โดยโหลดภาพใส่ฟอร์มหรือ Picture Box หรือ Image ด้วยวิธีกำหนดคุณสมบัติ Picture แล้วเลือกชื่อไฟล์หรืออาจใช้วิธี “คัด-ปะ” โดย copy ภาพที่ต้องการผ่านทางคลิปบอร์ดและเมื่อเลือกพารามิเตอร์ Picture แล้วให้เลือกหัวข้อ Paste จากเมนู Edit แทนที่จะระบุไฟล์

นอกจากการใส่รูปในออกแบบขณะออกแบบแล้ว การเพิ่มรูปในขณะที่โปรแกรมกำลังทำงานนั้นก็สามารถทำได้โดย Visual Basic ได้เตรียมคำสั่ง LoadPicture ไว้ให้ อย่างเช่นคำสั่ง

```
DisplayBox.Picture=LoadPicture ("win.bmp")
```

เป็นการโหลดภาพจาก File win.bmp ให้กับ DisplayBox

อีกวิธีหนึ่งคือการ Copy รูประหว่าง 2 ฟอร์มโดยใช้คำสั่งกำหนดค่าตามปกติคือ

```
DisplayBox2.Picture= DisplayBox1.Picture
```

ส่วนการยกเลิกภาพจากออกแบบขณะรันโปรแกรม ก็ยังคงใช้คำสั่ง LoadPicture เหมือนเดิม เพียงแต่ไม่ระบุชื่อ File เช่น

```
DisplayBox.Picture=LoadPicture ("")
```

นอกจากนั้นเรายังสามารถเก็บภาพลงเพิ่มข้อมูลได้อีกด้วย โดยใช้คำสั่ง SavePicture ดังนี้

```
SavePicture Picture1.Image, "c:\windows\desktop\mypic.bmp"
```

โดยรูปแบบของเพิ่มที่จัดเก็บนั้นจะเป็นรูปแบบเดียวกับภาพต้นฉบับ

2.43 การทำงานเกี่ยวกับออบเจกต์ของภาพ

Align เฉพาะ Picture Box สำหรับปรับให้ออบเจกต์ชิดขอบบน (1) หรือขอบล่าง (2)

Height กำหนดความสูงของออบเจกต์ซึ่งสามารถใช้ได้ทั้ง Run time และ Design time

Width กำหนดความกว้างของออบเจกต์ซึ่งสามารถใช้ได้ทั้ง Run time และ Design time

AutoSize เฉพาะ Picture Box หากมีค่าเป็น True จะทำให้ Visual basic ปรับขนาดของ Picture Box ให้เท่ากับขนาดของภาพ

Stretch เฉพาะ Image ใช้ในการปรับขนาดของภาพให้เท่ากับขนาดของ Image เมื่อมีค่าเป็นจริงจะย่อ/ขยายภาพให้ แต่หากเป็นเท็จจะเปลี่ยนขนาดของออบเจกต์ให้เท่ากับขนาดของภาพแทน

นอกจากจะเปลี่ยนขนาดแล้วการให้แสดงหรือไม่แสดงภาพก็ทำได้โดยเพียงแต่กำหนดที่คุณสมบัติ Visible ให้เป็นจริงหรือเท็จเท่านั้น เช่น `Picture1.Visible=True` ส่วนการเคลื่อนย้ายออบเจกต์จากจุดหนึ่งไปยังอีกจุด อาจใช้วิธีเปลี่ยนคุณสมบัติ Left หรือ Top ซึ่งเป็นตัวบอกระยะห่างระหว่างออบเจกต์เทียบกับขอบซ้ายและขอบบนของฟอร์ม หรือจะใช้เมธอด Move ก็ได้ สำหรับออบเจกต์ประเภท Line จะมีคุณสมบัติ X1, Y1, X2, Y2 ที่บอกจุดเริ่มต้นและจุดสิ้นสุดของเส้น ซึ่งไม่ว่าจะเป็นคุณสมบัติใด เราสามารถเขียนคำสั่งกำหนดค่าให้กับคุณสมบัติเพื่อเปลี่ยนตำแหน่งได้อย่างง่ายดาย ส่วนสีของภาพและพื้นกำหนดได้โดยใช้พรอพเพอร์ตี้ ForeColor และ BackColor ซึ่งหากกำหนดคอนทราสต์แบบ Visual Basic ก็จะแสดงตารางสีขึ้นมาให้เลือก

การเปลี่ยนค่าของคุณสมบัติ X, Y หรือ Left กับ Top นั้นแม้จะทำให้ออบเจกต์ย้ายตำแหน่งได้ก็ตาม แต่การเลื่อนตำแหน่งจะดูไม่ราบเรียบนัก โดยเฉพาะในกรณีของการเลื่อนในแนวทแยง เพราะจะต้องเปลี่ยนค่าทีละค่า ซึ่งเป็นการเลื่อนภาพ 2 ครั้ง ทำให้เกิดอาการกระตุก ดังนั้นจึงควรใช้เมธอด Move ซึ่งทำให้การเลื่อนตำแหน่งได้ราบรื่นกว่า รูปแบบ Move คือ

```
[object.]Move left[, top [, width [, height]]]
```

ออบเจกต์ที่ใช้อาจเป็นฟอร์มหรือคอนโทรลใดๆ ซึ่งหากไม่ระบุจะหมายถึงฟอร์มที่โค้ดนี้อยู่อย่างน้อยที่สุดเราจะต้องกำหนดค่าให้พรอพเพอร์ตี้ Left ไว้หนึ่งตัว ส่วนค่าอื่นๆ จะละไว้ก็ได้ สำหรับการเปลี่ยนตำแหน่งนั้น หากตำแหน่งใหม่เป็นแบบสัมพัทธ์ เช่น ย้ายไปทางขวา 100 ลงข้างล่าง 50 เราก็ต้องนำค่าในพรอพเพอร์ตี้ Left และ Top มาช่วยดังนี้

```
Picture1.Move Picture1.Left+100, Picture1.Top+50
```

ส่วนการทำภาพเคลื่อนไหวนั้นจะใช้วิธีโหลดภาพแต่ละเฟรมที่จะมาประกอบเป็นภาพเคลื่อนไหวไว้ในออบเจกต์ที่เก็บภาพได้ นั่นหมายถึงถ้าเรามีภาพที่จะใช้เป็นภาพเคลื่อนไหว 5 เฟรม เราก็จะต้องสร้างออบเจกต์สำหรับเก็บภาพไว้ 5 ออบเจกต์ พร้อมกับกำหนดพรอปเพอร์ตี้ Visible ให้เป็นเท็จเพื่อให้มองไม่เห็น หลังจากนั้นเมื่อต้องการแสดงภาพเคลื่อนไหวให้นำภาพที่เก็บไว้มา copy ให้กับ ออบเจกต์ที่เตรียมไว้สำหรับแสดงภาพโดยเฉพาะ ก็จะเห็นภาพเคลื่อนไหวตามต้องการ หากไม่ต้องการสร้างออบเจกต์ขึ้นมาหลายตัว อาจใช้ Control Array ก็ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนาโปรแกรม

ขั้นตอนทั้งหมดในงานวิจัยจะเกิดขึ้นไม่ได้หากขาดการออกแบบ การวางแผนที่ดีและมีเป้าหมายชัดเจน การดำเนินงานทั้งหมดจะถูกกระทำภายใต้แผนที่วางเอาไว้อย่างรัดกุมเพื่อนำความสำเร็จตามเป้าหมายที่วางเอาไว้ในแต่ละช่วง แผนงานทั้งหมดมีดังนี้

1. ขั้นตอนการออกแบบซึ่งอาจแบ่งย่อยได้อีกหลายส่วนงาน

- การออกแบบโปรแกรม
- การออกแบบอินเตอร์เฟซที่ใช้ติดต่อกับผู้ใช้งาน
- การออกแบบการทำงานในส่วนต่างๆของโปรแกรม

ในขั้นตอนการออกแบบนี้จะใช้เวลานานพอสมควร เพื่อให้ได้รูปแบบที่เป็นเอกลักษณ์และสอดคล้องกับความต้องการในการใช้งาน

2. ขั้นตอนการเขียนโปรแกรม

ขั้นตอนนี้ใช้เวลานานที่สุดในตลอดระยะเวลาทำวิจัย เพราะจำเป็นต้องศึกษารูปแบบวิธีการเขียนโปรแกรม เพื่อให้ได้โปรแกรมที่กระชับรัดกุมและทำงานได้อย่างถูกต้องเข้าใจง่าย

3.1 ขั้นตอนการออกแบบ

ขั้นตอนการออกแบบมีรายละเอียดดังนี้

3.1.1 การออกแบบโปรแกรม

การออกแบบโปรแกรมเป็นส่วนที่สำคัญที่สุดในขั้นตอนการออกแบบทั้งหมด เพราะโปรแกรมคือทุกสิ่งทุกอย่างตั้งแต่แนวความคิดของโปรแกรม การทำงานของโปรแกรม การทำงานของคอมพิวเตอร์ต่างๆของโปรแกรม โดยการออกแบบมีรายละเอียดต่างๆ ดังนี้

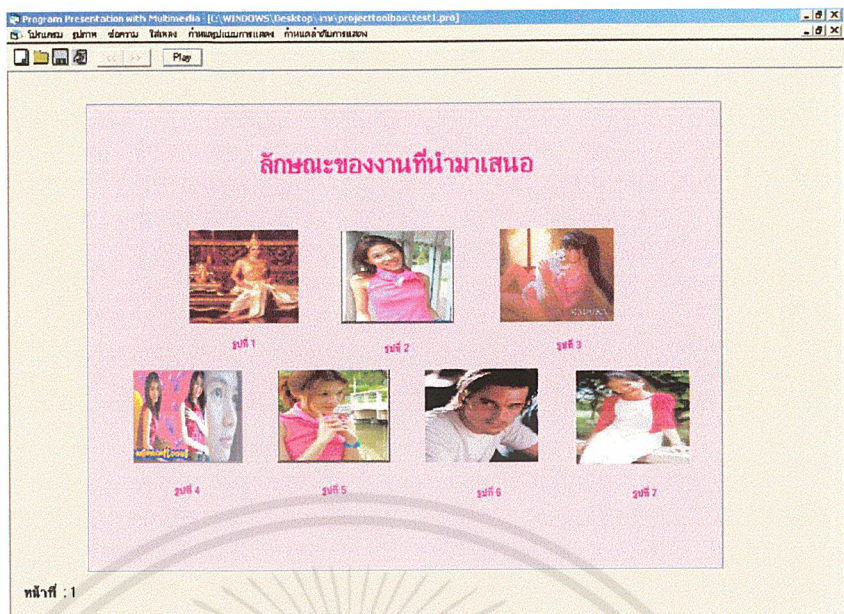
- การออกแบบทั่วไป
- การออกแบบเมนูและการรับข้อมูล

3.1.1.1 การออกแบบทั่วไป

การออกแบบทั่วไปเป็นการออกแบบที่เป็นลักษณะของโปรแกรม มีสิ่งต่างๆ ดังนี้

- รูปภาพ
- ข้อความ
- ปุ่มบังคับลำดับการทำงาน
- หน้าจอแสดงการนำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 รูปภาพ ข้อความ ปุ่มบังคับลำดับการทำงาน



รูปที่ 3.2 หน้าจอแสดงการนำเสนอ

3.1.1.2 การออกแบบเมนูและการรับข้อมูล

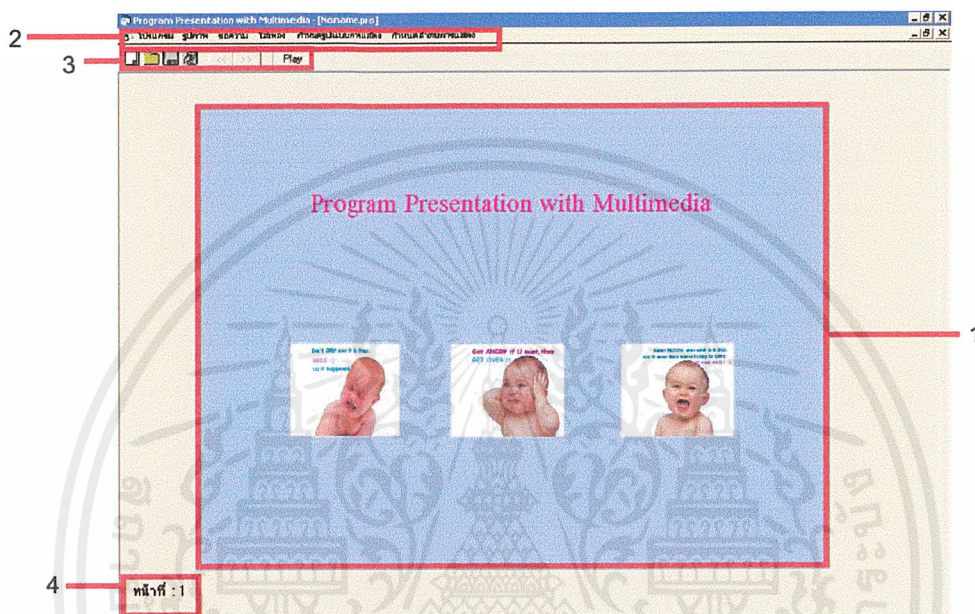
เป็นการออกแบบเมนูที่ใช้สำหรับกำหนดรายละเอียดต่างๆ สามารถใช้เมาส์เลือกเมนูและทำการกำหนดรายละเอียดต่างๆ ในแต่ละเมนูที่ต้องการได้ โปรแกรมจะทำการตรวจสอบข้อมูลที่รับเข้ามาและทำการเก็บข้อมูล และนำข้อมูลนั้นๆ ไปใช้ในการทำงานจริง ยกตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การกำหนดรูปแบบการเคลื่อนที่ให้กับรูปภาพ เราสามารถเลือกเมนูกำหนดรูปแบบการแสดง และเลือกเมนูย่อยกำหนดรูปแบบการแสดงภาพให้กับภาพที่ต้องการได้

3.1.2 การออกแบบอินเทอร์เฟซติดต่อกับผู้ใช้

ในส่วนการออกแบบอินเทอร์เฟซที่ใช้ติดต่อกับผู้ใช้ จะแสดงถึงสถานะต่างๆของโปรแกรม ขณะนั้น เช่น แถบของปุ่มคำสั่ง สถานะของหน้าที่ทำงานอยู่ บริเวณพื้นที่การทำงานของโปรแกรม โดยอินเทอร์เฟซต่างๆมีรูปแบบลักษณะ ดังรูป



รูปที่ 3.3 อินเทอร์เฟซที่ติดต่อกับผู้ใช้

จากรูปแสดงถึงส่วนสำคัญ 4 ส่วนดังนี้

- 1.บริเวณพื้นที่การทำงาน
- 2.แถบเมนูคำสั่ง
- 3.แถบของปุ่มคำสั่ง
- 4.ส่วนที่แสดงสถานะของหน้า

3.1.2.1 บริเวณพื้นที่การทำงาน

บริเวณพื้นที่การทำงานของโปรแกรมจะเป็นส่วนที่เราสามารถที่จะนำรูปภาพหรือข้อความต่าง ๆ มาวางไว้ในส่วนนี้ได้ และสามารถที่จะย้ายรูปภาพหรือข้อความต่าง ๆ ไปยังบริเวณต่างๆของพื้นที่ส่วนนี้ได้ โดยเราทำการคลิกเมาส์ปุ่มซ้ายค้างไว้ที่รูปภาพหรือข้อความที่ต้องการและทำการย้ายไปบริเวณที่ต้องการ และบริเวณส่วนนี้ยังสามารถที่จะใส่รูปภาพและข้อความได้หลายอัน

3.1.2.2 แอบเมนูคำสั่ง

ในส่วนนี้จะเป็นส่วนที่ให้ผู้ใช้งานสามารถใช้งานเพื่อช่วยให้สามารถสร้างงานได้ ในส่วนนี้จะไม่แสดงปุ่มคำสั่งการทำงาน โดยคำสั่งที่มีในส่วนนี้จะมีคำสั่งดังนี้

