



โปรแกรมระบบนำเสนอเรื่องราวหลายสื่อ
(Multimedia Story Presentation System)



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต
สาขาวิศวกรรมการวัดคุม
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม


คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมระบบนำเสนอเรื่องราวหลายสื่อ
(Multimedia Story Presentation System)

ผู้จัดทำ

นาย คุณากร	จารุประภัสสร	39013375
นาย ชัยวัฒน์	คันทวนิช	39013378
นาย สุระ	ศรีเข้มซ้อย	39013405


.....อาจารย์ที่ปรึกษา
(อาจารย์ ทรงชัย วีระทวีมาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมระบบนำเสนอเรื่องราวหลายสื่อ
(Multimedia Story Presentation System)

นาย คุณากร จารุประภัสสร

นาย ชัยวัฒน์ ตัณฑวนิช

นาย สุระ ศรีเข้มซ้อย

อาจารย์ ทรงชัย วีระทวีมาศ

อาจารย์ที่ปรึกษา

ปีการศึกษา 2538

บทคัดย่อ

โครงการสำหรับนำเสนอภาพและเสียงบนไมโครคอมพิวเตอร์นี้ จะทำการนำเสนอข้อมูลบนโปรแกรมวินโดว์ ซึ่งวินโดว์นั้นมีความสามารถในการนำเสนอภาพและเสียงได้หลากหลายเท่าที่ต้องการ โดยโปรแกรมนำเสนอภาพและเสียงนี้จะแบ่งเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของภาพและส่วนของเสียง ในส่วนของภาพจะมีลักษณะการนำเสนอแบบบิตแมพของภาพ ในส่วนของเสียง จะใช้ MCI (Multimedia Control Interface) ซึ่งเป็นชุดพัฒนาโปรแกรมสำหรับวินโดว์ มาใช้ในการบันทึกเสียง แล้วนำเสนอของภาพและเสียงมารวมกัน เพื่อนำเสนอพร้อมๆ กัน โดยการบันทึกข้อมูลของภาพและเสียงเก็บไว้ เพื่อให้ง่ายและสะดวกต่อการใช้งาน โดยสามารถกำหนดคุณสมบัติข้อมูลของภาพและเสียงได้ และสามารถแก้ไข, ลบ หรือเพิ่มลำดับการนำเสนอข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MULTIMEDIA STORY PRESENTATION SYSTEM

Chunagurn Charuprapatsorn

Chaiwat Tantavanich

Sura Srichamchoi

Advisor

Songchai Weerataweemas

1998

Abstract

The project presents the presentation of images and sounds on microcomputer which are operated on Windows system which is the operating system that has a capable of presenting everyday multimedia sources. The multimedia presenting program is divided into two main parts, images part and sounds part. The images part presents in bitmap form, which is the graphics property of windows, by applying the bitmaps' properties for editing images bitmap data. The Sound part uses MCI (Multimedia Control Interface), The program development kit for windows, for recordings. Then, Later, the images part and the sounds part, which have been recorded for ease of use, are mixed together for the synchronize of media. Image and sounds' properties can be changed or the sequence of presentation can be rearranged.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทหน้า	
บทที่ 1 กราฟฟิกส์ในคอมพิวเตอร์เฟส	1
1.1 DIB (Device Independent Bitmap)	2
1.2 การทำอินเตอร์เฟสกับ DIB	2
บทที่ 2 รูปภาพบิตแมพ (Bitmap) และการแสดงผล	3
2.1 การสร้างรูปบิตแมพ	3
2.2 การแสดงรูปบิตแมพ	8
2.3 การใส่สีให้กับรูปบิตแมพที่เป็นสีเขียว (ขาวดำ)	13
2.4 การลบรูปบิตแมพ	13
2.5 คอนเท็กซ์อุปกรณ์ (Device Context)	14
2.6 สี (Color)	15
2.7 จานสี(Palette)	16
บทที่ 3 MCI (Media Control Interface)	18
1.1 เกี่ยวกับ MCI	18
1.2 คำสั่ง Strings และ Messages ของ MCI	18
1.3 คำสั่ง Messages	19
1.4 การส่งคำสั่ง Messages	19
1.5 รูปแบบของคำสั่ง Messages	19
1.6 การแบ่งประเภทคำสั่งของ MCI	20
1.7 การใช้คำสั่ง Messages ของ MCI	22
บทที่ 4 การเขียนโปรแกรมบน Windows	30
4.1 ลักษณะของการเขียนโปรแกรมบน Windows	30
4.2 หลักการทำงานของโปรแกรม Multimedia Story Presentation System	32
4.3 บล็อกไดอะแกรมของโปรแกรม Multimedia Story Presentation System	33
4.4 การใช้งานโปรแกรม Multimedia Story Presentation System	39
บทที่ 5 สรุปผลและวิจารณ์	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ไม่ว่าจะเป็นด้านการติดต่อสื่อสาร ด้านการศึกษา หรือแม้แต่การนำคอมพิวเตอร์มาใช้ในด้านการนำเสนอ หรือแจ้งข่าวสาร ดังที่เห็นกันอยู่ทั่วไป

ในส่วนของตัวคอมพิวเตอร์เองนั้น ก็ได้มีการพัฒนาขึ้นเรื่อยๆ เป็นลำดับไป ทั้งในด้านของฮาร์ดแวร์ เช่น ความเร็ว หรือความสามารถในการเก็บข้อมูล และซอฟต์แวร์ ซึ่งได้มีการพัฒนาโปรแกรมเพื่อสนับสนุนในส่วนของฮาร์ดแวร์ด้วย ในส่วนของซอฟต์แวร์นี้ก็ได้มีการพัฒนาโปรแกรมเรื่อยๆ มา จนในปัจจุบันได้มีโปรแกรมที่พัฒนาจนได้รับความนิยมกันอย่างมากได้แก่โปรแกรม Windows ซึ่ง Windows นี้เองมีความสามารถต่างๆ มากมาย รวมทั้งความสามารถทางด้านกราฟิกในการนำเสนอภาพในลักษณะ Bitmap ที่คล่องตัวและความสามารถในการนำเสนอเสียง แบบ Waveform-audio ที่สะดวกในการใช้งาน

จากความสามารถของ Windows ดังที่กล่าวมาแล้วนั้น จึงได้มีการนำคุณสมบัติทางด้านกราฟิกและทางด้านเสียงของ Windows มาประยุกต์ใช้กับงานด้านการนำเสนอข้อมูลในรูปแบบของมัลติมีเดีย จึงได้ “ โปรแกรมนำเสนอเรื่องราวหลายสื่อ ”

บทที่ 1

กราฟิกดีไวซ์อินเตอร์เฟซ

GDI (Graphics Device Interface) คือหัวใจของความสามารถในการทำกราฟิกของ วินโดวส์ ฟังก์ชันการเขียนทั้งหมดสำหรับเส้นและที่เป็นรูปทรงจะอยู่ใน GDI เช่นเดียวกับฟังก์ชัน การจัดการเรื่องสีและฟอนต์ สมรรถภาพของวินโดวส์หลายอย่างผูกติดอยู่กับสมรรถภาพของ GDI และรหัส GDI จำนวนมากถูกทำขึ้นมาด้วยภาษาแอสเซมบลีของ 386 ที่ระดับแอฟพลีเคชัน วินโดวส์จะจัดหาออบเจ็กต์ตรรกะที่เรียกว่า ดีไวซ์คอนเท็กซ์ (*device contexts (DC)*) ที่บอกถึง สภาพปัจจุบันของเป้าหมายการเขียน GDI บางอย่าง DC สามารถอธิบายลักษณะอุปกรณ์เอาต์พุต หรือตัวแทนของอุปกรณ์ เช่น แอปพลีเคชันจะรับ DC สำหรับเครื่องพิมพ์เอาต์พุตหรือ สำหรับการ ปฏิบัติการที่มาจากฐานหน่วยความจำทั้งหมด แอปพลีเคชันทำการจัดการ DC โดยวิธีของ Win32 API เท่านั้น โครงสร้างข้อมูล DC ที่แท้จริงจะถูกซ่อนไว้จากแอปพลีเคชันเสมอในขณะใดขณะหนึ่ง DC จะบรรจุข้อมูลข่าวสารเกี่ยวกับออบเจ็กต์ต่างๆ เช่น ปากกาปัจจุบัน (สำหรับเขียนเส้นต่างๆ) แปรงปัจจุบัน (สำหรับเติมบริเวณต่างๆ) การเลือกสี และตำแหน่งและมิติต่างๆ ของเป้าหมายการ กวาดรูปทางตรรกะ

กฎเกณฑ์สำคัญในการใช้วินโดวส์ และแอปพลีเคชันวินโดวส์บนช่วงการกระจายที่กว้างขวาง ของฮาร์ดแวร์เป้าหมายคือ การเป็นอิสระจากอุปกรณ์ที่ฝังตัวอยู่ใน API ของวินโดวส์แอปพลีเคชัน จะใช้ DC และออบเจ็กต์ตรรกะอื่นๆ เมื่อทำการเรียกใช้ฟังก์ชัน GDI ตัว GDI เองจะจัดการกระบวนการ เปลี่ยนข้อมูลเข้าสู่รูปแบบที่เหมาะสมกับการใช้ของดีไวซ์ใดเวอร์บางอย่าง และไครเวอร์ จะ ไปจัดการในเรื่องการวางตัวแทนของคำขอบนอุปกรณ์เอาต์พุต ยกตัวอย่างแอปพลีเคชันอาจจะเรียก ระบบเพื่อขอหน้าต่างหลักมาทำการรีเพนต์ (Repainted) ช่วงการปฏิบัติรีเพนต์ในระหว่างคำขออื่นๆ GDI อาจจะบอกไครเวอร์ว่า บนจอภาพวาดเส้นความกว้างหนึ่งพิกเซลสีดำจากตำแหน่ง (0,48) ถึง ตำแหน่ง (639,48) ถ้าอุปกรณ์เป็นเครื่องพิมพ์แบบดอตเมทริกซ์ไม่สามารถทำการปฏิบัติการในการ เขียนเส้นเช่นนั้นได้ GDI ก็จะกระจายคำขอให้เป็นการปฏิบัติการที่ง่ายขึ้น เช่น ดีไวซ์ไครเวอร์จะได้รับชุดของการเรียกใช้ที่บอกให้มันเขียนแต่ละจุดขึ้นมา สถาปัตยกรรมแบบนี้จะปล่อยให้แอปพลีเคชันเป็นอิสระจากปัญหาที่ขึ้นอยู่กับอุปกรณ์ และยอมให้วินโดวส์ใช้ได้แม้แต่ฮาร์ดแวร์ที่ง่ายที่สุดดัง เช่นอุปกรณ์เอาต์พุตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIB (Device Independent Bitmap)

วินโดวส์ บิตแมพ (bitmap) คือตัวแทนที่มีรากฐานมาจากหน่วยความจำของลำดับการปฏิบัติการ GDI ที่สมบูรณ์ ออบเจ็กต์ที่ได้มาจะเหมาะสมสำหรับการแสดงผลในทันทีบนอุปกรณ์เอาต์พุตที่คอมแพคติเบิ้ล และในกรณีของบิตแมพที่เป็นอิสระจากอุปกรณ์การประมวลผลต่ำสุดเพิ่มเติม เป็นการเตรียมออบเจ็กต์สำหรับเอาต์พุตไปสู่อุปกรณ์ต่างๆ บิตแมพจะปรากฏออกมาในไฟล์ต่างๆ (เช่น วอลล์เปเปอร์เดสก์ทอป) เป็นเสมือนทรัพยากรของแอปพลิเคชัน (เช่นรูปภาพบนปุ่มทูลบาร์) และในหน่วยความจำที่แอปพลิเคชันและดีไวซ์ไครเวอร์สามารถสร้างและจัดการพวกมันได้โดยตรง การแสดงผลเดสก์ทอปทั่วทั้งหมด ก็คือตัวมันเองที่เป็นบิตแมพขนาดใหญ่และรหัสที่เกี่ยวข้องกับการปรับจอภาพให้ทันต่อเวลาจะมีความสำคัญต่อสมรรถภาพของระบบ

DIB ของวินโดวส์ ทราบถึงความจริงของสภาพปัจจุบันของฮาร์ดแวร์แสดงผล โดยการใช้ความสามารถในการจัดการบิตแมพที่ทำอะไรได้ผลกับอุปกรณ์เฟรมบัฟเฟอร์แบบราบเรื่องสี่ ในคำศัพท์ทางฮาร์ดแวร์ หมายความว่า อุปกรณ์เอาต์พุตจัดเตรียมพื้นที่ว่างหน่วยความจำขนาดใหญ่แบบเส้นตรง พร้อมด้วยแต่ละพิกเซลของจอภาพที่สามารถแอดเดรสโดยตรงได้เป็นเสมือนตำแหน่งหน่วยความจำ และที่รวมไปด้วยกับแต่ละพิกเซลก็คือสี ซึ่งแทนด้วยจำนวนบิต DIB จัดการ 1, 4, 8, 16 หรือ 32 บิตต่อสีพิกเซล เป็นการในช่วงการแสดงผลของมันจากการแสดงผลโมโนโครมง่ายๆ ไปยังอุปกรณ์เอาต์พุตที่ให้ผลออกมาในระดับสูงพร้อมด้วยความสามารถในการแสดงผลสีได้เป็นล้านๆ สี

การทำอินเตอร์เฟซกับ DIB

ในตอนแรกทีวินโดวส์ โทลคมินิไครเวอร์แสดงผล และทำการเรียกใช้รูทีนการกำหนดค่าเริ่มต้น DLL ของไครเวอร์ ไครเวอร์จะทำงานๆ โดยการรวบรวมข้อมูลข่าวสารเกี่ยวกับคอนฟิกูเรชันของมันเองจากไฟล์ SYSTEM.INI ต่อมาในโปรเซสการกำหนดค่าเริ่มต้นของระบบ GDI จะเรียกใช้อินเตอร์เฟซ Enable ของไครเวอร์สองครั้ง ครั้งแรกผ่านทางการเรียกใช้ไครเวอร์ DIB_Enable DIB จะยื่นพอยน์เตอร์คืนไปที่โครงสร้าง GDINFO ที่เหมาะสม ไครเวอร์บรรจุฟิลด์บางส่วนที่เป็นแบบขึ้นอยู่กัอุปกรณ์ (เช่น จำนวนบิตต่อพิกเซล) และส่งพอยน์เตอร์โครงสร้าง GDINFO คืนแก่ GDI การเรียกไปสู่ Enable ครั้งที่สองคือการทำงานในส่วนที่เหลือของการเริ่มต้นซึ่งทำให้แล้ว รวมไปถึงการเรียกใช้ VxD แสดงผลเพื่อกำหนดฮาร์ดแวร์เข้าสู่โหมดกราฟิกที่ถูกต้อง (ใช้ INT 10) และถ้าจำเป็นก็ส่งรหัส bank-switching ไปที่ VFLATD VxD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

รูปภาพบิตแมพ (Bitmap) และการแสดงผล

บิตแมพ (Bitmap) หากจะแปลกันตรงๆ ก็หมายถึง “การแทนจุดต่างๆ” ของรูปภาพด้วยการเรียงตัวของบิต” ใน Windows จะมีรูปบิตแมพอยู่สองอย่างคือ

1. รูปบิตแมพที่ขึ้นอยู่กับอุปกรณ์ (Device Dependent Bitmap - DDB) เป็นรูปบิตแมพที่ใช้แต่ละบิตในหน่วยความจำแสดงผลแทนจุดในภาพจริงๆ ดังนั้นหากใช้จอที่มีความละเอียดอย่างหนึ่งก็จะได้รูปที่มีขนาดหนึ่ง จึงได้ชื่อว่า “ขึ้นอยู่กับอุปกรณ์”

2. รูปบิตแมพแบบไม่ขึ้นกับอุปกรณ์ (Device Independent Bitmap - DIB) รูปแบบนี้จะเป็นการอธิบายลักษณะของภาพนั้น แทนที่จะใช้แต่ละบิตแทนแต่ละจุด ดังนั้นหากรู้จักลักษณะของจอจะสามารถรู้ชนิดนี้ให้เหมาะกับจออื่นๆ ได้

2.1 การสร้างรูปบิตแมพ

การสร้างรูปบิตแมพแบ่งได้เป็น 4 วิธีดังนี้

1. สร้างและโหลดรูปบิตแมพ

วิธีนี้เป็นวิธีที่สะดวกที่สุด โดยที่การสร้างนั้นหมายถึง ใช้แอฟพลิเคชันภายนอกสร้างรูปบิตแมพขึ้นมาก่อนและรวมภาพบิตแมพเป็นรีซอร์สของแอฟพลิเคชันและแอฟพลิเคชันจึงโหลดรูปเข้ามาแสดง แอฟพลิเคชันเหล่านี้ก็เช่น SDKPaint ของ SDK หรือ White Water Resource Toolbox (WRT) และอื่นๆ แต่ในที่นี้จะขออ้างอิงกับ SDKPaint เท่านั้น

