

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

คณะกรรมการอุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใบรับรองปริญญาโท

ปริญญาโท โปรแกรมช่วยสอนวิชาดิจิทัล

CAI FOR DIGITAL SUBJECT

ชื่อนักศึกษา 1.นางสาวจันทิมา บุทเสน รหัสประจำตัว 39031402
2.นางสาวจิราพร สะอาดยิ่ง รหัสประจำตัว 39031405
3.นางสาววันวิสา วัฒนาศิลป์ รหัสประจำตัว 39031427
4.นายสุรพันธ์ วงษ์ไพศาล รหัสประจำตัว 39031438

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญาโท

1.ดร.สุรสิทธิ์ ราตรี
2.อาจารย์กิติพงศ์ มะโน
3.อาจารย์สุชิน อางหาญ



คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1.อาจารย์กิติพงศ์ มะโน	
2.อาจารย์สุชิน อางหาญ	
3.อาจารย์พีระวุฒิ สุวรรณจันทร์	
4.อาจารย์อำพล ทองระอา	
5.อาจารย์ไพบุลย์ พวงวงศ์ตระกูล	

วันเดือนปีที่สอบ วันที่ 11 ธันวาคม 2540 เวลา 12.30 น.ถึง 14.30น.

สถานที่สอบ ห้อง 310 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

ดร.ธีระพล เทพหัสดิน ณ อยุธยา

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 13 เดือน ๑๓ พ.ศ. ๕๐/



คณะครุศาสตร์อุตสาหกรรม สจล.

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง

เลขที่.....
เลขทะเบียน..... 30153
วัน, เดือน, ปี..... ๑๓ ธ.ย. ๒๕๔๐

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

โปรแกรมช่วยสอนวิชาดิจิทัล
CAI FOR DIGITAL SUBJECT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาครุศาสตร์วิศวกรรม

ปริญญานิพนธ์

เรื่อง โปรแกรมช่วยสอนวิชาดิจิทัล

CAI FOR DIGITAL SUBJECT

ผู้จัดทำ

- 1.นางสาวจันทิมา บุทเสน
- 2.นางสาวจิราพร สะอาดยิ่ง
- 3.นางสาววันวิสา วัฒนาศิลป์
- 4.นายสุรพันธ์ วงษ์ไพศาล

อาจารย์ที่ปรึกษา

ลงนาม.....

(ดร.สุรสิทธิ์ รัตวี)

ลงนาม.....

(อาจารย์กิติพงศ์ มะโน)

ลงนาม.....

(อาจารย์สุชิน อางหาญ)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสติน ณ อยุธยา)

ปริญญานิพนธ์

เรื่อง โปรแกรมช่วยสอนวิชาดิจิทัล

CAI FOR DIGITAL SUBJECT

จุดประสงค์

1. เพื่อศึกษาการเขียนโปรแกรมสำหรับสร้างสื่อประสม HURESS
2. เพื่อออกแบบโปรแกรมช่วยสอนวิชาดิจิทัล
3. เพื่อสร้างโปรแกรมช่วยสอนวิชาดิจิทัล โดยใช้โปรแกรมสร้างสื่อประสม HURESS ของโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นเป็นเครื่องมือในการสร้างและพัฒนา
4. เพื่อให้โปรแกรมช่วยสอนวิชาดิจิทัลสามารถนำไปใช้สอนจริงได้
5. เป็นแนวทางในการพัฒนาโปรแกรมช่วยสอนวิชาอื่นๆ ที่จะใช้โปรแกรมสร้างสื่อประสม HURESS และเป็นเครื่องมือในการพัฒนาสำหรับผู้สนใจอื่นๆ ต่อไป

ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้มีความสามารถในการเขียนโปรแกรมสร้างสื่อประสม HURESS
2. ได้โปรแกรมช่วยสอนวิชาดิจิทัล
3. ทำให้การเรียนการสอนทางไกลสะดวกยิ่งขึ้น
4. ใช้เป็นต้นแบบของการศึกษาโปรแกรมสร้างสื่อประสม HURESS
5. ได้แนวทางสำหรับการพัฒนาวิชาอื่น ๆ เพื่อร่วมโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นต่อไปในอนาคต

โปรแกรมช่วยสอนวิชาดิจิทัล

นางสาวจันทิมา	บุทแสน
นางสาวจิราพร	สะอาดยิ่ง
นางสาววันวิสา	วัฒนาศิลป์
นายสุรพันธ์	วงษ์ไพศาล

อาจารย์ที่ปรึกษา

ดร.สุรสิทธิ์	ราตรี
อาจารย์กิติพงศ์	มะโน
อาจารย์สุชิน	อาจหาญ

ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอการสร้างบทเรียน โปรแกรมช่วยสอนวิชา ดิจิทัล ซึ่งมีเนื้อหาคำบรรยายเป็นภาษาอังกฤษ ทั้งนี้เพื่อพัฒนาเนื้อหาวิชาให้สอดคล้องกับระบบของโครงการวิจัยร่วมระหว่าง ไทย ญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบสองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ โดยใช้เครื่องมือการพัฒนา CAI (AUTHORING TOOLS) ชื่อโปรแกรม HURESS ซึ่งบทเรียน โปรแกรมคอมพิวเตอร์ช่วยสอนนี้สามารถนำไปใช้เป็นบทเรียนจริงหรือนำไปใช้เป็นบทเรียนเสริมวิชาดิจิทัล ซึ่งผู้เรียนสามารถเรียนรู้ได้ตามความซ้ำเร็วของตนเอง ทำให้การเรียนมีประสิทธิภาพมากขึ้นเนื่องจากข้อได้เปรียบในเรื่องของสี ภาพ การให้ข้อมูลป้อนกลับ การเปิดโอกาสให้ผู้เรียนได้มีส่วนร่วมในการเรียนการสอนและยังลดปัญหาความแตกต่างระหว่างบุคคล

การสร้างบทเรียนคอมพิวเตอร์ช่วยสอนนี้ สร้างขึ้นโดยโปรแกรม HURESS และโปรแกรมช่วยสร้างภาพอื่นๆ ทำให้สามารถช่วยออกแบบบทเรียน โปรแกรมคอมพิวเตอร์ช่วยสอนได้อย่างดี

CAI FOR DIGITAL SUBJECT

MISS JUNTIMA	BUTTASEN
MISS JEERAPORN	SAEATYING
MISS WANVISA	WATTANASIN
MR.SURAPHAN	WONGPHAISAN

ADVISORS

DR.SURASIT	RATREE
MR.KITIPONG	MANO
MR.SUCHIN	ADHAN

1997

ABSTRACT

The thesis presents the computer Assisted Instruction program used in the course 'Digital' instructed in English so as to develop the course content with the serve of Collaborative Research Project on Multimedia Human Resource Development Network Technology among Thailand, Japan and China. To develop human resource in tele-education on media via satellite with CAI (Authoring Tool) by HURESS, This program could be applied more effectively which leanness could be used individually. This bring about the more efficiency of learning by the advantages of colors, graphic animation, information retrieval, helping learners participate in teaching and learning and also lessening individual differences.

This computer Assisted Instruction program was produced by HURESS program and picture producer other programs. It can be design well of computer Assisted Instruction. picture - produced programs. It can help make computer Instruction program.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ผู้จัดทำขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ซึ่งให้โอกาสในการศึกษาและให้กำลังสนับสนุนทุนการศึกษาครั้งนี้ รวมทั้งความช่วยเหลือการให้คำปรึกษา ตลอดจนแนวความคิดต่างๆ พร้อมทั้งข้อเสนอแนะแนวทางแก้ไขปัญหาในการดำเนินงาน จากท่านอาจารย์ที่ปรึกษาปริญญาบัตร ดร.สุรสิทธิ์ ราตรี และอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน และขอขอบคุณ คุณสุวิทย์ ยิบมันตะศิริ ที่สละเวลาช่วยเหลือให้คำปรึกษาและอำนวยความสะดวกในการใช้ห้องสอนทางไกลสองทาง เครื่องมืออุปกรณ์ต่างๆ จึงทำให้โครงการนี้สำเร็จลุล่วงได้ด้วยดี

ผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญาบัตรฉบับนี้ จะให้ประโยชน์และคุณค่ากับผู้เรียนหรือผู้วิจัย และเป็นแนวทางในการพัฒนาสื่อการสอนในโครงการฯ ต่อไป



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญาโท	1
1.2 ขอบเขตของปริญญาโท	2
1.3 เนื้อหาโดยสังเขป	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 กล่าวนำ	4
2.2 ประวัติความเป็นมาของคอมพิวเตอร์ช่วยสอน	4
2.2.1 ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน	5
2.2.2 ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน	6
2.2.3 ข้อได้เปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน	7
2.2.4 ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน	7
2.2.5 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน	8
2.2.6 ลักษณะ โครงสร้างของบทเรียนคอมพิวเตอร์ช่วยสอน	8
2.2.7 โครงการวิจัยร่วมญี่ปุ่นและจีนในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทางพร้อม สื่อประสมดาวเทียมระหว่างประเทศ	8
2.3 หลักการของ โปรแกรม HURESS	11
2.4 ส่วนประกอบของ โปรแกรม HURESS	12
2.4.1 Books	12
2.4.2 Page	12

สารบัญ (ต่อ)

เรื่อง

หน้า

2.4.3 Media	13
2.4.4 Desk	13
2.4.5 Bookshelf	13
2.4.6 Entity	13
2.4.7 Procedures	13
2.4.8 Object	13
2.5 หลักการใช้โปรแกรม HURESS	14
2.5.1 การยืมและเปิดหนังสือ(Borrow and Operating a Book)	14
2.5.2 การเปิดหนังสือ Opening A Book	15
2.5.3 การทำงานเกี่ยวกับหน้าหนังสือ	16
2.5.4 การใช้งาน Desk Browser	17
2.5.5 การใช้หน้าสารบัญ	18
2.5.6 การใช้หน้าดัชนี	18
2.5.7 การคืนหนังสือ	19
2.6 การทำงานในส่วนของ Desk Main Menu	20
2.6.1 การทำงานในส่วนของเมนูย่อย [Desk]	21
2.6.2 การทำงานในส่วนของเมนูย่อย [Bookshelf]	22
2.6.3 การทำงานในส่วนของเมนูย่อย [Book]	22
2.6.4 การทำงานในส่วนของเมนูย่อย [Page]	23
2.6.5 การทำงานในส่วนของเมนูย่อย [Edit]	24
2.6.6 Media information Dialog Box	34
บทที่ 3 การออกแบบ การทำงาน และการสร้าง	40
3.1 เครื่องมือที่เกี่ยวข้องในการออกแบบและการสร้าง	40

VI

สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.1 ทางด้านฮาร์ดแวร์	40
3.1.2 ทางด้านซอฟต์แวร์	40
3.2 ขั้นตอนในการออกแบบโปรแกรม	41
3.3 ขั้นตอนในการสร้างบทเรียน	44
3.4 ขั้นตอนในการสร้างแบบฝึกหัด	44
3.5 วิธีในการสร้างส่วนของบทเรียนและแบบฝึกหัดด้วยโปรแกรม HURESS	45
3.5.1 การสร้างหนังสือ	45
3.5.2 การสร้าง Background (basic page)	48
3.5.3 การสร้าง Front Cover	48
3.5.4 การสร้างบทเรียน	49
3.5.5 การสร้าง Unit menu	51
3.5.6 การสร้าง Main menu	52
3.5.7 การสร้าง Index	55
3.5.8 การสร้างลำดับการทำงานของหนังสือ	58
3.5.9 การเขียนโปรแกรมเพื่อ SETUP การทำงานของหนังสือ	62
3.6 วิธีการสร้างส่วนของแบบฝึกหัดด้วยโปรแกรม HURESS	63
3.6.1 การสร้างแบบฝึกหัด	63
3.6.2 โปรแกรมในส่วนของ การสร้างแบบฝึกหัด	67
บทที่ 4 การทดลองและผลการทดลอง	70
4.1 การเข้าสู่โปรแกรม HURESS	70
4.2 การใช้งานของ โปรแกรม	71
4.2.1 ส่วนของเนื้อหา	71
4.2.2 ส่วนของแบบทดสอบ	76
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	81
5.1 ปัญหาที่พบในการทำโครงการ	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VII

สารบัญ (ต่อ)

เรื่อง	หน้า
5.3 การแก้ปัญหา	82
5.4 ข้อเสนอแนะ	83
5.5 ประโยชน์ที่ได้รับจากโครงการ	83
5.6 แนวทางการพัฒนา	83
ภาคผนวก ก รูปแบบการใช้งานของฟังก์ชันต่างๆ	84
ภาคผนวก ข การติดตั้งโปรแกรม HURESS	154
บรรณานุกรม	163
ประวัติผู้แต่ง	164



VIII

สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 2.1 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน	5
รูปที่ 2.2 ระบบสอนทางไกลแบบสองทางพร้อมสื่อประสมดาวเทียมระหว่างประเทศ	9
รูปที่ 2.3 เครือข่ายของระบบ	9
รูปที่ 2.4 การเขียนโปรแกรมแบบธรรมดาและการเขียนโปรแกรมแบบ Event-driven	12
รูปที่ 2.5 การข้มหนังสือผ่าน Desk Main Menu	14
รูปที่ 2.6 หนังสือที่ถูข้ม	14
รูปที่ 2.7 การข้มหนังสือผ่าน Bookshelf Browser	15
รูปที่ 2.8 การเปิดหนังสือ	16
รูปที่ 2.9 หนังสือที่ทำการเปิดได้แล้ว	16
รูปที่ 2.10 การทำงานเกี่ยวกับหน้าหนังสือ	17
รูปที่ 2.11 การใช้ Desk Browser	17
รูปที่ 2.12 การใช้หน้าสารบัญ	18
รูปที่ 2.13 การใช้หน้าดัชนี	19
รูปที่ 2.14 การคืนหนังสือ โดยปุ่ม Close Box	19
รูปที่ 2.15 การทำงานในส่วนของ Desk Main Menu	20
รูปที่ 2.16 การเข้าโปรแกรม HURESS	25
รูปที่ 2.17 การเปิดหนังสือ	26
รูปที่ 2.18 การข้มหนังสือ	26
รูปที่ 2.19 การกำหนดตำแหน่ง Selection Bar	26
รูปที่ 2.20 การ Set การแสดงของ Item ใน Section	27
รูปที่ 2.21 การกำหนดแบ็คกราวด์	27
รูปที่ 2.22 การเพิ่มชั้นวางหนังสือ	28
รูปที่ 2.23 การเพิ่มชั้นวางหนังสือ	28
รูปที่ 2.24 การสร้างหนังสือใหม่	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IX

สารบัญญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 2.25 การกำหนดขนาดของหนังสือ	29
รูปที่ 2.26 การกำหนดรายละเอียดในส่วนต่างๆของหนังสือ	30
รูปที่ 2.27 การยืมหนังสือ	30
รูปที่ 2.28 การลบหนังสือ	31
รูปที่ 2.29 การย้ายหนังสือ	31
รูปที่ 2.30 การคัดลอกหนังสือ	32
รูปที่ 2.31 การกำหนดรายละเอียดหน้าของเนื้อหา	32
รูปที่ 2.32 การกำหนดหน้าสารบัญ	33
รูปที่ 2.33 การกำหนดขนาดของหน้า	33
รูปที่ 2.34 การกำหนดรายละเอียดของหน้า	34
รูปที่ 2.35 การกำหนดดัชนี	34
รูปที่ 2.36 การทำรูปเคลื่อนไหว	35
รูปที่ 2.37 การเลือกคำตอบในแบบฝึกหัด	35
รูปที่ 2.38 การกำหนดรายละเอียดของหน้าดัชนี	36
รูปที่ 2.39 การเรียกภาพมาประกอบโปรแกรม	36
รูปที่ 2.40 การตั้งเวลา	37
รูปที่ 2.41 การสร้างข้อความ	37
รูปที่ 2.42 การสร้างปุ่ม	38
รูปที่ 2.43 การดึงรูปมาประกอบโปรแกรม	38
รูปที่ 2.44 การสร้างปุ่มทำงานต่างๆ	39
รูปที่ 3.1 ผังการทำงานของโปรแกรมช่วยสอนวิชาจิตตอล	41
รูปที่ 3.2 ผังขั้นตอนการออกแบบโปรแกรม	42
รูปที่ 3.2 (ต่อ) ผังขั้นตอนการออกแบบโปรแกรม	43
รูปที่ 3.3 แผนผังการสร้างบทเรียน	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.4 แผนผังการสร้างแบบฝึกหัด	45
รูปที่ 3.5 เมนูการสร้างหนังสือ	45
รูปที่ 3.6 TAO Desk Browser	46
รูปที่ 3.7 การสร้างหน้า Background	46
รูปที่ 3.8 การกำหนดชื่อหน้า Background	47
รูปที่ 3.9 การกำหนดรายละเอียดต่างๆ ของ Pict Media	47
รูปที่ 3.10 หน้า Background ของหนังสือ	48
รูปที่ 3.11 การกำหนดรายละเอียดของ Front Cover	48
รูปที่ 3.12 การเลือก Background ของหน้า Front Cover	49
รูปที่ 3.13 หน้า Front Cover ของหนังสือ	49
รูปที่ 3.14 การสร้างหน้าของหนังสือ	50
รูปที่ 3.15 การกำหนดรายละเอียดของหน้าบทเรียน	50
รูปที่ 3.16 หน้าของบทเรียน	51
รูปที่ 3.17 หน้าของ Unit Menu	51
รูปที่ 3.18 Dialog Box ของ Main Menu	52
รูปที่ 3.19 การเลือก Background และกำหนดรายละเอียดต่างๆของ Main Menu	52
รูปที่ 3.20 หน้าของ Main Menu	53
รูปที่ 3.21 Dialog Box ของ Main Menu	53
รูปที่ 3.22 Dialog Box ของการทำ Main Menu	54
รูปที่ 3.23 Dialog Box ของ Content Media	54
รูปที่ 3.24 หน้าของ Main Menu	55
รูปที่ 3.25 Dialog Box ของ Index	55
รูปที่ 3.26 การเลือก Background และกำหนดรายละเอียดต่างๆ ของ Main Menu	56
รูปที่ 3.27 หน้าของ Index	56
รูปที่ 3.28 Dialog Box ของ Main Menu	57

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.29 Dialog Box ของการทำ Index	57
รูปที่ 3.30 Dialog Box ของIndex Media	57
รูปที่ 3.31 หน้าของ Index	58
รูปที่ 3.32 โปรแกรมย่อยของมีเดีย How to read	58
รูปที่ 3.33 โปรแกรมย่อยของมีเดีย Content หรือ Main Menu	59
รูปที่ 3.34 โปรแกรมย่อยของมีเดีย [Close]	59
รูปที่ 3.35 โปรแกรมย่อยของมีเดีย [Backward]	59
รูปที่ 3.36 โปรแกรมย่อยของมีเดีย [Index]	60
รูปที่ 3.37 โปรแกรมย่อยของมีเดีย [Previous]	60
รูปที่ 3.38 โปรแกรมย่อยของมีเดีย [Forward]	60
รูปที่ 3.39 โปรแกรมย่อยของมีเดีย [Unit Menu]	61
รูปที่ 3.40 โปรแกรมย่อยของมีเดีย [Example]	61
รูปที่ 3.41 โปรแกรมย่อยของมีเดีย Voice Media]	61
รูปที่ 3.42 โปรแกรมย่อยของมีเดีย [SETUPBOOK]	62
รูปที่ 3.43 โปรแกรมย่อยของการเชื่อมต่อ Background	62
รูปที่ 3.44 การใช้ Combo Media สร้างแบบฝึกหัด	63
รูปที่ 3.45 การใช้ Radio Media สร้างแบบฝึกหัด	63
รูปที่ 3.46 การเลือกรูปแบบแสดงผล	64
รูปที่ 3.46 การเลือก Combo Media ผ่าน Evaluation Revised Media	64
รูปที่ 3.48 การ Add ตัวเลือกไปยัง Combo Media หรือ Radio Media	65
รูปที่ 3.49 การใส่ค่า Point ให้กับตัวเลือก	65
รูปที่ 3.50 การ Add Evaluation Revised Media ใน Response Controlled Media	66
รูปที่ 3.51 การใส่ค่า Point ให้กับ Evaluation Revised Media	66
รูปที่ 3.52 หน้าเฉลย	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.53 หน้าคำอธิบายที่ถูกต้อง	67
รูปที่ 3.54 โปรแกรมเช็คคำตอบ และเลือกแสดงคำตอบ	68
รูปที่ 3.55 โปรแกรมเช็คข้อผิดพลาดข้อสอบที่ถูกต้อง	68
รูปที่ 3.56 โปรแกรมเช็คข้อหนังสือ	69
รูปที่ 3.57 โปรแกรมเรียกใช้หนังสือเฉลย	69
รูปที่ 4.1 การเข้าสู่โปรแกรม HURESS	70
รูปที่ 4.2 การป้อน User Name	71
รูปที่ 4.3 การเลือกวิชาเรียน	71
รูปที่ 4.4 Font Cover	72
รูปที่ 4.5 How to read	72
รูปที่ 4.6 Unit Menu	73
รูปที่ 4.7 เนื้อหาของบทเรียนและการไปหน้าตัวอย่าง	73
รูปที่ 4.8 การใช้งานหน้าสารบัญ	74
รูปที่ 4.9 การใช้งานหน้าดัชนี	75
รูปที่ 4.10 การออกจากโปรแกรม	75
รูปที่ 4.11 หน้าจอหลักของส่วนแบบทดสอบ	76
รูปที่ 4.12 หน้าจอหลักของหัวข้อแบบทดสอบ	76
รูปที่ 4.13 หน้าจอแบบทดสอบหลักของบทเรียน	77
รูปที่ 4.14 คำถามหัวข้อที่ต้องการทดสอบ	77
รูปที่ 4.15 คำตอบที่ถูกเลือก	78
รูปที่ 4.16 คำตอบที่ถูกต้อง	78
รูปที่ 4.17 คำตอบที่ผิด	79
รูปที่ 4.18 แสดงคำอธิบายคำตอบเพิ่มเติม	79
รูปที่ 4.19 หน้า Question Menu หลังจากทำแบบทดสอบ	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ ข.1 การ Download Software	155
รูปที่ ข.2 ตำแหน่งที่ต้องการเก็บไฟล์	155
รูปที่ ข.3 การบันทึกโปรแกรม	156
รูปที่ ข.4 ตำแหน่งที่จะเก็บไฟล์	156
รูปที่ ข.5 การ Extract File	157
รูปที่ ข.6 ทำการ Extract เสร็จสมบูรณ์	157
รูปที่ ข.7 การ Extract File	158
รูปที่ ข.8 การ Extract เสร็จสมบูรณ์	158
รูปที่ ข.9 ไฟล์ใน Sub Directory temp	158
รูปที่ ข.10 การติดตั้ง ORACLE	159
รูปที่ ข.11 การติดตั้ง ORACLE เสร็จสมบูรณ์	160
รูปที่ ข.12 การติดตั้ง CAI Setup เสร็จสมบูรณ์	160
รูปที่ ข.13 Short Cut ของ HERESS	161
รูปที่ ข.14 Configuration Network	161
รูปที่ ข.15 Domain Sever	162
รูปที่ ข.16 Protocol TCP/IP	162
รูปที่ ข.17 Properties ของ Protocol TCP/IP	163

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญาโท

จากโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบสอนทางไกลแบบ สองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ ซึ่งมีหลักการและเหตุผลดังต่อไปนี้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ร่วมมือกับมหาวิทยาลัยฮอกไกโด ประเทศญี่ปุ่น ทำการศึกษาค้นคว้าวิจัยด้านการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทาง พร้อมสื่อประสม โดยติดต่อสื่อสารผ่านดาวเทียมระหว่างประเทศ JC SAT - 3 ของประเทศญี่ปุ่น โครงการนี้ได้รับพระมหากรุณาธิคุณจากสมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี เสด็จเปิดโครงการเมื่อวันที่ 4 กรกฎาคม 2539 ในงานนิทรรศการพระจอมเกล้าลาดกระบังนิทรรศน์เฉลิมพระเกียรติเทคโนโลยีเพื่อปวงชน ที่ผ่านมา โครงการนี้มีวัตถุประสงค์เพื่อ

1. พัฒนาทรัพยากรมนุษย์ด้านโทรคมนาคมในภูมิภาคเอเชียแปซิฟิก
2. เพื่อพัฒนาระบบเครือข่ายการสอนทางไกลด้วยสื่อระบบสองทาง ผ่านดาวเทียมที่มีประสิทธิภาพมากกว่าวิธีปกติ ที่ใช้สอนระหว่างมนุษย์ต่อมนุษย์โดยตรง
3. เพื่อประเมินผลของระบบดังกล่าวที่ถูกนำมาใช้ระหว่างประเทศญี่ปุ่นและประเทศในภูมิภาคเอเชียแปซิฟิก

ระยะเวลาของการดำเนินโครงการวิจัย 2 ปี โครงการได้เริ่มดำเนินการวิจัย โดยได้ทำการสอนครั้งแรกพร้อมกัน 3 ประเทศ คือระหว่างมหาวิทยาลัยฮอกไกโด ประเทศญี่ปุ่น มหาวิทยาลัยนานกิงประเทศจีน และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประเทศไทย ตั้งแต่วันที่ 6 พฤศจิกายน 2539 เป็นต้นมาโดยทำการสอนประมาณ สัปดาห์ละ 3 วัน วันละ 1 ชั่วโมงประกอบด้วยจำนวน 3 วิชาคือ

1. วิชา Telecommunication Network ตั้งแต่วันที่ 6-27 พฤศจิกายน 2539 จำนวน

10 บท

2. วิชา Online System Design ตั้งแต่วันที่ 3-25 ธันวาคม 2539 จำนวน 13 บท

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วิชา Network Theory ตั้งแต่วันที่ 9-30 มิถุนายน 2540 จำนวน 10 บท

นอกจากการศึกษาวิจัยถึงประสิทธิภาพการเรียนการสอนด้วยระบบวิธีการดังกล่าวแล้วโครงการวิจัยนี้ ยังมุ่งพัฒนาสื่อการเรียนการสอน ในลักษณะสื่อประสม ของเนื้อหาวิชาอื่นๆที่เกี่ยวข้องกับวิทยาศาสตร์และเทคโนโลยีเมื่อพัฒนาได้คุณภาพตามต้องการแล้ว จะสามารถนำไปใช้ในการเรียนการสอนร่วมกับมหาวิทยาลัยของประเทศอื่นๆในเอเชียแปซิฟิกอีกด้วย

โครงการนี้ได้รับความสนใจจากนักศึกษาและประชาชนทั่วไป และจะเป็นประโยชน์อย่างยิ่งต่อการพัฒนาการศึกษา และช่วยแก้ปัญหาการขาดแคลนอาจารย์ สร้างความเสมอภาคทางการศึกษา สามารถติดต่อกับมหาวิทยาลัยในต่างประเทศทั่วโลกได้ อันจะนำไปสู่การพัฒนาทางวิชาการต่อไป

ดังนั้นเพื่อให้สอดคล้องต่อโครงการดังกล่าวกลุ่มผู้เสนอหัวข้อปริญญานิพนธ์จึงได้มีแนวความคิดที่จัดสร้างวิชาใหม่ขึ้นมาอีกหนึ่งวิชาจากเดิมที่มีอยู่เพียง 3 วิชาโดยได้เลือกเนื้อหาวิชาที่ผู้ศึกษาคอมพิวเตอร์โดยทั่วไปจะต้องมีความรู้เหล่านี้เป็นพื้นฐานดังนั้นจึงได้เลือกวิชาที่เกี่ยวกับดิจิทัล ในการจัดทำโปรแกรมช่วยสอน

1.2 ขอบเขตของปริญญานิพนธ์

โปรแกรมช่วยสอนวิชาดิจิทัลจะแบ่งเนื้อหาออกเป็น 10 บท และแบ่งออกเป็นสองเล่มเป็นส่วนหนึ่งของเนื้อหาและส่วนของแบบฝึกหัด ในส่วนของแบบฝึกหัดจะมีบททดสอบเพื่อทดสอบความเข้าใจ โดยที่ผลการทดสอบของนักเรียนแต่ละคนจะปรากฏบนจอคอมพิวเตอร์ของผู้สอนด้วย ซึ่งผู้สอนจะเป็นคนเช็คได้ว่า นักเรียนคนไหนตอบผิดหรือตอบถูก

ในส่วนของบทเรียน โปรแกรมนี้จะมีเนื้อหาที่เป็นใจความสำคัญของหัวข้อนั้นๆ มีรูปภาพประกอบคำอธิบาย และสามารถเลือกฟังเสียงคำอธิบายได้ในแต่ละหัวข้อ ในแต่ละบทเรียนจะมีคู่มือที่เป็นเนื้อหาอธิบายหัวข้อต่างๆ ให้กับอาจารย์ผู้สอนใช้ประกอบการบรรยาย

โครงการนี้จะใช้โปรแกรม HURESS ของโครงการวิจัยร่วมระหว่างไทย ญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบ 2 ทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ ในการสร้างโปรแกรมช่วยสอนวิชาดิจิทัล

1.3 เนื้อหาโดยสังเขป

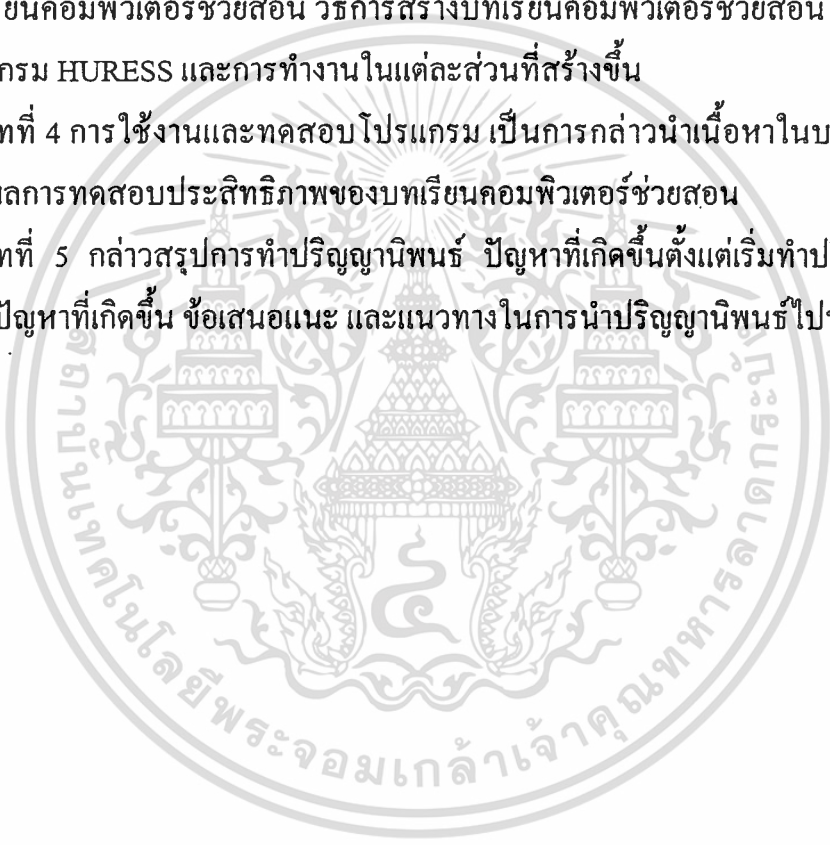
เนื้อหาทั้งหมดของปฏิญญานิพนธ์ฉบับนี้ ประกอบด้วย

บทที่ 2 กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงการ ประกอบด้วยประวัติความเป็นมา ความหมาย ประเภท การสร้างและประโยชน์ของคอมพิวเตอร์ช่วยสอน รวมถึงหลักการและรายละเอียดการใช้งานในโปรแกรม HURESS