- เปิดไฟล์ใหม่
- เปิดไฟล์
- บันทึกไฟล์
- บันทึกไฟล์เป็น
- ไล่ Background
- ใส่อารมณ์
- เปลี่ยนแปลงรูปภาพ
- เปลี่ยนแปลงขนาดรูปภาพ
- ลบรูปภาพ
- ใส่อารมณ์
- เปลี่ยนแปลงรูปแบบข้อความ
- ลบข้อความ
- ใส่เพลง
- กำหนดรูปแบบการแสดงผลรูปภาพ
- กำหนดรูปแบบการแสดงผลข้อความ
- กำหนดลำดับการทำงาน

โปรแกรม รูปภาพ ข้อความ ใส่เพลง กำหนดรูปแบบการแสดงผล กำหนดลำดับการแสดงผล

รูปที่ 3.4 แอบเมนูคำสั่ง

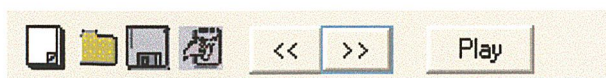
3.1.2.3 แอบของปุ่มคำสั่ง

ในส่วนนี้จะเป็นส่วนที่ใช้สำหรับทำงานต่างๆ โดยปุ่มคำสั่ง ซึ่งเป็นคำสั่งที่ใช้ในการทำงานของหน้าที่เรากำลังจะทำงาน โดยมีลักษณะการทำงานต่างๆเพื่อช่วยให้สามารถสร้างงานได้และในส่วนนี้ จะมีปุ่มคำสั่งต่างๆ ดังนี้

- ปุ่มการสร้างงานใหม่ (New)
- ปุ่มการเปิดงานที่มีการทำไว้แล้ว (Open)
- ปุ่มการบันทึกงาน (Save)
- ปุ่มการเพิ่มหน้าการทำงาน (Add Page)
- ปุ่มการย้อนกลับไปหน้าที่แล้ว (Back)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่มการไปสู่หน้าถัดไป (Next)
- ปุ่มการแสดงงาน (Play)



รูปที่ 3.5 แถบของปุ่มคำสั่ง

3.1.2.4 ส่วนที่แสดงสถานะของหน้า

ส่วนนี้จะเป็นการบอกถึงสถานะของหน้าที่มีการทำงานอยู่ในขณะนั้น

หน้าที่ : 1

รูปที่ 3.6 ส่วนที่แสดงสถานะของหน้า

3.1.3 การออกแบบการทำงานในส่วนต่างๆของโปรแกรม

ในส่วนนี้จะอธิบายถึงขั้นตอนในการออกแบบการทำงานในส่วนต่างๆของโปรแกรมโดยจะมีการมองภาพรวมของโปรแกรมว่าสามารถมีการทำอะไรได้บ้างอะไรใช้แนวคิดในการออกแบบส่วนต่างๆในการทำงานของโปรแกรมโดยมีรายละเอียดต่างๆดังนี้

3.1.3.1 ส่วนของการวางภาพ

โดยเมื่อมีการเพิ่มรูปภาพเข้ามาในโปรแกรมจะมีการกำหนดตำแหน่งการวางของรูปภาพไว้ที่บริเวณหนึ่ง โดยรูปภาพที่มีการเพิ่มเข้ามาในโปรแกรมจะมีลำดับการวาง โดยที่รูปภาพที่เพิ่มเข้ามาในโปรแกรมก่อนจะเป็นรูปภาพที่อยู่บริเวณล่างสุด เมื่อมีการเพิ่มรูปภาพถัดไปก็จะมีวางทับกับไปเรื่อยๆ และรูปภาพทุกรูปภาพเราสามารถที่จะเคลื่อนย้ายไปบริเวณต่างๆบนพื้นที่การออกแบบงานได้

3.1.3.2 ส่วนของการวางข้อความ

เมื่อมีการเพิ่มข้อความเข้ามาในโปรแกรม จะมีการกำหนดตำแหน่งของการวางข้อความไว้ที่บริเวณหนึ่ง โดยข้อความที่มีการเพิ่มเข้ามาในโปรแกรมจะมีลำดับการวาง โดยที่ข้อความที่เพิ่มเข้ามาในโปรแกรมก่อนจะเป็นข้อความที่อยู่บริเวณล่างสุด เมื่อมีการเพิ่มข้อความถัดไปก็จะมีวางทับกับไปเรื่อยๆ และข้อความทุกข้อความเราสามารถที่จะเคลื่อนย้ายไปบริเวณต่างๆบนพื้นที่การออกแบบงานได้

3.1.3.3 ขั้นตอนการบันทึกโปรแกรม

ในการบันทึกโปรแกรมมี 2 ประเภท คือ บันทึกไฟล์ใหม่ และบันทึกทับไฟล์เก่า

- บันทึกไฟล์ใหม่ จะทำการตรวจสอบว่าเป็นไฟล์ noname.pro หรือไม่ ถ้าใช่ จะทำการบันทึกโดยให้ใส่ชื่อไฟล์

- บันทึกทับไฟล์เก่า จะทำการ บันทึกทับ

รายละเอียดในการบันทึกไฟล์ทั้งสองประเภท มีดังนี้

1). Background

จะทำการบันทึก background ที่มีการแสดงอยู่โดยจะเก็บ path ของรูปและชื่อรูปไว้โดยจะมีคำอธิบายบอกไว้ว่าเป็น background และจะเก็บรายละเอียดว่าเป็น background รูปที่เท่าไร เช่น

C:\WINDOWS\Desktop\projecttoolbox\projecttoolbox\picture\picgreen.bmp background
+++++++ รูปที่ 0

C:\WINDOWS\Desktop\View-15.jpg background ++++++++ รูปที่ 1

2).รูปภาพ

จะทำการบันทึกรูปภาพที่มีการแสดงอยู่โดยจะเก็บ path และชื่อของรูปภาพไว้โดยจะมีคำอธิบายบอกว่าเป็นรูปภาพและเก็บว่าเป็นรูปภาพที่เท่าไร ซึ่งการเก็บรูปภาพนี้จะเก็บรายละเอียดต่างๆ เกี่ยวกับรูปภาพไว้ด้วย โดยจะเก็บตำแหน่งของรูปภาพ ความกว้างและความสูงของรูปภาพ เช่น

C:\WINDOWS\Desktop\BunBottom.jpg รูปที่ ++++++++ 0

1035 รูปที่ ++++++++ 0 (top,left,width,height)

6690 รูปที่ ++++++++ 0

4125 รูปที่ ++++++++ 0

3420 รูปที่ ++++++++ 0

C:\WINDOWS\Desktop\Classic_insert.jpg รูปที่ ++++++++ 10

5460 รูปที่ ++++++++ 10

4740 รูปที่ ++++++++ 10

1695 รูปที่ ++++++++ 10

2055 รูปที่ ++++++++ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\WINDOWS\Desktop\anchored.jpg รูปที่ ++++++++ 11

2025 รูปที่ ++++++++ 11

8955 รูปที่ ++++++++ 11

1695 รูปที่ ++++++++ 11

2055 รูปที่ ++++++++ 11

3). ข้อความ

จะทำการบันทึกข้อความที่มีการแสดงอยู่โดยจะเก็บข้อความต่างๆ ไว้และจะมีคำอธิบายบอกว่า เป็นข้อความและเก็บว่าเป็นข้อความที่เท่าไรซึ่งการเก็บข้อความนี้จะเก็บรายละเอียดต่างๆ เกี่ยวกับข้อความโดยจะเก็บ ตำแหน่งของข้อความ,ขนาดของ font,สีของแต่ละข้อความ,ประเภทของ font ด้วย เช่น

ข้อความที่ 1 หน้าที่ 1 ข้อความที่ ++++++++ 0

2460 ข้อความที่ ++++++++ 0

2670 ข้อความที่ ++++++++ 0

24 ข้อความที่ ++++++++ 0

-2147483640 ข้อความที่ ++++++++ 0

BrowalliaUPC ข้อความที่ ++++++++ 0

ข้อความแรก หน้าที่ 2 ข้อความที่ ++++++++ 10

3570 ข้อความที่ ++++++++ 10

3375 ข้อความที่ ++++++++ 10

24 ข้อความที่ ++++++++ 10

8404992 ข้อความที่ ++++++++ 10

AngsanaUPC ข้อความที่ ++++++++ 10

ข้อความที่ 2 หน้าที่ 2 ข้อความที่ ++++++++ 11

5250 ข้อความที่ ++++++++ 11

8925 ข้อความที่ ++++++++ 11

36 ข้อความที่ ++++++++ 11

33023 ข้อความที่ ++++++++ 11

AngsanaUPC ข้อความที่ ++++++++ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.4 รูปแบบการเคลื่อนที่ของรูปภาพ

รูปแบบการเคลื่อนที่ของรูปภาพจะดึงมาจากค่าที่เก็บไว้ในตัวแปรถ้าค่านั้นเก็บรูปแบบการเคลื่อนที่ไว้ ก็จะนำรูปแบบนั้นมาเก็บไว้ในไฟล์ โดยจะมีคำอธิบายบอกว่าเป็นรูปแบบการเคลื่อนที่ของรูปภาพและบอกว่าเป็นรูปภาพที่เท่าไร ถ้ารูปนั้นไม่มีการกำหนดรูปแบบการเคลื่อนที่ไว้ ค่าที่นำมาเก็บจะเก็บว่าไม่มีรูปแบบการเคลื่อนที่ เช่น

เคลื่อนที่จากขวาไปซ้าย การเคลื่อนที่รูปที่ ++++++++ 0

ไม่มีรูปแบบการเคลื่อนที่ การเคลื่อนที่รูปที่ ++++++++ 1

เคลื่อนที่จากล่างขึ้นบน การเคลื่อนที่รูปที่ ++++++++ 2

รูปภาพจะใช้ method move ในการแสดงการเคลื่อนที่ โดยจะทำการตรวจสอบว่ารูปภาพนั้นๆ มีรูปแบบการเคลื่อนที่เป็นอย่างไร ซึ่งถ้ารูปแบบการเคลื่อนที่เป็นแบบเคลื่อนที่จากซ้ายไปขวา จะกำหนดให้รูปภาพมีจุดเริ่มต้นในการแสดงอยู่ที่ซ้ายสุดของหน้าจอและทำการเคลื่อนที่ไปทางขวาทีละ 100 pixels จนกว่าจะถึงตำแหน่งที่กำหนดในหน้าจอของการออกแบบ

คลี่จากบนลงล่าง จะกำหนดให้รูปภาพมีขนาดความกว้างเท่ากับขนาดของรูปภาพในหน้าจอของการออกแบบ ส่วนความสูงจะกำหนดให้มีค่าเริ่มต้น = 0 เมื่อทำการแสดงงาน จะทำการเพิ่มขนาดความสูงของรูปภาพทีละ 100 pixels จนกระทั่งความสูงมีขนาดเท่ากับรูปภาพในหน้าจอการออกแบบ

คลี่จากซ้ายไปขวา จะกำหนดให้รูปภาพมีขนาดความสูงเท่ากับขนาดของรูปภาพในหน้าจอของการออกแบบ ส่วนความกว้างจะกำหนดให้มีค่าเริ่มต้น = 0 เมื่อทำการแสดงงาน จะทำการเพิ่มขนาดความกว้างของรูปภาพทีละ 100 pixels จนกระทั่งความกว้างมีขนาดเท่ากับรูปภาพในหน้าจอการออกแบบ

3.1.3.5 รูปแบบการเคลื่อนที่ของข้อความ

รูปแบบการเคลื่อนที่ของข้อความจะดึงมาจากค่าที่เก็บไว้ในตัวแปร ถ้าตัวแปรนั้นเก็บรูปแบบการเคลื่อนที่ไว้ ก็จะนำรูปแบบนั้นมาเก็บไว้ในไฟล์ โดยจะมีคำอธิบายบอกว่าเป็นรูปแบบการเคลื่อนที่ของข้อความและบอกว่าเป็นข้อความที่เท่าไร ถ้าข้อความนั้นไม่มีการกำหนดรูปแบบการเคลื่อนที่ไว้ ค่าที่นำมาเก็บจะเก็บว่าไม่มีรูปแบบการเคลื่อนที่ เช่น

เคลื่อนที่จากขวาไปซ้าย การเคลื่อนที่ข้อความที่ ++++++++ 10

เคลื่อนที่จากบนลงล่าง การเคลื่อนที่ข้อความที่ ++++++++ 11

ไม่มีรูปแบบการเคลื่อนที่ การเคลื่อนที่ข้อความที่ ++++++++ 12

ไม่มีรูปแบบการเคลื่อนที่ การเคลื่อนที่ข้อความที่ ++++++++ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความจะใช้ method move ในการแสดงการเคลื่อนที่ โดยจะทำการตรวจสอบว่าข้อความนั้นๆ มีรูปแบบการเคลื่อนที่เป็นอย่างไร ซึ่งถ้ารูปแบบการเคลื่อนที่เป็นแบบเคลื่อนที่จากซ้ายไปขวา จะกำหนดให้ข้อความมีจุดเริ่มต้นในการแสดงอยู่ที่ซ้ายสุดของหน้าจอและทำการเคลื่อนที่ไปทางขวาทีละ 100 pixels จนกว่าจะถึงตำแหน่งที่กำหนดในหน้าจอของการออกแบบ

3.1.3.6 ลำดับการแสดงผล

ลำดับการแสดงผลคำที่นำมาเก็บในส่วนลำดับการแสดงผลนี้จะดึงมาจากลำดับการเคลื่อนที่ในแต่ละหน้า ซึ่งจะมีคำอธิบายบอกว่าเป็นลำดับการเคลื่อนที่ของหน้าที่เท่าไร เช่น

รูปที่ 1 ลำดับการเคลื่อนที่หน้าที่ ++++++++ 2

ข้อความที่ 1 ลำดับการเคลื่อนที่หน้าที่ ++++++++ 2

รูปที่ 2 ลำดับการเคลื่อนที่หน้าที่ ++++++++ 2

ข้อความที่ 2 ลำดับการเคลื่อนที่หน้าที่ ++++++++ 2

รูปที่ 3 ลำดับการเคลื่อนที่หน้าที่ ++++++++ 2

ลำดับในการแสดงผลมีการทำงานคือ จะตรวจสอบว่าทำการแสดงงานในหน้าที่เท่าไร ก็จะไปตรวจสอบลำดับการแสดงผลของรูปภาพและข้อความในหน้านั้นว่า ประกอบด้วยรูปภาพและข้อความอะไรบ้างและมีลำดับการแสดงผลเป็นอย่างไร ซึ่งจะอ่านลำดับมาทีละค่าแล้วตรวจสอบว่าเป็นรูปภาพและข้อความที่เท่าไร แล้วจึงไปตรวจสอบรูปแบบการแสดงผลของรูปภาพและข้อความนั้นๆ แล้วจึงนำรูปภาพและข้อความนั้นไปแสดง

3.1.3.7 เพลง

เพลงที่นำมาเก็บจะดึงมาจากตัวแปรซึ่งจะเก็บ path และชื่อของเพลงนั้นไว้ โดยที่ถ้ามีเพลงที่กำหนดไว้ก็จะนำ path และ ไฟล์ เพลงนั้นมาเก็บ แต่ถ้าไม่ได้กำหนดเพลงที่นำเสนออีกก็จะไม่เก็บ path และชื่อเพลงไว้ เช่น

C:\Program+++++ไฟล์\Gravity\RagnarokOnline\BGM\09.mp3 เพลง ++++++++

3.1.3.8 รูปแบบการเคลื่อนที่ของรูปภาพ

ก่อนที่จะมีการแสดงการเคลื่อนที่ของรูปภาพ จะต้องทำการกำหนดรูปแบบการแสดงผลให้กับรูปภาพ ซึ่งรูปแบบมีดังนี้

- เคลื่อนที่จากซ้ายไปขวา
- เคลื่อนที่จากขวาไปซ้าย
- เคลื่อนที่จากบนลงล่าง
- เคลื่อนที่จากล่างขึ้นบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิปภาพไปทางขวา

- คลิปภาพลง

เมื่อเลือกรูปแบบการเคลื่อนที่แล้ว รูปแบบการเคลื่อนที่นั้นจะเก็บติดกับรูปภาพนั้น ไปตลอด

3.1.3.9 รูปแบบการเคลื่อนที่ของข้อความ

ก่อนที่จะมีการแสดงการเคลื่อนที่ จะต้องทำการกำหนดรูปแบบการแสดงผลให้กับข้อความ ซึ่งรูปแบบมีดังนี้