การสร้างและโหลดรูปบิตแมพมีขั้นตอนดังนี้

1. เรียกแอฟพลิเคชัน SDKPaint ในกลุ่มของ SDK
2. ใช้เครื่องมือที่มีให้ เช่น การลากเส้น การพ่นสี หรือการระบายสีในการสร้างรูปจากนั้นให้เซฟไว้โดยให้มีนามสกุลเป็น BMP
3. การทำให้แอฟพลิเคชันรู้จักไฟล์บิตแมพที่สร้างไว้ให้เพิ่มคำบอก BITMAP เข้าไปในไฟล์รีซอร์ส เช่น ถ้ารูปบิตแมพถูกเก็บไว้ในไฟล์ dog.bmp ก็ให้เพิ่มบรรทัดนี้เข้าไปในไฟล์ RC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ dog BITMAP DOG BMP ึ่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเรียกรูปบิตแมพเข้ามาในแอปพลิเคชันนั้น จะต้องมีการโหลดรูปเข้ามาในหน่วยความจำด้วยฟังก์ชัน LoadBitmap ซึ่งถ้าสามารถโหลดเข้ามาสำเร็จ ฟังก์ชันนี้ก็จะให้ค่าแฮนเดิลที่จะให้อ้างถึงรูปบิตแมพนั้นกลับมา ตัวอย่างเช่น

```
hDogBitmap = loadBitmap(hInstance, "dog");
```

5. จากนั้นก็ใช้ฟังก์ชัน SelectObject เพื่อเลือกรูปเข้าไปในคอนเท็กซ์อุปกรณ์

```
SelectObject(hMemoryDC, hDogBitmap);
```

6. สุดท้ายก็เป็นการแสดงรูปออกทางวินโดวส์โดยใช้ฟังก์ชัน BitBlt ดังนี้

```
BitBlt(hDC, 10, 10, 100, 150, hMemDC, 0, 0, SR_COPY);
```

จากตัวอย่างมี hDC เป็นแฮนเดิลของคอนเท็กซ์อุปกรณ์จริงๆ พิกัด 10,10 เป็นตำแหน่งที่จะวางรูปตำแหน่ง 100,150 เป็นความกว้างและยาวของรูปที่ต้องการแสดงตำแหน่ง 0,0 ตัวหลังเป็นตำแหน่งของรูปใน hMemoryDC ที่จะเริ่มต้นก๊อปปี้ไปยัง hDC และพารามิเตอร์สุดท้าย SR_COPY จะเป็นการบอกว่าให้ก๊อปปี้จากรูปต้นทางไปปลายทางโดยตรง (แต่หากเป็นวิธีอื่น เช่นตัวเลือก SRCAND จะเป็นการทำโลจิกแอนด์กันระหว่างต้นทางบิตต่อบิต)

2. วาดรูปบนพื้นที่ว่างที่สร้างขึ้น

วิธีนี้เป็นการสร้างพื้นที่ว่างขึ้นมาก่อน แล้วจึงใช้ฟังก์ชันของ GDI ในการวาดรูปลงไป การสร้างวิธีนี้มีขั้นตอนคือ

1. สร้างพื้นที่ว่างสำหรับบรรจุบิตแมพ โดยฟังก์ชัน CreateBitmap หรือ CreateCompatibleBitmap
2. เลือกบิตแมพเข้าไปในคอนเท็กซ์อุปกรณ์ด้วยฟังก์ชัน SelectObject
3. ใช้ฟังก์ชัน GDI วาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้เป็นารสร้างรูปดาวโดยใช้ฟังก์ชัน Polygon ของ GDI

```
HDC hDC;
```

```
HDC hMemoryDC;
```

```
HBITMAP hBitmap;
```

```
HBITMAP hOldBitmap;
```

```
POINT Points[5] = {32, 0, 16, 63, 16, 0, 16, 48, 63};
```

1. `hDC = GetDC(hWnd);`
2. `hMemoryDC = CreateCompatibleDC(hDC);`
3. `hBitmap = CreateCompatibleBitmap(hDC, 64, 64);`
4. `hOldBitmap = SelectObject(hMemoryDC, hBitmap);`
5. `PatBlt(hMemory, 0, 0, 64, 64, WHITNESS);`
6. `Polygon(hMemoryDC, Points, 5);`
7. `BitBlt(hDC, 0, 0, 64, 64, hMemoryDC, 0, 0, SRCCOPY);`
8. `SelectObject(hMemoryDC, hOldBitmap);`
9. `ReleaseDC(hWnd, hDC);`

จากตัวอย่าง

1. ฟังก์ชัน `GetDC` ใช้ในการหาค่าแฮนเดิลคอนเท็กซ์อุปกรณ์ของวินโดวส์นั้น
2. ใช้ `CreateCompatibleDC` สำหรับสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำที่เหมือนกับคอนเท็กซ์อุปกรณ์จริง
3. สร้างรูปบิตแมพเปล่าโดยใช้ฟังก์ชัน `CreateCompatibleBitmap` โดยที่ในที่นี้ขนาดเป็น 64x16 จุดสี แต่จำนวนบิตที่ใช้จริงนั้นขึ้นอยู่กับชนิดของจอ ถ้าเป็นจอสี ในหนึ่งจุดสีก็จะประกอบด้วยหลายชนิด
4. ฟังก์ชัน `SelectObject` จะเป็นการเลือกรูปบิตแมพที่สร้างไว้ เข้าไปในคอนเท็กซ์อุปกรณ์ โดยที่แฮนเดิลของรูปเก่าจะถูกเก็บไว้ในตัวแปร `hOldBitmap`
5. ฟังก์ชัน `PatBlt` จะเซตทุกจุดสีในรูป ให้เป็นสีขาว
6. ฟังก์ชัน `Polygon` จะใช้สำหรับสร้างรูปเหลี่ยมขึ้นมา ค่าพารามิเตอร์ที่ฟังก์ชันนี้ต้องการคือตำแหน่งของจุดปลายต่างๆ ในที่นี้ก็จะอยู่ในอะเรย์ `Points`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรทางการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ฟังก์ชัน BitBlt เป็นการก๊อปปี้รูปจากหน่วยความจำไปยังพื้นที่ใช้งานของวินโดวส์
8. ฟังก์ชัน SelectObject อีกครั้งจะคืนรูปบิตแมพเก่าเข้าไปยังคอนเท็กซ์อุปกรณ์ดั้งเดิม และ DelectDC จะทำการลบคอนเท็กซ์อุปกรณ์ที่สร้างจำลองไว้ในหน่วยความจำทิ้งไป
9. ReleaseDC จะเป็นการปล่อยการยึดคอนเท็กซ์อุปกรณ์เพื่อให้แอปพลิเคชันอื่นสามารถเรียกใช้
ได้บ้าง

3. การสร้างรูปบิตแมพสี

หลักการสร้างรูปบิตแมพสีคือ compatible บิตแมพวางขึ้นมาก่อน จากนั้นก็สร้างแปรงทาสี
ต่างๆ (แดง เขียว และน้ำเงิน) มาวาดรูปลงในบิตแมพนั้น หากเจอแสดงผลเป็นแบบโมโนโครม
Windows ก็จะสร้างรูปแบบของจุดสีที่ใช้แทนสีต่างๆ เหล่านั้นขึ้นมา

ตัวอย่างการสร้างรูปบิตแมพเป็นดังนี้

```
HDC HDC ;
HDC hMemoryDC;
HBITMAP hBitmap;
HBITMAP hOldBitmap;
HBITMAP hRedBrush;
HBITMAP hGreenBrush;
HBITMAP hBlueBrush;
HBITMAP hOldBrush;

HDC = GetDc(hWnd);
HmemoryDC = CreatCompatibleDC(hDC);
Hbitmap = CreatCompatibleBitmap(hDC, 64, 31);
HOldBitmap = SelectObject(hMemoryDC, hBitmap);
HRedBrush =CreatSolidBrush(RED);
HGreenBrush =CreatSolidBrush(GREEN);
HBlueBrush =CreatSolidBrush(BLUE);
PetBlt(hMemoryDC, 0, 0, 64, 32, BLACKNESS);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่หรือเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HOldBrush = SelectObject(hMemoryDC, hRedBrush);
PatBlt(hMemoryDC, 0, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 40, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 24, 22, 24, 11, PATORDEST);
SelectObject(hMemoryDC, hGreenBrush);
PatBlt(hMemoryDC, 24, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 0, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 24, 22, 24, 11, PATORDEST);
SelectObject(hMemoryDC, hBlueBrush);
PatBlt(hMemoryDC, 40, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 24, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 0, 22, 24, 11, PATORDEST);
BitBlt(hDC, 0, 0, 64, 32, hMemoryDC, 0, 0, SR CCOPY);
SelectObject(hMemoryDC, hOldBrush);
DeleteObject(hRedBrush);
DeleteObject(hGreenBrush);
DeleteObject(hBlueBrush);
SelectObject(hMemoryDC, hOldBrush);
DeleteDo(hMemory, hOldBitmap);
ReleaseDC(hWnd, hDC);

```

จากตัวอย่างฟังก์ชัน CreatSolidBrush ทำหน้าที่สร้างแปรงทาสี สีต่างๆ ที่ต้องการใช้ในการวาดรูป โดยที่พารามิเตอร์ 3 ตัวที่ฟังก์ชันนี้ต้องการเป็นความเข้มของสีแดง เขียว น้ำเงิน ตามลำดับ จากนั้นก็ใช้ SelectObject เพื่อเลือกแปรงทาสีที่ต้องการ และใช้ PatBlt สำหรับทำการวาด โดย PATORDEST จะเป็นตัวที่บอกถึงลักษณะการวาดสี โดยจะมีสีใหม่ลงไปทับกับสีที่อยู่เดิมโดยการ OR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การแสดงรูปบิตแมพ

เมื่อทำการสร้างรูปบิตแมพได้ตามวิธีทั้ง 4 อย่างดังที่กล่าวมาแล้ว ก็มาถึงการแสดงรูปบิตแมพนั้น Windows ได้จัดวิธีแสดงรูปออกเป็นอีก 4 วิธีดังนี้

1. โดยการใช้ฟังก์ชัน BitBlt

ก่อนใช้ฟังก์ชันจะต้องสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำเสียก่อน แล้วจึงใช้ฟังก์ชันนี้ก็อปปีรูปลงในหน่วยความจำส่วนที่สร้างขึ้น

```
HDC Hdc, hMemoryDC;
HDC = GetDC(hWnd);
HMemoryDC = CreatCompatibleDC(hDC);
HOldBitmap = SelectObject(hMemory, hBitmap);
If(hOldBitmap){
    BitBlt(hDC, 100, 30, 64, 32, hMemoryDC, 0, 0, SRCCOPY);
    SelectObject(hMemoryDC, hOldBitmap);
}
DelectDC(hMemoryDC);
ReleaseDC(hWnd,hDC);
```

การทำงานของสามฟังก์ชันแรก ก็เหมือนกับที่เคยอธิบายในตัวอย่างที่กล่าวมาแล้ว ส่วน BitBlt จะทำการก็อปปีรูปจากคอนเท็กซ์อุปกรณ์ในหน่วยความจำไปยังคอนเท็กซ์อุปกรณ์ของจอ (จาก hMemoryDC ไปยัง hDC) โดยพิกัดตำแหน่งที่จะนำไปแสดงคือ (100,30) ความกว้างและยาวของรูปคือ (64,32) ส่วน SRCCOPY จะบอกให้เป็นการก็อปปีแบบเหมือนต้นฉบับทุกประการ

การบอกความกว้างและยาวของรูปนั้น นอกจากกำหนดลงไปโดยตรงดังตัวอย่างข้างต้นแล้วยังสามารถใช้ฟังก์ชัน GetObject มาช่วยได้

BITMAP Bitmap

ตัวแปรโครงสร้าง Bitmap จะเก็บคุณสมบัติต่างๆ ของรูปบิตแมพนั้นไว้รวมทั้งความกว้างและความยาวด้วย ทำให้เมื่อนำไปใช้ ในฟังก์ชัน Bitblt ก็จะได้เป็นดังนี้

BitBlit(hDC,100, 30, Bitmap.bmWidth, Bitmap.bmHeight, hMemoryDC,0,0,SRCCOPY);

การใช้ฟังก์ชันนี้นั้นไม่ต้องคำนึงถึงว่ารูปเป็นรูปกลายสีหรือสีเขียว แต่หากชนิดของรูปที่ค้นทาง (hMemoryDC) ไม่เหมือนกับอุปกรณ์ที่แสดงผล (hDC) แล้ว ก็จะเกิดการแปลงค่าบางประการเช่น ถ้ารูปเป็นสีแต่จอเป็นโมโนโครม รูปที่ปรากฏออกมาจะมีฉากหลังเป็นสีขาว และจุดอื่นๆ เป็นสีดำทั้งหมด

2. โดยการใช้ฟังก์ชัน StretchBit

รูปภาพที่แสดงออกทางอุปกรณ์ต่างๆ นั้น ไม่จำเป็นที่จะต้องมีความเหมือนที่ต้นฉบับหากมีการใช้ฟังก์ชัน StretchBit แทนฟังก์ชัน แล้วก็สามารถที่จะย่อหรือขยายเป็นรูปนั้น ได้ง่ายเพียงแค่พารามิเตอร์ BitBlit ไม่มีเข้าไปอีกสองตัว คือความกว้างและยาวของรูปที่เอาต์พุตเช่น

StretchBit(hDC,100,30,128,64 ,hMemoryDC,0,0,64,32,SRCCOPY);

ค่าพิกัด (128,64) ที่ปรากฏในตัวอย่างจะเป็นตัวบอกความกว้างยาวของรูปที่จะออกมาส่วน (64,32) จะเป็นความกว้างจริงของรูป

ในการย่อขนาดรูปนั้น แน่แน่นอนว่ารายละเอียดต่างๆ ของรูปต้องหดหายไปด้วย การลดการบ่ยเบนตรงนี้สามารถแก้ไขได้โดยบอก StretchBit ให้ใช้วิธีการแสดงผลต่างๆ กันดังนี้

โหมด	จุดประสงค์
WHITEONBLACK	สร้างรูปที่เป็นสีขาว โดยมีพื้นสีดำ
BLACKONWHITE	สร้างรูปที่เป็นสีดำ โดยมีพื้นสีขาว
COLORONCOLOR	สร้างรูปสี

ฟังก์ชันที่ใช้ตั้งโหมดเหล่านี้ให้ StretchBit รู้จักคือ SetStretchBitMode เช่น ถ้าต้องการให้ใช้โหมด

WHITEONBLACK ก็ทำได้ดังนี้

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SetStretchBitMode(hDC, WHITEONBLACK);
```

3. โดยการสร้างเป็นลักษณะของแปรงทาสี

อีกทางหนึ่งที่สามารถใช้รูปบิตแมพได้คือ นำไปสร้างเป็นลักษณะหนึ่งของแปรงทาสีไว้สำหรับทาสี (fill) ในรูปบิตต่างๆ เมื่อสิ่งเหล่านี้ลงในรูปบิตนั้น Windows จะก๊อปปี้รูปบิตนั้นไปทางตั้งและทางนอนจนกว่าจะเต็มรูปบิต

บิตแมพที่ต้องการนำมาใช้งานอย่างนี้ควรสร้างให้มีความกว้างยาว 8x8 จุดสี ทั้งนี้เนื่องจาก Windows จะใช้เพียงรูปขนาด 8x8 จุดสีที่ตำแหน่งมุมบนซ้ายเท่านั้น และการสร้างอาจเป็นการกำหนดที่ละจุด (hard code) โดยการสร้างและบรรทัด หรือจะใช้การโหลดด้วยไฟล์รีซอร์สก็ได้ไม่ว่าจะเป็นการสร้างด้วยวิธีใดก็ตาม จะได้เป็นเส้นเคิลของ รูปบิตแมพนั้นกลับมา จากนั้นก็นำเส้นเคิลนั้นผ่าน ไปให้กับฟังก์ชัน CreatePatternBrush ดังตัวอย่างต่อไปนี้

```
hBitmap = LoadBitmap(hInstance,"Checks");
```

```
hBrush = CreatCompatibleBrush(hBitmap);
```

//hBrush เป็นเส้นเคิลของแปรงที่สร้างขึ้น การนำไปใช้งานให้ฟังก์ชัน SelectObject ดังนี้

```
hOldBrush = SelectObject(hDC,hBrush);
```

ลักษณะการใช้ฟังก์ชันนี้จะถือว่าบิตแมพที่เลือกใหม่นั้นเป็นส่วนหนึ่งของแปรงทาสีดังนั้นจะไม่กระทบกับรูปบิตแมพอื่นที่ถูกเลือกไว้ก่อนแล้ว

จากนั้นก็เป็นการนำไปใช้งาน ตัวอย่างต่อไปนี้เป็นการระบายสีที่มุมบนซ้ายของวินโดวส์

```
PatBlt(hDC, 0, 0, 100, 100, PATORDEST);
```

ส่วนที่เป็นพื้นของวินโดวส์ของแอปพลิเคชัน ก็สามารถเซตให้เป็นรูปแบบที่ต้องการได้โดยเปลี่ยนโครงสร้างวินโดวส์คลาส ดังนี้

```
pWndClass->hbrGround = CreatPatternBrush(hBitmap);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ ไม่ก็ใช้ฟังก์ชัน SetClassWord

```
SetClassWord(hWnd, GCW_HBRBACKGROUND,hBrush);
```

แต่วิธีนี้ก็มีข้อเสีย การเปลี่ยนรูปแบบของพื้นนั้นจะเกิดขึ้นกับทุกวินโดว์ที่ใช้คลาสนี้แต่ หากต้องการเปลี่ยนเฉพาะบางวินโดว์ก็จะต้องมีการกำหนดลงไปอย่างชัดเจน โดยจะกระทำต่อเมื่อ ได้รับเมสเสจ WM_ERASEBKGD

```
RECT Rect;
```

```
HBRUSH hOldBrush;
```

```
Case WM_ERASEBKGD:
```

```
UnRealizeObject(hMyBkgnBrush);
```

```
HOldBrush = SelectObject(wParam, hMyBkgnBrush);
```

```
GetUpdateRect(wParam,(LPRECT) & Rect, FALSE);
```

```
PatBlt(wParam,Rect.left,Rect.top,Rec.right-Rect.left,
```

```
Rect.buttom-Rect.top,PATCOPY);
```

```
SelectObject(wParam,hOldBrush);
```

```
Break;
```