บทที่ 3 การออกแบบ การสร้าง การทำงาน โดยกล่าวถึงการอธิบาย ขั้นตอนการออกแบบบทเรียนคอมพิวเตอร์ช่วยสอน วิธีการสร้างบทเรียนคอมพิวเตอร์ช่วยสอนในแต่ละส่วนด้วยโปรแกรม HURESS และการทำงานในแต่ละส่วนที่สร้างขึ้น

บทที่ 4 การใช้งานและทดสอบโปรแกรม เป็นการกล่าวนำเนื้อหาในบทเรียนแต่ละส่วนและผลการทดสอบประสิทธิภาพของบทเรียนคอมพิวเตอร์ช่วยสอน

บทที่ 5 กล่าวสรุปการทำปฏิญญานิพนธ์ ปัญหาที่เกิดขึ้นตั้งแต่เริ่มทำปฏิญญานิพนธ์ วิธีการแก้ปัญหาที่เกิดขึ้น ข้อเสนอแนะ และแนวทางในการนำปฏิญญานิพนธ์ไปพัฒนาต่อไป



บทที่ 2

ทฤษฎี และหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญาบัตรในบทนี้เป็นทฤษฎีและหลักการที่นำมาใช้ประกอบการสร้างโครงการ โดยประกอบด้วย ประวัติความเป็นมาและหลักการของบทเรียนคอมพิวเตอร์ช่วยสอน และหลักการรายละเอียดการใช้งานในแต่ละส่วนของโปรแกรม HURESS ซึ่งเนื้อหาในบทนี้จะใช้เป็นพื้นฐานในการนำไปสร้างและออกแบบโปรแกรมช่วยสอนซึ่งมีเนื้อหาในบทต่อไป

2.2 ประวัติความเป็นมาของคอมพิวเตอร์ช่วยสอน

วิวัฒนาการของบทเรียนคอมพิวเตอร์ช่วยสอนมีลำดับดังนี้
ปี ค.ศ. 1950 ศูนย์วิจัยของ IBM ได้ริเริ่มนำบทเรียนคอมพิวเตอร์ช่วยสอนเพื่อใช้กับงานวิจัยด้านจิตวิทยา นับว่าเป็นการบุกเบิกด้านนี้เป็นแห่งแรก

ต่อมา ปี ค.ศ. 1958 มหาวิทยาลัย Florida ได้พัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนเพื่อใช้ทบทวนวิชาฟิสิกส์และสถิติ พร้อมๆ กับมหาวิทยาลัย Standford ได้นำคอมพิวเตอร์มาช่วยสอนวิชาภาษาอังกฤษและคณิตศาสตร์ในระดับมัธยมศึกษา

ปี ค.ศ. 1960 มหาวิทยาลัย Illinois ได้จัดทำบทเรียนคอมพิวเตอร์ช่วยสอนในสาขาวิชาจิตวิทยาการศึกษา และด้านวิศวกรรมศาสตร์ โดยตั้งชื่อว่า PLATO CAI (Programmed Learning for Automated Teaching Operations CAI)

ต่อมาเมื่อคอมพิวเตอร์แพร่หลายมากขึ้น ได้มีการส่งเสริมพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนมากขึ้น ในปี ค.ศ. 1967 มีบทเรียนคอมพิวเตอร์ช่วยสอนจำหน่ายมากกว่า 1500 เรื่อง

ในปี ค.ศ. 1970 บทเรียนคอมพิวเตอร์ช่วยสอน ได้เริ่มแพร่หลายในทวีปยุโรปองค์การ French National Experiment in Educational Computing ของฝรั่งเศส ได้เริ่มพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนสำหรับผู้สอนในระดับมัธยมศึกษาพร้อมๆ กับประเทศอังกฤษได้ริเริ่มโครงการพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนมากกว่า 35 โครงการ

ปี ค.ศ. 1971 มหาวิทยาลัย Brigham Young ร่วมกับมหาวิทยาลัย Texas ได้พัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนสำหรับกับมินิคอมพิวเตอร์โดยผสมผสานคอมพิวเตอร์กับโทรทัศน์เพื่อช่วยสอนวิชาภาษาอังกฤษและคณิตศาสตร์ ตั้งชื่อว่า TICCIT (Time Shared Interactive Computer Controlled Information Television)

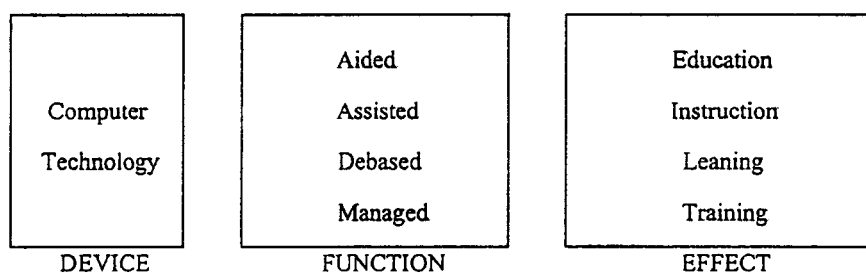
พัฒนาการของบทเรียนคอมพิวเตอร์ช่วยสอนกับเครื่องคอมพิวเตอร์ขนาดใหญ่ นับตั้งแต่ต้นมา จะประสบความสำเร็จในระดับหนึ่งเท่านั้น ต่อมาเมื่อไมโครคอมพิวเตอร์เริ่มเข้ามามีบทบาทในสถานศึกษามากขึ้น จึงมีการใช้บทเรียนคอมพิวเตอร์ช่วยสอนกันอย่างจริงจัง เช่น มหาวิทยาลัย Osaka ในประเทศญี่ปุ่น เป็นต้น

ปัจจุบันบทเรียนคอมพิวเตอร์ช่วยสอนได้มีบทบาทสำคัญต่อการเรียนการสอนทั้งในห้องเรียน และการฝึกอบรมในสถานประกอบการ โดยเฉพาะเทคโนโลยีคอมพิวเตอร์ในปัจจุบันสามารถประยุกต์ใช้งานได้ทั้งภาพ เสียง และการโต้ตอบ ในลักษณะสื่อประสม (Multi Media) ทำให้แนวโน้มของบทเรียนคอมพิวเตอร์ช่วยสอน ได้แสดงบทบาทในการประยุกต์ใช้งานได้อย่างเต็มที่

2.2.1 ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน

Computer Assisted Instruction:CAI เป็นคำศัพท์เดิมที่นิยมใช้ในสหรัฐอเมริกามีความหมายว่า การสอนโดยใช้คอมพิวเตอร์เป็นเครื่องช่วยสอน แต่ปัจจุบันมีผู้นิยมใช้ คำว่า CBT (Computer Based Teaching หรือ Computer Based Training) มากกว่า ความหมายก็คือ การสอนหรือการฝึกอบรมโดยใช้คอมพิวเตอร์เป็นหลัก

นอกจากนี้ยังมีคำอื่นๆ ที่กำหนดขึ้นมาอีก ซึ่งโดยส่วนใหญ่แล้วจะประกอบด้วย 3 ส่วนผสมกัน ได้แก่ เครื่องมือ (Device) หน้าที่ (function) และผลที่เกิด (Effect) ซึ่งสรุปได้ดังนี้



รูปที่ 2.1 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน จึงตรงกับคำต่อไปนี้

Computer Assisted Instruction หรือ Computer Aided Instruction:CAI

Computer Based Teaching หรือ Computer Based Training:CBT

Computer Based Education:CBE

Computer Managed Instruction:CMI

Computer Managed Learning:CML

สำหรับประเทศไทยนั้นจะคุ้นเคยกับคำว่า CAI มากกว่าคำอื่นๆ ส่วนภาษาไทยนั้น จะใช้คำว่าบทเรียนคอมพิวเตอร์ช่วยสอน บทเรียนช่วยสอนด้วยคอมพิวเตอร์ บทเรียนโปรแกรมคอมพิวเตอร์ บทเรียนสำเร็จรูปด้วยคอมพิวเตอร์หรืออื่นๆ

ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน คือ บทเรียนสำเร็จรูปที่นำเสนอเนื้อหา สื่อ กิจกรรม การตรวจปรับการประเมินผล และกระบวนการเกี่ยวกับการเรียนรู้อื่นๆ ด้วยคอมพิวเตอร์

2.2.2 ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน

บทเรียนคอมพิวเตอร์ช่วยสอน เป็นการประยุกต์ใช้คอมพิวเตอร์เพื่อการศึกษาในรูปแบบของบทเรียนสำเร็จรูป เพื่อสนับสนุนการเรียนการสอนรายบุคคลทั้งในห้องเรียนของสถาบันศึกษาและการฝึกอบรมในสถานประกอบการ ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอนจึงเหมาะสมกับสถานการณ์ต่อไปนี้

1. ใช้เพื่อสอนแทนผู้สอนทั้งในและนอกห้องเรียน เช่น การสอนแทนผู้สอนโดยตรง การสอนทบทวน การสอนเสริม เป็นต้น
2. ใช้เพื่อการศึกษาทางไกลผ่านสื่อโทรคมนาคม เช่น การเรียนการสอนทางไกล(Distance Learning)
3. ใช้กับเนื้อหาการสอนที่ซับซ้อน ไม่สามารถศึกษาได้จากของจริงโดยตรง เช่นการเปลี่ยนแปลงภายในโมเลกุล
4. ใช้กับลักษณะงานที่อันตราย มีความเสี่ยงต่อความเสียหายสูง โดยเฉพาะอย่างยิ่งการฝึกอบรม เช่น การจำลองระบบการบิน (Flight Simulator)
5. ใช้กับสาระเนื้อหาที่ต้องการแสดง ให้เห็นลำดับขั้นตอนการเปลี่ยนแปลงที่ละขั้นๆ โดยการจำลองจากเหตุการณ์จริงที่เกิดขึ้นเร็วไปหรือช้าเกินไป

6. ใช้ในการฝึกอบรมพนักงานใหม่ โดยไม่ต้องเสียเวลาเริ่มงานเหมือนกับการฝึกอบรมแบบปกติ

7. ใช้เพื่อคงความเป็นมาตรฐาน ทั้งหลักสูตรการสอนและการฝึกอบรมให้เหมือนกันทุกแห่งที่ใช้หลักสูตรเดียวกัน

2.2.3 ข้อได้เปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน

จากผลการวิจัยในการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน สรุปได้ว่า

1. บทเรียนคอมพิวเตอร์ช่วยสอน ทำให้ผลสัมฤทธิ์ทางการเรียนสูงขึ้นเมื่อเปรียบเทียบกับการเรียนการสอนในห้องเรียน

2. ลดเวลาเรียนลง เมื่อเปรียบเทียบกับการเรียนการสอนในห้องเรียน

3. ผู้เรียนสนใจการเรียนมากขึ้นเมื่อเรียนด้วยบทเรียนคอมพิวเตอร์ช่วยสอน

4. ผู้เรียนจะมีปฏิสัมพันธ์ กับบทเรียนอย่างแท้จริง โดยมีการโต้ตอบซึ่งกันและกัน ทำให้เกิดการเรียนรู้อย่างลึกซึ้ง

5. ผู้เรียนเป็นผู้ควบคุมบทเรียนด้วยตนเอง นับตั้งแต่การจัดการบทเรียนเลือกเรียนกิจกรรมที่ตนถนัด จนถึงการประเมินผลการเรียนด้วยตนเอง

6. บทเรียนคอมพิวเตอร์ช่วยสอนเสนอเนื้อหาได้รวดเร็วฉับไว เก็บเนื้อหาไว้ได้มากกว่า เสนอรูปภาพที่เคลื่อนไหวซับซ้อนและมีเสียงประกอบได้

7. สามารถนำติดตัวไปเรียนในสถานที่ต่างๆ โดยไม่มีข้อจำกัดด้านเวลาและสถานที่

2.2.4 ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน

ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน เมื่อเปรียบเทียบกับการเรียนการสอนปกติ จำแนกออกเป็น 4 ประเด็น ได้ดังนี้

1. ค่าใช้จ่ายการพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน จำเป็นต้องลงทุนค่าใช้จ่ายก่อนข้างสูง ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์

2. ผู้เชี่ยวชาญ ต้องจัดเตรียมผู้เชี่ยวชาญหลายด้านมาระดมความคิดเพื่อพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน ทั้งด้านหลักสูตร การเรียนการสอน สื่อการสอน การวัดและประเมินผล และด้านการโปรแกรมคอมพิวเตอร์

3. ระยะเวลาในการพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน จะต้องใช้เวลามากสำหรับการพัฒนา การทดสอบ และการปรับปรุงบทเรียน

4. ความยากในการออกแบบ เนื่องจากบทเรียนคอมพิวเตอร์ช่วยสอนจะต้องออกแบบให้ยืดหยุ่นต่อการใช้งาน มีทางเลือกที่เหมาะสมกับผู้เรียนที่มีความถนัดแตกต่างกันจึงเป็นการยากที่จะออกแบบเนื้อหาให้สอดคล้องกับกลุ่มเป้าหมายที่มีความแตกต่างกัน

2.2.5 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน ประกอบด้วย

1. ฮาร์ดแวร์ ได้แก่ ตัวเครื่องและอุปกรณ์ที่ใช้เป็นทางผ่านของบทเรียนประกอบด้วย จอภาพ แป้นพิมพ์ เครื่องจับแผ่น Diskette เครื่องอ่าน CD-ROM ลำโพง และอื่นๆ
2. ซอฟต์แวร์ ได้แก่ โปรแกรมที่ใช้จัดการและนำเสนอบทเรียน เรียกว่า Authoring System เป็นโปรแกรมสำเร็จรูปที่ออกแบบมาเพื่อการออกแบบบทเรียนคอมพิวเตอร์ช่วยสอนโดยตรง เช่น Authorware, Icon Author, Tencor หรืออาจใช้ภาษาคอมพิวเตอร์พัฒนาบทเรียนก็ได้

2.2.6 ลักษณะโครงสร้างของบทเรียนคอมพิวเตอร์ช่วยสอน

บทเรียนคอมพิวเตอร์ช่วยสอนเป็นการนำคอมพิวเตอร์มาใช้ในการเรียนการสอนการทบทวนการทำแบบฝึกหัดหรือการวัดผล ผู้เรียนแต่ละคนจะนั่งอยู่หน้าเครื่องคอมพิวเตอร์เรียกโปรแกรมบทเรียนที่เตรียมไว้สำหรับการสอนในวิชานั้นๆ ขึ้นมาแสดงบนหน้าจอภาพและอ่านทำความเข้าใจโต้ตอบกับบทเรียน ตามการจัดการของบทเรียน จนจบบทเรียน

ลักษณะสำคัญของบทเรียนคอมพิวเตอร์ช่วยสอนบทเรียนหนึ่งๆ จะประกอบด้วยสาระสำคัญๆ 5 ประการดังนี้

1. การสนับสนุนการเรียนการสอนรายบุคคล
2. การโต้ตอบระหว่างผู้เรียนกับบทเรียน
3. รูปแบบการนำเสนอบทเรียน
4. การจัดการบทเรียน
5. ประสิทธิภาพในการใช้งานของบทเรียน