- เคลื่อนที่จากซ้ายไปขวา

- เคลื่อนที่จากขวาไปซ้าย

- เคลื่อนที่จากบนลงล่าง

- เคลื่อนที่จากล่างขึ้นบน

เมื่อเลือกรูปแบบการเคลื่อนที่แล้ว รูปแบบการเคลื่อนที่นั้นจะเก็บติดกับข้อความนั้น ไปตลอด

3.1.3.10 ลำดับการแสดง

จะทำการตรวจสอบจากลำดับที่ได้กำหนดไว้ ในแต่ละหน้า เมื่อเราทำการแสดงงานที่นำเสนอ ก็จะตรวจสอบว่าลำดับการแสดงเป็นอย่างไร ซึ่งจะดึงลำดับการแสดงในหน้านั้นมาตรวจสอบว่าเป็นรูปภาพหรือข้อความ ถ้าเป็นรูปภาพจะทำการตรวจสอบว่าเป็นรูปภาพที่เท่าไร แล้วดึงรูปแบบการแสดงผลนั้นออกมา ในกรณีเดียวกัน ถ้าลำดับที่ดึงมาได้เป็นข้อความก็จะตรวจสอบว่าเป็นข้อความที่เท่าไร ซึ่งจะทำงานเริ่มต้นทำงานตั้งแต่ลำดับแรกที่เราอ่านได้จนถึงลำดับสุดท้ายในหน้านั้น

3.1.3.11 การลบรูปภาพ

จะลบรูปแบบการเคลื่อนที่ของรูปภาพนั้น และให้รูปภาพนั้นไม่มีการแสดงผลและจะกำหนดจุดเริ่มต้นใหม่ ขนาดความกว้างและความยาวของรูปเป็นค่าเริ่มต้น

3.1.3.12 การลบข้อความ

จะลบรูปแบบการเคลื่อนที่ของข้อความนั้น และให้ข้อความนั้นไม่มีการแสดงผลและกำหนดจุดเริ่มต้นใหม่ ค่าต่างๆ เช่น สีของข้อความ , Font ของข้อความ ,ขนาดของข้อความจะถูกกำหนดให้เป็นค่าเริ่มต้น

3.1.3.13 การเปิดไฟล์

จะไปทำการตรวจสอบที่ละบรรทัดว่าเป็น background,รูปภาพ,ข้อความ,รูปแบบการเคลื่อนที่, ลำดับการเคลื่อนที่ที่หน้าต่าง ๆ และเพลง ซึ่งรายละเอียดในไฟล์จะบอกไว้อยู่แล้ว

ถ้าค่าที่อ่านมาได้เป็น background จะไปตรวจสอบว่ามีรูปอยู่ใน path ที่อ่านเข้ามาได้หรือไม่ ถ้ามีก็จะนำ background นั้นมาแสดงตามลำดับ background ที่อ่านมาได้

ถ้าเป็นรูปภาพ รายละเอียดในไฟล์จะบอกว่าเป็นรูปภาพอะไร อยู่ใน path ไหนและเป็นภาพที่เท่าไร และจะเก็บตำแหน่งและขนาดความกว้างและสูงของรูปภาพไว้ด้วย จะอ่านออกมาว่าเป็นภาพ

ที่เท่าไรและ path ของรูปภาพคืออะไร ต่อมาจะแสดงรูป ณ ตำแหน่งที่อ่านมาได้ และแสดงรูปตามขนาดความกว้างและสูงตามค่าที่อ่านมาได้

ถ้าเป็นข้อความ รายละเอียดในไฟล์จะบอกว่าเป็นข้อความอะไร และจะเก็บตำแหน่งของข้อความ Font ของข้อความ, สีของข้อความ และขนาดของข้อความ จะอ่านออกมาว่าเป็นข้อความที่เท่าไร มาจัดแสดงข้อความ ณ ตำแหน่งที่อ่านมาได้และแสดงข้อความโดยมีขนาดของ Font, รูปแบบของ Font และสีของข้อความตามค่าที่อ่านมาได้

ถ้าเป็นรูปแบบการเคลื่อนที่ของรูปภาพ รายละเอียดในไฟล์จะบอกว่ารูปภาพนั้นมีรูปแบบการเคลื่อนที่เป็นอย่างไรมาก่อน จะอ่านออกมาว่าเป็นการเคลื่อนที่ของรูปที่เท่าไร ซึ่งถ้ามีรูปแบบการเคลื่อนที่ก็จะนำค่าที่อ่านมาได้ไปเก็บไว้ สำหรับแต่ละรูป แต่ถ้าค่าที่อ่านได้ไม่มีรูปแบบการเคลื่อนที่ ก็จะไม่นำค่าใดไปเก็บไว้

ถ้าเป็นการเคลื่อนที่ของข้อความ รายละเอียดในไฟล์จะบอกว่าข้อความนั้นมีรูปแบบการเคลื่อนที่เป็นอย่างไรมาก่อน จะอ่านออกมาว่าเป็นการเคลื่อนที่ของข้อความที่เท่าไร ซึ่งถ้ามีรูปแบบการเคลื่อนที่ก็จะนำค่าที่อ่านมาได้ไปเก็บไว้ สำหรับแต่ละข้อความ แต่ถ้าค่าที่อ่านได้ไม่มีรูปแบบการเคลื่อนที่ ก็จะไม่นำค่าใดไปเก็บไว้

ถ้าเป็นลำดับการแสดง ก็จะอ่านมาว่าเป็นลำดับการแสดงของหน้าที่เท่าไร แล้วนำรายการลำดับการแสดงของแต่ละหน้าที่อ่านมาได้นำมาเก็บไว้ ซึ่งค่าที่นำมาเก็บไว้นี้จะนำไปใช้ในการกำหนดลำดับการแสดงรูปภาพและข้อความต่อไป

ถ้าเป็นเพลง จะทำการอ่านว่าเพลงนั้นอยู่ใน path ใดและนำค่าที่อ่านได้นั้นมาเก็บไว้ ถ้าค่าที่อ่านมาได้ไม่มีการกำหนดเพลงไว้ ก็จะไม่นำค่าใดไปเก็บ

3.2 ขั้นตอนการเขียนโปรแกรม

ส่วนใหญ่แล้วขั้นตอนต่างๆ จะมีการทำงานเป็นเส้นตรงหมายถึงเมื่อทำงานเสร็จงานหนึ่งแล้ว ก็ทำงานต่อไปโดยไม่มีการทำงานวนกลับมาที่งานนั้น แต่ขั้นตอนการเขียนโปรแกรมมีการทำงานที่ต้องวนมาศึกษาเรื่อยๆ และทำการเขียนโปรแกรม และหากเขียนไม่สำเร็จก็ต้องทำการศึกษาเพิ่มเติม ดังแผนภาพต่อไปนี้

เหตุผลหลักของผู้ทำโครงการ คือแบ่งการทำงานออกเป็นหลายๆส่วน เพื่อต้องการจัดแบ่งหน้าที่การทำงาน ให้เป็นประเภทเดียวกัน ทำให้ง่ายต่อการตรวจสอบ แก้ไข หรือเพิ่มเติม หลักการแนวความคิดและรูปแบบของข้อมูลในแต่ละส่วนการทำงาน

ตัวอย่าง การเขียน โปรแกรมแสดงรูปภาพ

// กำหนดและเก็บรายละเอียดก่อนแสดงการเคลื่อนที่

```
Image2.Picture = Form1.Image2(0 + add).Picture
Form2.Image2.Visible = False
diff_left = Form1.Image2(0 + add).Left - Form1.bk(0).Left
diff_top = Form1.Image2(0 + add).Top - Form1.bk(0).Top
Image2.Left = diff_left * (Image1.Width / Form1.imgbk.Width)
Image2.Top = diff_top * (Image1.Height / Form1.imgbk.Height)
left_pic = Image2.Left
top_pic = Image2.Top
Image2.Width = Form1.Image2(0 + add).Width * (Image1.Width / Form1.imgbk.Width)
Image2.Height = Form1.Image2(0 + add).Height * (Image1.Height / Form1.imgbk.Height)
Form2.Image2.Visible = True
```

จากตัวอย่าง เป็นการเขียนโปรแกรมเพื่อแสดงรูปภาพ 1 ภาพหรือภาพนิ่งเท่านั้น แต่รายการแสดงภาพเคลื่อนที่ จำเป็นต้องมีการคำนวณเข้ามาเกี่ยวข้องการเคลื่อนที่ของภาพได้กล่าวมาแล้วว่า แต่ละภาพห่างกัน 100 pixels ซึ่งหากต้องการแสดงรูปภาพเคลื่อนที่ที่เราจึงทำการบวกค่าของระยะห่างแต่ละรูปภาพเพิ่มเข้าไป

// แสดงการทำงานของภาพเคลื่อนที่

```
If move1add = "เคลื่อนที่จากซ้ายไปขวา" Then
  xx = Form2.Image2.Left
  Form2.Image2.Left = 0
  yy = 0
  Do While yy < xx
    Form2.Image2.Left = Form2.Image2.Left + 100
    yy = yy + 100
  Loop
ElseIf move1add = "เคลื่อนที่จากขวาไปซ้าย" Then
  xx = Form2.Image2.Left
  Form2.Image2.Left = Form2.Width
  yy = Form2.Image2.Left
  Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Form2.Image2.Left = Form2.Image2.Left - 100
yy = yy - 100
Loop Until yy < xx
ElseIf move1add = "เคลื่อนที่จากบนลงล่าง" Then
xx = Form2.Image2.Top
Form2.Image2.Top = 0
yy = 0
Do While yy < xx
Form2.Image2.Top = Form2.Image2.Top + 100
yy = yy + 100
Loop
ElseIf move1add = "เคลื่อนที่จากล่างขึ้นบน" Then
xx = Form2.Image2.Top
Form2.Image2.Top = Form2.Height
yy = Form2.Image2.Top
Do
Form2.Image2.Top = Form2.Image2.Top - 100
yy = yy - 100
Loop Until yy < xx
ElseIf move1add = "คลิกภาพไปทางขวา" Then
xx = Form2.Image2.Width 'Image1.Width
Form2.Image2.Width = 0
yy = 0
Do While yy < xx
Form2.Image2.Width = Form2.Image2.Width + 100
yy = yy + 100
Loop
ElseIf move1add = "คลิกภาพลง" Then
xx = Form2.Image2.Height 'Image1.Height
Form2.Image2.Height = 0
yy = 0
Do While yy < xx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form2.Image2.Height = Form2.Image2.Height + 100
```

```
yy = yy + 100
```

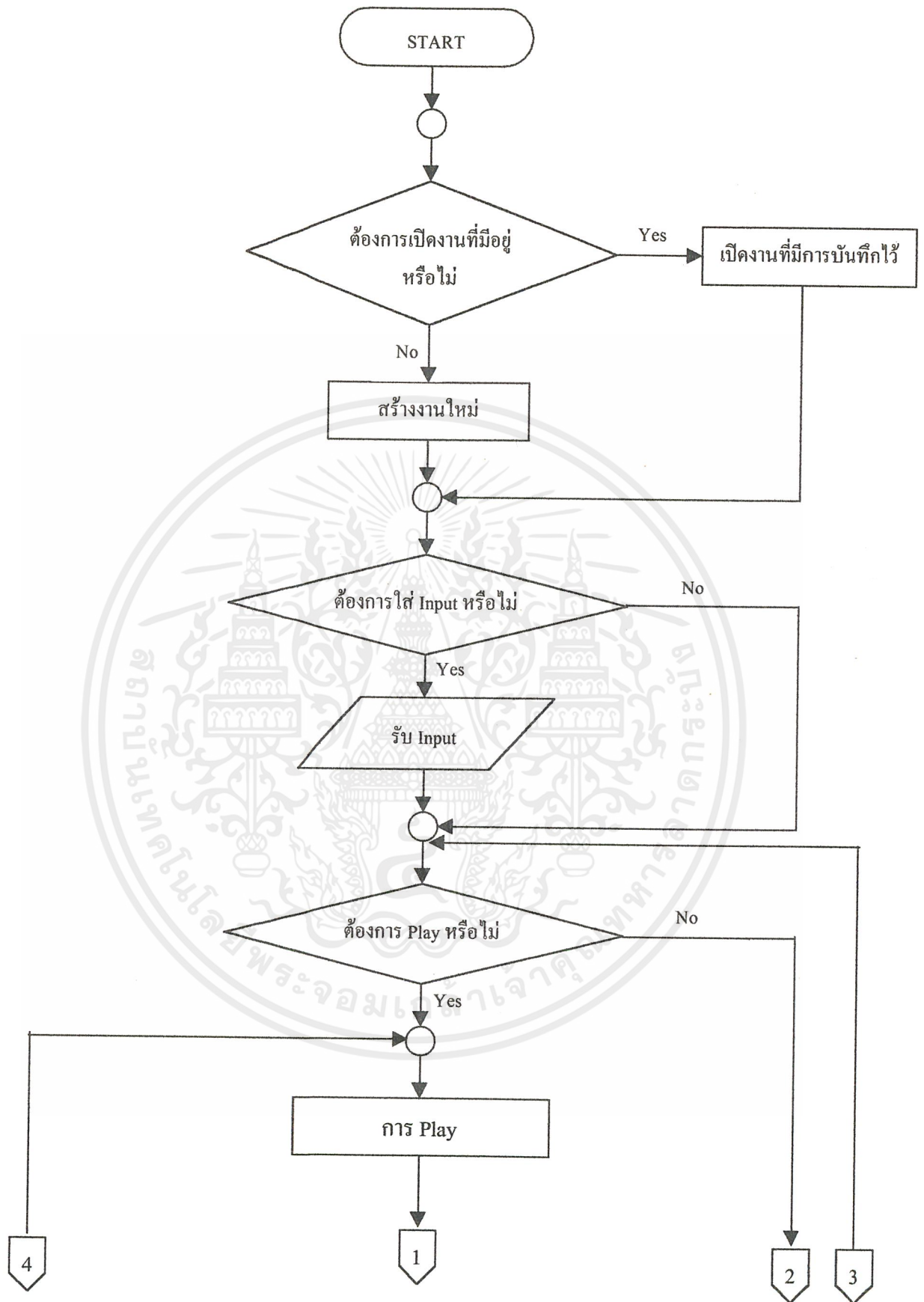
```
Loop
```

```
End If
```

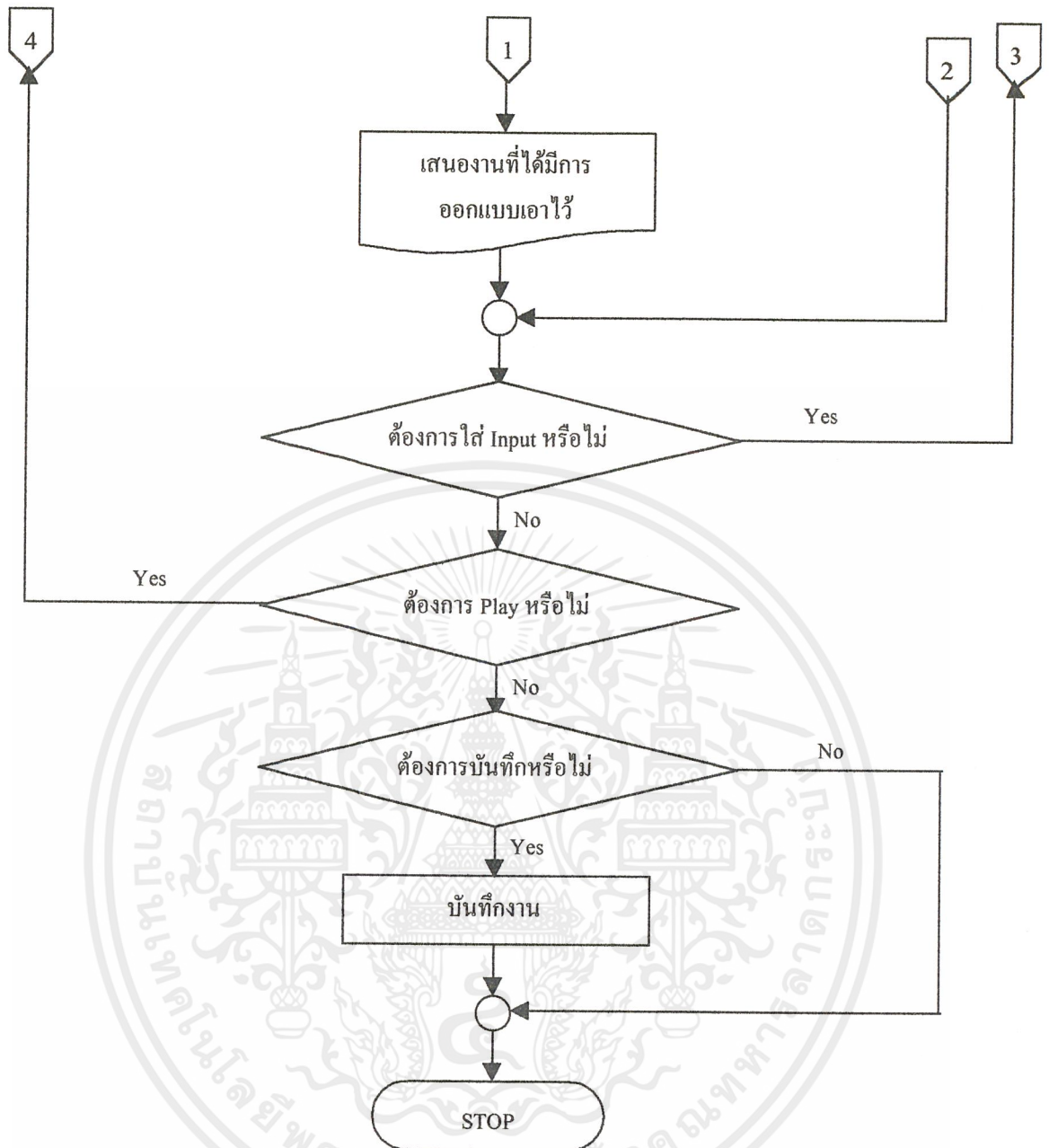
ในการสร้างภาพเคลื่อนที่นั้นอาศัยการคำนวณเพียงเล็กน้อยแต่อาจสร้างความสับสนให้บ้าง แต่หากได้ลองศึกษาและทดลองจนเข้าใจแล้วจะสามารถสร้างภาพเคลื่อนไหวได้ด้วยโปรแกรมเพียงเล็กน้อย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