ขณะที่ได้รับเมสเสจ WM-ERASEBKGD ก็จะได้แชนดิลของคอนเท็กซ์อุปกรณ์มาในตัวแปร wParam ด้วยฟังก์ชัน GetUpdateRect จะให้รายละเอียดของพื้นที่ที่ต้องถูกลบเพื่อวาดใหม่กลับมาในโครงสร้าง Rect ซึ่งถูกนำไปใช้ฟังก์ชัน PatBlt เพื่อวาดรูปบิตแมพลงในพื้นที่นั้นส่วนฟังก์ชัน UnRealize Object ทำหน้าที่ในการปรับจุดเริ่มต้นของบิตแมทที่วาดใหม่เมื่อวินโดว์นั้นถูกเคลื่อนย้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การแสดงรูปบิตแมพแบบไม่ขึ้นกับอุปกรณ์

ประโยชน์อย่างหนึ่งของการแสดงผลแบบนี้คือ ไม่จำเป็นต้องสร้างคอนเท็กซ์อุปกรณ์จำลองในหน่วยความจำก่อน ทำให้หน่วยความจำที่ต้องการลดลงไปเล็กน้อย ฟังก์ชันที่นำมาใช้คือ SetDIBitsToDevice ส่วนค่าพารามิเตอร์ต่างๆ ที่ต้องการคือ

- คอนเท็กซ์อุปกรณ์แสดงผลที่ต้องการแสดง
- ตำแหน่งที่ต้องการแสดงรูป
- ขนาดของรูปภาพที่ต้องการแสดง
- จำนวนเส้นสแกนของรูปต้นทาง
- ตำแหน่งของจุดสีแรก(มุมซ้ายบน) ในรูปต้นทางที่เริ่มแสดง
- โครงสร้างที่เก็บข้อมูลของรูปบิตแมพ รวมทั้งบัพเฟอร์ที่เก็บรูปต้นทางอยู่ด้วย
- ค่าที่บอกว่าข้อมูลที่อยู่ในจานสีหรือตารางสี (palette table) ที่ใช้นั้นเป็นการบอกค่าสี RGB จริงหรือเป็นดัชนีชี้ค่าของจานสีที่ใช้อยู่ในขณะนั้น

ตัวอย่างการใช้ฟังก์ชันเป็นดังนี้

```
SetDIBitsToDevice(hdc,0,0,lpb->bmciHeader.bcWidth,lpbi->bmciHeader.bcHeight,0,0,0,lpbi->bmciHeader.bcHeight,Pbuf,(LPBITMAPINFO)lpbi,DIB_RGB_COLORS);
```

พารามิเตอร์ตัวแรกเป็นแฮนเดิลของคอนเท็กซ์อุปกรณ์ สองตัวอย่างต่อมาเป็นจุดเริ่มต้นของรูปที่จะแสดง ในกรณีก็จะเป็นมุมบนซ้ายของอุปกรณ์นั้นๆ ตัวแปรโครงสร้างสองตัวถัดมาเป็นความกว้างและความสูงของรูปที่ต้องการแสดง

เลข 0,0 ที่ตามมาเป็นจุดเริ่มของรูปต้นทาง (0,0 จะเป็นการเริ่มต้นตั้งแต่มุมบนซ้ายของภาพ) เลข 0 ตัว ต่อมาเป็นจุดเริ่มต้นของการทำ banding (ศึกษาเพิ่มเติมได้เรื่อง "การพิมพ์") ส่วนตัวที่ตามมาเป็นความกว้างของแบนด์ที่ต้องการ ในที่นี้หมายถึง ในหนึ่งแบนด์ครอบคลุมทั้งรูป

lpbi เป็นบัพเฟอร์ที่เก็บรูปต้นทาง บัพเฟอร์นี้อยู่ในตัวแปร โครงสร้าง BITMAPINFO ส่วนพารามิเตอร์ตัวสุดท้ายเป็นไปได้สองอย่างคือ DIB_RGB_COLORS (ตามตัวอย่าง) จะบอกว่าตารางสีที่ใช้นั้นเก็บข้อมูลการผสมสีระหว่าง แดง เขียว และน้ำเงิน จริง แต่หากเป็น

DIB_RGB_COLORS จะเป็นดัชนี (อะเรย์ 16 บิต) ชี้ตำแหน่งในจานสี (palette) ที่ใช้งานอยู่ในขณะ

นั้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การใส่สีให้กับรูปบิตแมพที่เป็นสีเขียว(ขาวดำ)

สำหรับรูปบิตแมพที่เป็นรูปสีเขียว เราสามารถที่จะกำหนดสีจุดที่เป็นสีดำ (ตัวรูป) กับจุดที่เป็นสีขาว(พื้น) ได้ว่าให้แสดงด้วยสีอะไรไทย การตั้งสีของรูปนั้นจะใช้ฟังก์ชัน SetTextColor การเซตสีของพื้นใช้ฟังก์ชัน SetBkColor

ตัวอย่างต่อไปนี้เป็น การเซตสีเพื่อให้ตัวรูปเป็นสีแดง ส่วนพื้นที่เป็นสีเขียว

```
SetTextColor(hDC, RGB(255,0,0));
```

```
SetBkColor(hDC, RGB(0,255,0));
```

ฟังก์ชัน RGB จะรับค่าความเข้มของสีแดง เขียว และน้ำเงิน (เรียงตามลำดับของพารามิเตอร์) มาสร้างรหัสของสีนั้นกลับมา โดยค่าความเข้มที่น้อยที่สุดคือ 0 และความเข้มที่มากที่สุดเป็น 255 ดังนั้น RGB(255,255,0) จะมีสีแดงเข้มเท่ากับสีเขียว สีที่ออกมาจึงเป็นสีเหลือง

2.4 การลบรูปบิตแมพ

รูปบิตแมพก็เหมือนกับรีซอร์สอย่างอื่น คือ ต้องมีการจองหน่วยความจำมาเรียกใช้ ดังนั้นเมื่อเลิกใช้ก็ต้องมีการลบทิ้ง ฉะนั้นเพื่อให้หน่วยความจำส่วนนั้นว่างลง และสามารถนำไปใช้งานอย่างอื่นได้ การลบต้องให้คอนเท็กซ์อุปกรณ์เลือกรูปบิตแมพเดิมเสียก่อน โดยใช้

```
SelectObject(hMemoryDC,hOldBitmap);
```

แล้วจึงลบแฮนเดิลที่ชี้รูปบิตแมพนั้นทิ้ง

```
DeleteObject(hBitmap);
```

การที่ต้องการเลือกรูปอื่นเข้าไปในคอนเท็กซ์อุปกรณ์ก่อน เพราะ Windows จะไม่ยอมให้ลบแฮนเดิลที่กำลังถูกใช้อยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 คอนเท็กซ์อุปกรณ์ (Device Context)

คอนเท็กซ์อุปกรณ์เป็นสะพานเชื่อมข้อมูลระหว่างแอปพลิเคชัน ไดรเวอร์และอุปกรณ์เข้าด้วยกัน เมื่อแอปพลิเคชันที่ต้องการแสดงผลทางอุปกรณ์ต่างๆ ก็สามารถเรียกใช้งานฟังก์ชันใช้งานกราฟิก (GDI) โดยไม่ต้องสนใจในลักษณะของอุปกรณ์หรือสนใจเพียงเล็กน้อย ซึ่งฟังก์ชันเหล่านี้ก็จะผ่านข้อมูลเหล่านั้นไปให้ไดรเวอร์ของอุปกรณ์ และไดรเวอร์ของอุปกรณ์นั้นจะไปเรียกใช้งานอุปกรณ์ต่างๆ อื่นๆ

- การติดต่อกับอุปกรณ์แสดงผล

แอปพลิเคชันของ WINDOWS จะสามารถเรียกใช้ฟังก์ชันใน GDI สำหรับติดต่อกับอุปกรณ์แสดงผลฟังก์ชันของ GDI (ซึ่งทำงานไม่ขึ้นกับอุปกรณ์) จะผ่านค่าต่างๆ ที่ได้จากแอปพลิเคชันไปยังไดรเวอร์โดยพิจารณาถึงลักษณะของอุปกรณ์ และแปลงการเรียกใช้ฟังก์ชันของ GDI ให้เป็นกระบวนการสำหรับอุปกรณ์นั้น

- การเก็บคอนเท็กซ์อุปกรณ์

ในบางครั้งอาจจำเป็นต้องมีการเก็บค่าต่างๆ ที่คอนเท็กซ์อุปกรณ์นั้นกำลังใช้งานอยู่ เพื่อนำมาใช้ภายหลังฟังก์ชันที่ทำหน้าที่ในการเก็บและการนำกลับสถานะของดีไวน์คอนเท็กซ์คือ SaveDC และ RestoreDC

- การลบคอนเท็กซ์อุปกรณ์

เราสามารถลบคอนเท็กซ์อุปกรณ์ที่สร้างขึ้นหลังจากที่ไม่ต้องการใช้งานแล้วได้ด้วยฟังก์ชัน DeleteDC ซึ่งจะเป็นการคืนทรัพยากรต่างๆ กลับให้ WINDOWS

- คอนเท็กซ์อุปกรณ์ในหน่วยความจำ

การสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำ จะเป็นการกันหน่วยความจำไว้ส่วนหนึ่งเป็นอุปกรณ์เทียม โดยให้มีลักษณะและการปฏิบัติเหมือนคอนเท็กซ์อุปกรณ์ที่ต่ออยู่กับอุปกรณ์จริง จากนั้นเมื่อมีการสร้างรูปปิดแมพลงไปในคอนเท็กซ์อุปกรณ์แบบนี้ ก็สามารถใช้ฟังก์ชันเพื่อก๊อปปี้ข้อมูลไปยังคอนเท็กซ์อุปกรณ์จริงๆ ไคท์นที หรือในทางกลับกันจะนำข้อมูลของคอนเท็กซ์อุปกรณ์จริงมาลงในคอนเท็กซ์อุปกรณ์ในหน่วยความจำก็ได้ ฟังก์ชันที่ใช้ในการสร้างดีไวซ์คอนเท็กซ์นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท็กซ์แบบนี้คือ CreatCompatibleDC แต่อุปกรณ์ที่จะถ่ายเทข้อมูลกับคอนเท็กซ์อุปกรณ์แบบหน่วยความจำนี้ได้ต้องสนับสนุนสร้างบิตแมพฟังก์ชัน BitBlt เท่านั้น

2.6 สี (Color)

ฟังก์ชัน GDI ที่มีการสร้างปากกาหรือแปรงทาสีส่วนมากต้องการข้อมูลที่ใช้ในการระบุสีของเครื่องมือ โดยอยู่ในโครงสร้างข้อมูล COLORREF วิธีในการบอกสีมี 3 วิธีคือ

- ระบุค่า RGB(Red Green Blue)
- ระบุจานสี (Color Palette) ที่ต้องการใช้
- ระบุค่า RGB ที่สัมพันธ์กับจานสี

วิธีแรกนั้นจะกำหนดเป็นความเข้มของสีแดง เขียวและน้ำเงินที่จะมาผสมเป็นสีที่ต้องการ โดยค่าความเข้มต่ำสุดคือ 0 และสูงสุดคือ 255 ไบต์หนึ่ง (ไบต์ต่ำสุด) จะเป็นค่าความเข้มของสีแดง ไบต์ที่สองเป็นค่าความเข้มของสีเขียว ไบต์ที่สามเป็นค่าความเข้มของสีน้ำเงิน และไบต์ที่สี่ต้องมีค่าเป็น 0 เสมอ ตัวอย่างเช่น ถ้าต้องการสีน้ำเงินก็จะมีค่าเป็น 0X00FF0000 หรือถ้าเป็นสีเขียวก็จะเป็น 0X0000FF00 เป็นต้น การกำหนดสีวิธีนี้ฟังก์ชัน GDI จะส่งไปยัง ไดรเวอร์ของอุปกรณ์โดยตรง ซึ่ง ไดรเวอร์จะเป็นตัวเลือกสีที่ใกล้เคียงกับที่ต้องการมากที่สุดมาใช้ การขอค่าของสีที่ใกล้เคียงค่าของสีที่ต้องการให้ใช้ฟังก์ชัน GetNearestColor

แต่ถ้าอุปกรณ์ที่ต้องการเป็นพล็อตเตอร์ ตัวไดรเวอร์ก็จะแปลงสีที่ต้องการให้เป็นหมายเลขของปากกาพล็อตเตอร์ที่มีสีใกล้เคียงแทน

ในบางอุปกรณ์ที่ไม่สามารถแสดงสีได้มากนัก ก็จะใช้วิธี "dither" คือการสร้างรูปแบบของจุดเข้าไปในสีด้วย เพื่อสร้างความแตกต่างได้ด้วยสีเดียวกัน ถ้าอุปกรณ์นั้นเป็น โมโน โครมลักษณะที่แสดงออกจะเป็นสามลักษณะคือ ขาว ดำ และรูปแบบของจุด โดยที่ค่าผลรวมของค่า RGB เป็น 0 สีที่แสดงออกมาจะเป็นสีดำ (เลือกใช้แปรงสีดำ) ถ้ามีค่าเป็น 765 ก็จะเป็นสีขาว แต่ถ้าอยู่ระหว่าง 0 และ 765 จะแสดงเป็นรูปแบบของจุดฟังก์ชันที่ดึงค่าสีแดง เขียว และน้ำเงิน ออกจากโครงสร้าง COLORREF คือ GetValue, GetBValue และ GetGValue

2.7 จานสี(Palette)

ปกติแล้วจอภาพส่วนมากสามารถแสดงสีได้จำนวนมาก แต่จำนวนสีที่แสดงได้พร้อมกันในหนึ่งหน้าจอจะมีไม่มากนัก เช่น จอที่แสดงสีได้ 262,000 สี อาจแสดงสีได้พร้อมกันเพียง 256 สีเท่านั้น ทั้งนี้เพียงมาจากข้อจำกัดของการ์ดแสดงผล ในกรณีส่วนแสดงผลจะมี “จานสี” (Palette) เพื่อเก็บว่ามีอะไรที่กำลังถูกแสดงอยู่เมื่อใดก็ตามที่มีการร้องขอใช้สีมากกว่าที่จานสีจะรับได้ สีที่ร้องขอมาใหม่ก็จะไปทับของเดิม ทำให้สภาพที่กำลังแสดงอยู่บนจอมีสีที่ผิดเพี้ยนไปได้

ในสภาวะของ WINDOWS จะมีหลายแอปพลิเคชันทำงานพร้อมๆ กัน ซึ่งแต่ละแอปพลิเคชันก็จะมีวินโดว์ของตนเอง ปัญหาอยู่ตรงที่ว่าวินโดว์อาจไม่สามารถใช้สีมากเท่าที่มีอยู่ในจานสีได้ ทั้งนี้ก็เพราะมีวินโดว์อื่นกำลังใช้งานสีต่างกัน ไปอยู่เช่นกัน ดังนั้น WINDOWS จึงให้วินโดว์ที่ไม่ได้แอกทีฟนั้นจะถูกเปลี่ยนสีไปให้ใกล้เคียงกับสีที่มีอยู่ในจานสีปัจจุบัน (ที่วินโดว์ที่แอกทีฟใช้งานอยู่)

● เรื่องของจานสี

จานสีเป็นเครื่องมือไม่ขึ้นกับอุปกรณ์แสดงผลอีกอย่างหนึ่ง ทำหน้าที่เกี่ยวกับการจัดการให้แอปพลิเคชันสามารถใช้สีที่มีอยู่จำกัดในจานสีของระบบ(ของส่วนแสดงผลของเครื่อง) ได้อย่างเต็มที่

การที่จะให้แต่ละแอปพลิเคชันสามารถใช้งานจานสีของระบบได้เต็มที่นั้น WINDOWS จานสีทางตรรก (logical palette) ขึ้นมาให้แก่วินโดว์ของแอปพลิเคชันและเมื่อต้องการใช้งานสีของแอปพลิเคชันก็จะมาเพิ่มสีในจานสีทางตรรกของวินโดว์นั้นแน่ แต่ว่าอาจจะมีหลายวินโดว์ทำงานอยู่พร้อมกันในขณะหนึ่งๆ ทำให้ผลรวมของจำนวนสีในแต่ละจานสีของวินโดว์อาจมีมากกว่าจานสีของระบบก็ได้ WINDOWS จึงต้องมีการจัดการในเรื่องนี้ด้วย โดยจะพยายามจับคู่สีที่มีอยู่ในจานสีของระบบเข้ากับจานสีของวินโดว์ ถ้ามีไม่ตรงกันก็จะเพิ่มสีนั้นลงไปในจานสีของระบบ แต่ถ้าจานสีของระบบมีสีอยู่เต็มแล้วก็จะใช้วิธีหาสีที่ใกล้เคียงกันมากที่สุด (จะมีสี 20 สีมาตรฐานที่สามารถจับคู่ได้เสมอ) โดยปกติ WINDOWS จะพยายามให้วินโดว์ที่แอกทีฟอยู่ได้สีที่ใกล้เคียงความต้องการของตนมากที่สุด ส่วนที่ให้ความสำคัญรองลงมาคือวินโดว์ที่ได้อินพุทโฟกัสอยู่