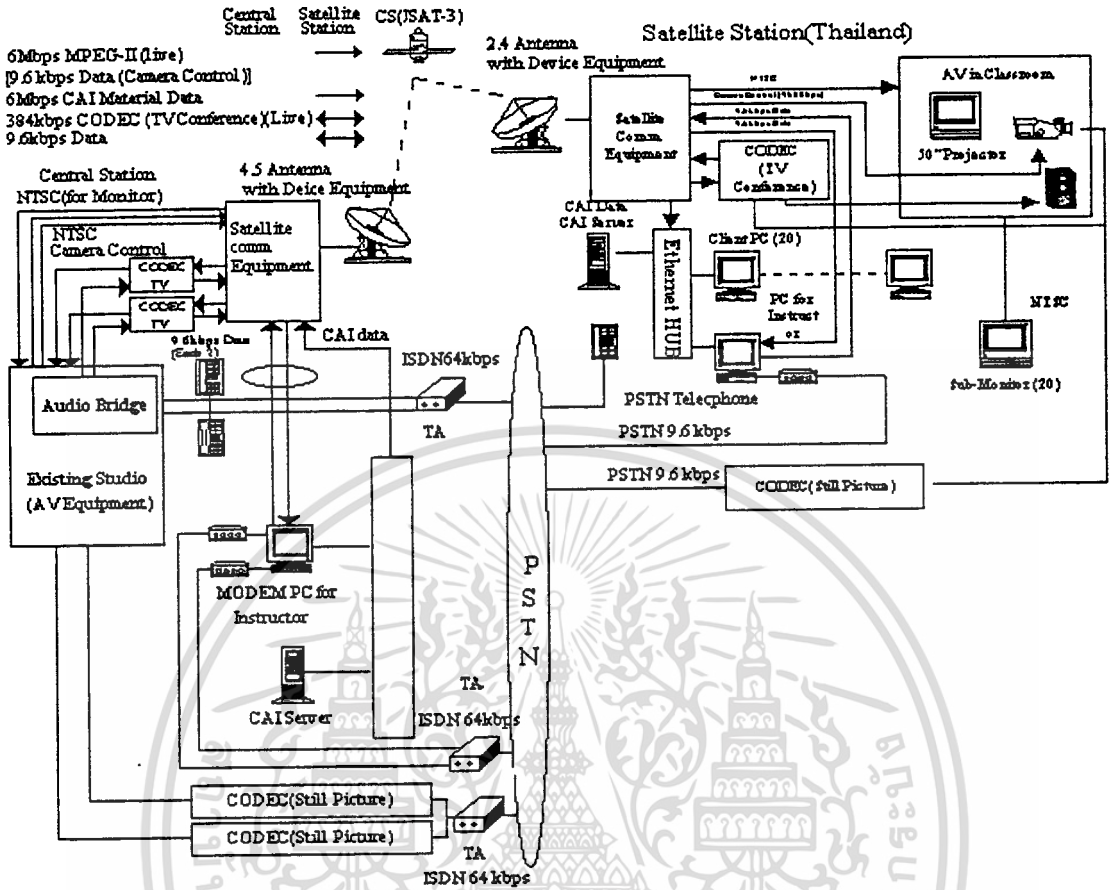
2.2.7 โครงการวิจัยร่วมญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการ

สอนทางไกลแบบสองทางพร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ

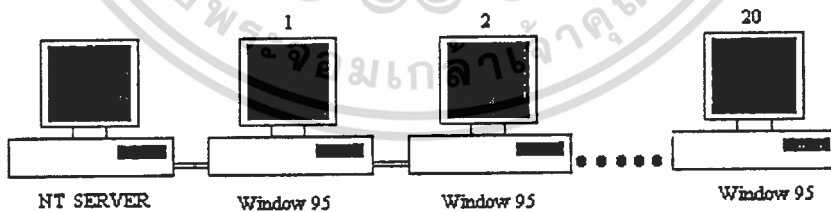
เทคโนโลยีทางการสื่อสาร โทรคมนาคมและคอมพิวเตอร์ได้ถูกพัฒนาขึ้นเป็นอย่างมากในช่วง 10 ปีที่ผ่านมา ไม่ว่าจะเป็นโทรศัพท์มือถือ เครื่องเล่นวีดีโอแบบเลขเซอร์ โทรทัศน์ระบบดิจิทัล หรือ HDTV (High Definition TeleVision) ระบบดาวเทียมแบบ

ดิจิทัล

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ระบบการสอนทางไกลแบบสองทางพร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ



รูปที่ 2.3 เครื่องข่ายของระบบ

หรือ DSS (Digital Satellite System) คอมพิวเตอร์แบบสื่อประสม (Multimedia computer) ตลอดจนระบบเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่รู้จักกันดีในนามของ อินเทอร์เน็ต (Internet) เทคโนโลยีเหล่านี้ต่างๆ ได้รับการได้รับการพัฒนาขึ้นเพื่อการให้บริการและตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนองความต้องการของมนุษย์ในรูปแบบต่างๆ รวมทั้งเพื่อการแข่งขันกันทางการตลาด แห่งยุคโลกาภิวัตน์ในปัจจุบัน

ในหน่วยงานทั้งรัฐบาลและเอกชน ต่างก็พยายามที่จะปรับปรุงรูปแบบการทำงานและการให้บริการของตน ให้มีลักษณะของการนำเทคโนโลยีสารสนเทศ (Information technology) เข้ามาใช้ในองค์กรกันมากขึ้น โดยพยายามนำเทคโนโลยีสื่อสารและคอมพิวเตอร์เข้ามาประยุกต์ เพื่อให้เกิดประสิทธิภาพการทำงานในองค์กร โดยเฉพาะอย่างยิ่งหน่วยงานทางการศึกษา ที่ต่างก็พยายามพัฒนารูปแบบการเรียนการสอนของตน ให้สามารถบริการนักศึกษา และลดปัญหาต่างๆ ที่เป็นอุปสรรคต่อการเรียนการสอนลง เช่น ปัญหาการขาดแคลนอาจารย์ ปัญหาเรื่องระยะทาง เป็นต้น รวมทั้งเพื่อตอบสนองนโยบายของรัฐบาลในการที่กระจายโอกาสทางการศึกษาไปสู่ชนบทได้อย่างทั่วถึง โดยอาศัยเทคโนโลยีเข้ามาช่วยขยายบริการเพื่อให้ผู้เข้าเรียนเข้าถึงการบริการการศึกษามากที่สุด

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหนึ่งหน่วยงานการศึกษาของรัฐที่ได้พยายามพัฒนารูปแบบการเรียนการสอนให้สอดคล้องกับการเปลี่ยนแปลงของเทคโนโลยีเพื่อตอบสนองนโยบายของรัฐบาล โดยทำการวิจัยเรื่องระบบการเรียนการสอนทางไกลในโครงการต่างๆ ด้วยเทคโนโลยีที่แตกต่างกันไป เช่น การใช้ระบบการประชุมแบบเห็นภาพทางเครื่องคอมพิวเตอร์ (Desktop video conferencing) บนเครือข่ายผ่านดาวเทียมในโครงการ POST - Partners การใช้ระบบการสอนทางไกลผ่านเครือข่ายใยแก้วนำแสงในโครงการวิทยาเขตสารสนเทศของทบวงมหาวิทยาลัย และการใช้ระบบการสอนทางไกลแบบสื่อประสมผ่านดาวเทียมในโครงการวิจัยร่วมระหว่างไทย-ญี่ปุ่น เป็นต้น นอกจากนี้สถาบันฯยังได้นำเทคโนโลยีเครือข่ายคอมพิวเตอร์เข้ามาใช้เพื่อพัฒนารูปแบบการศึกษาและสร้างงานวิจัยใหม่ๆ ขึ้นในสถาบัน อันได้แก่ เทคโนโลยีเครือข่ายระบบเปิด (Open System) และระบบเครือข่ายภายในสถาบันฯ (Campus network) เป็นต้น เทคโนโลยีดังกล่าวนี้เป็นโครงสร้างพื้นฐานสำหรับการเชื่อมโยงเครือข่ายคอมพิวเตอร์ภายในสถาบันฯ ให้สามารถติดต่อสื่อสารกับอินเทอร์เน็ต (Internet) และโลกภายนอกโดยตรง

โครงการสำหรับนวัตกรรมและเทคโนโลยีการศึกษา เป็นหน่วยงานใหม่ของสถาบันฯ ที่ถูกจัดตั้งขึ้นเพื่อตอบสนองนโยบายและแนวทางดังกล่าว โดยใช้เป็นศูนย์กลางในการให้บริการผลิตโปรแกรมบทเรียน เพื่อสนับสนุนการสอนแก่คณาจารย์ในสถาบันฯ รวมทั้งเป็นศูนย์วิจัยพัฒนาและให้บริการทางด้านเทคโนโลยีทางการศึกษา เพื่อพัฒนาศักยภาพเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

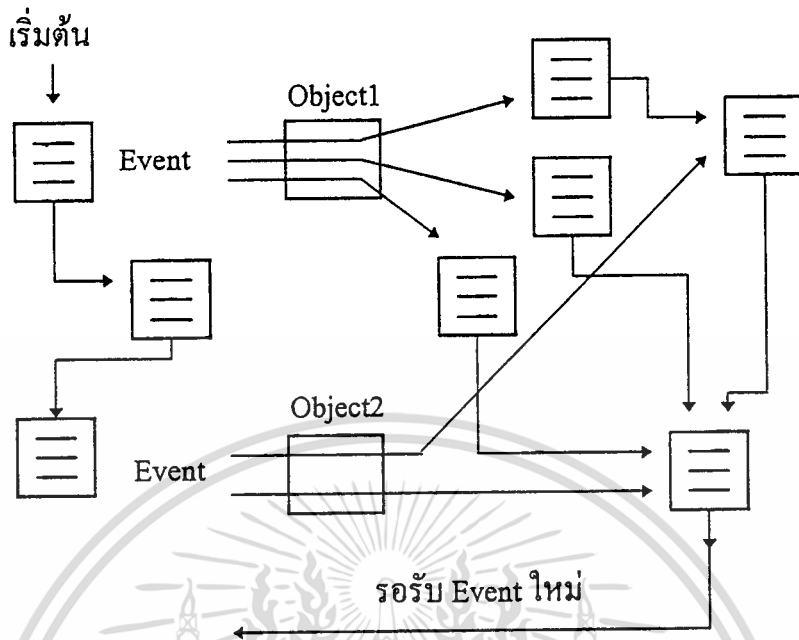
และคุณภาพทรัพยากรมนุษย์ให้เกิดประโยชน์ ในการถ่ายทอดความรู้ทางวิชาการสูงสุด ปัจจุบันโครงการ สำนักนวัตกรรมฯ รับผิดชอบในการพัฒนาทรัพยากรมนุษย์ด้วยสื่อประสมผ่านดาวเทียม ไทย-ญี่ปุ่น ซึ่งโครงการทั้งสองเป็นโครงการที่จะนำเทคโนโลยีเข้ามาใช้เพื่อประสิทธิภาพทางการศึกษาเป็นหลัก และพัฒนาการเรียนการสอนของสถาบันฯ

จากโครงสร้างของแต่ละระบบเดิม ทำให้การใช้งานถูกจำกัดอยู่เฉพาะในแต่ละส่วน โดยไม่สามารถกระจายโอกาสไปยังกลุ่มเป้าหมายได้เท่าเทียมกันทุกภาควิชาและคณะในสถาบันฯ เนื่องจากขาดการรวมหรือเชื่อมต่อกับเทคโนโลยีต่างๆ ที่มีอยู่เข้าด้วยกันเพื่อให้เกิดการใช้งานได้สูงสุด ดังนั้นโครงการสำนักนวัตกรรมฯ จึงได้มีแนวคิดที่จะสร้างระบบเครือข่ายสำนักนวัตกรรมฯ ขึ้น โดยรวมเทคโนโลยีต่างๆ ที่มีอยู่เดิมด้วยการเชื่อมต่อเครือข่ายของแต่ละระบบในแต่ละโครงการเข้าด้วยกัน พร้อมนำเสนอการใช้งานและการให้บริการทางการศึกษาในรูปแบบต่างๆ ที่จะเกิดขึ้นบนเครือข่ายใหม่ เช่น บริการด้านการเรียนการสอนทางไกลผ่านเครือข่ายในรายวิชาต่างๆ บริการสื่อการสอน บริการสอนทางไกลผ่านอินเทอร์เน็ต บริการการประชุมแบบเห็นภาพบนเครื่องคอมพิวเตอร์ บริการวีดิโอตามคำขอ และบริการมหาวิทยาลัยเสมือน เป็นต้น

2.3 หลักการของโปรแกรม HURESS

โปรแกรม HURESS เป็นโปรแกรมที่ถูกออกแบบและพัฒนาขึ้นมาเพื่อใช้ในโครงการวิจัยร่วมญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทางพร้อมสื่อประสมดาวเทียมระหว่างประเทศ โดยเฉพาะ โปรแกรม HURESS อาศัยการทำงานที่เรียกว่า อีเวินไครฟ์เวิน (Event-Driven) คือ ขึ้นอยู่กับเหตุการณ์ (Event) โดยเริ่มจากการสร้างวินโดว์ย่อยและวินโดว์จะประกอบด้วย อปเจ็คต่างๆ เช่น ช่องรับข้อความ Scroll Bar หรือปุ่ม (Button) เมื่อกำหนดสิ่งเหล่านี้ครบตามความต้องการแล้วจึงระบุว่าจะประกอบแต่ละอย่างจะทำงานอย่างไร โดยเขียนโปรแกรมย่อยๆ ปะเข้าไปอปเจ็คเหล่านี้

จากลักษณะการทำงานแบบอีเวิน คือ กำหนดหน้าตาของจอภาพหรือที่เรียกว่าส่วนติดต่อกับผู้ใช้ และระบุว่าถ้าเกิดเหตุการณ์อย่างนี้กับสิ่งนี้จะต้องทำอะไร ซึ่งโปรแกรมที่เราจะเขียนก็คือ วนที่เราจะต้องเขียนโปรแกรมเข้าไปนั่นเอง



รูปที่ 2.4 การเขียนโปรแกรมแบบธรรมดาและ การเขียนโปรแกรมแบบ Event-driven

โดยสรุปแล้วรูปแบบของหลักการ HURESS คือ เริ่มจากออกแบบจอภาพและเขียนโปรแกรมให้ทำงานตามเหตุการณ์ที่เกิดขึ้น โดยเขียนโปรแกรมสำหรับแต่ละเหตุการณ์ ปะเข้าไปยัง ออปเจ็ค ต่างๆ ให้ทำงานตามเหตุการณ์ที่เกิดขึ้น

2.4 ส่วนประกอบของโปรแกรม HURESS

2.4.1 Books

ส่วนที่ถือว่าเป็นส่วนสำคัญหลักในโปรแกรม HURESS ก็คือ หนังสือ (Book) หนังสือจะถูกเก็บไว้ในชั้นหนังสือ ที่เรียกว่า Bookshelves หนังสือในโปรแกรมนี้นี้ ถือว่าเป็นหนังสืออิเล็กทรอนิกส์ ซึ่งในแต่ละหน้าจะสามารถเรียกใช้อุปกรณ์ต่าง ๆ เพื่อสร้าง ตัวหนังสือ รูปภาพ เสียง รวมถึงการเรียกใช้ Background ต่าง ๆ ได้

2.4.2 Page

ในส่วนของหน้าหนังสือ จะเป็นส่วนที่แสดงเนื้อหาต่าง ๆ ของหนังสือ ในแต่ละหน้าหนังสือ สามารถที่จะเรียกใช้เครื่องมือในการสร้าง ตัวหนังสือ ภาพ เสียง ปุ่มต่าง ๆ ได้ตามแต่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของประเภทหน้าหนังสือในโปรแกรม HURESS นั้น จะประกอบไปด้วย 5 ประเภทด้วยกัน คือ

- หน้าปกหนังสือ (COVER)
- หน้าสารบัญ (Table of Contents)
- หน้าเนื้อหาหลัก (Main Text)
- หน้าดัชนี (Index)
- หน้าปิดท้ายของหนังสือ (Back Cover)

2.4.3 Media

ในส่วนนี้จะเป็นส่วนของเครื่องมืออำนวยความสะดวก ในการสร้าง Media ใช้งานต่าง ๆ ในโปรแกรม เช่น ตัวอักษร รูปภาพ เสียง ปุ่มการใช้งาน ฯลฯ

2.4.4 Desk

ในส่วนนี้จะเป็นส่วนหลักในการใช้งานโปรแกรมในลักษณะโครงสร้างโปรแกรม เช่น การยืมหนังสือ การสร้างชั้นหนังสือ การปิดชั้นหนังสือ การเซ็ทหน้าจอ การเซ็ทเมนูบาร์ต่าง ๆ ฯลฯ