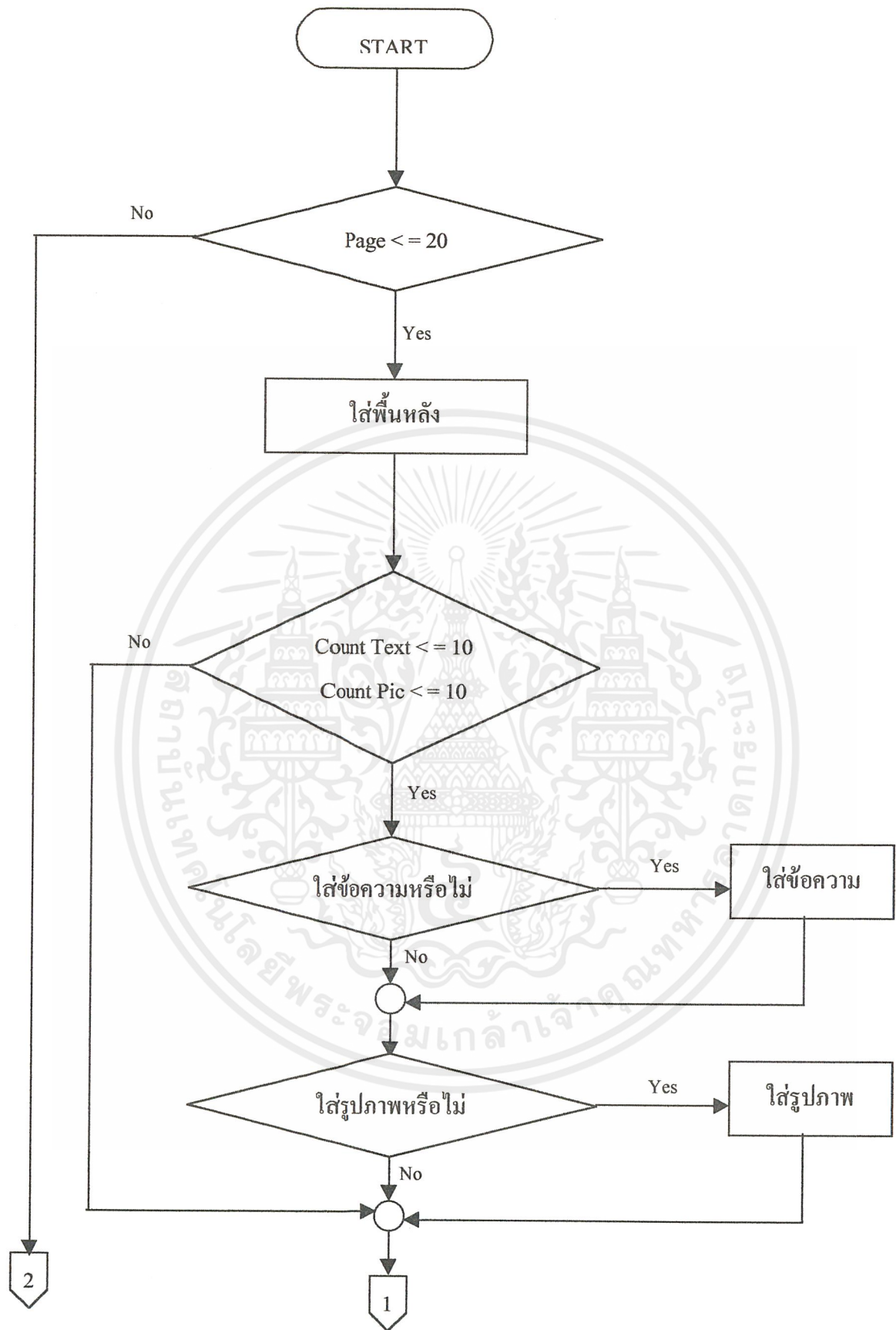


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

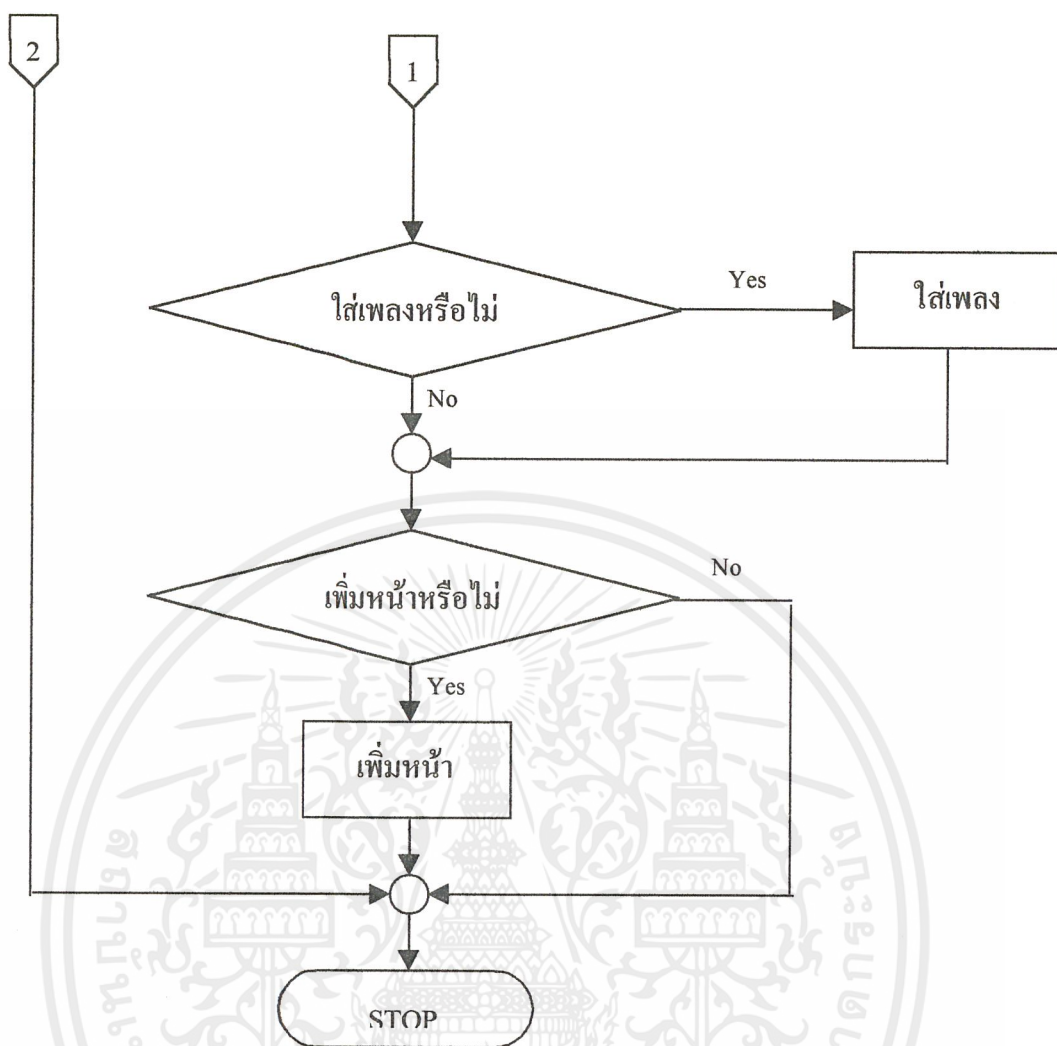


รูปที่ 3.7 แผนภาพแสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

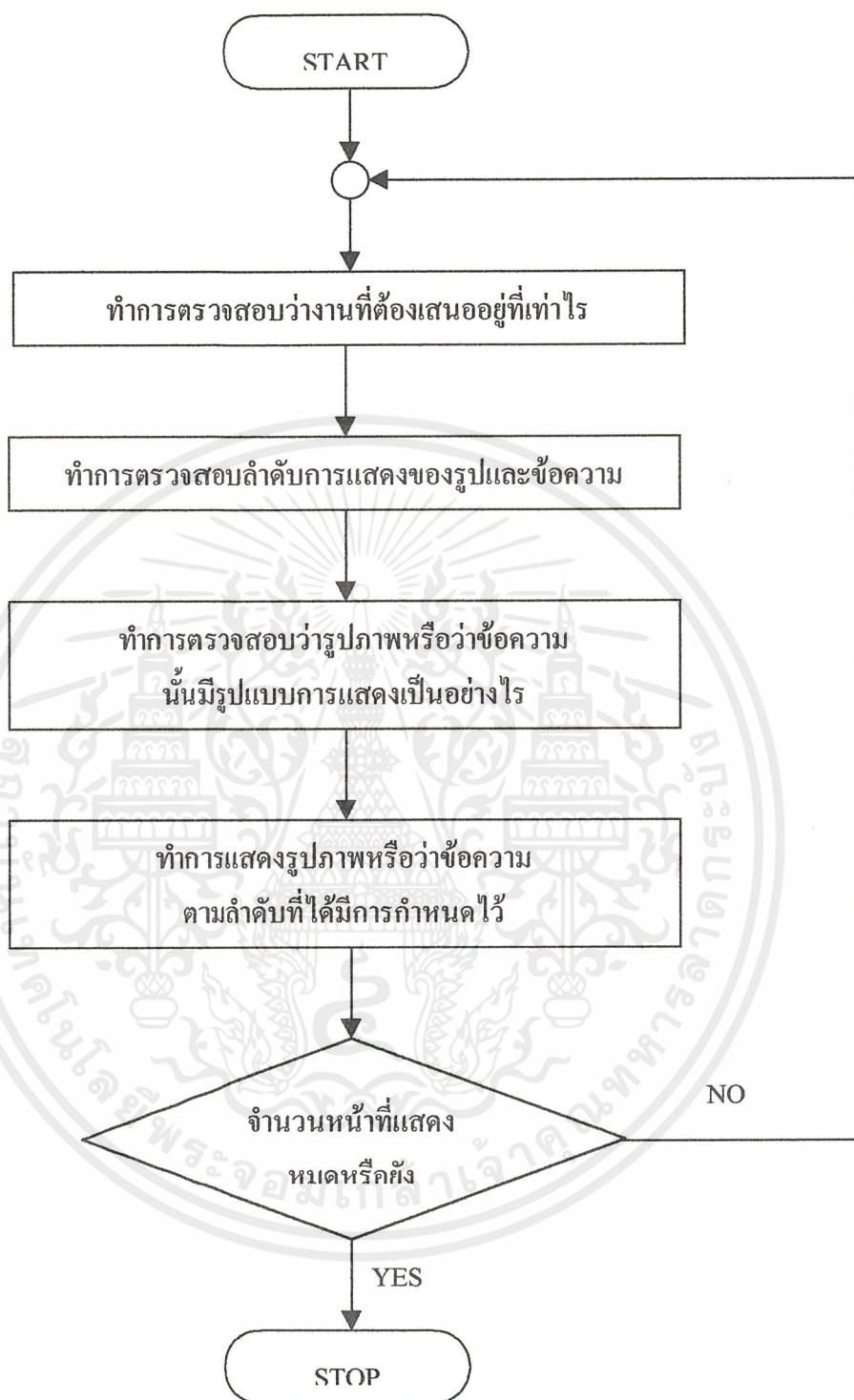


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



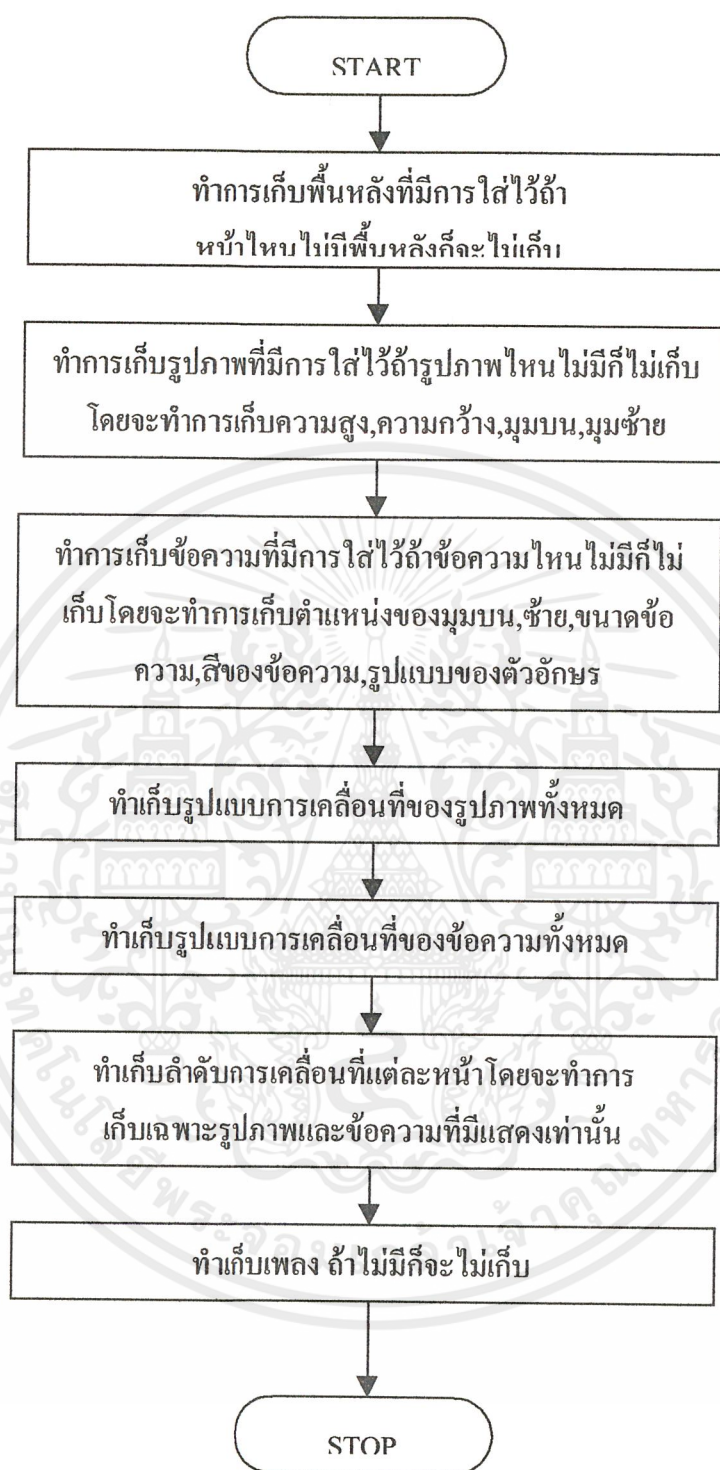
รูปที่ 3.8 แผนภาพแสดงการรับ Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แผนภาพแสดงขั้นตอนการ Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แผนภาพแสดงขั้นตอนการ Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