* เพิ่มเติม * จานสีทางตรรก (logical Palette) นั้นเป็นจานสีที่ให้วินโดว์ใช้อย่างเต็มที่และสามารถใส่สีที่ได้ตามความพอใจ แต่จะมีใช้งานจริงที่อยู่บนการ์ดแสดงผลซึ่งจะมีเพียงอันเดียว WINDOWS จะคอยจัดการให้ทุกวินโดว์รู้สึกเหมือนกันว่าได้ใช้งานมีเพียงผู้เดียว โดยใช้วิธีการหาสีที่จะมาทดแทนได้ตามวิธีที่กล่าวมาแล้ว

เอกสารนี้เป็นเอกสารทศวงวิสาทรสำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การใช้งานงานสี

ก่อนที่จะวาดสิ่งต่างๆ ลงบนอุปกรณ์แสดงผล ต้องมีการสร้างงานสีของวินโดวส์ขึ้นมาด้วย ฟังก์ชัน CreatPalette และเลือกเข้าไปในคอนเท็กซ์อุปกรณ์ด้วย SelectPalette เสียก่อน ฟังก์ชันที่สามารถรับพารามิเตอร์เป็นค่าสีได้ก็สามารถรับเป็นดัชนีของงานสีได้เช่นกัน การกำหนดเป็นดัชนีของงานสีจะใช้ตัวแปร long โดยที่บิตแรกของไบต์สูงมีค่าเป็นหนึ่งในและดัชนีของงานสีก็จะเก็บอยู่ในไบต์ต่ำ เช่น 0X01000005 จะเป็นการเลือกใช้สีของงานสีในช่องที่ 5 จากนั้นใช้มาโครที่มีชื่อ PALETTEINDEX ในการเปลี่ยนดัชนีงานสีให้เป็นค่าของสี COLORREF เพื่อใช้เป็นพารามิเตอร์ให้กับฟังก์ชัน GDI ต่างๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

MCI

(Media Control Interface)

จะจัดเตรียมคำสั่งมาตรฐานสำหรับการใช้ควบคุมอุปกรณ์มัลติมีเดียและการบันทึกเพิ่มข้อมูลมัลติมีเดีย คำสั่งเหล่านี้โดยทั่วไปจะใกล้เคียงกับชนิดของอุปกรณ์มัลติมีเดีย

3.1 เกี่ยวกับ MCI

MCI จะเตรียมแอฟพลิเคชันกับอุปกรณ์ที่เป็นอิสระ ซึ่งสามารถควบคุม Audio เสมือนอยู่รอบนอกแอฟพลิเคชันของคุณสามารถใช้ MCI เพื่อจะควบคุมอุปกรณ์มัลติมีเดียใดๆ ที่ให้การสนับสนุนรวมทั้งอุปกรณ์ Wave form audio, MIDI Sequencers, อุปกรณ์ CD audio, และ Digital – video

3.2 คำสั่ง Strings และ Messages ของ MCI

MCI สนับสนุนคำสั่ง Strings และคำสั่ง Messages จะสามารถใช้คำสั่ง Strings หรือ Messages ใดๆอย่างหนึ่งหรือทั้งสองในแอฟพลิเคชัน MCI

- The Command-Messages interface ประกอบด้วยค่าคงที่ (Constant) โครงสร้าง (Constructures) จะใช้ฟังก์ชัน `mciSendCommand` เพื่อส่ง Message ไปยังอุปกรณ์ MCI
- The Command-strings interface คำแปลในต้นแบบของคำสั่ง Messages จะใช้ฟังก์ชัน `mciSendCommand` เพื่อส่ง strings ไปยังอุปกรณ์ MCI คำสั่ง String จะจำลองหน้าที่ของคำสั่ง Messages ระบบปฏิบัติการ Microsoft windows จะแปลงคำสั่ง Strings เป็นคำสั่ง Messages ก่อนที่จะส่งไปยัง MCI driver สำหรับดำเนินการต่อไป

คำสั่ง Messages นั้นจะทำการแก้ไขข้อมูลในรูปแบบของโครงสร้างที่ซึ่งง่ายต่อการแปลในแอฟพลิเคชัน โครงสร้างเหล่านี้สามารถบรรจุข้อมูลในรูปแบบที่แตกต่างกันของอุปกรณ์ คำสั่ง String จะแก้ไขข้อมูลที่ทำให้รูปแบบของ string และสามารถแก้ไขได้เพียงหนึ่ง String ในหนึ่งครั้งเท่านั้น แอฟพลิเคชันจะต้องกระจายหรือทดลอง string ก่อนที่จะแปลอาจจะใช้คำสั่ง Messages ซึ่งจะใช้ง่ายกว่าคำสั่ง String แต่คำสั่ง String จะง่ายต่อการจัดและการแปลของแอฟพลิเคชันของ MCI ใช้คำสั่ง String เมื่อไม่มีการส่งค่ากลับและคำสั่ง Messages จะใช้เมื่อแก้ไขข้อมูลจากอุปกรณ์

* ท้ายเหตุ * ในปฏิญานพนธ์นี้จะอธิบายแต่คำสั่ง Messages เท่านั้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3.3 คำสั่ง Messages

The Command-Messages interface คือการออกแบบที่ถูกใช้โดยแอปพลิเคชันที่ต้องการติดต่อกับภาษาซี เพื่อควบคุมอุปกรณ์มัลติมีเดีย มันใช้ Messages-passing paradigm เพื่อสื่อสารกับอุปกรณ์ MCI คุณสามารถส่งคำสั่ง โดยการใช้ฟังก์ชัน mciSendCommand

3.4 การส่งคำสั่ง Messages

ระบบปฏิบัติการวินโดวส์จะเตรียมการฟังก์ชันสำหรับรับคำสั่ง Messages ไปยังอุปกรณ์และตรวจสอบข้อมูลที่ผิดพลาด ฟังก์ชัน mciSendCommand จะส่งคำสั่ง Messages ไปยังอุปกรณ์ MCI ฟังก์ชัน mciSendCommand จะส่งคำสั่ง error string ที่ตรงกับ error number กลับ

ฟังก์ชัน mciSendCommand จะส่งค่า 0 กลับถ้าสำเร็จ ถ้าฟังก์ชันล้มเหลว low-order word ของการส่งค่ากลับจะบรรจุ error code คุณสามารถผ่าน error code นี้ไป mciGetErrorString เพื่อให้แสดงข้อความ

3.5 รูปแบบของคำสั่ง Messages

คำสั่ง Messages MCI ประกอบด้วยองค์ประกอบ 3 อย่างดังนี้

- ค่าคงที่ของ Messages
- โครงสร้างที่บรรจุพารามิเตอร์สำหรับคำสั่ง
- Set flage เพื่อระบุทางเลือกของคำสั่ง และให้ field ในพารามิเตอร์บล็อกไม่เป็นโมฆะ จากตัวอย่างจะส่งคำสั่ง MCI_PLAY ไปยังอุปกรณ์ที่ถูกระบุโดย device identifier

```

mciSendCommand(wDeviceID, //device identifier
               MCI_PLAY, //คำสั่ง Messages
               0,
               (DWORD)(LPVOID) & mclpalyParams); //parameter block device

```

device identifier จะอยู่ในพารามิเตอร์แรก จะถูกเรียกมาใช้เพื่ออุปกรณ์ถูกเปิดโดยการใช้คำสั่ง MCI_OPEN พารามิเตอร์ ถัดมาเป็นส่วนของโครงสร้าง MCI_PLAY_PARAMS ที่ซึ่งบรรจุข้อมูลเกี่ยวกับจุดเริ่มต้นหรือจุดสุดท้ายของการ playback คำสั่ง Messages ของ MCI ส่วนมากจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างเพื่อบรรจุพารามิเตอร์ของชนิดนี้ ในสมาชิกแรกของแต่ละ โครงสร้างเหล่านี้ระบุให้
 วินโดว์รับข่าวสาร MM MCINOTIFY เมื่อปฏิบัติการเสร็จ

3.6 การแบ่งประเภทคำสั่งของ MCI

MCI จะแบ่งคำสั่งเป็น 4 ประเภท :System Required, Basic และ Extened มีรายการดังต่อไปนี้

- System Commands จะรอควบคุมโดย MCI โดยตรง มากกว่าการใช้ driver
- Required Command จะถูกควบคุมโดย driver MCI driver ทั้งหมดจะสนับสนุน required commands และ flages
- Basic Commands จะถูกใช้โดยบางอุปกรณ์ ถ้าอุปกรณ์นั้นสนับสนุนคำสั่งพื้นฐาน
- Entended Commands จะระบุถึงเปิดหาอุปกรณ์หรือ driver

1. System Commands

<u>Messages</u>	<u>Description</u>
MCI-BREAK	กำหนดคีย์เบรกสำหรับ MCI drivice
MCI-SYSINFO	ส่งข้อความ

2. Required Commands

อุปกรณ์ MCI ทั้งหมดจะสนับสนุน Required Command ดังนี้

<u>Messages</u>	<u>Description</u>
MCI_GETDEVCAPS	ความสามารถของอุปกรณ์
MCI_CLOSE	ปิดอุปกรณ์
MCI_INFO	ข้อมูลเดิมจากอุปกรณ์
MCI_OPEN	กำหนดค่าเริ่มต้นของอุปกรณ์
MCI_STATUS	บอกสถานะของข้อมูลจากอุปกรณ์เดิม,บท flages ของคำสั่งนี้ ไม่ต้องการใช้ ดังนั้น มันจึง เป็นคำสั่งพื้นฐานอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Basic Commands

หัวข้อย่อยของคำสั่งพื้นฐาน : อุปกรณ์ MCI จะใช้คำสั่งเหล่านี้เป็นทางเลือก

<u>Messages</u>	<u>Description</u>
MCI_LOAD	โหลดข้อมูลจากไฟล์
MCI_PAUSE	หยุดการเล่น การเล่นกลับหรือบันทึก สามารถที่จะกลับคืนยังตำแหน่งเดิมได้
MCI_PLAY	เริ่มการส่งข้อมูลของ output
MCI_RECORD	เริ่มการบันทึกข้อมูลของ input
MCI_RESUME	กลับคืนไปสู่การเล่นหรือบันทึกจากที่หยุดอุปกรณ์ไว้
MCI_SAVE	บันทึกข้อมูลลงดิสก์ forward หรือ backward
MCI_SEEK	Set ชั้นรายการทำงานของอุปกรณ์
MCI_SET	บอกสถานะข้อมูลของอุปกรณ์
MCI_STOP	หยุดการเล่น

ถ้า driver สนับสนุนคำสั่งพื้นฐาน มันจะต้องสนับสนุนการกำหนด มาตรฐานของ flages สำหรับคำสั่งอีกด้วย

4. Extended Command

<u>Messages</u>	<u>Device types</u>	<u>Description</u>
MCI_DELETE	wave audio	ลบข้อมูลบางส่วนจากไฟล์
MCI_SETAUDIO	digitalvideo	ตั้งพารามิเตอร์ของ audio สำหรับ video
MCI_SETVIDEO	digitalvideo	ตั้งพารามิเตอร์ของ video
MCI_UPDATE	digitalvideo	เขียน frame เดิมซ้ำ ไปยัง device context
MCI_WINDOW	digitalvideo,overlay	ควบคุมการแสดงของวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การใช้คำสั่ง Messages ของ MCI

ตัวอย่างเหล่านี้จะแสดงให้เห็นถึงรูปแบบของคำสั่งสำหรับการใช้งาน

➤ การปิดอุปกรณ์ MCI ที่ถูกใช้ทั้งหมดโดยแอฟพลิเคชัน

ตัวอย่างการปิดอุปกรณ์ MCI ทั้งหมดที่ถูกเปิด โดยแอฟพลิเคชัน

```
UINT wDeviceID;
```

```
DWORD dwReturn;
```

```
//ปิดอุปกรณ์ MCI ทั้งหมดที่ถูกเปิด โดยแอฟพลิเคชัน
```

```
//รอจนกระทั่งอุปกรณ์ถูกปิดก่อนจึงจะส่งค่ากลับ
```

```
if(dwReturn = mciSendCommand(MCI_ALL_DEVICE_ID,  
MCI_CLOSE,  
MCI_WAIT,  
NULL))
```

```
//ผิดพลาด,ไม่สามารถปิดอุปกรณ์ทั้งหมดได้
```

➤ การเปิด Compound device โดยการเรียกชื่อ ไฟล์

ตัวอย่างนี้เป็นการเปิดอุปกรณ์ waveform-audio โดยการระบุ file Waveform-audio ที่ชื่อ

“TIMPANI.WAV”

```
UINT wDeviceID;
```

```
DWORD dwReturn;
```

```
MCI_OPEN_PARAMS mciOpenParams;
```

```
//การเปิดอุปกรณ์ Waveform-audio โดยระบุอุปกรณ์และชื่อไฟล์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่รับผิดชอบต่อการใช้งานที่ไม่ถูกต้องของเอกสารนี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mciOpenParams.lpstrElementName = "timpani.wav";
```

```
if(dwReturn = mciSendCommand(NULL,MCI_OPEN,  
MCI_OPEN_TYPE | MCI_OPEN_ELEMENT,  
(DWORD)(LPVOID) & mciOpenParams))
```

```
//Error ไม่สามารถเปิดอุปกรณ์ได้
```

```
//ถ้าเปิดอุปกรณ์สำเร็จ จะให้ ID ของอุปกรณ์
```

```
wDeviceID = mciOpenParams.wDeviceID;
```

➤ การควบคุมความผิดพลาด MCI

จะต้องตรวจสอบการส่งค่าของฟังก์ชัน mciSendCommand เสมอ ถ้าแสดงคำผิดพลาดสามารถใช้ฟังก์ชัน mciGetErrorStrings ให้แสดงข้อความของข้อผิดพลาดได้

จากตัวอย่างจะให้รหัสของ MCI Error ที่ถูกระบบโดย dwError ไปยัง mciGetErrorString และเมื่อแสดงผลของคำผิดพลาดโดยการ ใช้ฟังก์ชัน Messages Prox

```
//ใช้ mciGetErrorString เพื่อแสดงข้อความผิดพลาดของ MCI
```

```
//แสดงข้อผิดพลาด โดยใช้ Messages Box
```

```
void ShowError(DWORD dwError){  
    char szErrorBuf[MAXERRORLENGTH];  
    MessagesBeep(MB_ICONEXCLAMATION);  
    If(mciGetErrorString(dwError,(LPSTR) szErrorBuf, MAXERRORLENGTH))  
        MessagesBox(hMainWnd,szErrorBuf, "MCI Error",  
MB_ICONEXCLAMATION);  
    else  
        MessagesBox(hMainWnd, "Unknown Error", "MCI Error",  
MB_ICONEXCLAMATION);  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➤ การเล่นไฟล์ Waveform-Audio

จากตัวอย่างเป็นการเปิด Waveform-Audio และการเล่นไฟล์ Waveform ที่ถูกระบุโดยพารามิเตอร์ lpszWAVEFile

```
DWORD playWAVEFile(HWND hWndNotify,LPSTR lpszWAVEFileName){
    UINT wDeviceID;
    DWORD dwReturn;
    MCI_OPEN_PARMS mciOpenParms;
    MCI_PLAY_PARMS mciplayParms;

    //เปิดอุปกรณ์โดยระบุอุปกรณ์และไฟล์
    //MCI จะเลือกอุปกรณ์ที่สามารถเล่น โดยระบุชื่อไฟล์

    mciOpenParms.lpstrDeviceType = "waveaudio";
    mciOpenParms.lpstrElementName = lpszWAVEFileName;
    if(dwReturn = mciSendCommand(0,MCI_OPEN, MCI_OPEN_TYPE |
        MCI_OPEN_ELEMENT,(DWORD)(LPVOID) &
        mciOpenParms)){

        //ถ้าเปิดอุปกรณ์ล้มเหลว มันจะไม่ปิดจนกว่าจะส่งค่าผิดพลาด
        return(dwReturn);
    }
}
```

//ถ้าเปิดอุปกรณ์สำเร็จ จะให้ ID ของอุปกรณ์

```
wDeviceID = mciOpenParams.wDeviceID;
```

//เริ่ม playback ฟังก์ชันการปฏิบัติการของวินโดว์สำหรับ parent จะไม่แจ้งข่าวสาร

//MM_MCI_NOTIFY เมื่อ play back สำเร็จ

//การเล่นที่ระบุไฟล์ waveform-audio โดยการใช้ MCI_OPEN and MCI_PLAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ส่งกลับเมื่อเริ่ม playback ,ส่ง OL ถ้าสำเร็จ , อื่นๆ
```

```
//ส่งรหัสผิดพลาดของ MCI
```

```
mciPlayParams.dwCallback = (DWORD) hWndNotify;
```

```
if(dwReturn = mciSendCommand(wDeviceID,MCI_PLAY,MCI_NOTIFY,  
                             (DWORD)(LPVOID) &mciPlayParams)){
```

```
    mciSendCommand(wDeviceID,MCI_CLOSE,0,NULL);
```

```
    return(dwReturn);
```

```
}
```

```
return(OL);
```

```
}
```

➤ การเปลี่ยน Current Position

การเปลี่ยน Current Position ในส่วนต่างๆ ของอุปกรณ์ ใช้คำสั่ง Messages MCI_SEEK พร้อมด้วย flag MCI_TO และ โครงสร้าง MCI_SEEK_PARMS ถ้าใช้สมาชิก dwTo เพื่อระบุค้นหาตำแหน่ง โดยใช้ MCI_SEEK จะต้องถาม format time และตั้งค่าให้มัน ถ้าจำเป็นในการเพิ่มเพื่อระบุตำแหน่ง โดยใช้ dwTo สมาชิกสามารถระบุ flags MCI_SEEK_TO_START หรือ MCI_SEEK_TO_END สำหรับพารามิเตอร์ dwParam 1 ของฟังก์ชัน mciSendCommand เพื่อค้นหาตำแหน่งเริ่มแรกและสุดท้ายของส่วนต่างๆ ของอุปกรณ์ ถ้าใช้แฟล็กใดแฟล็กหนึ่ง จะต้องไม่ระบุ flag MCI_TO