2.4.5 Bookshelves

เป็นส่วนในการใช้เก็บหนังสือ ชั้นหนังสือสามารถสร้างขึ้นกี่ชั้นก็ได้ แล้วแต่เราต้องการเก็บหนังสืออย่างไร

2.4.6 Entity

เป็นส่วนของรูปแบบฟังก์ชันการใช้งานในลักษณะต่าง ๆ ตามเมนูเลือกใช้งานหรือในรูปแบบของไอคอน ซึ่งการใช้งานจะทำงานตามฟังก์ชัน โดยการใช้เมาส์คลิกที่ฟังก์ชันนั้น

2.4.7 Procedures

เป็นส่วนการลำดับการทำงานของแต่ละฟังก์ชัน โดยในส่วนนี้จะมีการเขียนลำดับการทำงานในรูปแบบของการเขียนโปรแกรมการทำงาน ให้กับฟังก์ชันที่เราต้องจะให้ทำ

2.4.8 Objects

โปรแกรม HURESS จะ Support กับ OLE2, ซึ่ง OLE2 เป็นลักษณะฟังก์ชันการแลกเปลี่ยนข้อมูลข่าวสารต่าง ๆ ใน windows อื่น ๆ ได้

2.5 หลักการใช้โปรแกรม HURESS

ในหัวข้อนี้จะกล่าวถึงการใช้หนังสือ ในโปรแกรม HURESS ซึ่งจะมีลำดับการใช้เป็นลำดับตามหัวข้อต่อไปนี้

2.5.1 การยืมและเปิดหนังสือ (Borrowing and Operating a Book)

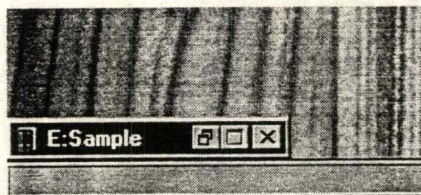
1. การยืมหนังสือผ่าน Desk Main Menu

การยืมในส่วนนี้จะยืมจาก ชั้นวางหนังสือ (Bookshelf) บน Desk Main Menu เมื่อทำการยืมจะปรากฏ Dialog Box ดังรูปที่ 2.5



รูปที่ 2.5 การยืมหนังสือผ่าน Desk Main Menu

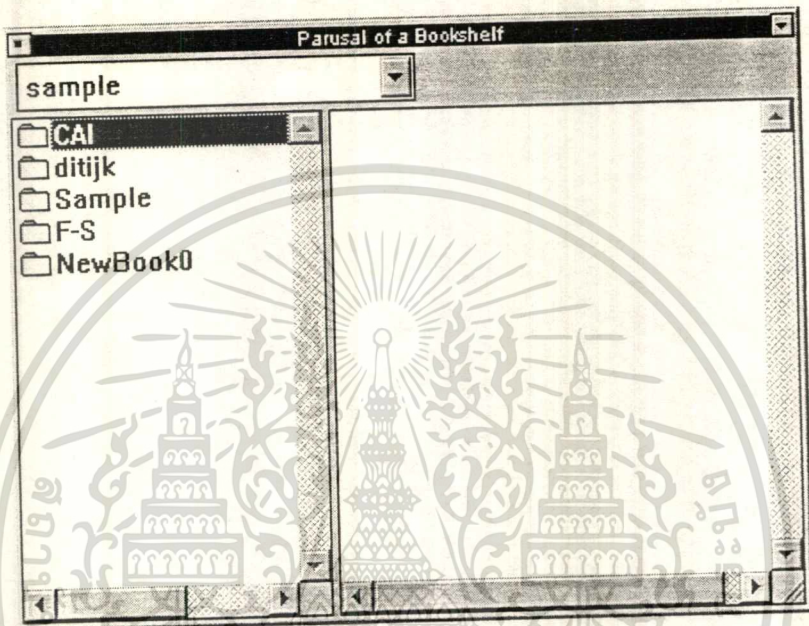
จากชั้นวางหนังสือ (Bookshelf) หากยืมหนังสือ Sample ในชั้นวางหนังสือ Sample หนังสือที่ถูกยืมก็จะปรากฏเป็น Dialog Box ตรงมุมล่างซ้ายของ Desk Screen ในลักษณะ Minimize Form ดังรูปที่ 2.6



รูปที่ 2.6 หนังสือที่ถูกยืม


2. การยืมหนังสือโดยผ่าน Bookshelf Browser

จาก Desk Main Menu นำเมาส์ไปคลิกที่ Bookshelf แล้วคลิกเลือกที่ Peruse Bookshelf ก็จะปรากฏหนังสือที่เราต้องการยืมดังรูปที่ 2.7



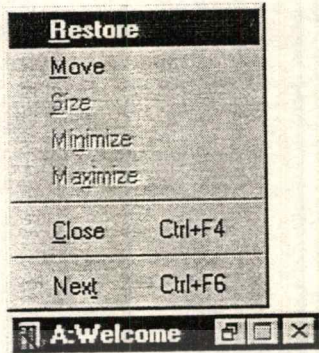
รูปที่ 2.7 การยืมหนังสือผ่าน Bookshelf Browser

3. การยืมโดยใช้ เมนู Selection Bar

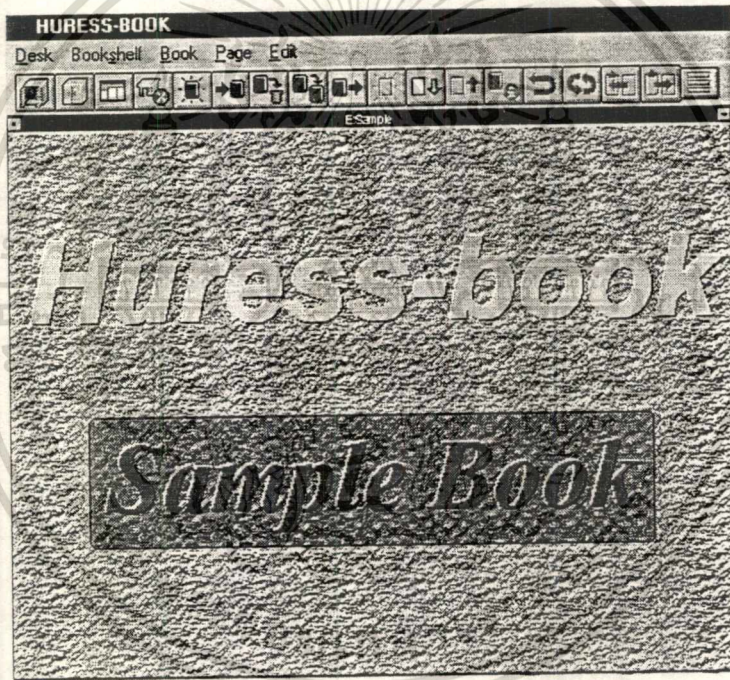
นำเมาส์ไปคลิกที่ ปุ่มดังรูป  ก็จะเข้าสู่การยืมหนังสืออีกแบบหนึ่ง

2.5.2. การเปิดหนังสือ Opening A Book

นำเมาส์ไปคลิกตรง Minimize Book มุมล่างด้านซ้ายของ Windows Desktop แล้วทำการคลิกปุ่ม Restore ดังรูปที่ 2.8 และหนังสือที่ถูกยืมก็จะปรากฏบนหน้าจอดังรูปที่ 2.9



รูปที่ 2.8 การเปิดหนังสือ

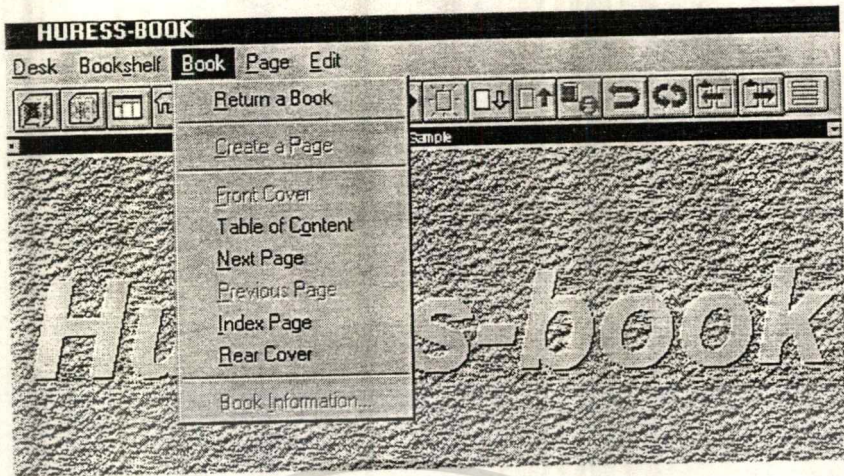


รูปที่ 2.9 หนังสือที่ทำการเปิดได้แล้ว

2.5.3 การทำงานเกี่ยวกับหน้าหนังสือ

คลิกเมาส์ไปที่ Book บน Desk Main Menu ก็จะปรากฏหน้าจอดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

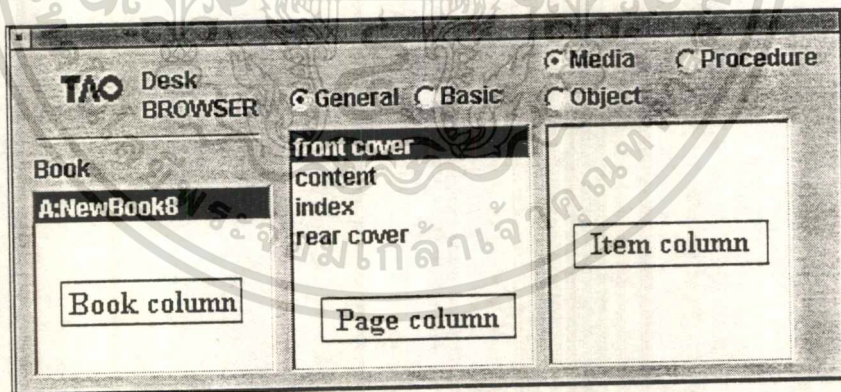


รูปที่ 2.10 การทำงานเกี่ยวกับหนังสือ

ทำการคลิกที่ Next Page หรือ Previous Page ในการเปลี่ยนหน้า Next Page จะเป็นการเลื่อนหน้าต่อไป ส่วน Previous Page จะเป็นการเลื่อนมายังหน้าก่อน

2.5.4 การใช้งาน Desk Browser

Desk Browser จะเป็นตัวแสดงรายละเอียดของหนังสือที่เราชม ดังรูปที่ 2.11



รูปที่ 2.11 การใช้ Desk Browser

Book Column จะแสดงชื่อหนังสือที่เราเปิดอ่าน

Page Column จะแสดงหน้าต่างๆของหนังสือ

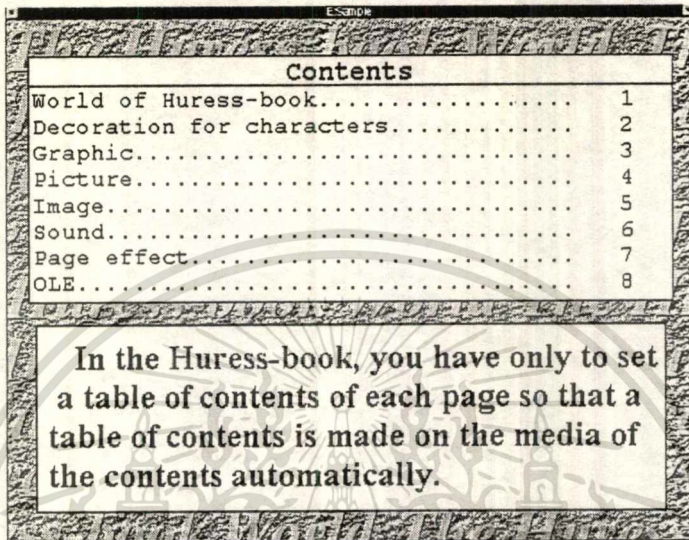
Item Column จะแสดงสื่อต่างๆที่เราใช้ประกอบในหนังสือเล่มนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 การใช้หน้าสารบัญ

คลิกเมาส์ไปหน้าสารบัญ (Contents) ก็จะปรากฏหน้าจอดังรูปที่ 2.12



Contents	
World of Huress-book.....	1
Decoration for characters.....	2
Graphic.....	3
Picture.....	4
Image.....	5
Sound.....	6
Page effect.....	7
OLE.....	8

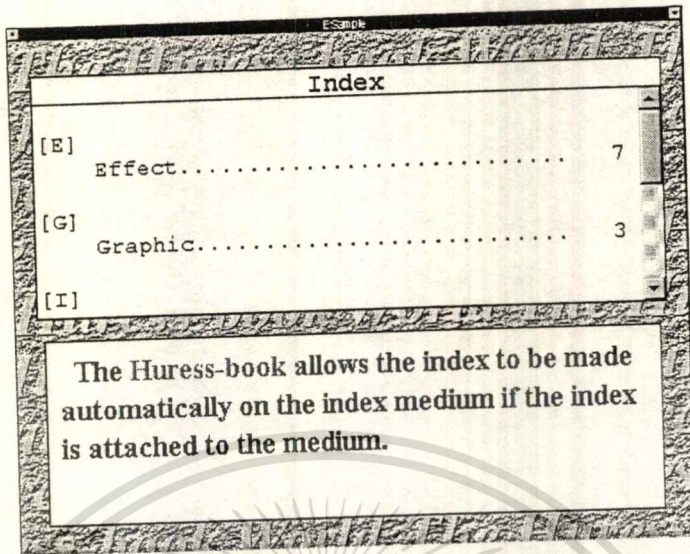
In the Huress-book, you have only to set a table of contents of each page so that a table of contents is made on the media of the contents automatically.