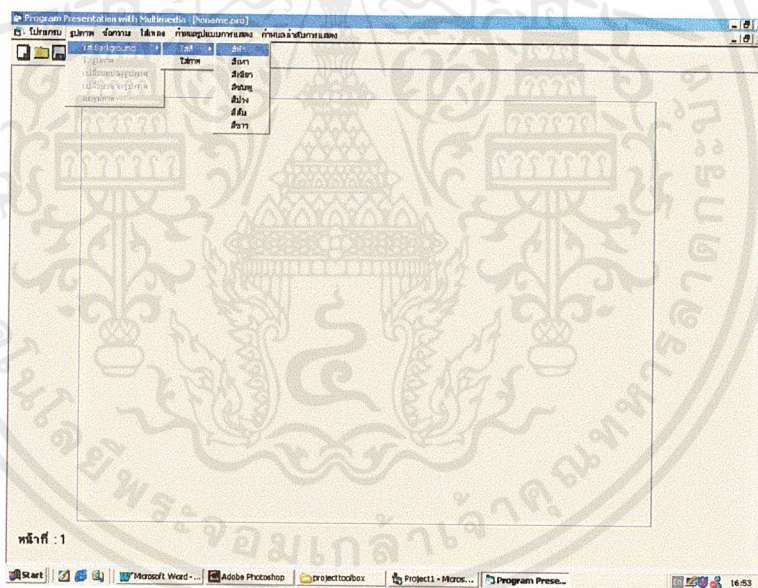
การใช้โปรแกรม

ลักษณะในการใช้งานของ โปรแกรมนำเสนอจากรูปแบบมัลติมีเดียสามารถใช้งานได้หลายรูปแบบ โดยที่ลักษณะของงานจะอยู่ในลักษณะของการนำเสนอรูปภาพและข้อความที่มีการกำหนดลักษณะการเคลื่อนที่ให้กับรูปภาพและข้อความ โดยจะมีลักษณะการใช้งานดังนี้

4.1 การใส่พื้นหลัง

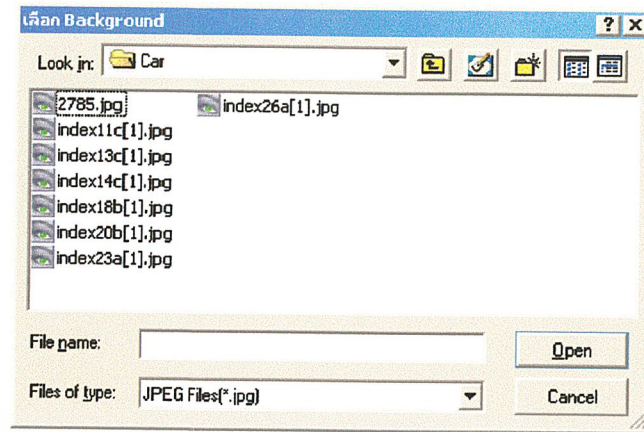
ในโปรแกรมก่อนที่จะทำงานอื่นๆ จะต้องมีการใส่พื้นหลังก่อนเป็นอย่างแรก โดยลักษณะของการใส่พื้นหลังจะสามารถใส่ได้ 2 ลักษณะ คือ

4.1.1 พื้นหลังแบบมีสี โดยลักษณะของพื้นหลังแบบมีสีนี้จะเป็นรูปแบบสำเร็จที่โปรแกรมกำหนดมาให้ โดยผู้ใช้สามารถที่จะทำการเลือกสีได้หลายสี วิธีการใช้งานทำได้โดยการเลือกเมนูรูปภาพ > ใส่ Background > ใส่สี ดังรูป



รูปที่ 4.1 วิธีการใส่พื้นหลังแบบมีสี

4.1.2 พื้นหลังแบบรูปภาพ โดยลักษณะของพื้นหลังแบบนี้จะเป็นพื้นหลังที่เป็นรูปภาพ ซึ่งเกิดจากผู้ใช้เลือกรูปภาพนั้นๆ มาใส่ วิธีการใช้งานสามารถทำได้โดยเลือกที่เมนูรูปภาพ > ใส่ Background > ใส่ภาพ หลังจากที่เราทำการเลือกเมนูนี้แล้วจะเกิดไดอะล็อกบ็อกซ์ขึ้นเพื่อให้ผู้ใช้สามารถเลือกรูปที่จะนำมาเป็นพื้นหลังได้ตามต้องการ ดังรูป

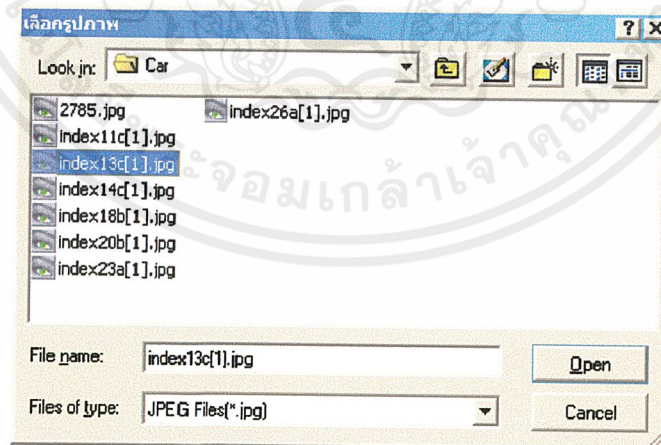


รูปที่ 4.2 วิธีการใส่พื้นหลังแบบรูปภาพ

โดยการใส่พื้นหลังทั้งสองวิธีนี้ ผู้ใช้สามารถทำการเปลี่ยนพื้นหลังได้ หลังจากมีการเลือกพื้นหลังไปแล้ว โดยจะทำในลักษณะเดียวกันกับที่ได้มีการกล่าวไว้ในตอนต้น

4.2 การใส่รูป

ผู้ใช้สามารถทำการใส่รูปภาพในการนำเสนองานได้ โดยรูปภาพที่จะนำมาใส่สามารถทำการใส่ได้หลายรูปในหนึ่งหน้าของงานในการนำเสนอ การใส่รูปภาพผู้ใช้สามารถทำได้โดยเลือกที่เมนูรูปภาพ > ใส่รูปภาพ เมื่อทำการเลือกเมนูดังกล่าวแล้วจะปรากฏไดอะล็อกบ็อกซ์ซึ่งผู้ใช้สามารถที่จะทำการเลือกรูปได้ตามต้องการ ดังรูป

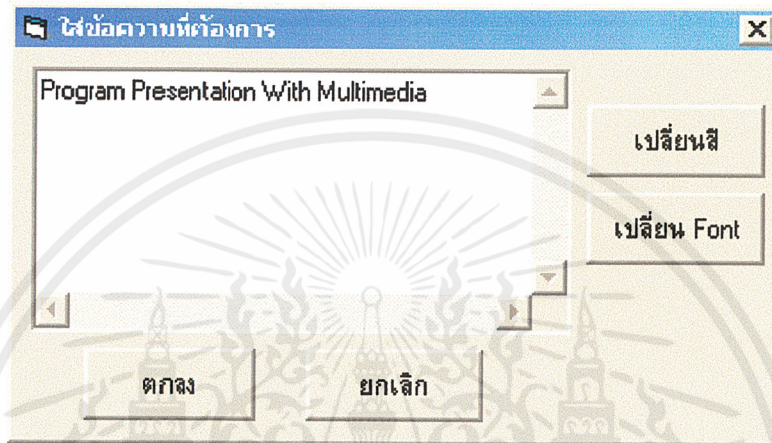


รูปที่ 4.3 การใส่รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การใส่ข้อความ

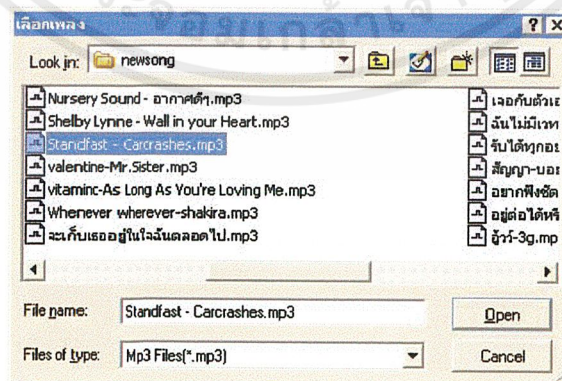
ผู้ใช้สามารถทำการใส่ข้อความในการนำเสนองานได้ โดยเลือกเมนูข้อความ>ใส่ข้อความ เมื่อทำการเลือกเมื่อดังกล่าวแล้วจะเกิดหน้าต่างข้อความขึ้นมาเพื่อให้ผู้ใช้สามารถทำการใส่ข้อความได้ตามต้องการและในหน้าต่างของข้อความนี้ ผู้ใช้สามารถทำการเลือกรูปแบบอักษร ขนาดตัวอักษร และสีของตัวอักษรได้ และในการใส่ข้อความนี้ถ้าผู้ใช้ต้องการใส่ข้อความเพื่อนำเสนอานมากกว่าหนึ่งข้อความก็สามารถทำได้ในลักษณะเดียวกับที่กล่าวไว้ในตอนต้นของการใส่ข้อความ ดังรูป



รูปที่ 4.4 การใส่ข้อความ

4.4 การใส่เพลง

ในการนำเสนอานนั้น ถ้าผู้ใช้ให้งานที่จะนำเสนอมีเพลงประกอบ ผู้ใช้สามารถทำการเลือกเมนูใส่เพลง เมื่อทำการเลือกเมื่อดังกล่าวจะปรากฏไดอะล็อกบ็อกซ์ของการใส่เพลงขึ้นมาเพื่อให้ผู้ใช้สามารถทำการเลือกเพลงได้



รูปที่ 4.5 การใส่เพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การกำหนดรูปแบบการเคลื่อนที่ให้กับรูปภาพ

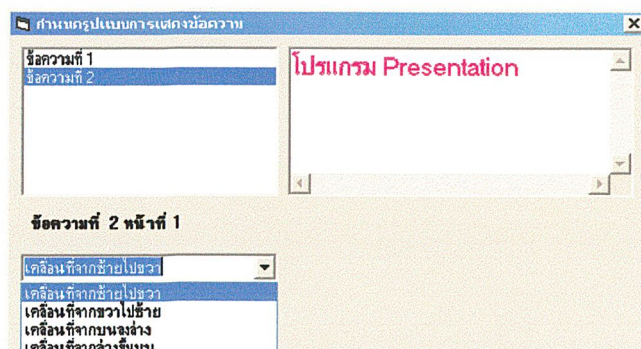
ผู้ใช้สามารถทำการกำหนดรูปแบบการเคลื่อนที่ให้กับรูปภาพเพื่อทำให้ลักษณะของงานมีความสวยงามมากขึ้น โดยสามารถทำการกำหนดรูปแบบการเคลื่อนที่ของรูปภาพได้โดยเลือกเมนูกำหนดรูปแบบการแสดง>กำหนดรูปแบบการแสดงรูปภาพ เมื่อผู้ใช้ทำการเลือกเมนูดังกล่าว จะปรากฏหน้าต่างของการกำหนดรูปแบบการเคลื่อนที่ให้กับรูปภาพ โดยรูปแบบการเคลื่อนที่ของรูปภาพจะมีหลายลักษณะผู้ใช้สามารถทำการเลือกให้เหมาะกับลักษณะของงานที่ต้องการได้ ดังรูป



รูปที่ 4.6 การกำหนดรูปแบบการแสดงผลของรูปภาพ

4.6 การกำหนดรูปแบบการเคลื่อนที่ให้กับข้อความ

ผู้ใช้สามารถทำการกำหนดรูปแบบการเคลื่อนที่ให้กับข้อความ เพื่อทำให้ลักษณะของงานมีความสวยงามมากขึ้น โดยสามารถทำการกำหนดรูปแบบการเคลื่อนที่ของข้อความ ได้โดยเลือกเมนูกำหนดรูปแบบการแสดง>กำหนดรูปแบบการแสดงผลข้อความ เมื่อผู้ใช้ทำการเลือกเมนูดังกล่าว จะปรากฏหน้าต่างของการกำหนดรูปแบบการเคลื่อนที่ให้กับข้อความ โดยรูปแบบการเคลื่อนที่ของรูปภาพจะมีหลายลักษณะผู้ใช้สามารถทำการเลือกให้เหมาะกับลักษณะของงานที่ต้องการได้ ดังรูป

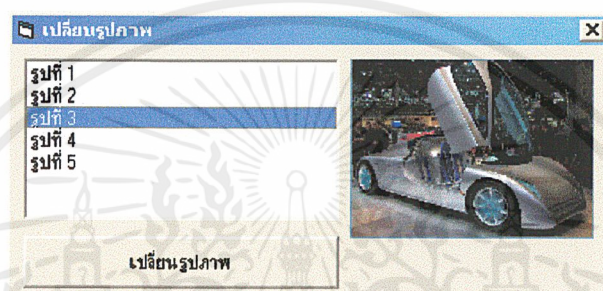


รูปที่ 4.7 การกำหนดรูปแบบการเคลื่อนที่ของข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การเปลี่ยนแปลงรูปภาพ

ผู้ใช้งานสามารถทำการเปลี่ยนแปลงรูปภาพที่ได้มีการเลือกไว้ได้ เนื่องจากรูปภาพที่เลือกอาจไม่ตรงกับความต้องการของผู้ใช้เอง โดยผู้ใช้งานสามารถทำการเปลี่ยนแปลงได้โดยเมนูรูปภาพ > เปลี่ยนแปลงรูปภาพ เมื่อทำการเลือกเมนูดังกล่าวแล้วจะเกิดหน้าต่างการเปลี่ยนแปลงรูปภาพเกิดขึ้น ซึ่งในหน้าต่างนี้จะปรากฏรูปภาพทั้งหมดที่ผู้ใช้งานมีการเลือกไว้ในการทำงานทั้งหมดและผู้ใช้สามารถทำการเปลี่ยนแปลงรูปได้โดยจะต้องทำการเลือกที่รูปที่ต้องการเปลี่ยนแปลง จากนั้นคลิกที่ปุ่มเปลี่ยนรูปภาพ เมื่อคลิกปุ่มดังกล่าวจะปรากฏไอคอนลูกบอกรูปภาพขึ้น ทำให้ผู้ใช้งานสามารถเลือกรูปภาพที่จะนำมาเปลี่ยนได้ ดังรูป



รูปที่ 4.8 การเปลี่ยนแปลงรูปภาพ

4.8 การเปลี่ยนแปลงข้อความ

ผู้ใช้งานสามารถทำการเปลี่ยนแปลงข้อความที่ได้มีการสร้างไว้ เนื่องจากข้อความที่สร้างไว้อาจไม่ตรงกับความต้องการของผู้ใช้เอง โดยผู้ใช้งานสามารถทำการเปลี่ยนแปลงได้โดยเมนูข้อความ > เปลี่ยนแปลงข้อความ เมื่อทำการเลือกเมนูดังกล่าวแล้วจะเกิดหน้าต่างการเปลี่ยนแปลงข้อความเกิดขึ้น ซึ่งในหน้าต่างนี้จะปรากฏข้อความทั้งหมดที่ผู้ใช้งานมีการสร้างไว้ทั้งหมดและผู้ใช้สามารถทำการเปลี่ยนแปลงข้อความได้โดยจะต้องทำการเลือกที่ข้อความที่ต้องการเปลี่ยนแปลง จากนั้นทำการพิมพ์ข้อความใหม่ลงในช่องข้อความและในหน้าต่างนี้ก็สามารถทำการเปลี่ยนแปลงรูปแบบของตัวอักษรขนาดและสีของข้อความได้