➤ การตั้ง Time Format(Setting the Time Format)

ใช้คำสั่ง Messages MCI_SET พร้อมด้วยโครงสร้าง Messages MCI_SET_PARMS เพื่อตั้งค่า time format สำหรับการเปิดอุปกรณ์ ตั้งค่าสมาชิก dwTimeFormat ค่าใดค่าหนึ่งดังนี้

Constant

MCI_FORMAT_BYTES

MCI_FORMAT_MILLISECONDS

MCI_FORMAT_MSF

Time format

Byte(in pulse code modulated[PCM] format files)

Milliseconds

Minute/second/frame

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

MCI_FORMAT_SAMPLES	Samples
MCI_FORMAT_SMPTE_24	SMPTE,24 frame
MCI_FORMAT_SMPTE_25	SMPTE,25 frame
MCI_FORMAT_SMPTE_30	SMPTE,30 frame
MCI_FORMAT_SMPTE_30DROP	SMPTE,30 frame drop
MCI_FORMAT_TMSF	Track/Minute/second/frame
MCI_SEQ_FORMAT_SONGPTR	MIDI song pointer

จากตัวอย่างการตั้ง time format แบบ milliSeconds บนอุปกรณ์ที่ถูกระบุโดยตัวแปล

```

UINT wDeviceID;
MCI_SETPARMS mciSetParms;

//ตั้งค่า time format แบบ milliSeconds

mciSetParms.dwTimeFormat = MCI_FORMAT_MILLISECONDS;
if(mciSendCommand(wDeviceID,MCI_SET, MCI_SET_TIME_FORMAT,(DWORD) &
mciSetPams))

//ผิดพลาด ไม่สามารถตั้งค่า Time format ได้
return FALSE;

else
//ตั้งค่า time format สำเร็จ
return TRUE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➤ การใช้ Flag MCI_NOTIFY (Using the MCI_NOTIFY Flag)

จากตัวอย่างแสดง Flag MCI_NOTIFY ที่ถูกใช้โดยคำสั่ง MCI_PLAY การควบคุมการปฏิบัติของวินโดวนั้นจะทำการสร้าง Messages MM_MCINOTIFY ที่ถูกระบุใน hwnd

```
MCI_DGV_PLAY_PARMS mciPlay;
DWORD dwFlags;
mciPlay.dwCallback = MAKELONG(hwnd,0);
DwFlages = MCI_NOTIFY;
mciSendCommand(wMCIDeviceID,MCI_PLAY,dwFlags,(DWORD)(LPSTR)&mciPlay);
```

➤ การบันทึกเสียงโดยอุปกรณ์ Waveform-audio

จากตัวอย่างเปิดอุปกรณ์ Waveform-audio ด้วยไฟล์ใหม่ บันทึกเสียงสำหรับระยะเวลาทำการบันทึก และเตือนให้ผู้ใช้บันทึกข้อมูลถ้าต้องการ

```
//ใช้คำสั่ง MCI_OPEN,MCI_RECORD และ MCI_SAVE เพื่อบันทึกเสียงและเก็บ
//ไฟล์ Waveform-audio จะส่งค่า OL จะทำสำเร็จ,อื่นๆ จะส่งรหัส MCI error
```

```
DWORD recordWAVEFile(DWORD dwMilliSeconds){
    UINT wDeviceID;
    DWORD dwReturn;
    MCI_OPEN_PARMS mciOpenParms;
    MCI_RECORD_PARMS mciRecordParms;
    MCI_SAVE_PARMS mciSaveParms;
    MCI_PLAY_PARMS mciplayParms;

    //เปิดอุปกรณ์ Waveform-audioกับไฟล์ใหม่ สำหรับบันทึกเสียง
    mciOpenParms.lpstrDeviceType = "waveaudio";
    mciOpenParms.lpstrElementName = "";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (dwReturn = mciSendCommand(0,MCI_OPEN,MCI_OPEN_ELEMENT |
                                MCI_OPEN_TYPE,
                                (DWORD)(LPVOID)&mciOpenParms)){
    //ถ้าการเปิดอุปกรณ์ล้มเหลว ; มันจะไม่ปิด จนกว่าจะส่งค่าผิดพลาด
    return(dwReturn);
}

```

//ถ้าเปิดอุปกรณ์สำเร็จ; จะให้ ID device

```
wDeviceID = mciOpenParms.wDeviceID;
```

//เริ่มการบันทึกเสียง และบันทึกจำนวนเวลาที่ถูกระบุ รอการบันทึกเสียงจนกว่าสำเร็จก่อนที่จะทำ

//ต่อไป

```
mciRecordParams.dwTo = dwMilliseconds;
```

```
if (dwReturn = mciSendCommand(wDeviceID,MCI_RECORD,
                                MCI_TO|MCI_WAIT,
                                (DWORD)(LPVOID)&mciRecordParms)){

```

```
    mciSendCommand(wDeviceID,MCI_CLOSE,0,NULL);
```

```
    return(dwReturn);
}

```

//บันทึกเสียงและถามผู้ใช้งานว่าเก็บไฟล์หรือไม่

```
mciplayParms.dwForm = OL;
```

```
if (dwReturn = mciSendCommand(wDeviceID,MCI_PLAY,
                                MCI_FROM |MCI_WAIT,
                                (DWORD)(LPVOID)&mciPlayParms)){

```

```
    mciSendCommand(wDeviceID,MCI_CLOSE,0,NULL);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(dwReturn);
}
if(MassegesBox(hMainWnd, "Do you want to save this recording?",
    "",MB_YESNO)==IDNO){
mciSendCommand(wDeviceID,CLOSE,0,NULL);
return(OL);
}

//บันทึกเสียง TEMPFILE.WAV รอจนกว่าจะทำงานสำเร็จก่อนที่จะต่อ

mciSaveParms.lpfilename= "tempfile.wav";

if (dwReturn = mciSendCommand(wDeviceID,MCI_SAVE,
    MCI_SAVE_FILE |MCI_WAIT,
    (DWORD)(LPVOID)&mciSaveParms)){
mciSendCommand(wDeviceID,MCI_CLOSE,0,NULL);
return(dwReturn);
}
return(OL);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การเขียนโปรแกรมบน Windows

4.1 ลักษณะของการเขียนโปรแกรมบน Windows

ในการเขียนโปรแกรมบนคอมพิวเตอร์ โปรแกรมเมอร์ภาษา C ส่วนมากมักจะติดต่อกับ I/O และ หน่วยความจำโดยฟังก์ชันมาตรฐานในไลบรารีภาษา C ฟังก์ชันเหล่านี้จะมอง I/O เป็นอุปกรณ์แบบ ตัวหนังสือเท่านั้น ส่วนการติดต่อกับหน่วยความจำก็จะติดต่อโดยตรงอย่างอิสระ การกระทำเหล่านี้ จะทำไม่ได้ใน Windows ที่ซึ่งแอปพลิเคชันแต่ละแอปพลิเคชันจะต้องป็นส่วนต่างๆ ใช้ร่วมกัน เช่น คีย์บอร์ด หน่วยความจำ พอร์ตต่างๆ จอภาพ หรือแม้กระทั่งตัวประมวลผลหรือ CPU โดยจะถือว่า สิ่งเหล่านี้คือทรัพยากร (Resource) ที่ต้องแบ่งกันใช้

ความแตกต่างระหว่างการเขียนโปรแกรมบนคอมพิวเตอร์กับ Windows ได้แก่

1. ระบบติดต่อกับผู้ใช้ จุดประสงค์หลักอันหนึ่งของการออกแบบ Windows ก็เพื่อที่จะให้ ผู้ใช้ สามารถใช้งานแอปพลิเคชันหลายๆ แอปพลิเคชันพร้อมกัน รวมทั้งยังให้ผู้ใช้สามารถมองเห็น การทำงานของทุกๆ แอปพลิเคชันได้พร้อมกันอีกด้วย ด้วยเหตุที่ว่าใช้งานหลายๆ แอปพลิเคชัน พร้อมกันนั้น แต่ละแอปพลิเคชันจึงต้องแบ่งปันจอภาพเพื่อแสดงผลต่อผู้ใช้ สำหรับระบบงานที่ คล้ายกันบางระบบจะเจาะจงยกจอภาพทั้งจอให้แก่แอปพลิเคชันเดียวใช้งาน การแสดงของแอปพลิเคชันอื่นๆ จะถูกซ่อนไว้ข้างหลัง (หรือเรียกว่า Background Process) แต่ในระบบ Windows จะแตกต่างออกไปแอปพลิเคชันที่ทำงานบน Windows ทุกๆ ตัว จะได้ใช้จอภาพแสดงผลกันทั่วหน้า ใหญ่ บ้าง เล็กบ้างตามแต่ผู้ใช้ต้องการ และสามารถแสดงให้แก่ผู้ใช้ตลอดเวลา

แต่ละแอปพลิเคชันจะใช้จอภาพแสดงผลร่วมกับแอปพลิเคชันอื่นๆ โดยใช้งานเฉพาะใน ส่วนที่อยู่ในกรอบสี่เหลี่ยมของคนที่เรียกว่า วินโดว์ (Windows) หากมองกันผิวเผินวินโดว์จะเป็น รูปสี่เหลี่ยมที่มีกรอบธรรมดาเท่านั้น แต่ที่จริงแล้ววินโดว์ไม่เพียงแต่จะเป็นอุปกรณ์แสดงผลขั้นพื้นฐานของแอปพลิเคชันเท่านั้น แต่ยังเป็นสิ่งที่ผู้ใช้สามารถติดต่อกับแอปพลิเคชันได้เป็นอย่างดีโดย ผ่านส่วนประกอบต่างๆ ของวินโดว์ เช่น เมนู ปุ่มบังคับ สไลด์บาร์ (Scroll bar) เป็นต้น

เนื่องจาก Windows มีวินโดว์ให้แอปพลิเคชันหลายแบบ Windows จึงไม่สร้างวินโดว์ล่วงหน้าให้กับแอปพลิเคชันใดๆ แอปพลิเคชันเหล่านั้นจะต้องสร้างวินโดว์ของตน ในแบบที่ตน ต้องการเอง ซึ่งต่างจกคอมพิวเตอร์ที่จะเตรียมจอภาพพื้นฐานให้ได้ทันที แต่ Windows ก็จะมีระบบจัดการ พื้นฐานสำหรับวินโดว์อำนวยความสะดวกสำหรับผู้ใช้อยู่แล้วเช่น การเคลื่อนย้าย การเปลี่ยนขนาด การลดขนาดให้เป็นไอคอน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อได้เปรียบของ Windows อีกประการคือ จากการใช้แต่ละแอปพลิเคชันมีวินโดว์เป็นของตนเอง จึงสามารถแน่ใจได้เลยว่าแอปพลิเคชันจะไม่แย่งกันใช้จอภาพ จนเกิดความสับสนในการทำงาน

2. ระบบการรับข้อมูลแบบใหม่ ในระบบคอมพิวเตอร์เมื่อแอปพลิเคชันจะรับข้อมูลจากผู้ใช้งาน เช่น ผ่านทางฟังก์ชัน `getchar()` ก็จำเป็นต้องรอจนกว่าผู้ใช้งานจะได้ข้อมูลจริงๆ เสียก่อน หากในระหว่างนั้นผู้ใช้งานจะใส่ข้อมูลทางอื่น เช่น เม้าส์ ก็จะไม่สามารถรับข้อมูลเข้ามาได้ แต่ใน Windows จะแตกต่างกันไป Windows จะรวบรวมข้อมูลเข้าจากทุกๆ อุปกรณ์ แล้วเก็บไว้ในคิวเมสเสจ (Message Que) เมื่อแอปพลิเคชันต้องการอ่านเข้าก็เพียงแค่อ่านเมสเสจคิวเท่านั้นแอปพลิเคชันจะได้รับข้อมูลจากผู้ใช้งานโดยไม่มีการสูญหายของข้อมูล

ข้อมูลที่อยู่ในคิวเมสเสจจะอยู่ในรูปของเมสเสจที่มีขนาดในการเก็บเท่ากันและมีรูปแบบการใช้งานคล้ายกันในการอ่านค่าแต่ละครั้งการพิจารณาข้อมูลก็จะใช้โปรแกรมในลักษณะที่คล้ายกัน ทำให้การเขียนแอปพลิเคชันง่ายและสะดวกขึ้น

การอ่านข้อมูลอุปกรณ์ต่างๆ ก็เช่นกัน ในระบบ Windows การอ่านค่าเมสเสจคิวใช้ฟังก์ชันเพียงฟังก์ชันเดียวเท่านั้น คือ `GetMessage()` แต่ในคอสมอสจะยุ่งยากมากกว่า คือหากเราต้องการอ่านค่าคีย์บอร์ดเป็นตัวหนังสือ ก็ต้องใช้ฟังก์ชัน `keypressed()` การที่ต้องใช้หลายฟังก์ชันในการทำงานที่มีลักษณะคล้ายกันนั้น เมื่อเทียบกับการใช้ฟังก์ชัน `GetMessage()` เพียงฟังก์ชันเดียว จะเห็นชัดว่าการเขียนแอปพลิเคชัน Windows นั้นเรียบง่ายและสะดวกสบายมากกว่า

นอกจากนี้ข้อมูลที่เกิดจากผู้เลือกเมนูหรือปุ่มบังคับต่างๆ ก็ยังบรรจุเอาไว้ในคิวเมสเสจด้วย ดังนั้นการรับรู้การกระทำทั้งหมดของผู้ใช้ทุกอย่างจึงใช้ฟังก์ชัน `GetMessage()` เพียงฟังก์ชันเดียวเท่านั้น

3. ระบบกราฟิก จากระบบคอสมอสที่ไม่มีการสนับสนุนระบบกราฟิกเลย มาสู่ Windows ที่เต็มไปด้วยคำสั่งและวิธีการอันหลากหลาย ซับซ้อน และมีประสิทธิภาพ สำหรับการจัดการระบบกราฟิกไม่ว่าจะเป็น เส้นตรง วงกลม รูปเหลี่ยม รูปทึบ หรือเส้นโค้ง ไม่เพียงเท่านั้นคำสั่งทุกคำสั่งยังสามารถใช้ได้กับอุปกรณ์แสดงผลทุกชนิดที่สามารถผลเป็นกราฟิกได้ เราสามารถใช้คำสั่งเพียงคำสั่งเดียวกัน รูปแบบเดียวกัน วาดรูปต่างๆ ได้ทั้งบนจอภาพ เครื่องพิมพ์ (กราฟิก) พล็อตเตอร์ ฯลฯ

4. ระบบการจัดงาน ในการทำงานในคอสมอสนั้นจะไม่มีระบบจัดการป็นส่วนทรัพยากรใดๆ เมื่อแอปพลิเคชันหนึ่งๆ ทำงานก็ถือเสมือนว่าแอปพลิเคชันนั้นเป็นเจ้าของทรัพยากรทั้งหมด เช่น คีย์บอร์ด จอภาพ หน่วยความจำ หรือแม้กระทั่งซีพียูสำหรับ Windows จะมีการบังคับให้แอปพลิเคชันมีการป็นส่วนทรัพยากรต่างๆ ร่วมกันเสมอ ซึ่งต่างจากแอปพลิเคชันในคอสมอสที่แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถนำหน่วยความจำที่ไม่ได้ยึดครอง โดยคอสนำไปใช้ทั้งหมดโดยไม่ต้องคำนึงถึงแอปพลิเคชันอื่นๆ

แต่ใน Windows อันเป็นระบบหลายงาน (Multitasking) นั้นแตกต่างออกไป พื้นที่หน่วยความจำซึ่งเป็นทรัพยากรอันจำกัดนั้นจะต้องปันส่วนกันใช้ หากทุกแอปพลิเคชันต่างใช้สิ้นเปลืองหน่วยความจำอันจำกัดก็จะหมดไปอย่างรวดเร็ว จึงทำให้แอปพลิเคชันต่างๆ ทำงานพร้อมกันได้ไม่มาก เพียงหนึ่งหรือสองแอปพลิเคชัน ฉะนั้นในการทำงานจริงของ Windows จึงมีการเคลื่อนย้ายสลับเปลี่ยน (Swap) หรือทิ้ง (Discard) ส่วนต่างๆ ของหน่วยความจำตลอดเวลาเพื่อให้การทำงานทั้งระบบมีประสิทธิภาพสูงสุด การใช้หน่วยความจำแต่ละครั้งมักจะพบว่ามียู่อหลายครั้งที่เดียวที่ข้อมูลที่ได้เข้าจะไม่อยู่ที่ตำแหน่งแอดเดรสเดิม