รูปที่ 2.12 การใช้หน้าสารบัญ

หน้าสารบัญนี้หากต้องการไปที่หน้าไหนก็ทำการ Double Click ที่หน้าที่ต้องการในตาราง Contents ได้เลย

2.5.6 การใช้งานหน้าดัชนี

คลิกเมาส์ไปที่หน้าดัชนี (Index) เมื่อต้องการจะไปที่หัวข้อใดในดัชนีก็ทำการ Double Click ที่หัวข้อในตารางดัชนีได้เลย ดังแสดงในรูปที่ 2.13



รูปที่ 2.13 การใช้หน้าต่างนี้

2.5.7 การคืนหนังสือ

ให้ทำการคลิกเมาส์ไปยัง Return A Book บน Desk Main Menu หรือคลิกเมาส์ไปที่



Menu Selection Bar หรือจะทำการคลิกซ้ายที่ Close Box บนมุมซ้ายบนของหนังสือ

ดังรูปที่ 2.14



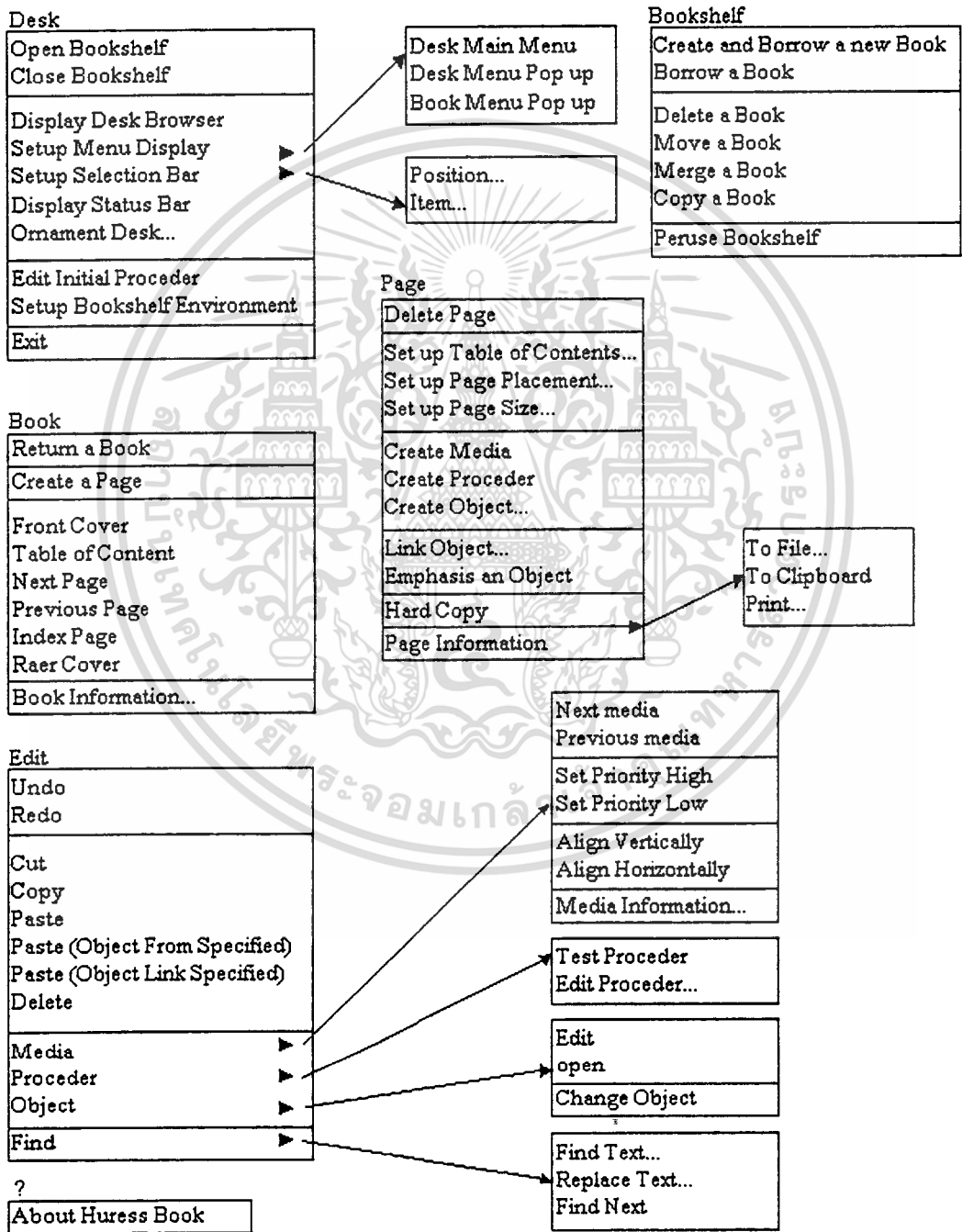
รูปที่ 2.14 การคืนหนังสือ โดย ปุ่ม Close Box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การทำงานในส่วนของ Desk Main Menu

การทำงานทั้งหมดของโปรแกรม HURESS โดยผ่าน Desk Main Menu จะเป็นดังรูป

ที่ 2.15



รูปที่ 2.15 การทำงานในส่วนของ Desk Main Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การทำงานในส่วนของเมนูย่อย [Desk]

1. Open Bookshelf

เป็นเมนูที่ใช้สำหรับการเปิดชั้นวางหนังสือก่อนที่จะเปิดหนังสือที่เราต้องการในชั้นนั้น ๆ

2. Close Bookshelf

เป็นเมนูที่ใช้สำหรับการปิดชั้นหนังสือ ก่อนการปิดโปรแกรม HURESS ทุกครั้ง

3. Display Desk Browser

เมื่อทำการเลือกเมนูนี้ ส่วนของ Desk Browser ก็จะปรากฏ

4. Setup Menu Display - Desk Main Menu

เป็นส่วนที่ใช้ setup Desk Main Menu ว่าจะให้แสดงผลส่วนไหนของจอภาพ

5. Setup Menu Display - Desk Menu Pop up

เป็นส่วนที่ใช้ Setup Desk Main Menu ว่าจะให้ปรากฏบนจอภาพหรือไม่

6. Setup Menu Display - Book Menu pop up

เป็นเมนูที่ใช้ setup Book Menu pop up ว่าจะให้ปรากฏบนจอภาพหรือไม่

7. Setup Selection Bar - Position

เป็นเมนูส่วนที่ใช้ setup ตำแหน่งของ Selection Bar ว่าจะให้ปรากฏบนหรือล่างของจอภาพ

8. Setup Selection Bar - Item

เป็นเมนูส่วนที่ใช้ set buttons ว่าจะให้ปรากฏบน menu selection bar หรือไม่

9. Display Status bar

เป็นเมนูที่ใช้ setup bar แสดงสถานะของเครื่องมือ

10. Ornament Desk

เป็นเมนูที่ใช้ setup background ของ โปรแกรม HURESS

11. Edit Initial Procedure

เป็นเมนูส่วนที่ใช้เขียนลำดับการทำงานของหนังสือเล่มนั้น ๆ ที่กำลังใช้อยู่ ตั้งแต่เริ่มเปิดหนังสืออ่านจนกว่าจะคืนหนังสือ

12. Set up Bookshelf Environment

เป็นเมนูส่วนที่ใช้ในการสร้างชั้นหนังสือ

13. Exit

เป็นเมนูส่วนที่ใช้ในเปิดโปรแกรม HURESS

2.6.2 การทำงานในส่วนของเมนูย่อย [Bookshelf]

1. Create and Borrow a New Book

เป็นเมนูส่วนที่ใช้ในการสร้างหนังสือขึ้นใหม่ และถือว่าการยืมหนังสือเล่มนั้นด้วย

2. Borrow a Book

เป็นเมนูส่วนที่ใช้ในการยืมหนังสือ ในชั้นหนังสือนั้น ๆ

3. Delete a Book

เป็นเมนูส่วนที่ใช้ในการลบหนังสือเล่มนั้น ๆ ที่เราต้องการจะลบ

4. Move a Book

เป็นเมนูในส่วนที่ต้องการจะย้ายหนังสือ ไปไว้ยังชั้นหนังสืออื่นที่เราต้องการจะนำหนังสือเล่มนั้นไปเก็บไว้

5. Merge a Book

เป็นเมนูที่ใช้ในการรวมหนังสือ 2 เล่มเข้าด้วยกัน เพื่อสร้างเป็นหนังสือเล่มใหม่ขึ้นมา

6. Copy a Book

เป็นส่วนของเมนูที่ใช้ในการ copy หนังสือที่เราต้องการ

7. Peruse Bookshelf

เป็นเมนูแสดงจำนวน Bookshelf ว่ามีกี่เล่ม

2.6.3 การทำงานในส่วนของเมนูย่อย [Book]

1. Return a book

เป็นเมนูที่ใช้ในการคืนหนังสือ

2. Create a Page

เป็นเมนูที่ใช้ในการสร้างหน้าที่เราต้องการ

3. Front Cover, Table of Contents, Next Page, Previous Page,

Index Page, Rear Cover

เป็นเมนูแสดงสถานะของหน้าหนังสือที่กำลังอ่านอยู่ ถ้าหากต้องการจะไปหน้าใด ก็คลิกเมาส์มายังเมนูนี้เลือกหน้าที่ต้องการได้เลย

4. Book Information

เป็นเมนูส่วนที่ใช้เช็คแก้ไขข้อมูลเกี่ยวกับหนังสือ เช่น ขนาดหนังสือ เป็นต้น

2.6.4 การทำงานในส่วนของเมนูย่อย [Page]

1. Delete Page

เป็นเมนูส่วนที่ใช้ในการลบหน้าที่ต้องการจะลบ ในหนังสือเล่มนั้น ๆ ที่กำลังเปิดอ่านอยู่

2. Set up Table of Contents

เป็นเมนูส่วนที่ใช้ในการเซ็ทหน้าสารบัญ (Contents) ของหนังสือเล่มนั้น ๆ

3. Set up Page Placement

เป็นส่วนของเมนูที่ใช้ในการเซ็ทวางตำแหน่งของหน้ากระดาษ

4. Set up Page Size

เป็นส่วนที่ใช้ในการเซ็ทขนาดของหน้ากระดาษ

5. Create Media

เป็นส่วนในการสร้าง Media ที่ต้องการจะสร้างในโปรแกรม

6. Create Procedure

เป็นส่วนที่ใช้ในการสร้างลำดับคำสั่งในการทำงานของ Media ที่เราต้องการ

7. Create Object

เป็นส่วนที่แสดงการเพิ่ม Dialog Box ของ Object ในการสร้าง Objects

8. Link Object

เป็นส่วนที่ใช้ในการเชื่อมโยงส่วนของ Dialog Box เข้าด้วยกัน

9. Emphasis and Object

เป็นส่วนที่ใช้ในการสร้างวงกลมในหน้านั้น ๆ

10. Hard Copy to File

เป็นส่วนในการใช้ในการ copy รูปภาพ ให้เป็นไฟล์

11. Hard copy to Clipboard

เป็นส่วนที่ใช้ Copy รูปภาพลง Clipboard

12. Hard copy to Print

เป็นส่วนที่ใช้ในการนำหน้าปัจจุบันพิมพ์ออกทางเครื่องพิมพ์

13. Page Information

เป็นส่วนที่ใช้เกี่ยวกับข้อมูลต่าง ๆ ของหน้ากระดาษ

2.6.5 การทำงานในส่วนของเมนูย่อย [Edit]**1. Undo**

เป็นเมนูที่ยกเลิกการกระทำครั้งสุดท้าย

2. Redo

เป็นส่วนที่เรียกใช้ข้อมูลที่ทำการ Undo ไป ให้ปรากฏขึ้นมาใหม่

3. Cut

เป็นส่วนที่ลบ Media ที่ต้องการ

4. Copy

เป็นส่วนที่ใช้ในการคัดลอก Media ที่เลือก

5. Paste

เป็นส่วนที่ใช้ในการวาง Media ที่คัดลอกมาในครั้งล่าสุด

6. Paste (Object Link Specified)

เป็นส่วนที่ใช้ในการวาง object ที่คัดลอกมาในครั้งล่าสุด

7. Paste (Object Form Specified)

เป็นส่วนที่ใช้ในการวาง Form ที่คัดลอกมาในครั้งล่าสุด

8. Delete

เป็นส่วนที่ใช้ในการลบ Media และอื่น ๆ ที่ต้องการ

9. Media - Next Media

เป็นการเลือก Media ถัดไป

10. Media - Previous media

เป็นการเลือกใช้ Media ก่อน Media ปัจจุบัน

11. Media - Set Priority High

เป็นการเพิ่มลำดับความสำคัญของ Media ขึ้น 1 ลำดับ

12. Media - Set Priority Low

เป็นการลดลำดับความสำคัญของ Media ลง 1 ลำดับ

13. Media - Information

เป็นการเซ็ทรายละเอียดเกี่ยวกับ Media

14. Media - Procedure Test Procedure

เป็นการทดสอบการทำงานของโปรแกรมย่อย

15. Media - Procedure Edit

เป็นการแก้ไขโปรแกรมย่อย

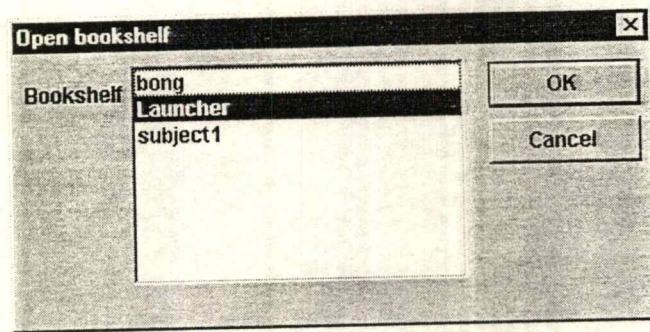
2.6.6 การทำงานในส่วนของ Dialog Box ต่าง ๆ ในโปรแกรม HURESS**1. HURESS-Book User Validation Dialog Box**

Dialog Box ในรูปที่ 2.16 ใช้เพื่อป้อน User name และ Password เพื่อที่จะเข้าสู่โปรแกรม HURESS

รูปที่ 2.16 การเข้าโปรแกรม HURESS

2. Dialog Box to open a Bookshelf

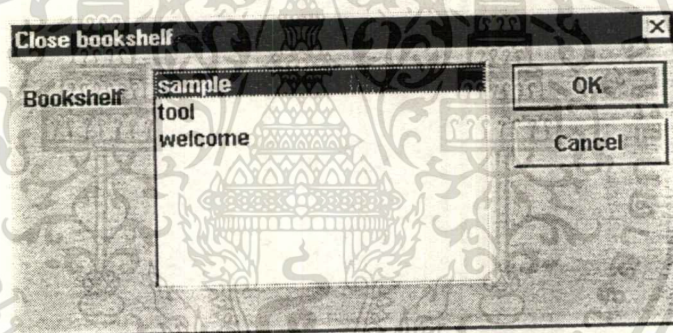
Dialog Box ในรูปที่ 2.17 จะใช้เพื่อเลือกชั้นวางหนังสือ เมื่อเราต้องการเลือก กด OK เพื่อใช้ชั้นหนังสือนั้น



รูปที่ 2.17 การเปิดหนังสือ

3. Dialog Box to Close a Bookshelf

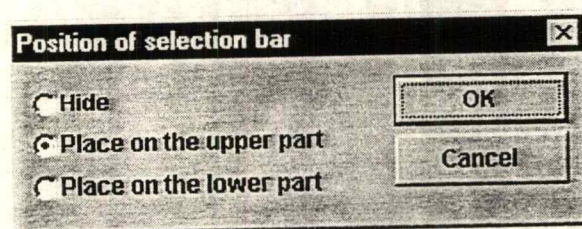
Dialog Box ในรูปที่ 2.18 ใช้เพื่อปิดชั้นหนังสือ กด OK เพื่อทำการปิดหนังสือนั้น ๆ



รูปที่ 2.18 การขีมนหนังสือ

4. Dialog Box to Set up Position of selection bar

Dialog Box ในรูปที่ 2.19 ใช้ในการกำหนดตำแหน่งของ Selection bar

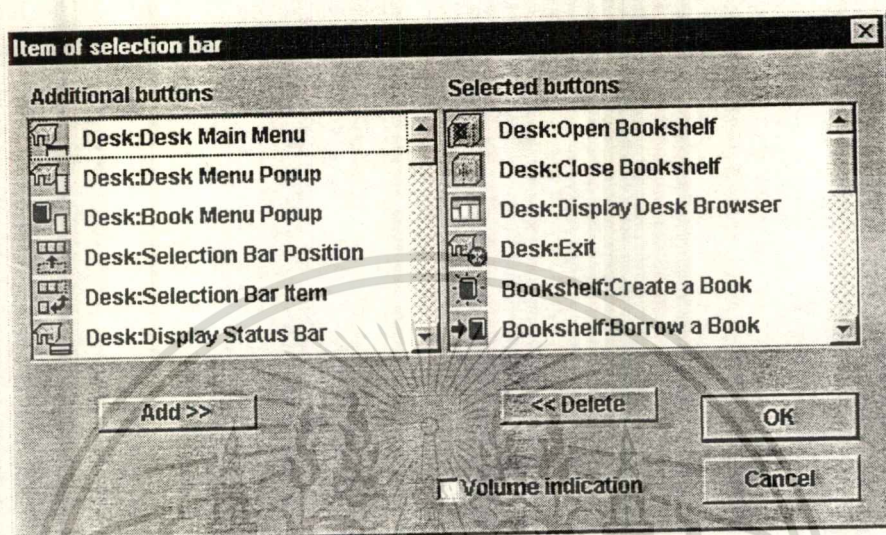


รูปที่ 2.19 การกำหนดตำแหน่ง Selection bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Dialog Box to set up Item of selection bar