4.9 การเปลี่ยนแปลงขนาดรูปภาพ

ผู้ใช้งานสามารถทำการเปลี่ยนแปลงขนาดของรูปภาพที่ได้มีการสร้างไว้แล้วได้ โดยการเลือกเมนูรูปภาพ>เปลี่ยนขนาดรูปภาพ เมื่อเลือกเมนูนี้แล้วจะปรากฏหน้าต่างของการเปลี่ยนแปลงของรูปภาพซึ่งผู้ใช้งานสามารถทำการเปลี่ยนแปลงขนาดของรูปภาพได้

การเปลี่ยนแปลงขนาดของรูปภาพนั้น เพื่อความสะดวกของผู้ใช้งาน ผู้ใช้งานสามารถทำการเปลี่ยนแปลงขนาดของรูปภาพได้ที่หน้าจอของการออกแบบงาน โดยทำการเลือกภาพที่ต้องการจะทำการ

เปลี่ยนแปลงขนาด ที่บริเวณรูปนั้นจะเกิดกรอบของรูปขึ้น ผู้ใช้สามารถคลิกเมาส์ซ้ายค้างไว้ที่บริเวณกรอบของรูปและทำการลากเพื่อทำการเปลี่ยนแปลงขนาดตามความต้องการของผู้ใช้ได้

4.10 การลบรูป

เมื่อผู้ใช้ต้องการทำการลบรูปภาพที่ได้มีการสร้างไว้แล้ว ก็สามารถทำการลบรูปได้โดยการเลือกเมนูรูปภาพ > ลบรูปภาพ เมื่อเลือกเมนูนี้แล้วจะปรากฏหน้าต่างของการลบรูปภาพซึ่งผู้ใช้สามารถทำการเลือกรูปภาพที่จะทำการลบได้

4.11 การลบข้อความ

เมื่อผู้ใช้ต้องการทำการลบข้อความที่ได้มีการสร้างไว้แล้ว ก็สามารถทำการลบข้อความได้โดยการเลือกเมนูข้อความ > ลบข้อความเมื่อเลือกเมนูนี้แล้วจะปรากฏหน้าต่างของการลบข้อความซึ่งผู้ใช้สามารถทำการเลือกข้อความที่จะทำการลบได้

4.12 การเปลี่ยนแปลงลำดับการแสดง

ผู้ใช้สามารถเปลี่ยนแปลงลำดับการแสดงของรูปภาพและข้อความได้ เพื่อความเหมาะสมกับงานนั้นๆ โดยผู้ใช้สามารถทำการเปลี่ยนแปลงลำดับการแสดงโดยเลือกเมนูกำหนดลำดับการแสดง เมื่อเลือกเมนูนี้แล้วจะปรากฏหน้าต่างการเปลี่ยนแปลงลำดับการแสดงขึ้น ผู้ใช้สามารถทำการเปลี่ยนแปลงลำดับการแสดงได้ โดยเลือกปุ่มขึ้นหรือลงเพื่อเปลี่ยนแปลงลำดับการแสดง

4.13 การเพิ่มหน้า

เมื่อผู้ใช้ต้องการสร้างงานเพื่อนำเสนอ โดยที่ลักษณะของงานจะต้องมีการนำเสนอมากกว่า 1 หน้า ผู้ใช้สามารถทำการเพิ่มหน้าในการสร้างงานเพื่อนำเสนอได้โดยกดปุ่มเพิ่มหน้าเมื่อมีการเพิ่มหน้าแล้ว ผู้ใช้สามารถทำการสร้างงานในหน้าต่างๆ ได้โดยใช้ปุ่มย้อนกลับและไปข้างหน้าเพื่อไปสู่หน้าที่ต้องการสร้างงาน

4.14 การนำเสนองาน

เมื่อผู้ใช้ได้ทำการสร้างงานสอดคล้องกับความต้องการแล้วผู้ใช้สามารถทำการทดสอบหรือทำการนำเสนอได้โดยกดปุ่ม Play เพื่อทำการดูงานที่ได้มีการสร้างไว้

4.15 การบันทึกงาน

เมื่อผู้ใช้มีการสร้างงานเพื่อจะนำเสนอแล้ว ผู้ใช้สามารถทำการบันทึกงานเพื่อนำไปเปิดในคราวหลังได้ สามารถทำได้โดยการเลือกเมนู โปรแกรม > บันทึกไฟล์ เมื่อเลือกเมนูนี้แล้ว จะปรากฏไดอะล็อกบ็อกซ์ของการบันทึกงาน ผู้ใช้สามารถทำการบันทึกงานได้

4.16 การเปิดไฟล์

เมื่อผู้ใช้ต้องการเปิดงานที่ได้มีการบันทึกไว้ เพื่อนำเสนอ ผู้ใช้สามารถทำการเปิดได้โดยทำการเลือกเมนู โปรแกรม > เปิดไฟล์ เมื่อเลือกเมนูนี้แล้ว จะปรากฏไดอะล็อกบ็อกซ์ของการเปิดไฟล์ ผู้ใช้สามารถทำการเปิดไฟล์ที่ต้องการได้

4.17 การออกจากโปรแกรม

เมื่อผู้ใช้มีการสร้างงานแล้วต้องการจะออกจากโปรแกรม ผู้ใช้สามารถทำได้โดยเลือกเมนู โปรแกรม > ออกจากโปรแกรมหรือกดที่เครื่องหมายกากบาทที่มุมขวาด้านบนของหน้าต่างของตัวโปรแกรม



บทที่ 5

สรุปและข้อเสนอแนะ

5.1 บทสรุป

จากโปรแกรมนำเสนองานในรูปแบบมัลติมีเดีย มีข้อสรุปดังนี้

5.1.1 ในโปรแกรมสามารถทำการใส่รูปภาพเป็น Background และเป็น Background สีพื้นซึ่งมีการกำหนดมาให้ในโปรแกรมได้

5.1.2 สามารถทำการใส่รูปภาพ ข้อความ บนหน้าจอของการออกแบบงาน โดยสามารถทำการใส่รูปภาพและข้อความได้หลายอัน โดยที่ผู้ใช้สามารถเลือกที่จะวางรูปภาพและข้อความไว้ที่บริเวณใด ๆ ของพื้นที่การทำงานได้ และสามารถทำการเคลื่อนย้ายรูปภาพและข้อความไปไว้ ณ ตำแหน่งต่าง ๆ ของพื้นที่การออกแบบงานโดยใช้ Mouse ในการเคลื่อนย้ายรูปภาพและข้อความ

5.1.3 สามารถกำหนดรูปแบบการเคลื่อนที่ของรูปภาพ และข้อความได้ และเมื่อมีการนำเสนองานรูปภาพและข้อความสามารถเคลื่อนที่ได้ถูกต้องตามที่ได้กำหนดไว้ในหน้าของการออกแบบงาน

5.1.4 ในกรณีที่ต้องการขยายรูปภาพและข้อความ สามารถที่จะทำได้โดยใช้คำสั่งที่มีการกำหนดมาให้ หรือจะทำการย่อขยายโดยใช้ Mouse โดยสามารถทำการย่อขยายรูปภาพและข้อความได้อย่างถูกต้องตามที่ได้มีการกำหนดเอาไว้

5.1.5 เนื่องจากการออกแบบงาน จะมีการเลือกรูปภาพหรือใส่ข้อความผิด ในโปรแกรมสามารถที่จะทำการลบรูปภาพหรือข้อความนั้น ๆ ได้อย่างถูกต้อง

5.1.6 ในกรณีที่ต้องการเปลี่ยนแปลงรูปภาพหรือข้อความที่ได้มีการออกแบบไว้แล้ว เราสามารถทำการเปลี่ยนแปลงได้อย่างถูกต้อง

5.1.7 ในการนำเสนองาน สามารถที่จะเลือกเพลงที่นำมาใช้ในการนำเสนอได้ โดยไฟล์เพลงที่เลือกนั้นจะต้องอยู่ในรูปแบบไฟล์นามสกุล .Mp3 และเมื่อมีการแสดงสามารถเล่นเพลงได้อย่างถูกต้องตามที่ได้เลือกไว้

5.1.8 เมื่อมีการออกแบบงานภายในหน้าของการออกแบบงานและทำการนำเสนองาน รูปภาพและข้อความที่มีการกำหนดไว้อยู่ในตำแหน่งที่ถูกต้องตามที่ได้มีการกำหนดไว้ในหน้าของการออกแบบงานรวมถึงการเคลื่อนที่ต่างๆของรูปภาพและข้อความ เป็นไปอย่างถูกต้อง

5.1.9 ในการออกแบบงานนั้น สามารถที่จะทำการเพิ่มหน้าของการนำเสนองานได้ โดยในแต่ละหน้าสามารถใส่รูปภาพและข้อความ สามารถกำหนดรูปแบบการเคลื่อนที่ ของรูปภาพและข้อความได้โดยลำดับการเคลื่อนที่และรูปแบบการเคลื่อนที่ในแต่ละหน้าของการออกแบบ มีความเป็นอิสระต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.10 เมื่อมีการออกแบบงานที่ต้องการนำเสนอไปแล้ว สามารถที่จะทำการบันทึกงานนั้นไว้ได้ โดยมีการเก็บรายละเอียดของงานนั้น ๆ ได้อย่างถูกต้อง และเมื่อมีการเปิดงานที่ได้เคยมีการบันทึกไว้ ปรากฏว่ารายละเอียดต่าง ๆ ภายในแต่ละงานสามารถแสดงได้อย่างถูกต้องทั้งหมด เช่น ตำแหน่งของรูปภาพ และข้อความ การกำหนดรูปแบบการเคลื่อนที่ของรูปภาพและข้อความ ลำดับในการนำเสนองาน จำนวนหน้าของงานที่ได้มีการออกแบบไว้ และขนาดของรูปภาพก็มีความถูกต้อง และเมื่อมีการทดสอบ ก็สามารถทำการนำเสนองานได้จริง

5.2 ข้อจำกัดในการใช้งานโปรแกรม

การพัฒนาโปรแกรมนำเสนอแบบมัลติมีเดีย มีข้อจำกัดทางด้านฮาร์ดแวร์ และ ซอฟต์แวร์ ดังนี้

5.2.1 คุณสมบัติที่ใช้ในการทดสอบ

- ระบบปฏิบัติการ Microsoft Windows 98
- หน่วยความจำขนาด 64 MB
- หน่วยประมวลผลกลางที่มีความเร็วตั้งแต่ 300 MHz
- ความละเอียดของหน้าจอมีความละเอียด 1024*768

5.2.2 ข้อจำกัดของโปรแกรม

5.2.2.1 ข้อจำกัดในเรื่องหน่วยความจำ

เมื่อมีจำนวนของภาพและข้อความมากขึ้นจะทำให้มีการใช้หน่วยความจำของระบบมากขึ้นตามไปด้วยซึ่งจะทำให้เกิดปัญหาหน่วยความจำไม่เพียงพอต่อการใช้งานของโปรแกรมซึ่งจะทำให้โปรแกรมหยุดทำงานทันที

สาเหตุของการใช้หน่วยความจำมากเกินไปเนื่องมาจากการเขียนโปรแกรมที่ไม่กระชับ มีการวนลูปตรวจสอบค่ามากเกินไปและมีการสร้างวัตถุมารองรับล่วงหน้ามากเกินไปดังนั้นวัตถุในส่วนที่ไม่ได้ถูกใช้งาน ก็จะต้องมีการจองพื้นที่หน่วยความจำเช่นเดียวกันทำให้เป็นการสิ้นเปลืองหน่วยความจำ

5.2.2.2 การเคลื่อนที่ของรูปภาพและข้อความ

ปัญหาที่สำคัญอีกปัญหาหนึ่งก็คือ การเคลื่อนไหวของวัตถุ ซึ่งจะมีการกระพริบ ชกตัวอย่างเช่น เมื่อมีการกำหนดการเคลื่อนไหวให้กับรูปภาพ เมื่อทำการทดสอบภาพจะทำการวิ่งตามที่กำหนดไว้แต่ภาพที่วิ่งจะมีลักษณะกระพริบ

สาเหตุของการกระพริบของวัตถุ เนื่องจากโปรแกรมต้องมีการวาดภาพพื้นหลังและมีการวาดภาพที่นำมาเสนอสลับกันอยู่ตลอดเวลาในการเคลื่อนที่ ดังนั้นโปรแกรมจะต้องมีการทำการสลับภาพตลอดเวลา ดังนั้นจึงทำให้เกิดปัญหาการกระพริบของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.3 จำนวนรูปภาพและข้อความ

เนื่องจากภายในโปรแกรมการนำเสนองานแบบมัลติมีเดียนี้ จะมีการจำกัดจำนวนรูปภาพและข้อความที่จะนำเสนอ คือ สามารถที่จะทำการใส่รูปภาพและข้อความ ได้หน้าละ 10 รูปและ 10 ข้อความ จะไม่สามารถใส่รูปภาพและข้อความได้เกินกว่าที่กำหนดไว้

5.2.2.4 จำนวนหน้าของการนำเสนองาน

ภายในโปรแกรมการนำเสนองานแบบมัลติมีเดียนี้ จะมีการกำหนดจำนวนหน้าของการนำเสนอไว้ได้มากที่สุดเพียง 20 หน้าต่อ 1 งาน ดังนั้นจะไม่สามารถใช้กับงานที่มีจำนวนหน้าในการนำเสนอมากกว่า 20 หน้า

5.2.2.5 ไฟล์เพลงที่ใช้ในการนำเสนองาน

การนำไฟล์เพลงที่นำมาใช้เพื่อจะทำการนำเสนอ นั้น จะสามารถเลือกไฟล์เพลงมาใส่ได้ในกรณีที่ไฟล์เพลงอยู่ในประเภทของไฟล์นามสกุล .MP3 เท่านั้น ถ้าอยู่ในไฟล์ประเภทอื่น ๆ จะไม่สามารถนำมาใส่เพื่อใช้ในการประกอบการนำเสนองานได้

5.3 บทการวิจารณ์และแนวทางการพัฒนา

5.3.1 ข้อจำกัดในการแสดงรูปภาพและข้อความ

จากที่ได้มีการกล่าวไว้ว่าในโปรแกรมการนำเสนอแบบมัลติมีเดียนี้ จะสามารถสร้างงานโดยใส่รูปภาพและข้อความได้เพียงอย่างละ 10 รูปภาพและ 10 ข้อความต่อ 1 หน้า เนื่องจากขั้นตอนในการพัฒนาโปรแกรมเป็นแบบ Static คือ มีการสร้างรูปภาพและข้อความขึ้นมาทั้งหมดไว้ก่อนทุกครั้งที่มีการสร้างงาน ไม่ว่าจะผู้ใช้จะมีการใช้งานรูปภาพหรือข้อความจำนวนเท่าไรก็ตามในหน้านั้น แต่ต้องไม่เกินจำนวนที่ได้กำหนดไว้ จึงควรจะมีการพัฒนาโปรแกรมให้อยู่ในรูปแบบของ Dynamic คือเมื่อต้องการที่จะสร้างงานและมีการใส่รูปภาพและข้อความ จะมีการสร้างตามจำนวนที่มีการใช้งานจริงและในการพัฒนาโปรแกรมแบบ Dynamic นี้จะทำให้ผู้ใช้สามารถใส่รูปภาพและข้อความได้อย่างไม่จำกัด และทำให้ขนาดของงานไม่ใหญ่เกินไป ถ้าเทียบกับการพัฒนาโปรแกรมแบบ Static

5.3.2 ข้อจำกัดของจำนวนหน้า

ในการสร้างงานจะสามารถสร้างงานได้ 20 หน้าต่อการเสนองาน 1 งาน เนื่องจากการพัฒนาโปรแกรมเป็นไปในลักษณะเช่นเดียวกับการสร้างรูปภาพและข้อความ จึงควรจะมีการแก้ไขในลักษณะเดียวกับรูปภาพและข้อความ คือมีการพัฒนาให้โปรแกรมให้อยู่ในรูปแบบ Dynamic เพื่อให้การสร้างงานแต่ละงานไม่มีข้อจำกัดในเรื่องของจำนวนหน้าของการนำเสนองานเพื่อให้สอดคล้องกับความต้องการของผู้ใช้