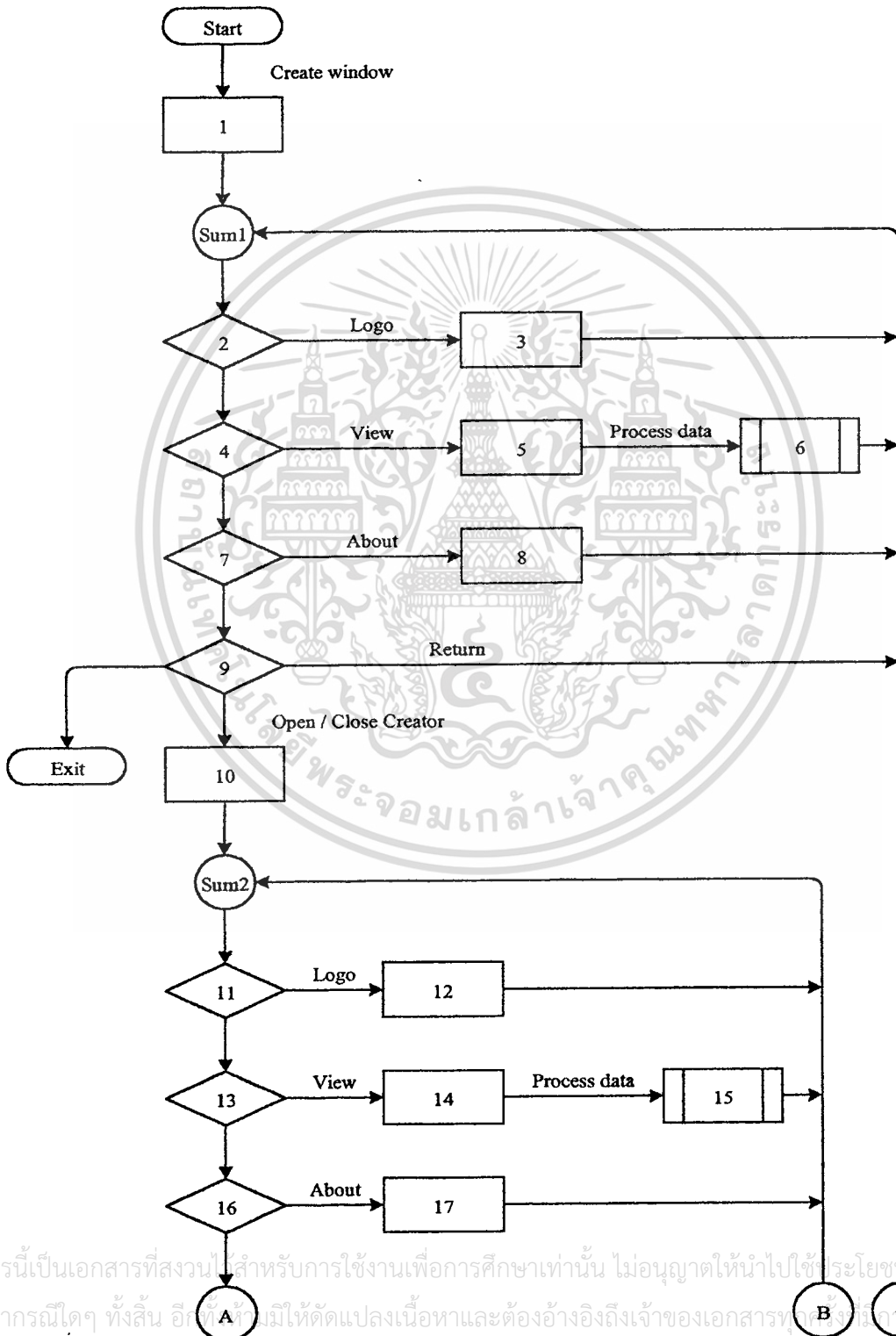
อีกตัวอย่างคือการจัดจอภาพที่จะต้องแบ่งกันใช้ในหลายๆ แอปพลิเคชัน ในระบบคอสจะยกจอภาพทั้งจอไปขึ้นอยู่กับการจัดการของแอปพลิเคชันเพียงแอปพลิเคชันเดียว ไม่ว่าแอปพลิเคชันนั้นจะทำการคัดแปลงจอภาพอย่างไร เช่นเปลี่ยนความละเอียดในการแสดงผล เปลี่ยนสีที่ใช้ในการแสดงผล ฯลฯ แต่ในระบบ Windows จะเปลี่ยนไป แอปพลิเคชัน จะต้องใช้จอภาพร่วมกันเพราะฉะนั้นแอปพลิเคชันที่ทำงานบน Windows จะต้องไม่มีการคัดแปลงการใช้งานจอภาพโดยตรงข้ามมีการขอใช้โดยผ่าน Windows เสมอ

4.2 หลักการทำงานของโปรแกรม Multimedia Story Presentation System

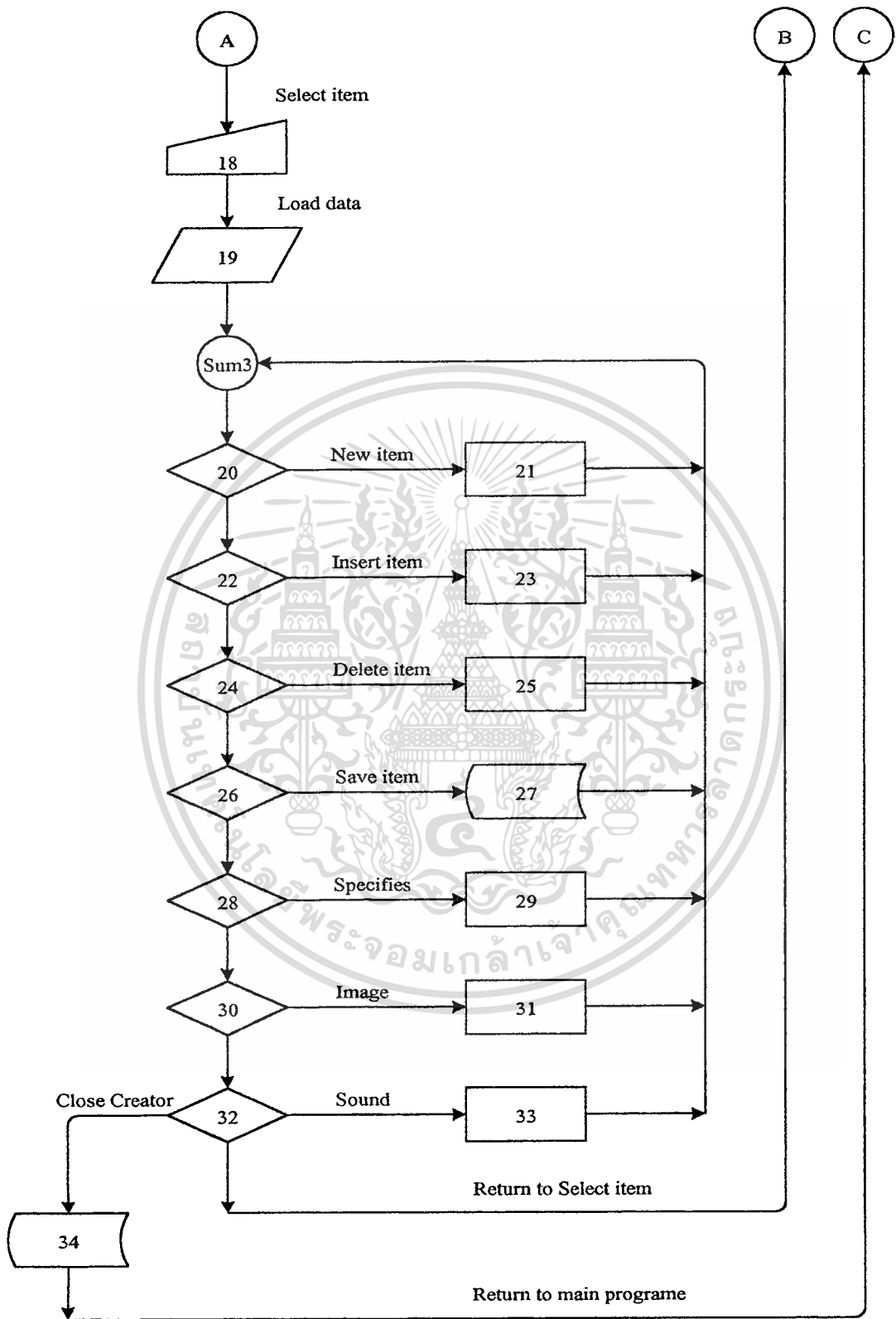
โปรแกรม Multimedia Story Presentation System เป็นโปรแกรมที่ให้นำเสนอข้อมูลข่าวสารในรูปแบบของภาพและเสียง และต้องให้ข้อมูลของเราเป็นสิ่งที่น่าสนใจมากขึ้น ดังนั้นการนำเสนอข้อมูลจึงต้องมีรูปแบบแปลกใหม่ ดังนั้นโปรแกรม Multimedia จะเน้นที่รูปของการนำเสนอข้อมูล เช่น ในของภาพที่ใช้ในการนำเสนอข้อมูลจะสามารถนำเสนอในลักษณะมีดๆ และค้อยๆ ซัดเงินขึ้นเป็นปกติ แสดงเป็นจุดเล็กๆ เพิ่มจนเป็นรูปปกติ, ภาพหมุน เป็นต้น ซึ่งนอกจากนี้ยังมีข้อความประกอบคำอธิบาย หรือเป็นเสียงที่บันทึกไว้บรรยายประกอบ ซึ่งคุณสมบัติต่างๆ ของภาพและเสียงเหล่านี้ จะสามารถกำหนดได้ว่าต้องการแสดงแบบใด และยังสามารถบันทึกข้อมูลเหล่านี้สามารถบันทึกเก็บเป็นไฟล์ได้ด้วย เพื่อสะดวกในการใช้งานในภายหลัง และสามารถที่จะนำข้อมูลเหล่านี้มาแก้ไขบางส่วนที่ต้องการแก้ไขได้เช่นกัน ซึ่งการคัดแปลงแก้ไขก็สามารถทำได้ง่ายเพียงเปิดไฟล์ข้อมูลขึ้นมาแล้วกำหนดคุณสมบัติของภาพและเสียงเหล่านั้นเป็นค่าใหม่ตามที่ต้องการซึ่งจากที่กล่าวมาโปรแกรมจะประกอบด้วยส่วนการทำงานสองส่วนคือ ส่วนที่ให้นำเสนอข้อมูลและส่วนที่เพิ่ม ลด คัดแปลง แก้ไขข้อมูล และลักษณะการใช้งานของโปรแกรมจะเป็นแบบ Virtual คือเมื่อมีการเพิ่มลดคัดแปลงแก้ไขข้อมูล จะเห็นการเปลี่ยนแปลงตลอดทำให้ง่ายต่อการใช้งาน

4.3 บล็อกไดอะแกรมของโปรแกรม Multimedia Story Presentation System

บล็อกไดอะแกรมการทำงานของโปรแกรม ระบบนำเสนอเรื่องราวหลายสื่อ สามารถแสดงได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารที่ต้นฉบับไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ทางวิศวกรรมท่าอากาศยานภูเก็ตนำมาใช้ประโยชน์ด้านการค้า
 รูปที่ 4.3 - 1 บล็อกไดอะแกรมของโปรแกรม
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปของบล็อกไดอะแกรมสามารถอธิบายบล็อกไดอะแกรมหมายเลขต่างๆได้

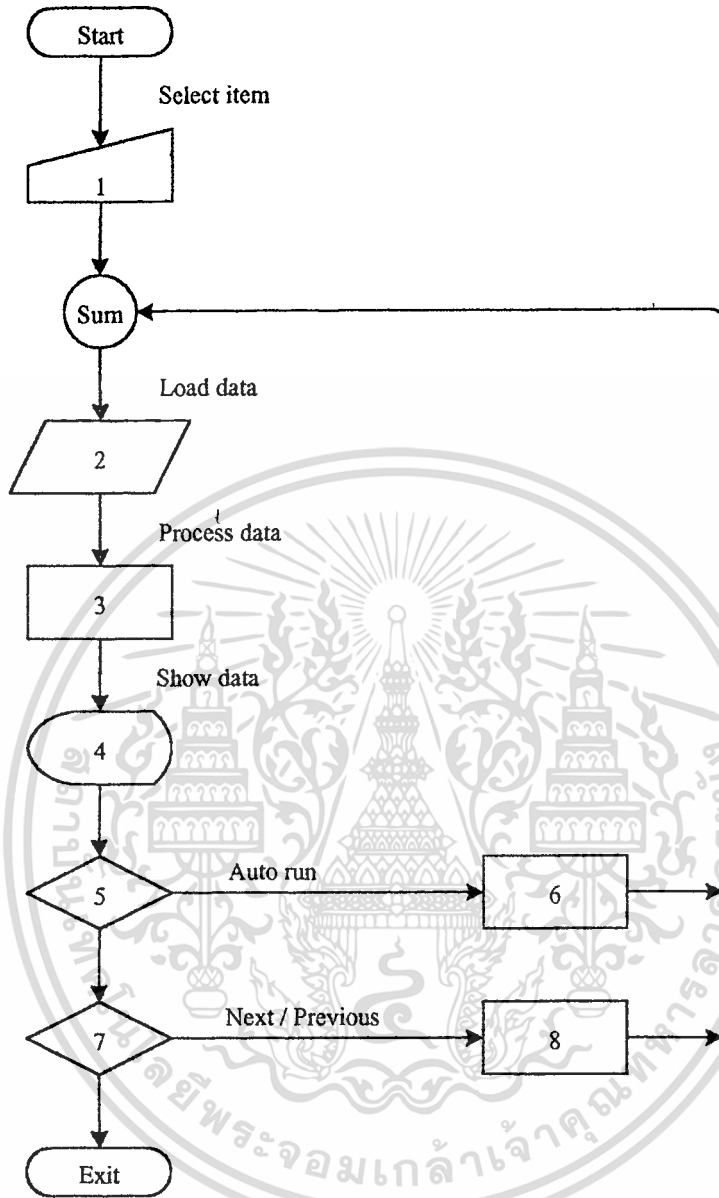
1. สร้างวินโดว์หลัก
2. เลือกการทำงานไปยังส่วนการทำงานของ Logo
3. ทำการแสดง Logo ของโปรแกรม
4. เลือกการทำงานไปยังส่วนของ View
5. แสดง Icon ของรายการข้อมูลที่บันทึกไว้
6. นำข้อมูลของ Icon ที่ถูกเลือกมาประมวลผลเพื่อจะนำข้อมูลออกมาแสดงในลักษณะของภาพและเสียง
7. เลือกการทำงานไปยังส่วนของ About
8. ทำการแสดงข้อมูลเกี่ยวกับ โปรแกรม
9. เลือกการทำงานไปยังส่วนการสร้าง, แก้ไขข้อมูล หรือออกจากการทำงานของโปรแกรม หรือ กลับไปเลือกรายการอื่นๆ
10. เปิดหรือปิดการทำงานในส่วนของการสร้างแก้ไขข้อมูล
11. เลือกการทำงานไปยังส่วนการทำงานของ Logo
12. ทำการแสดง Logo ของโปรแกรม
13. เลือกการทำงานไปยังส่วนของ View
14. แสดง Icon ของรายการข้อมูลที่บันทึกไว้
15. นำข้อมูลของ Icon ที่ถูกเลือกมาประมวลผลเพื่อจะนำข้อมูลออกมาแสดงในลักษณะของภาพและเสียง
16. เลือกการทำงานไปยังส่วนของ About
17. ทำการแสดงข้อมูลเกี่ยวกับ โปรแกรม
18. เลือกรายการ (Icon) เพื่อสร้าง,แก้ไขข้อมูล
19. ทำการโหลดข้อมูลของรายการที่ถูกเลือกมาแสดงเพื่อตรวจสอบ หรือแก้ไขข้อมูล
20. เลือกการทำงานไปยังส่วนการทำงานของ View
21. ทำการสร้างรายการของข้อมูลใหม่โดยการต่อท้ายข้อมูลเดิม
22. เลือกการทำงานไปยังส่วนของการทำงาน Insert
23. ทำการสร้างรายการของข้อมูลใหม่โดยการแทรกรายการข้อมูลใหม่ตรงตำแหน่งรายการข้อมูลที่ถูกเลือก
24. เลือกการทำงานไปยังส่วนของการทำงาน Delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

25. ทำการลบรายการของข้อมูลตรงตำแหน่งรายการข้อมูลที่ถูกเลือก
26. เลือกการทำงานไปยังส่วนของการทำงาน Save
27. ทำการบันทึกรายการของข้อมูล, ข้อมูลที่เกิดจากการเปลี่ยนแปลงของการสร้าง, แก้ไขข้อมูล
28. เลือกการทำงานไปยังส่วนของการทำงาน Specifies
29. เป็นส่วนการทำงานที่ใช้กำหนด, แก้ไขคุณสมบัติต่างๆของหัวข้อของรายการที่ถูกเลือก
30. เลือกการทำงานไปยังส่วนของการทำงาน Image
31. เป็นส่วนที่ใช้ทำการเพิ่ม, เปลี่ยนรูปภาพของข้อมูล
32. เลือกการทำงานไปยังส่วนของการทำงาน Sound หรือปิดการทำงานในส่วน of Creator
33. เป็นส่วนของการทำงานที่ใช้บันทึกเสียง, เพิ่มหรือเปลี่ยนเสียงดนตรีประกอบ
34. ทำการบันทึกข้อมูลที่มีการเปลี่ยนแปลง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