Dialog Box ในรูปที่ 2.20 ใช้เมื่อต้องการ set การแสดงของ Item ใน selection bar



รูปที่ 2.20 การ set การแสดงของ Item ใน Selection bar

6. Dialog Box for Ornament Desk

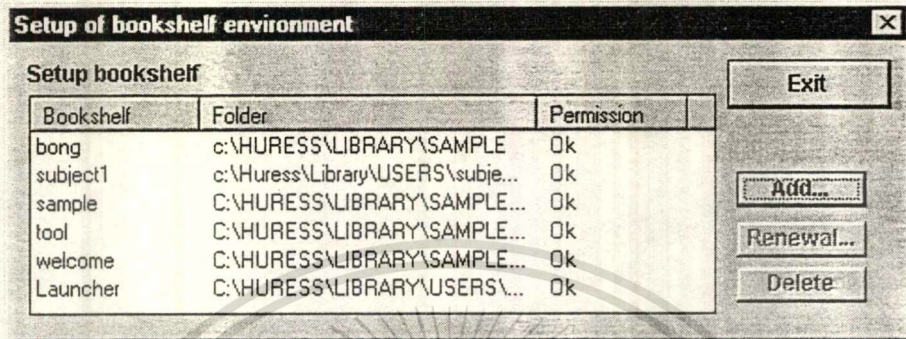
Dialog Box ในรูปที่ 2.21 ใช้เพื่อกำหนดการแสดงของแบ็คกราวนด์ในรูปแบบต่างๆ



รูปที่ 2.21 การกำหนดแบ็คกราวนด์

7. Dialog Box to Setup Bookself Environment

Dialog Box ในรูปที่ 2.22 ใช้เพื่อต้องการเพิ่มชั้นวางหนังสือ



รูปที่ 2.22 การเพิ่มชั้นวางหนังสือ

8. Additional Dialog box for Bookshelf

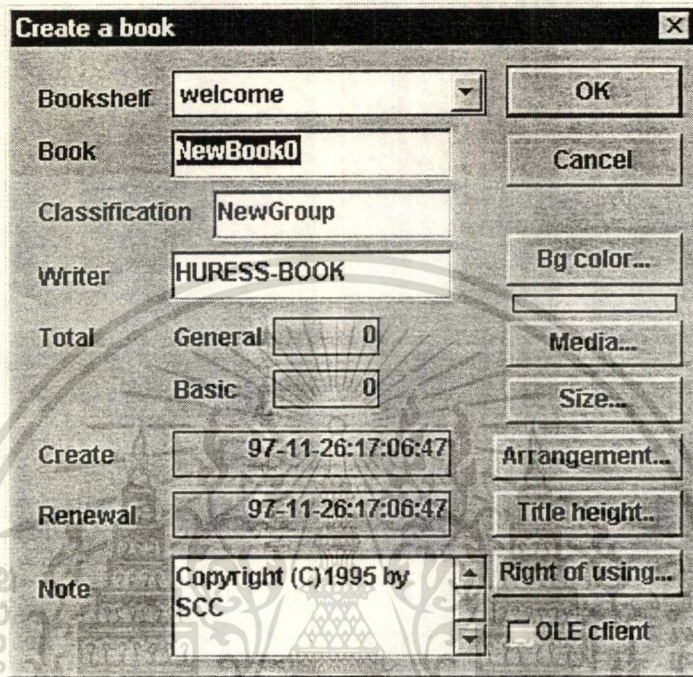
Dialog Box ในรูปที่ 2.23 ใช้เมื่อต้องการจะเพิ่มชั้นวางหนังสือ



รูปที่ 2.23 การเพิ่มชั้นวางหนังสือ

9. Dialog Box for creating a Book

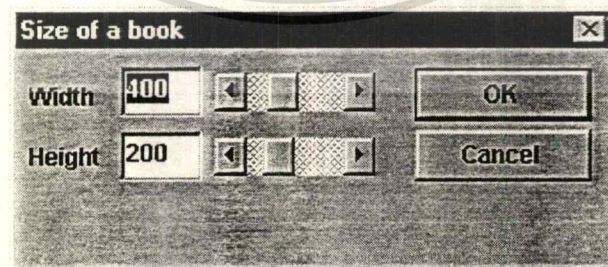
Dialog box ในรูปที่ 2.24 ใช้เพื่อสร้างหนังสือเล่มใหม่



รูปที่ 2.24 การสร้างหนังสือเล่มใหม่

10. Dialog Box to set up the size of a book

Dialog Box ในรูปที่ 2.25 ใช้เพื่อกำหนดขนาดของหนังสือ



รูปที่ 2.25 การกำหนดขนาดของหนังสือ

11. Information Dialog Box for Books

Dialog Box ในรูปที่ 2.26 ใช้เพื่อกำหนดรายละเอียดในส่วนต่าง ๆ ของหนังสือที่สร้างขึ้นใหม่

The 'Information of book' dialog box is a standard Windows-style window with a title bar and a close button. It is divided into several sections:

- Bookshelf:** A text box containing 'welcome'.
- Book:** A text box containing 'NewBook0'.
- Classification:** A text box containing 'NewGroup'.
- Writer:** A text box containing 'HURESS-BOOK'.
- Total:** Two spinners. 'General' is set to 46 and 'Basic' is set to 10.
- Create:** A text box containing the timestamp '97-11-14:15:29:18'.
- Renewal:** A text box containing the timestamp '97-11-26:16:27:41'.
- Note:** A text box containing 'Copyright (C)1995 by SCC'.
- Buttons:** A vertical column of buttons on the right side: 'OK', 'Cancel', 'Bg color...', 'Media...', 'Size...', 'Arrangement...', 'Title height...', 'Right of using...', and a checkbox for 'OLE client'.

รูปที่ 2.26 การกำหนดรายละเอียดในส่วนต่างๆ ของหนังสือ

12. Dialog box for Book Borrowing

Dialog Box ในรูปที่ 2.27 ใช้ในการยืมหนังสือ

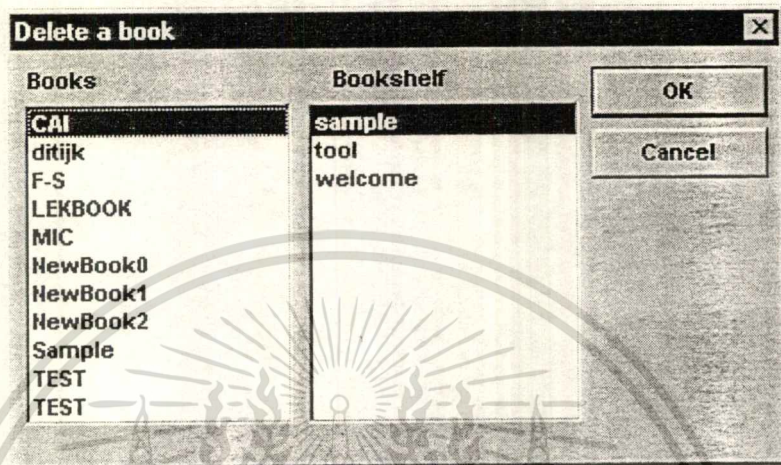
The 'Borrow book' dialog box features two list boxes and three buttons:

- Books List:** Contains the items 'Answer', 'F-S', 'TEST', and 'Welcome'.
- Bookshelf List:** Contains the items 'sample', 'tool', and 'welcome'.
- Buttons:** 'Peruse', 'Edit', and 'Cancel' are arranged vertically on the right side.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 2.27 การยืมหนังสือ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. Dialog Box to Delete a Book

Dialog Box ในรูปที่ 2.28 ใช้เพื่อต้องการจะลบหนังสือ



รูปที่ 2.28 การลบหนังสือ

14. Dialog Box to move a Book

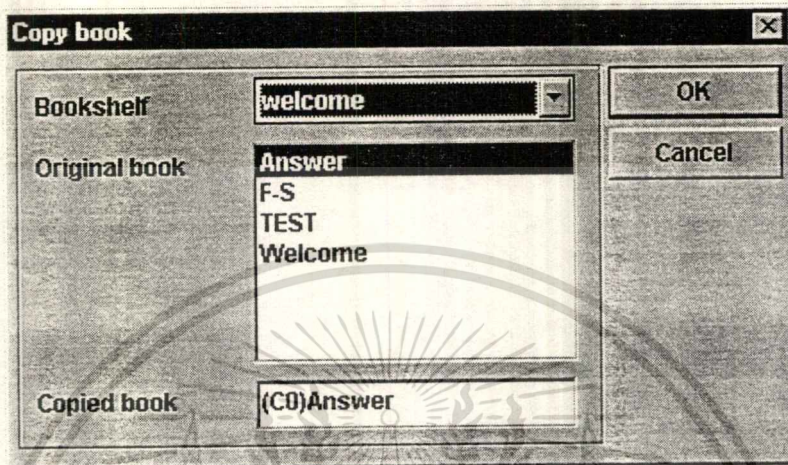
Dialog Box ในรูปที่ 2.29 ใช้เพื่อทำการย้ายหนังสือ



รูปที่ 2.29 การย้ายหนังสือ

15. Dialog Box to copy a Book

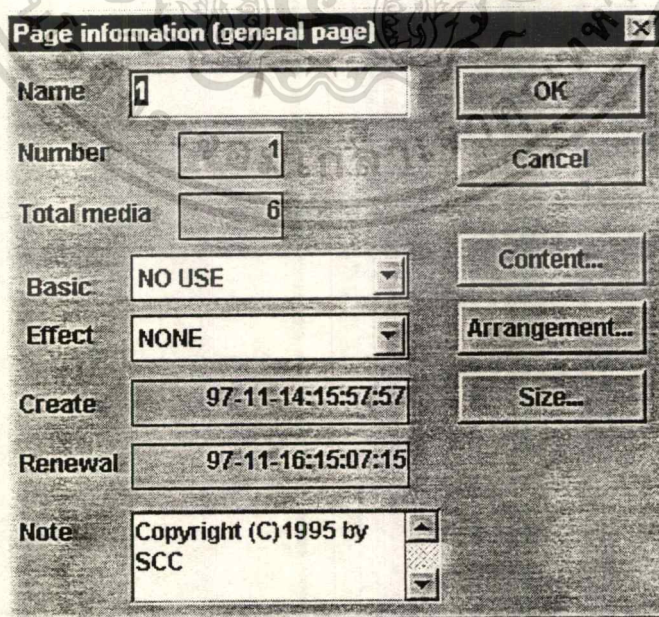
Dialog Box นี้ ใช้เพื่อทำการ copy หนังสือ



รูปที่ 2.30 การคัดลอกหนังสือ

16. Page Information (General Page) Dialog Box

Dialog Box ในรูปที่ 2.31 ใช้เพื่อกำหนดรายละเอียดหน้าของเนื้อหา

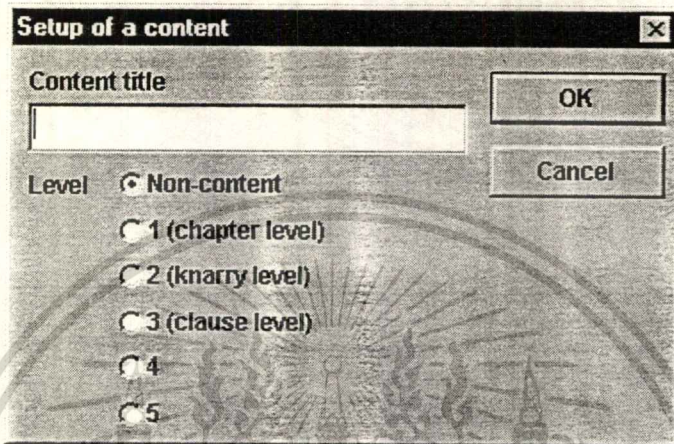


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องสมุดเท่านั้น เมื่อผู้ใช้เห็นข้อผิดพลาดประการใด กรุณาแจ้งให้ทราบโดยด่วน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. Dialog box to set up a Table of contents

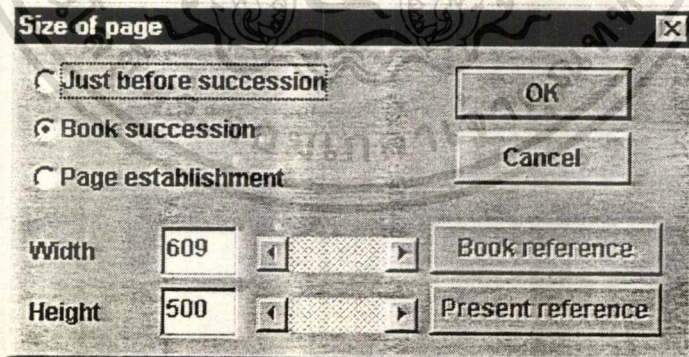
Dialog Box ในรูปที่ 2.32 ใช้เพื่อต้องการ set หน้า นั้น ๆ ให้อยู่ในรายการของหน้าสารบัญ



รูปที่ 2.32 การกำหนดหน้าสารบัญ

18. Dialog Box to set up Page Size

Dialog Box ในรูปที่ 2.33 ใช้เพื่อต้องการกำหนดขนาดของหน้า



รูปที่ 2.33 การกำหนดขนาดของหน้า

19. Page Information (Basic Page) Dialog Box

Dialog Box ในรูปที่ 2.34 ใช้เพื่อกำหนดรายละเอียดของหน้า Background

The dialog box titled "Page information (basic page)" has the following fields and values:

- Name: [Empty text box]
- Number: 3
- Total media: 5
- Create: 97-11-15:18:30:39
- Renewal: 97-11-22:21:01:28
- Note: Copyright (C)1995 by SCC

Buttons: OK, Cancel

รูปที่ 2.34 การกำหนดรายละเอียดของหน้า

2.6.7 Media information Dialog Box

1. Index Resistration Dialog Box

Dialog box ในรูปที่ 2.35 ใช้เพื่อกำหนดดัชนี

The dialog box titled "Registration of index" has the following fields and buttons:

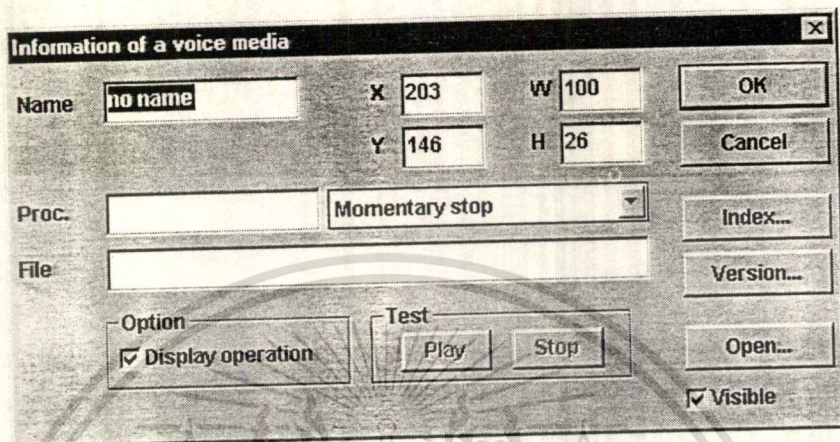
- Reading: [Empty text box]
- Index: [Empty text box]

Buttons: OK, Cancel

รูปที่ 2.35 การกำหนดดัชนี

2. Information Dialog Box for Vedio media

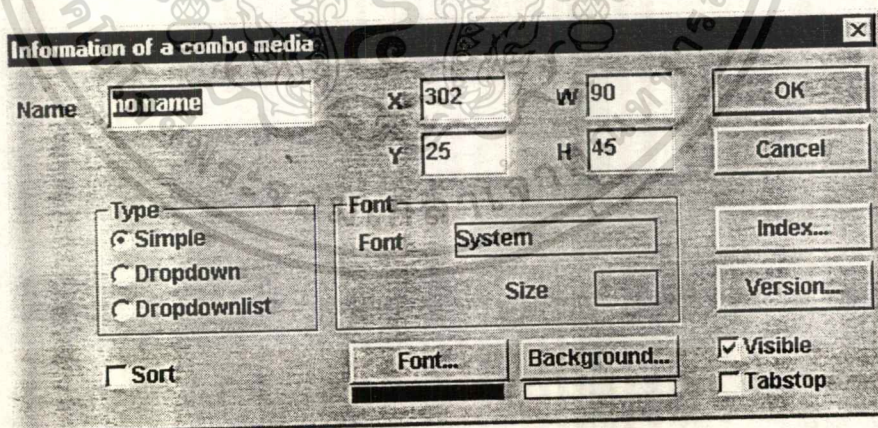
Dialog Box ในรูปที่ 2.36 ใช้เพื่อทำรูปภาพเคลื่อนไหว



รูปที่ 2.36 การทำรูปภาพเคลื่อนไหว

3. Information Dialog Box for Combo media

Dialog Box ในรูปที่ 2.37 ใช้เพื่อใช้ในการเลือกคำตอบ ของแบบทดสอบ



รูปที่ 2.37 การเลือกคำตอบในแบบฝึกหัด

4. Information Dialog Box for Index media

Dialog Box ในรูปที่ 2.38 ใช้เพื่อกำหนดรายละเอียดของ Index

รูปที่ 2.38 การกำหนดรายละเอียดของหน้าดัชนี

5. Information Dialog Box for Figure media

Dialog Box ในรูปที่ 2.39 ใช้เพื่อการวาดเกี่ยวกับเส้นต่าง ๆ รูปวงกลม วงรี ลูกศร

รูปที่ 2.39 การเรียกภาพมาประกอบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Information Dialog Box for timer media

dialog Box ในรูปที่ 2.40 ใช้เพื่อการตั้งเวลา

Information of a timer media

Name: X: Y: W: H:

Procedure: leftClick

Timer value: Min Sec 1/10sec

Type: Onece timer Cyclic timer

Visible

Buttons: OK, Cancel, Index..., Version...