5.3.3 การกระพริบของรูปภาพในการเคลื่อนที่

ในการกำหนดรูปแบบการเคลื่อนที่ให้กับรูปภาพและข้อความ เมื่อมีการแสดงรูปภาพหรือข้อความในหน้าจอของการนำเสนอ งาน รูปภาพและข้อความจะมีการกระพริบ เนื่องจากโปรแกรมจะต้องมีการวาดภาพ Background และวาดภาพที่จะแสดงสลับกันตลอดเวลา ทำให้เกิดการกระพริบของรูปภาพ จึงควรจะมีการพัฒนาโปรแกรมในส่วนนี้ โดยทำการนำภาพ Background เก็บลงใน Buffer โดยสามารถใช้ Double Buffer ในการแก้ปัญหาเพื่อไม่ให้เกิดปัญหาการสลับรูปภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. วิธีการติดตั้งโปรแกรม

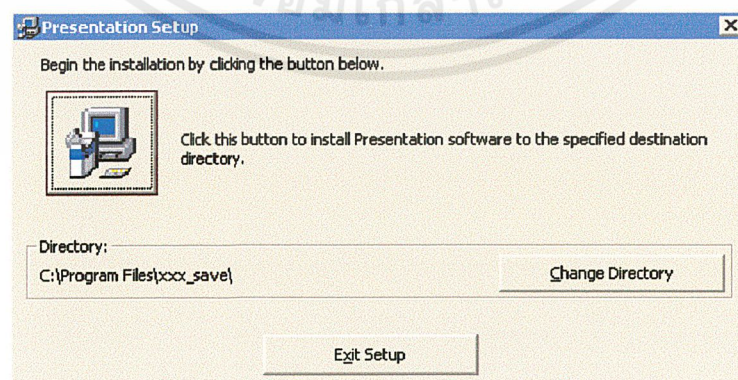
ขั้นตอนการติดตั้ง

1. บรรจแผ่นซีดีลงไปที่ซีดีรอม โปรแกรมติดตั้งจะทำงานอัตโนมัติ (Autorun)
2. หน้าจอจะปรากฏไดอะล็อกบ็อกซ์ดังรูป ให้ทำการเลือก Ok เพื่อเริ่มทำการติดตั้ง



รูปที่ ก.1 ไดอะล็อกบ็อกซ์เพื่อทำการติดตั้งโปรแกรม

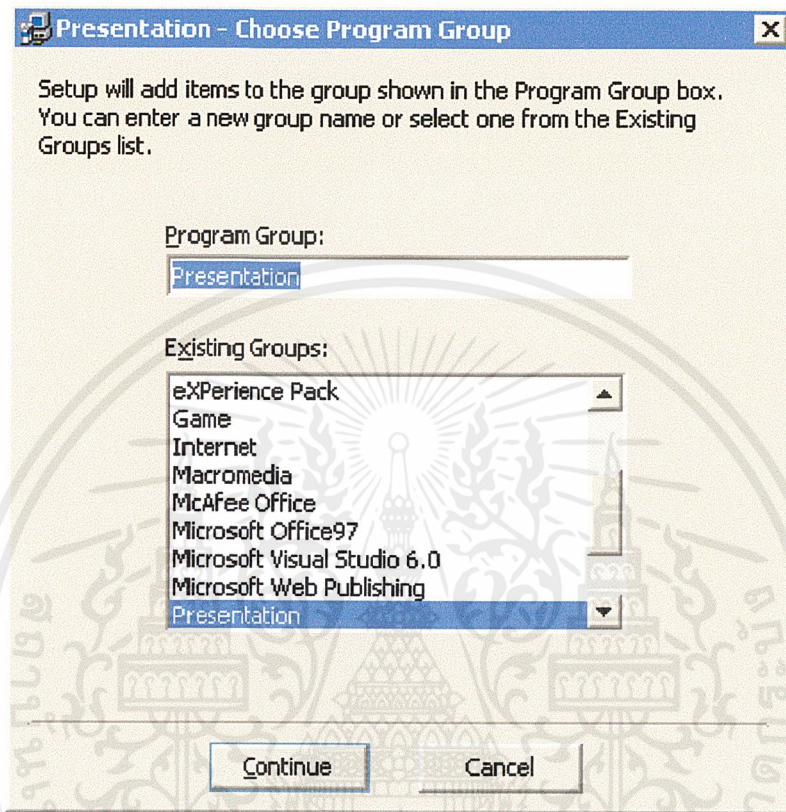
3. เมื่อเลือก OK แล้ว โปรแกรมจะให้เลือก Directory เพื่อทำการติดตั้งโปรแกรม โดยเราสามารถเลือก Change Directory เพื่อทำการเลือก Directory ที่จะติดตั้งโปรแกรมได้ แล้วคลิกที่รูปภาพทางด้านซ้ายเพื่อทำการติดตั้งโปรแกรม หรือสามารถเลือก Exit Setup เพื่อยกเลิกการติดตั้งโปรแกรม



รูปที่ ก.2 การเลือก Directory เพื่อทำการติดตั้งโปรแกรม

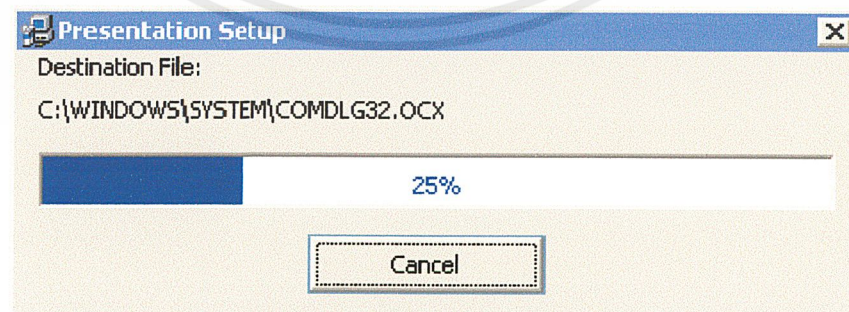
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากนั้น ให้เราเลือกว่าต้องการใส่โปรแกรมไว้ที่ Start Menu ที่ใด ถ้าเราต้องการใช้ Start Menu ที่สร้างไว้ก่อนแล้ว เราสามารถเลือกรายการใน List ได้เลย แต่ถ้าต้องการสร้างชื่อใหม่ ให้เราสร้างชื่อใหม่ในช่อง Program Group



รูปที่ ก.3 การเลือก Program Group

5. เริ่มการติดตั้งโปรแกรม



รูปที่ ก.4 เริ่มการติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. โปรแกรมทำการติดตั้งเรียบร้อยแล้ว Ok เพื่อกลับสู่หน้าจอวินโดว



รูปที่ ก.5 โปรแกรมติดตั้งเสร็จสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

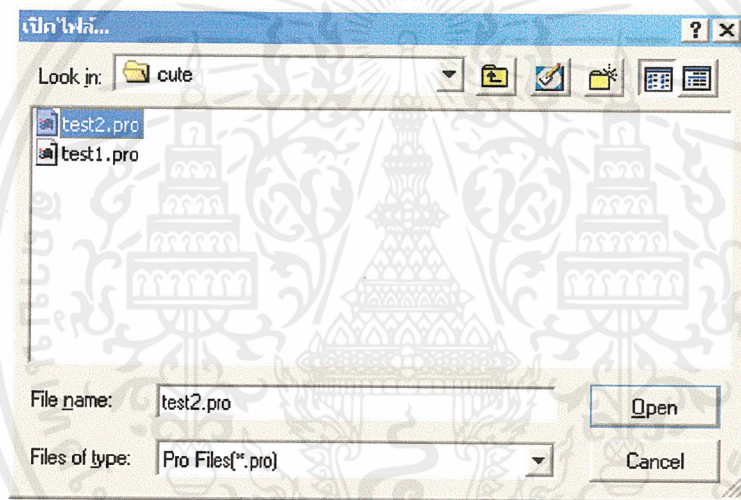
ภาคผนวก ข.

การจัดแบ่ง Menu

Menu ของโปรแกรม

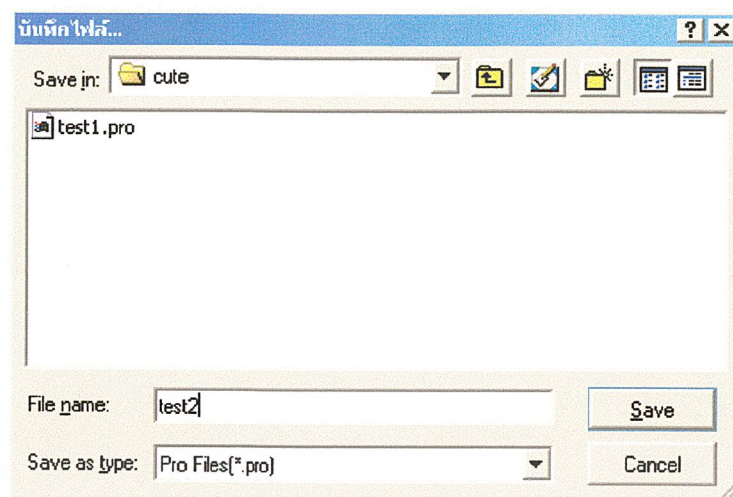
ประกอบด้วย

- การเปิดไฟล์ใหม่ ทำหน้าที่สร้างไฟล์ใหม่ เพื่อทำการออกแบบตั้งแต่เริ่มต้น
- การเปิดไฟล์ ทำหน้าที่เปิดไฟล์ เพื่อนำไฟล์ที่เปิดมาดูหรือมาทำการแก้ไขใหม่ จะมี dialog box เพื่อแสดงรายชื่อไฟล์ที่จะสามารถเปิดได้ ซึ่งไฟล์ที่สามารถจะเปิดมาทำการดูหรือแก้ไขจะมีนามสกุล .pro



รูปที่ ข.1 การเปิดไฟล์

- การบันทึกไฟล์ ทำหน้าที่บันทึกไฟล์หรืองานที่เราได้สร้างเอาไว้ ซึ่งการบันทึกจะมี 2 ลักษณะ คือ ถ้าเป็นไฟล์ที่ยังไม่เคยบันทึกมาก่อน จะสร้าง Dialog box ให้ผู้สร้างไฟล์สามารถจำแนกชื่อไฟล์ที่จะทำการบันทึกและอีกลักษณะคือเป็นไฟล์ที่เคยบันทึกมาก่อนแล้ว ไฟล์ที่ทำการบันทึกใหม่นี้ ข้อมูลของไฟล์ใหม่จะไปทับข้อมูลไฟล์เก่า ซึ่งไฟล์ที่ทำการบันทึกจะมีนามสกุล .pro



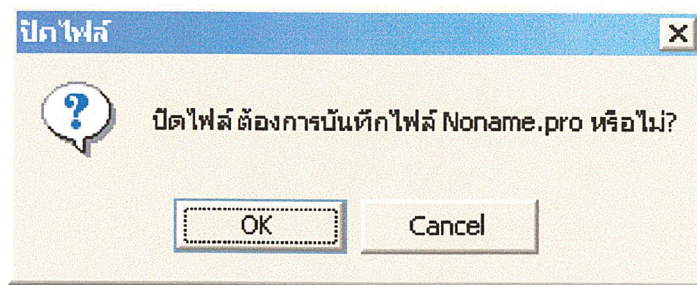
รูปที่ ข.2 การบันทึกไฟล์

- การบันทึกไฟล์เป็น ทำหน้าที่บันทึกไฟล์เช่นเดียวกัน แต่จะเป็นการบันทึกไฟล์แบบที่เป็นไฟล์ที่เคยบันทึกมาก่อนอยู่แล้ว และต้องการบันทึกเป็นไฟล์ชื่อใหม่ จะมี dialog box ให้ผู้สร้างงานที่ต้องการบันทึกไฟล์สามารถกำหนดไฟล์ชื่อใหม่ได้ โดยไฟล์ที่บันทึกมีนามสกุลเป็น .pro



รูปที่ ข.3 การบันทึกไฟล์เป็น

- ออกจากโปรแกรม จะทำหน้าที่ปิดโปรแกรม แต่ถ้ามีงานที่ทำการออกแบบค้างไว้อยู่ โปรแกรมจะมี msgbox ถามว่าต้องการบันทึกไฟล์ก่อนออกจากโปรแกรมหรือไม่ โดยถ้าเป็นไฟล์ที่ยังไม่เคยบันทึกมาก่อน จะสร้าง dialog box เพื่อให้ทำการกำหนดชื่อ file ก่อนปิดโปรแกรม หรือถ้าเป็นไฟล์ที่เคยบันทึกมาแล้ว จะทำการบันทึกทับ file เก่า



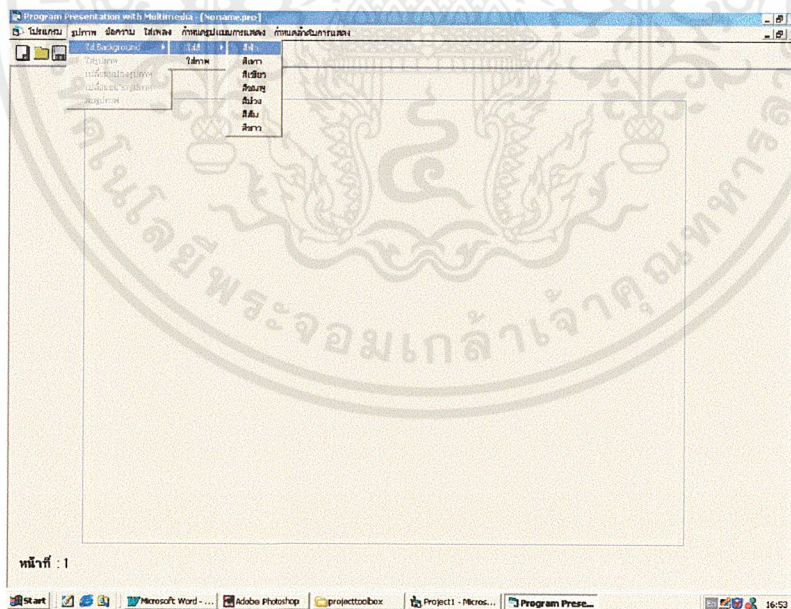
รูปที่ ข.4 การปิดไฟล์

Menu รูปภาพ

ประกอบด้วย

- ไล่ Background จะทำหน้าที่ไล่จากหลังให้กับแต่ละหน้า สามารถนำจากหลังมาไล่ได้ 2 รูปแบบ คือ

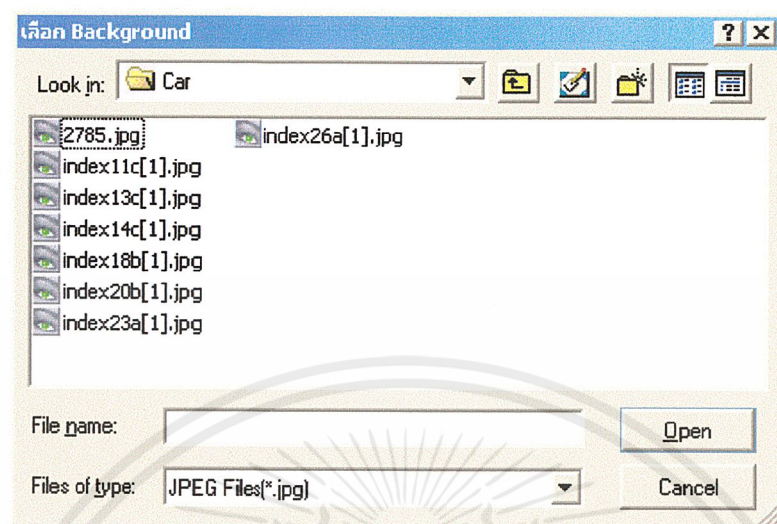
1). การไล่ Background ที โดยใช้ Background ที่โปรแกรมมีการกำหนดมาให้ซึ่งจะมีสีให้เลือกหลายสี คือ สีฟ้า เทา เขียว ชมพู ม่วง ส้ม และ สีขาว