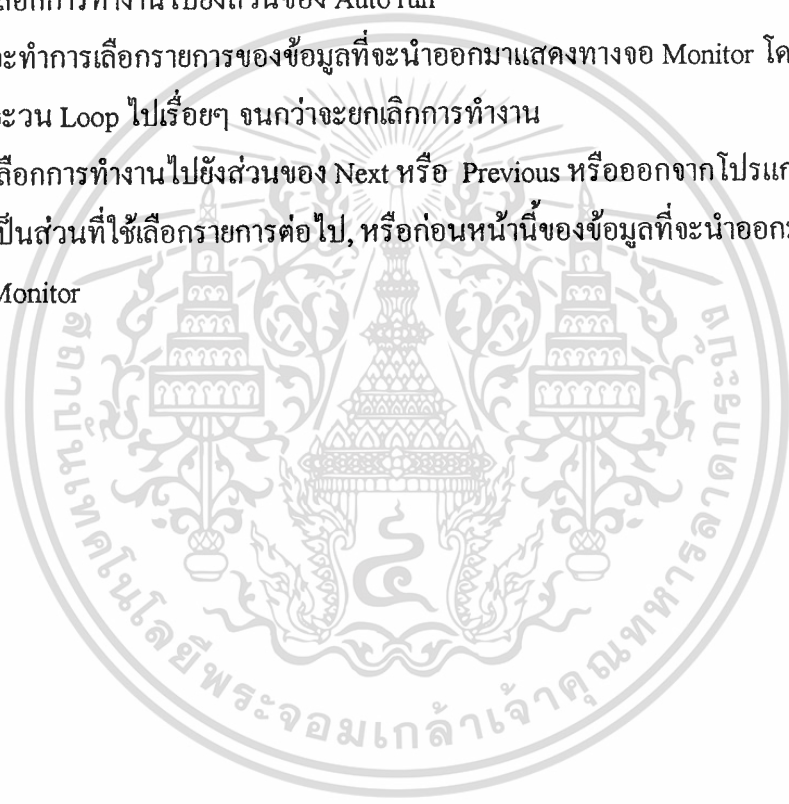


รูปที่ 4.3 - 2 บล็อกไดอะแกรมของโปรแกรมย่อย Process data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาคารรูปของบล็อกไดอะแกรมย่อย Process data สามารถอธิบายบล็อกไดอะแกรม
หมายเลขต่างๆได้ดังนี้

1. ทำการรับข้อมูลจากรายการของข้อมูลที่ถูกเลือก
2. ทำการโหลดข้อมูลจากรายการที่ถูกเลือก
3. ทำการประมวลผลข้อมูลที่ประกอบไปด้วยข้อมูลของภาพ,เสียง, และภาพเคลื่อนไหว
4. แสดงผลของข้อมูลที่ประกอบไปด้วยรูปภาพ, ภาพเคลื่อนไหว, และเสียงออกสู่จอ Monitor
5. เลือกการทำงานไปยังส่วนของ Auto run
6. จะทำการเลือกรายการของข้อมูลที่จะนำออกมาแสดงทางจอ Monitor โดยอัตโนมัติ โดยจะวน Loop ไปเรื่อยๆ จนกว่าจะยกเลิกการทำงาน
7. เลือกการทำงานไปยังส่วนของ Next หรือ Previous หรือออกจากโปรแกรมย่อย
8. เป็นส่วนที่ใช้เลือกรายการต่อไป, หรือก่อนหน้าของข้อมูลที่จะนำออกมาแสดงทางจอ Monitor



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การใช้งานโปรแกรม Multimedia Story Presentation System

โปรแกรม Multimedia Story Presentation System จะประกอบด้วยส่วนของวินโดว์สองส่วนดังนี้ คือ ส่วนของวินโดว์หลัก และ ส่วนของการสร้าง, แก้ไขข้อมูล (Tool box)



รูปที่ 4.4-1 แสดงวินโดว์หลักของโปรแกรม



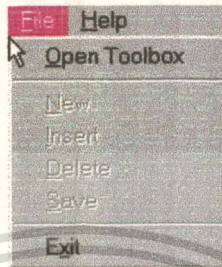
รูปที่ 4.4-2 แสดงวินโดว์ Toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 วินโดว์หลักของโปรแกรม ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

➤ **Menu** ประกอบด้วยเมนู ดังต่อไปนี้

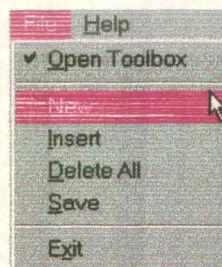
- ◆ **File** ประกอบด้วยตัวเลือกต่างๆ เพื่อเลือกรายการในทำงานดังต่อไปนี้



- **Open Toolbox** เพื่อเลือกการทำงานในส่วนของการสร้าง, แก้ไข รายละเอียด, บันทึก ข้อมูลของ ภาพ และ เสียง

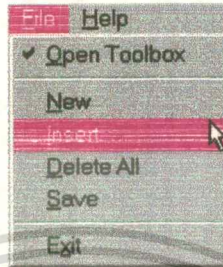


- **New** เพื่อเลือกการทำงานในส่วนของการสร้าง ข้อมูลของรายการใหม่โดย ต่อท้าย Child ของรายการชุดเดิม ข้อมูลชุดใหม่จะประกอบด้วยข้อมูล ภาพ และ เสียง

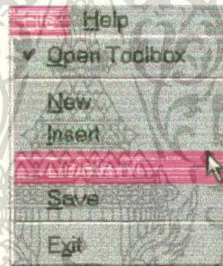


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

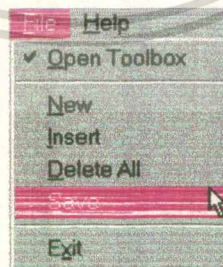
- **Insert** เพื่อเลือกการทำงานในส่วนของการสร้างข้อมูลของรายการใหม่
แทรกตรงตำแหน่งที่ถูกเลือก ข้อมูลใหม่จะประกอบด้วยข้อมูล ภาพ และ
เสียง



- **Delete หรือ Delete All** เพื่อเลือกการทำงานในส่วนของการลบ ข้อมูลของ
ภาพ, เสียง และรายละเอียดอื่นๆ ของรายการที่ถูกเลือก

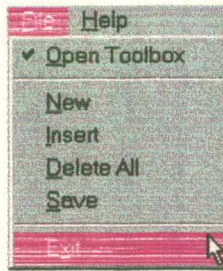


- **Save** เพื่อเลือกการทำงานในส่วนของการบันทึก ข้อมูลของภาพ, เสียง และ
รายละเอียดอื่นๆ ของรายการเมื่อมีการเปลี่ยนแปลง



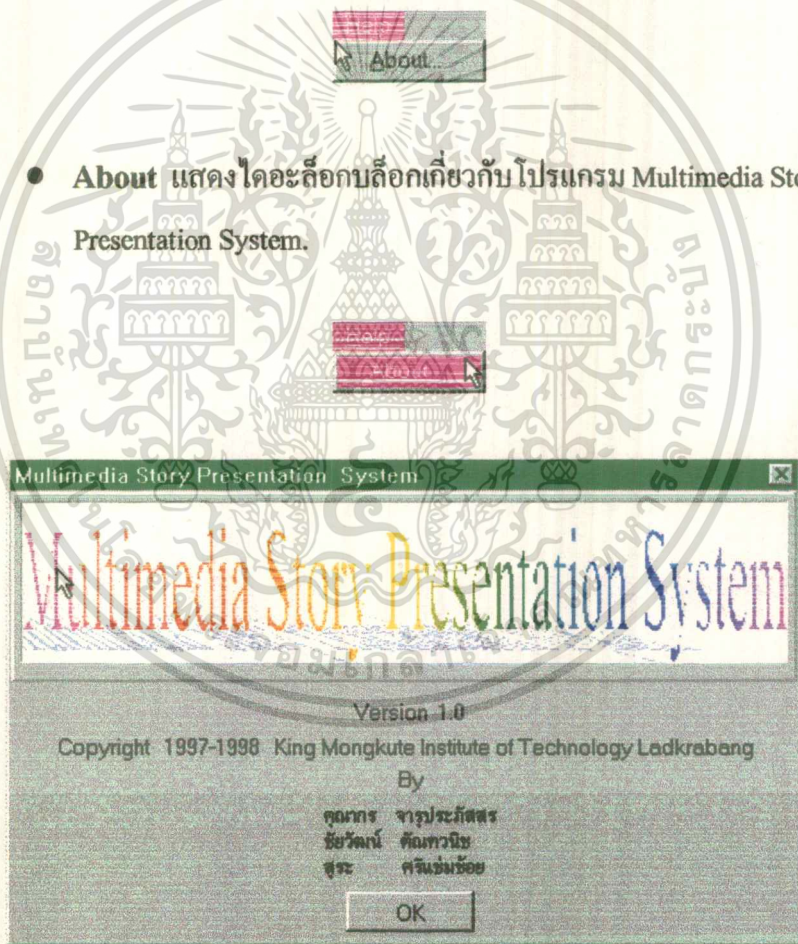
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Exit** ออกจากการทำงานของโปรแกรม



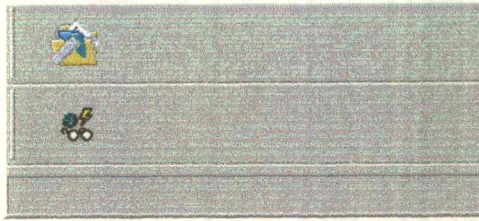
- ◆ **Help** ประกอบด้วยตัวเลือก เพื่อเลือกรายการในการทำงานดังนี้

- **About** แสดงไดอะล็อกบ็อกเกี่ยวกับ โปรแกรม Multimedia Story - Presentation System.

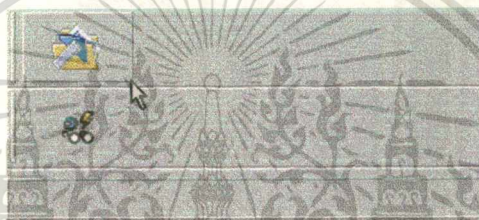


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

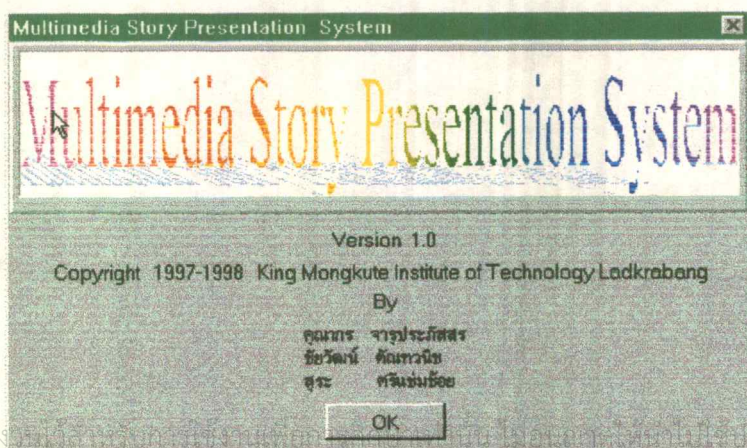
➤ **Toolbar** ประกอบด้วยปุ่ม (Button) ต่างๆเพื่อเลือกการทำงานดังต่อไปนี้



◆ **Toolbox** เพื่อเลือกการทำงานในส่วนของการสร้าง, แก้ไขรายละเอียด, บันทึก ข้อมูลของ ภาพ และ เสียง



◆ **About** แสดงรายละเอียดเกี่ยวกับ โปรแกรม Multimedia Story - Presentation System.



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับงานเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➤ **Tab Display** ประกอบด้วยปุ่ม Tab ต่างๆ เพื่อแสดงการทำงานดังต่อไปนี้

◆ **Logo** แสดง logo ของโปรแกรม



◆ **View** แสดงรายการข้อมูลที่ถูกสร้างขึ้นเพื่อใช้ในการนำเสนอข้อมูล



เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการนำเสนอข้อมูล
ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในสิ่งที่ปรากฏและขอสงวนสิทธิ์ในการนำไปใช้

นอกจากนี้ในส่วนของการแสดงข้อมูล ยังประกอบไปด้วยปุ่มต่างๆ ดังนี้

- **Previous** ใช้สำหรับแสดงข้อมูลชุดก่อนหน้า



- **Next** ใช้สำหรับแสดงข้อมูลชุดถัดไป



- **Auto Run** ใช้สำหรับแสดงข้อมูลชุดถัดไปแบบอัตโนมัติ โดยจะวนลูปต่อไปเรื่อย จนกว่าจะยกเลิกคำสั่ง



- **Exit** ออกจากส่วนของการแสดงข้อมูล



4.3.2 วินโดว์ของ Toolbox

ใช้สำหรับการสร้าง, แก้ไข และกำหนดรายละเอียดต่างๆ ของข้อมูล ซึ่งประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

➤ **Toolbar** ประกอบด้วยปุ่ม (Button) ต่างๆ เพื่อเลือกการทำงานดังต่อไปนี้

- ◆ **New** เพื่อเลือกการทำงานในส่วนของการสร้าง ข้อมูลของรายการใหม่โดยต่อท้ายข้อมูลชุดเดิม ข้อมูลชุดใหม่จะประกอบด้วยข้อมูล ภาพ และ เสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ◆ **Insert** เพื่อเลือกการทำงานในส่วนของการสร้าง ข้อมูลของรายการใหม่ตรงตำแหน่งที่ถูกเลือก ข้อมูลใหม่จะประกอบด้วยข้อมูล ภาพ และ เสียง



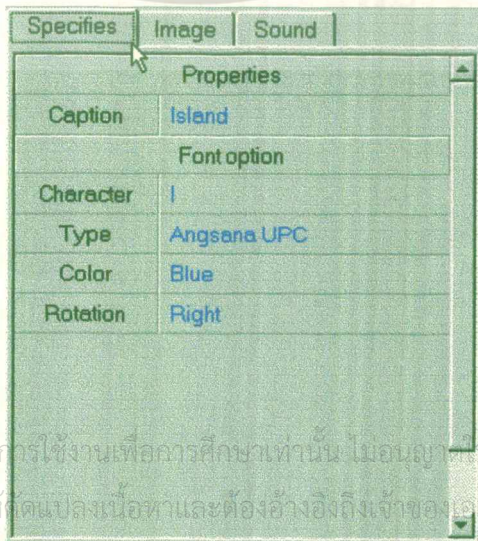
- ◆ **Delete หรือ Delete All** เพื่อเลือกการทำงานในส่วนของการลบ ข้อมูลของภาพ, เสียง และรายละเอียดอื่นๆ ของรายการที่ถูกเลือก



- ◆ **Save** เพื่อเลือกการทำงานในส่วนของการบันทึก ข้อมูลของภาพ, เสียง และรายละเอียดอื่นๆ ของรายการเมื่อมีการเปลี่ยนแปลง



- **Tab Properties** ใช้กำหนดรายละเอียดต่างๆ ของรายการที่ถูกสร้างขึ้น มี Tab ดังต่อไปนี้
- ◆ **Tab Specifies** ใช้กำหนดรายละเอียดเกี่ยวกับ Caption ของรายการที่ถูกสร้างขึ้น ซึ่งประกอบด้วยส่วนต่างๆ ดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Caption** ใช้แก้ไขชื่อของรายการที่ถูกสร้างขึ้น

Caption	Island
---------	--------

- **Character** ใช้เลือกตัวอักษรของชื่อรายการเพื่อใช้ในการกำหนดรายละเอียดของตัวอักษร

Character	
-----------	--

- **Type** ใช้กำหนดชนิดตัวพิมพ์ของตัวอักษรของชื่อรายการที่ถูกสร้าง

Type	Angsana UPC
------	-------------

Type	
------	--

Cordia UPC

ค่าเริ่มต้น คือ Angsana UPC

- **Color** ใช้กำหนดสีของตัวอักษรของชื่อรายการที่ถูกสร้าง

Color	Blue
-------	------

Color	
-------	--

- Green
- Orange
- Purple
- Red
- Yellow

ค่าเริ่มต้น คือ Blue

- **Rotation** ใช้กำหนดทิศทางการหมุนของตัวอักษรของชื่อรายการที่ถูกสร้าง

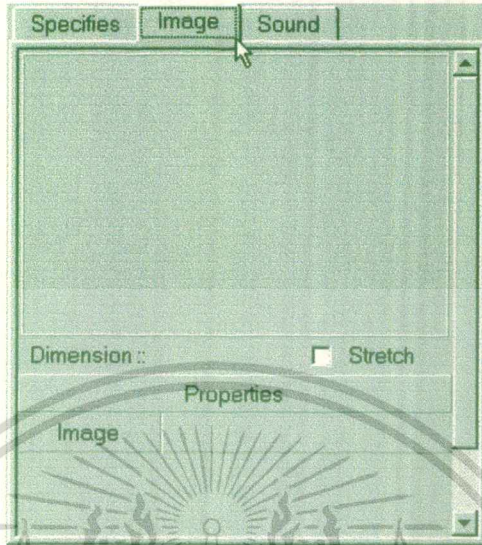
Rotation	Right
----------	-------

Rotation	
----------	--

Left

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ◆ **Tab Image** ใช้เพิ่มรูปภาพเพื่อใช้ประกอบ รายการที่ถูกสร้างขึ้น

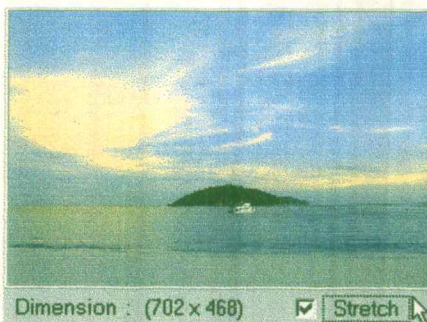


ซึ่งประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

- **Preview** ใช้แสดงรูปภาพที่ถูกเพิ่มเข้ามา

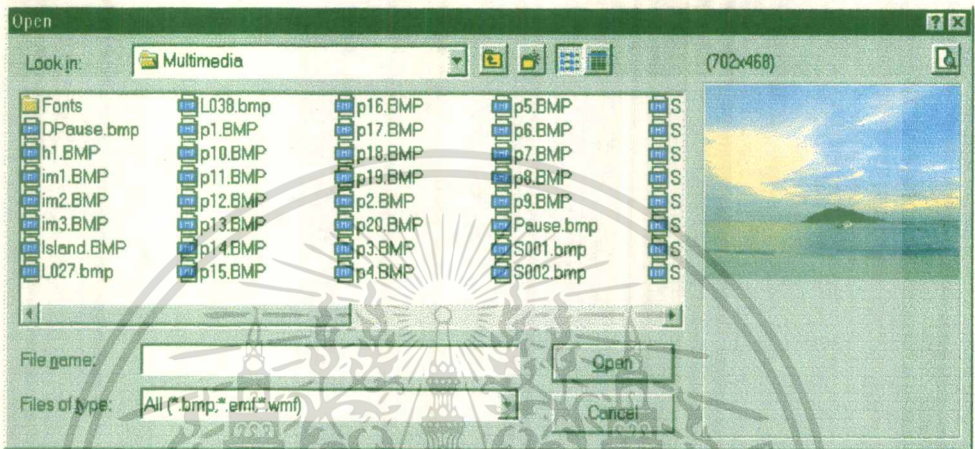
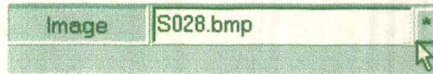


- **Stretch** ใช้ปรับขนาดรูปภาพที่เลือกเข้ามาให้พอดีกับพื้นที่ที่ใช้แสดงภาพ

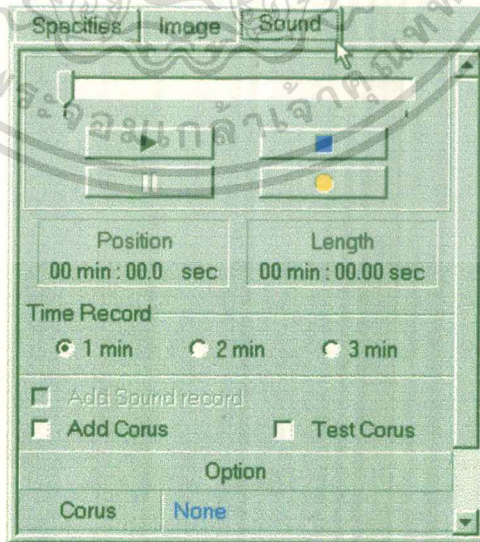


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Image Button** ใช้เปิดไดอะล็อกบ็อกเพื่อเลือกไฟล์ภาพเพื่อใช้ประกอบรายการ

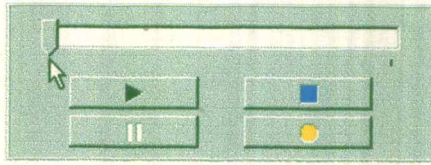


- ◆ **Tab Sound** ใช้บันทึกเสียงเพื่อใช้ประกอบการบรรยายของรายการที่สร้างขึ้น หรือเพิ่มดนตรีประกอบรายการ ซึ่งประกอบด้วยส่วนต่างๆ ดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **TrackBar** แสดงตำแหน่งของเวลาของการ เล่นเพลง หรือ การบันทึกเสียง



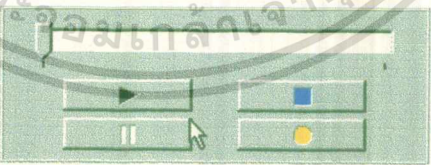
- **Play** ใช้เล่นเพลงที่ใช้เพิ่มประกอบรายการ หรือ ใช้เล่นเพลงที่เกิดจากการ บันทึกเสียง



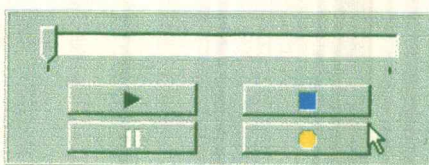
- **Stop** ใช้ยกเลิกการเล่นเพลง, การหยุดเพลง, การบันทึกเสียง



- **Pause** ใช้หยุดการเล่นเพลง หรือ การบันทึกเสียง ไว้ชั่วคราว

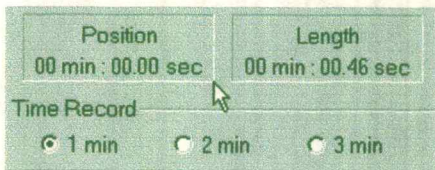


- **Record** ใช้บันทึกเสียงเพื่อบรรยายรายละเอียดของรายการ

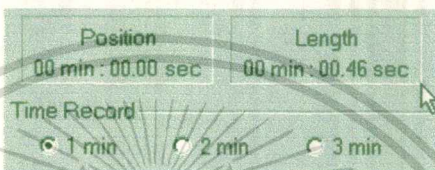


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Position** ใช้แสดงตำแหน่งของเวลาขณะเล่นเพลง หรือ บันทึกเสียง



- **Length** ใช้แสดงระยะเวลาทั้งหมดของเพลงขณะเล่น หรือ บันทึกเสียง



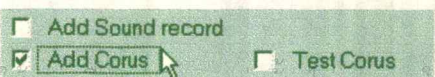
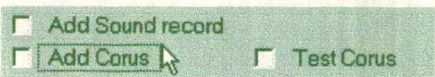
- **Time Record** ใช้กำหนดระยะเวลาที่ใช้สำหรับการบันทึกเสียง



- **Add Sound record** ใช้กำหนดการเพิ่ม หรือ เอาออก ของเสียงที่ถูกบันทึก

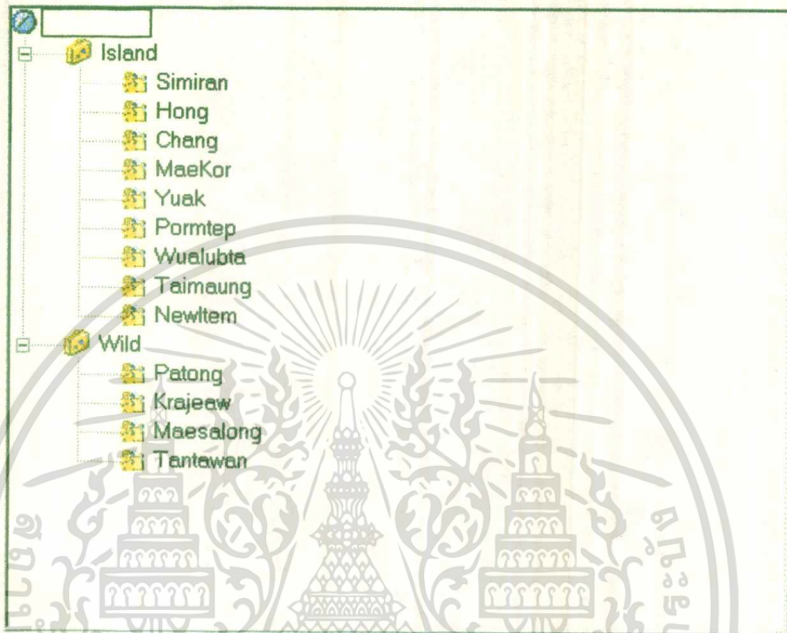


- **Add Corus** ใช้กำหนดการเพิ่ม หรือ เอาออก ของเสียงดนตรีประกอบ

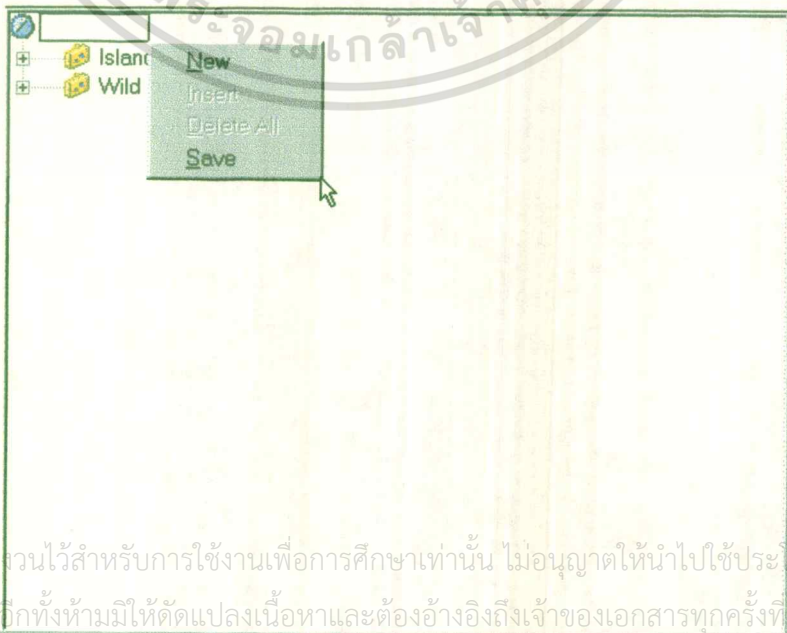


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

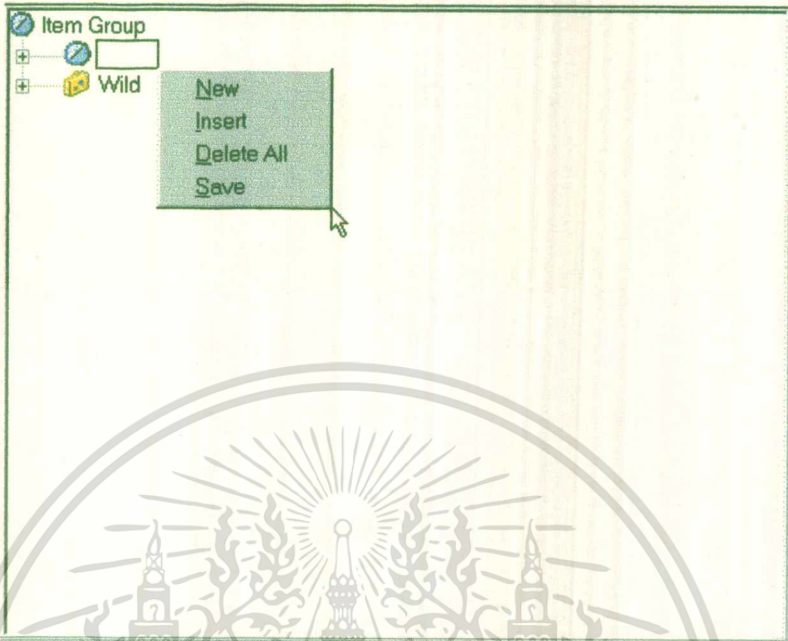
- **Tree View** ใช้แสดงชื่อรายการของข้อมูลที่ใช้ในการนำเสนอในรูปแบบของภาพและเสียง



ในส่วน **Tree View** นี้ ยังสามารถใช้ **Popup menu** ในการสร้าง, ลบ, บันทึก รายการข้อมูลได้เช่นเดียวกับปุ่ม (Button) บน **Toolbar**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

โครงการนี้ถูกสร้างขึ้น โดยมีวัตถุประสงค์ดังนี้

1. นำความรู้ทางด้านการเขียน โปรแกรมภาษาซี มาประยุกต์ใช้ในงานด้านมัลติมีเดีย
2. สามารถออกแบบ โปรแกรมทางด้านมัลติมีเดีย เพื่อนำเสนอข้อมูลที่เป็นภาพและเสียงได้

จากโครงการ ระบบนำเสนอเรื่องราวหลายสื่อ สามารถสรุปหลักการการทำงานของ โปรแกรม ได้ดังนี้

1. การสร้างรายการ

ในการสร้างรายการใหม่นั้น เราสามารถใช้คำสั่ง New หรือ Insert เพื่อสร้างรายการใหม่ โดยสามารถเรียกใช้คำสั่งได้จาก Menu , Toolbar หรือ Popuption menu รายการใหม่ที่ถูกสร้างขึ้น จะถูกกำหนดให้มีข้อมูลเริ่มต้นโดยอัตโนมัติ ซึ่งค่าเริ่มต้นของข้อมูล สามารถดูได้จากแถบ Specifies, Image, Sound และชื่อของรายการใหม่จะถูกแสดงที่ Tree View โดยถ้าเป็นคำสั่ง New ชื่อของรายการใหม่ จะถูกเพิ่มต่อท้ายรายการเก่าที่เป็น Child ของรายการที่ถูกเลือก แต่ถ้าเป็นคำสั่ง Insert ชื่อของรายการใหม่จะถูกแทรกลงในตำแหน่งรายการที่ถูกเลือก ในการสร้างรายการใหม่ถ้าเป็นรายการหลัก ชื่อของรายการจะถูกส่งไปแสดงที่แถบ View ของ Tab Display ด้วย

2. การแก้ไขข้อมูลของรายการ

การแก้ไขข้อมูลของรายการสามารถทำได้ โดยการเลือกรายการที่ Tree View ข้อมูลของรายการที่ถูกเลือกจะถูกโหลดมาแสดงที่ Tap Properties และถ้าต้องการแก้ไขเกี่ยวกับรายละเอียดของชื่อรายการให้เลือกไปที่แถบ Specifies ถ้าเกี่ยวกับภาพก็เลือกไปที่แถบ Image ถ้าเกี่ยวกับเสียงก็เลือกไปที่แถบ Sound ของ Tab Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การลบรายการ

การลบรายการสามารถกระทำได้โดยใช้คำสั่ง Delete หรือ Delete All จาก Menu, Toolbar หรือ Popupmenu โดยเลือกรายการที่ต้องการจะลบได้จาก Tree View โดยถ้าแต่ละรายการไม่มีรายการย่อยการลบจะกระทำได้โดยคำสั่ง Delete และถ้าแต่ละรายการที่ต้องการลบมีรายการย่อยก็จะเป็นคำสั่ง Delete All

4. การบันทึกข้อมูลของรายการ

การบันทึกข้อมูลของแต่ละรายการสามารถกระทำได้โดยใช้คำสั่ง Save จาก Menu, Toolbar หรือ Popupmenu ข้อมูลของแต่ละรายการจะถูกบันทึกเก็บในรูปแบบของ Text file โดยปกติแล้วข้อมูลของแต่ละรายการจะถูกบันทึกโดยอัตโนมัติ เมื่อมีการคลิกเพื่อเปลี่ยนรายการใหม่ใน Tree View ดังนั้นเมื่อมีการเปลี่ยนแปลงรายละเอียดของข้อมูลที่ Tab Properties แล้วต้องการแสดงผลของข้อมูลที่แถบ View ก็ควรบันทึกข้อมูลก่อนด้วยคำสั่ง Save.

5. การนำเสนอข้อมูล

การนำข้อมูลออกมาแสดงสามารถกระทำได้โดยการเลือกไปที่แถบ View เมื่อมีการดับเบิลคลิกไปที่ Icon ข้อมูลของรายการที่มีดัชนีตรงกับดัชนีของ Icon ก็จะถูกโหลดขึ้นมาเพื่อประมวลผลว่าจะต้องนำข้อมูลใดบ้างเช่น ข้อมูลของชื่อรายการ, ข้อมูลภาพ และ ข้อมูลเสียง แสดงออกทางจอคอมพิวเตอร์ และในส่วนของการนำเสนอข้อมูลนี้ สามารถที่จะเลือกแสดงผลข้อมูลย่อยถัดไปหรือก่อนหน้า หรือให้มีการแสดงผลแบบอัตโนมัติก็ได้.

ข้อเสนอแนะและการปรับปรุง

1. ในส่วนของชื่อรายการจะถูกจำกัดขนาดไว้ที่ 10 ตัวอักษร เนื่องจากอักษรที่เป็นภาพเคลื่อนไหวมีขนาดความกว้าง 80 พิกเซล ดังนั้นเมื่อใช้โหมดการแสดงผลขนาด 800 x 600 พิกเซล จะใช้จำนวนอักษรได้เพียง 10 ตัวอักษร

2. โหมดที่ใช้ในการแสดงผลของจอคอมพิวเตอร์ ต้องใช้โหมดการแสดงผลขนาด 800 x 600 พิกเซล เพราะจะมีการกำหนดขนาดของวินโดว์ไว้คงที่ เพื่อประโยชน์จัดรูปแบบของโปรแกรม ดังนั้นถ้ากำหนดโหมดการแสดงผลอื่นๆ จะทำให้รูปแบบของวินโดว์ผิดเพี้ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในกรณีที่ชื่อของรายการมีการใช้อักษรซ้ำกัน จะต้องกำหนดรายละเอียดของอักษรของชื่อรายการในแถบ Specifics ให้แตกต่างกัน มิฉะนั้นจะไม่สามารถโหลดไฟล์ของอักษรตัวนั้นได้

4. ในการโหลดไฟล์อักษรของชื่อรายการ จะมีความล่าช้ามาก เนื่องจากจะต้องโหลดไฟล์อักษรเท่ากับจำนวนของอักษรของชื่อรายการ ทำให้ Timer ทำงานผิดพลาดไปด้วย ทำให้จังหวะการหมุนของตัวอักษรแต่ละตัวทำงานไม่สอดคล้องกัน

5. ในโปรแกรมระบบนำเสนอเรื่องราวหลายสื่อ ยังขาดในส่วนของการแสดงข้อความ เพื่อใช้ประกอบการอธิบายข้อมูล และ ส่วนของการจัดการ การวางตำแหน่งของข้อมูล เช่น ตำแหน่งของภาพ, ตำแหน่งของชื่อรายการ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ ทรงชัย วีระทวิมาศ ที่ให้คำปรึกษาแนวทางในการแก้ไขปัญหา ตลอดจนหนังสือเพื่อให้ใช้ในการทำโครงการนี้ และขอบคุณเพื่อน ๆ ทุกคนที่ให้การสนับสนุนช่วยเหลือในการทำโครงการจนสำเร็จลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. พ. อ. เจนวิทย์ เหลืองอร่าม, การเขียนโปรแกรมแบบ OOP ด้วย Borland C++
2. พ. อ. เจนวิทย์ เหลืองอร่าม, การเขียนโปรแกรมสำหรับ Windows ด้วย Borland C++
3. Kent Reisdorph และ Ken Henderson, Teach Yourself Borland C++ Builder in 21 day



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้