รูปที่ ข.5 การเลือก Background แบบที่ 1

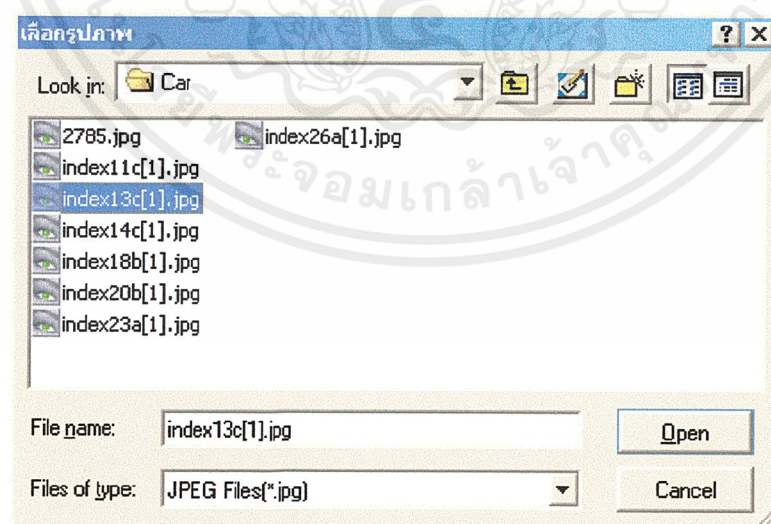
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2). การใส่ Background โดยใส่รูปเป็นฉากหลัง ซึ่งการนำรูปภาพที่นำมาใช้เราจะนำไฟล์ที่มีนามสกุลเป็น .jpg , .bmp



รูปที่ ข.6 การเลือก Background แบบที่ 2

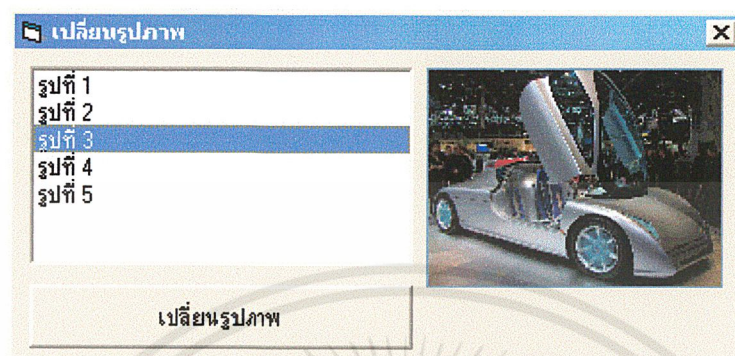
- ใส่รูปภาพ จะทำหน้าที่ใส่ภาพเพื่อทำการเสนอ จะแสดง Dialog box เพื่อให้เลือกรูปภาพที่จะนำมาแสดงได้ โดยรูปภาพที่นำมาแสดงนี้จะมีนามสกุลเป็น .jpg , .bmp และผู้ออกแบบสามารถลากรูปภาพไปวาง ณ ตำแหน่งต่างๆ ที่ต้องการได้



รูปที่ ข.7 การเลือกรูปภาพ

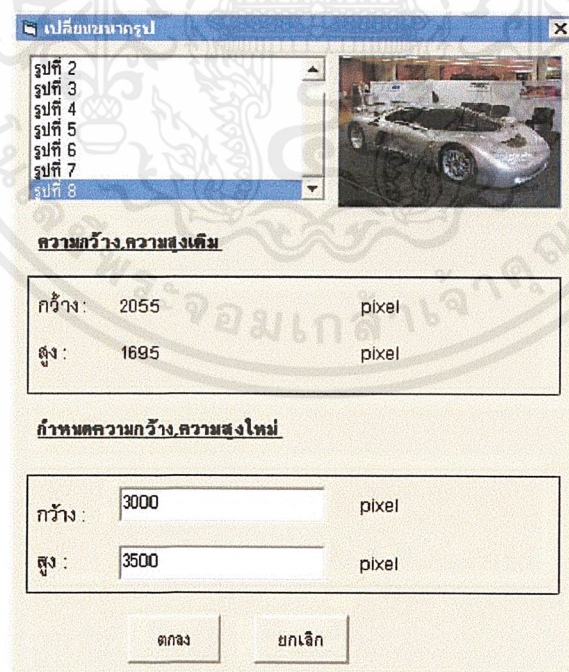
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **เปลี่ยนแปลงรูปภาพ** จะทำหน้าที่เปลี่ยนรูปภาพ โดยรูปภาพที่นำมาเปลี่ยนจะมีนามสกุล .jpg , .bmp โดยผู้ออกแบบจะต้องเลือกรูปภาพที่จะทำการเปลี่ยนก่อน และโปรแกรมจะแสดง dialog box เพื่อให้เลือกรูปภาพที่จะนำมาเปลี่ยน



รูปที่ ข.8 การเปลี่ยนแปลงรูปภาพ

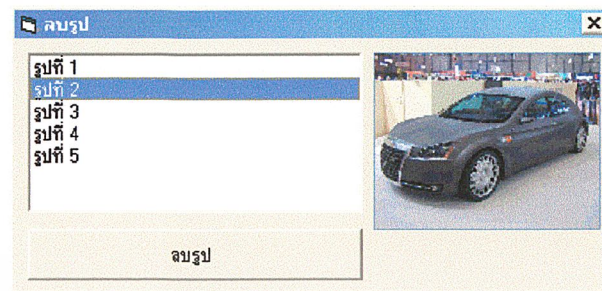
- **เปลี่ยนขนาดรูปภาพ** จะทำหน้าที่เปลี่ยนขนาดความกว้างและความยาวของรูปภาพ โดยหน่วยความกว้างและความยาวของรูปภาพจะอยู่ในหน่วย pixel ผู้ออกแบบจะต้องเลือกรูปภาพที่จะทำการเปลี่ยนขนาดก่อน ซึ่งจะแสดงความกว้าง,ยาว มาให้แล้วผู้ออกแบบสามารถใส่ความกว้าง,ยาวใหม่ตามที่ต้องการได้



รูปที่ ข.9 การเปลี่ยนแปลงขนาดรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ลบรูปภาพ** จะทำหน้าที่ลบรูปภาพที่ไม่ต้องการ โดยผู้ออกแบบต้องเลือกรูปภาพที่ต้องการจะลบก่อนจึงสามารถทำการลบได้



รูปที่ ข.10 การลบรูป

Menu ข้อความ

ประกอบด้วย

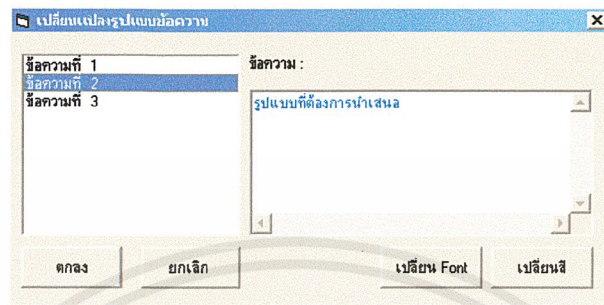
- **การใส่ข้อความ** จะทำหน้าที่ใส่ข้อความที่เป็นคำอธิบาย เพื่อให้เข้าใจงานที่นำเสนอได้ดียิ่งขึ้น โดยผู้ออกแบบสามารถใส่ข้อความที่ต้องการและใส่สีและ Font ให้กับข้อความนั้นได้และผู้ออกแบบสามารถลาข้อความไปวาง ณ ตำแหน่งต่าง ๆ ที่ต้องการได้



รูปที่ ข.11 การใส่ข้อความ

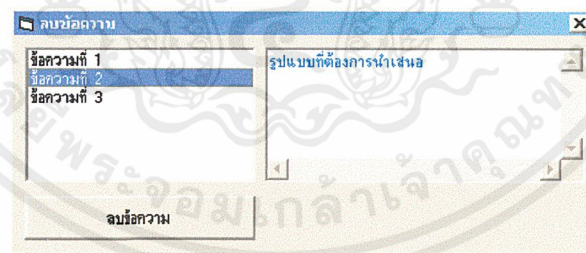
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเปลี่ยนแปลงรูปแบบข้อความ ทำหน้าที่เปลี่ยนแปลงรูปแบบข้อความในลักษณะต่าง ๆ เช่น เปลี่ยนคำในข้อความ , เปลี่ยน font , เปลี่ยนสี โดยก่อนที่จะทำการเปลี่ยนแปลงรูปแบบข้อความได้ จะต้องเลือกข้อความที่จะเปลี่ยนก่อน จึงจะสามารถเปลี่ยนแปลงได้



รูปที่ ข.12 การเปลี่ยนแปลงรูปแบบข้อความ

- ลบข้อความ ทำหน้าที่ลบข้อความที่ไม่ต้องการ โดยผู้ออกแบบต้องทำการเลือกข้อความที่ต้องการจะลบก่อน จึงจะสามารถลบได้

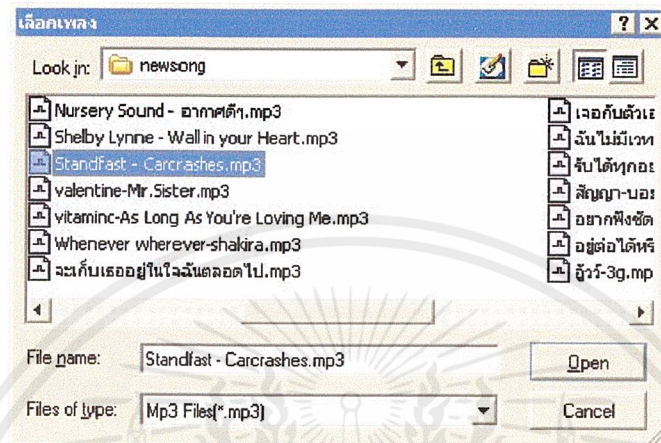


รูปที่ ข.13 การลบข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Menu ใส่เพลง

ทำหน้าที่เลือกเพลงที่จะนำมาใช้ในการนำเสนองาน โดยเสนอที่จะนำมาใช้ในการนำเสนอที่จะมีนามสกุล .mp3 จะแสดง dialog box เพื่อให้เลือกเพลงที่จะนำมาใช้ในการนำเสนอได้



รูปที่ ข.14 การเลือกเพลง

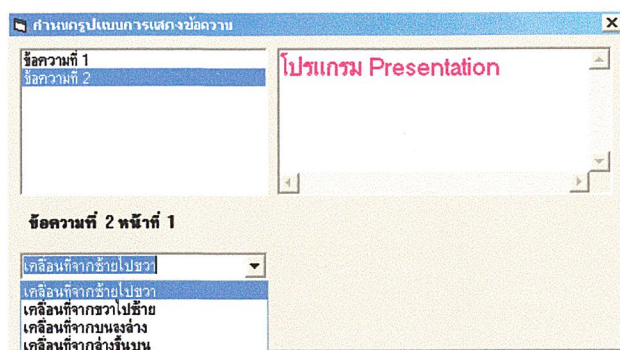
Menu กำหนดรูปแบบการเคลื่อนที่

ประกอบด้วย

- กำหนดรูปแบบการแสดงความเคลื่อนไหว ทำหน้าที่เลือกรูปภาพการเคลื่อนที่ให้กับแต่ละข้อความ ซึ่งรูปแบบการเคลื่อนที่มีดังนี้

- เคลื่อนที่จากซ้ายไปขวา
- เคลื่อนที่จากขวาไปซ้าย
- เคลื่อนที่จากบนลงล่าง
- เคลื่อนที่จากล่างขึ้นบน

โดยก่อนที่จะเลือกรูปแบบการเคลื่อนที่ ต้องเลือกข้อความที่จะทำการกำหนดรูปแบบการเคลื่อนที่ก่อนจึงสามารถเลือกรูปแบบการเคลื่อนที่แบบต่าง ๆ ได้

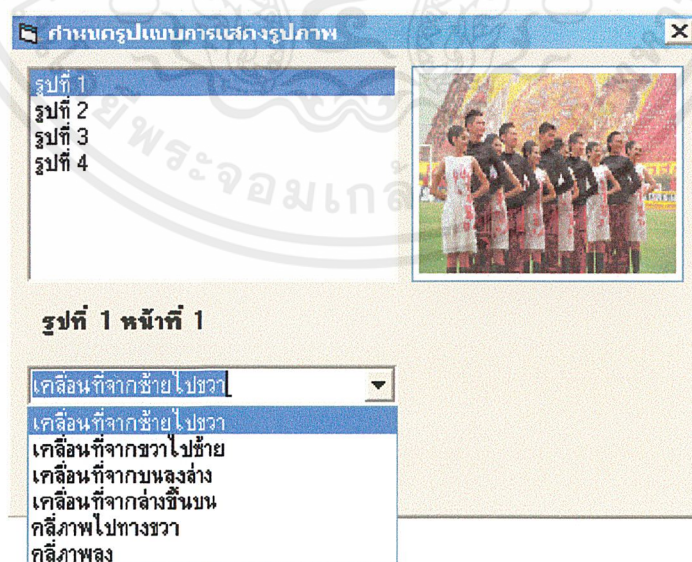


รูปที่ ข.15 การกำหนดรูปแบบการแสดงผลข้อความ

- กำหนดรูปแบบการแสดงผลรูปภาพ ทำหน้าที่เลือกรูปแบบการเคลื่อนที่ให้กับแต่ละรูปภาพ ซึ่งรูปแบบการเคลื่อนที่มีดังนี้

- เคลื่อนที่จากซ้ายไปขวา
- เคลื่อนที่จากขวาไปซ้าย
- เคลื่อนที่จากบนลงล่าง
- เคลื่อนที่จากล่างขึ้นบน
- คลี่ภาพไปทางขวา
- คลี่ภาพลง

โดยก่อนที่จะเลือกรูปแบบการเคลื่อนที่ ต้องเลือกรูปภาพก่อนที่จะทำการกำหนดรูปแบบการเคลื่อนที่ก่อนจึงสามารถเลือกรูปแบบการเคลื่อนที่แบบต่าง ๆ ได้

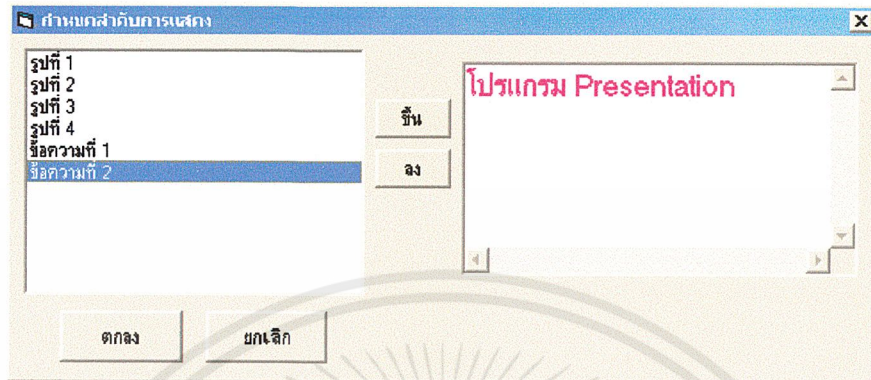


รูปที่ ข.16 การกำหนดรูปแบบการแสดงผลรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Menu กำหนดลำดับการแสดงผล

ทำหน้าที่แสดงลำดับการแสดงผลรูปภาพและข้อความในการนำเสนองาน ซึ่งผู้ออกแบบสามารถกำหนดลำดับการแสดงผลได้โดยต้องเลือกรูปภาพก่อนแล้วทำการสลับลำดับความต้องการ



รูปที่ ข.17 การกำหนดลำดับการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- พ.อ.เจนวิทย์ เหลืองอร่าม และปิยวิทย์ เหลืองอร่าม. 2543. การเขียนโปรแกรมสำหรับ Application ด้วย Visual Basic 6. กรุงเทพฯ : ซีเอ็ดยูเคชั่น
- ฉัททวุฒิ พิษผล และพิชิต สันติภูพานนท์. 2542. คู่มือเรียน Visaul Basic 6. กรุงเทพฯ : ซีเอ็ดยูเคชั่น
- ธงชัย พยุงกร และบุษกร กู้หลี. 2544. Q&A 500 ถาม-ตอบ Microsoft Visual Basic. กรุงเทพฯ : เอ็ม ไอ เอส ซอฟท์เทค



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้