รูปที่ 2.40 การตั้งเวลา

7. Information Dialog Box for Text media

Dialog Box ในรูปที่ 2.41 ใช้เพื่อการสร้างข้อความที่ต้องการ และสามารถกำหนดรูปแบบของตัวอักษร สี ขนาด ได้

Information of text media

Name: X: Y: W: H:

Procedure: leftClick

Indication option: Display frame No background color

Letter decoration: No Shadow White extraction

Arrangement: Left adjust Center adjust Right adjust

Font: Size:

Visible

Buttons: OK, Cancel, Index..., Version..., Font..., Background...

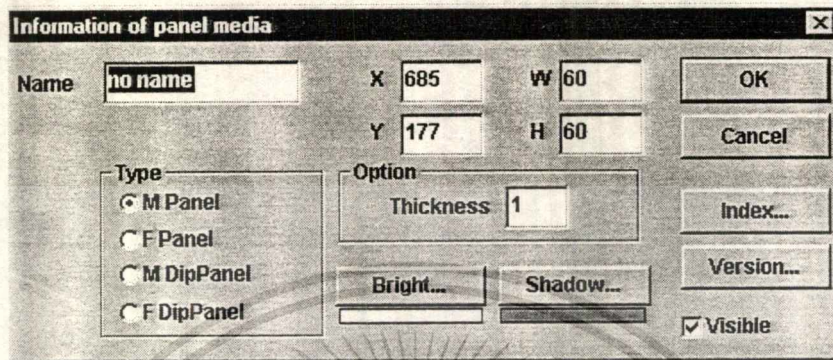
Input text:

รูปที่ 2.41 การสร้างข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Information Dialog Box for Panel Media

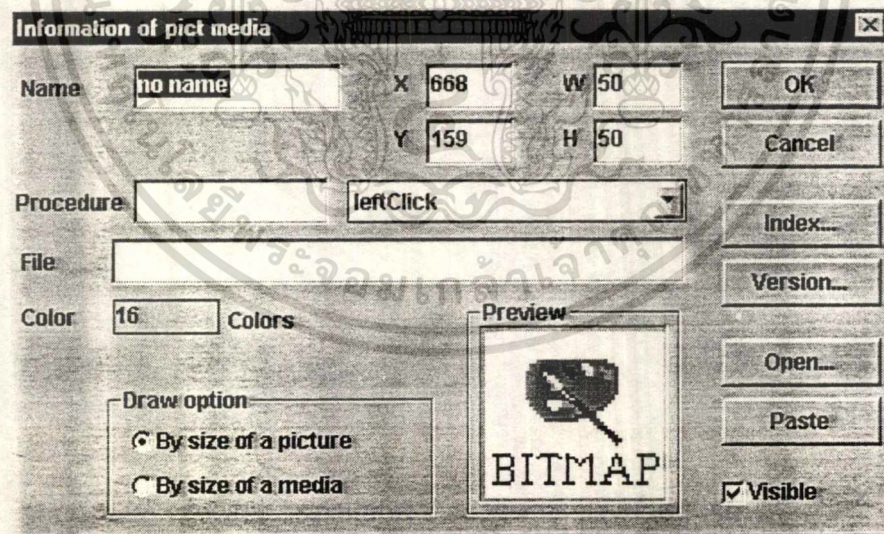
Dialog Box ในรูปที่ 2.42 ใช้เพื่อการสร้างปุ่ม ที่สามารถกำหนดรูปแบบของปุ่มนั้น ๆ



รูปที่ 2.42 การสร้างปุ่ม

9. Information Dialog Box for Pict Media

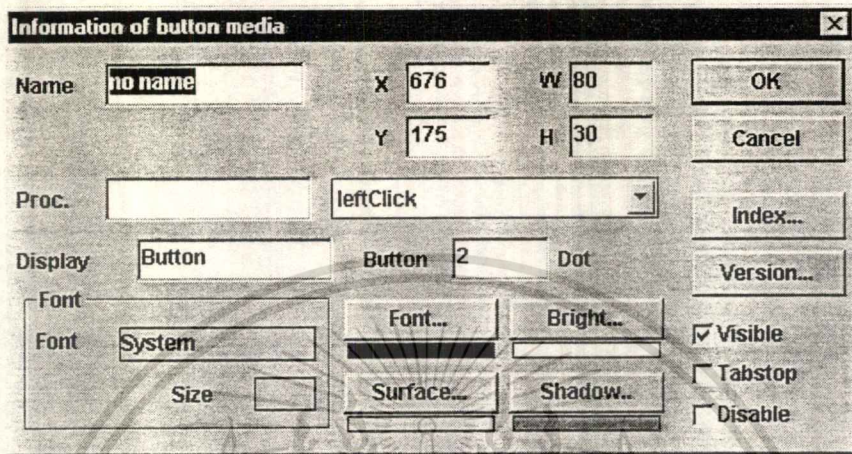
Dialog Box ในรูปที่ 2.43 ใช้เพื่อดีงรูปภาพ ที่ต้องการมาประกอบหนังสือ



รูปที่ 2.43 การดึงรูปมาประกอบ โปรแกรม

10. Information Dialog Box for Button Media

Dialog Box ในรูปที่ 2.44 ใช้เพื่อสร้างปุ่มที่สามารถกำหนดรูปแบบได้



รูปที่ 2.44 การสร้างปุ่มทำงานต่างๆ

บทที่ 3

การออกแบบ การทำงาน และการสร้าง

การออกแบบและการสร้างโปรแกรมช่วยสอนวิชา ดิจิตอล ซึ่งเป็นสื่อประกอบการเรียนการสอนในโครงการสอนระบบทางไกลแบบ 2 ทาง ซึ่งเป็นบทเรียนช่วยสอนที่สร้างขึ้นโดยโปรแกรม HURESS เน้นสื่อความหมายรูปแบบที่อ่านง่าย และในทุกๆบทจะประกอบด้วยคำอธิบายที่สำคัญหลักๆของแต่ละหัวข้อและรูปภาพที่เกี่ยวข้องในการอธิบาย ส่วนแบบฝึกหัดก็จะมีการถามตอบและมีการเช็คคำตอบระหว่างผู้สอนและผู้เรียนได้โดยผลการทำแบบฝึกหัดจะตรวจได้ด้วยอาจารย์ผู้สอน เทคนิคและวิธีการที่ใช้สร้างบทเรียนและแบบฝึกหัดโดยโปรแกรม HURESS ได้นำเสนอรวมในส่วนของการสร้างบทเรียนและแบบฝึกหัด

3.1 เครื่องมือที่เกี่ยวข้องในการออกแบบและการสร้าง

3.1.1 ทางด้านฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์ Pentium-100 NEC PowerMate V
- เครื่องคอมพิวเตอร์ Pentium-133 MHz
- หน่วยความจำหลัก 32 MB
- ฮาร์ดดิสก์ 1.2 GB
- Mouse
- Sound Card

3.1.2 ทางด้านซอฟต์แวร์ (Software)

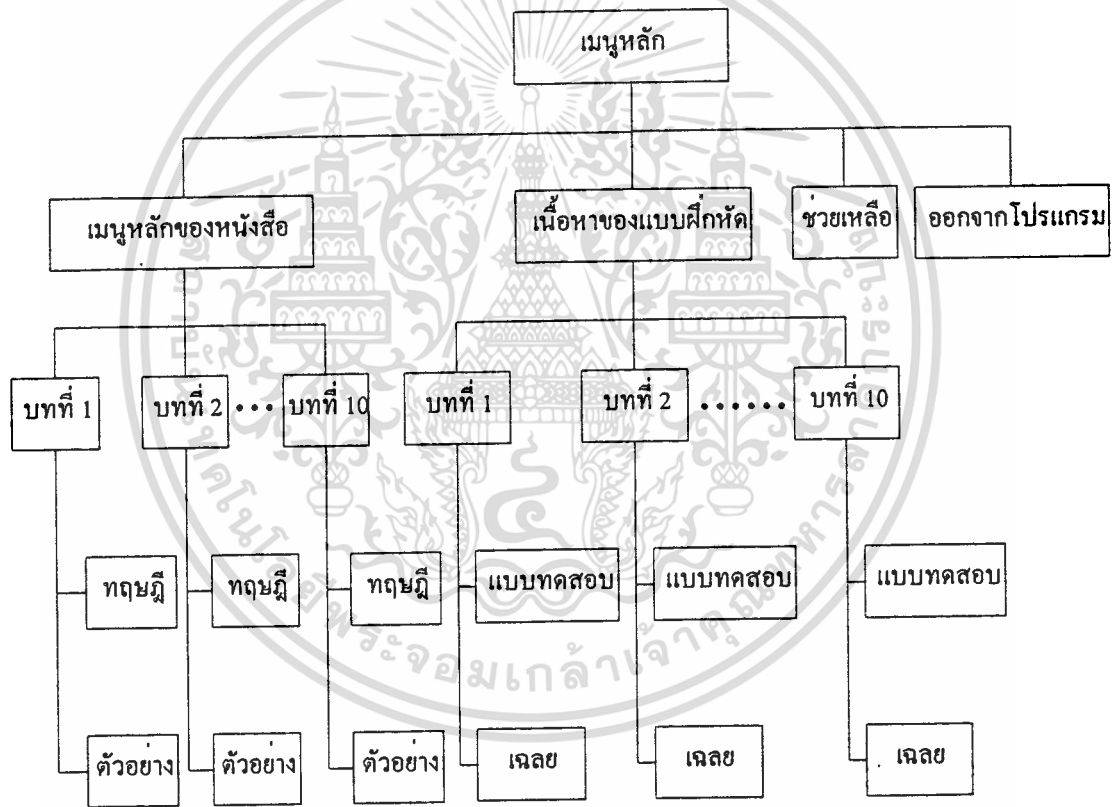
- โปรแกรม MS-WINDOWS NT version 3.5
- โปรแกรม MS-WINDOWS 95
- โปรแกรม HURESS
- โปรแกรม GIFCON
- โปรแกรม ACDSSee 95

3.2 ขั้นตอนในการออกแบบโปรแกรม

โปรแกรมช่วยสอนวิชา คณิตศาสตร์ ประกอบด้วยบทเรียนทั้งหมด 10 บทเรียน ซึ่งในแต่ละบทเรียนจะประกอบไปด้วย 2 ส่วนหลัก ๆ คือ

ส่วนที่หนึ่ง เป็นทฤษฎีของบทเรียนต่าง ๆ ซึ่งจะเป็นเนื้อหาหลัก ๆ ของแต่ละหัวข้อ และมีรูปภาพประกอบคำอธิบาย

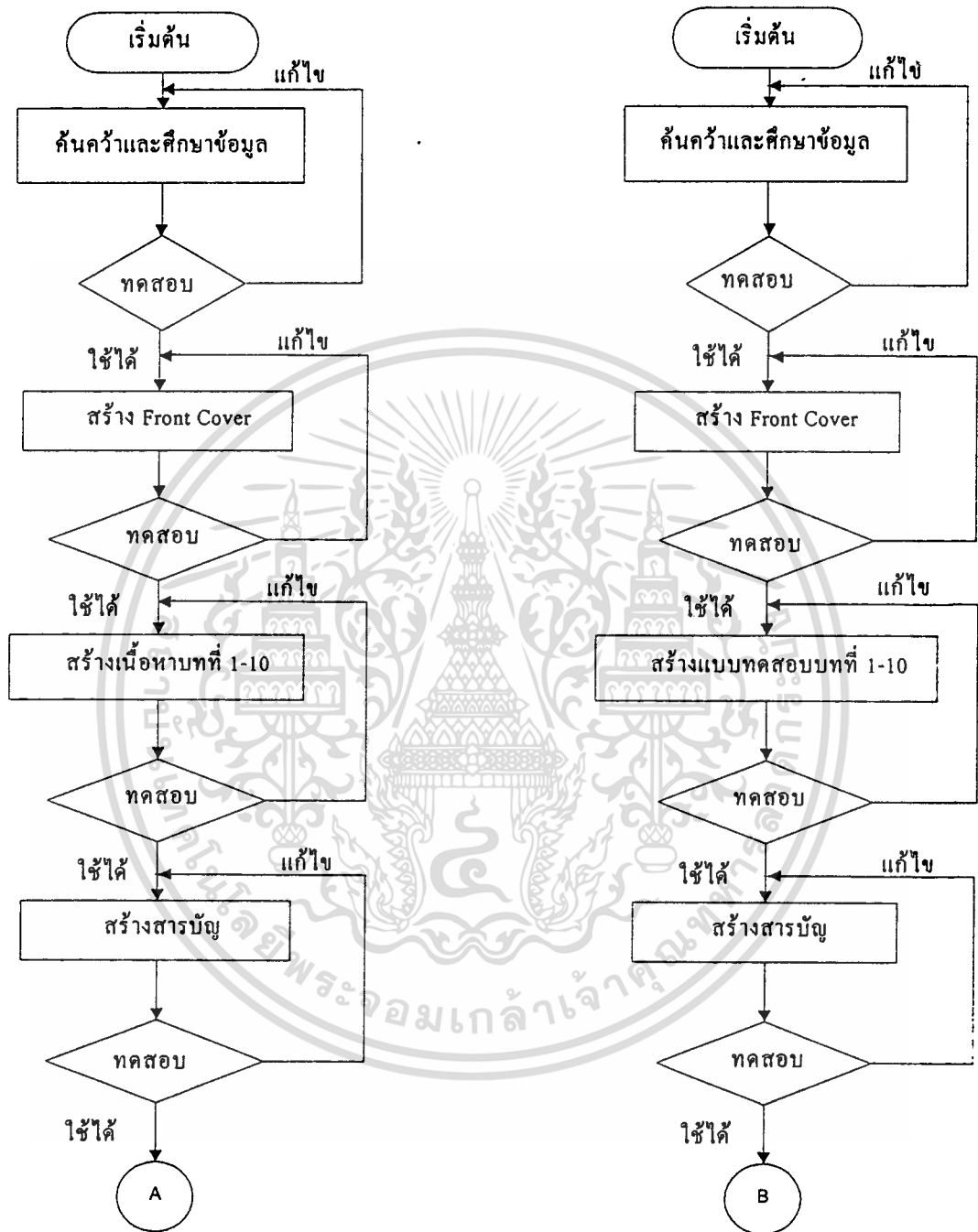
ส่วนที่สอง เป็นส่วนของแบบฝึกหัด ซึ่งผู้เรียนในแต่ละประเทศและอาจารย์ผู้สอนสามารถทำแบบฝึกหัดและตรวจคำตอบพร้อมกันได้



รูปที่ 3.1 ผังการทำงานของโปรแกรมช่วยสอนวิชาคณิตศาสตร์

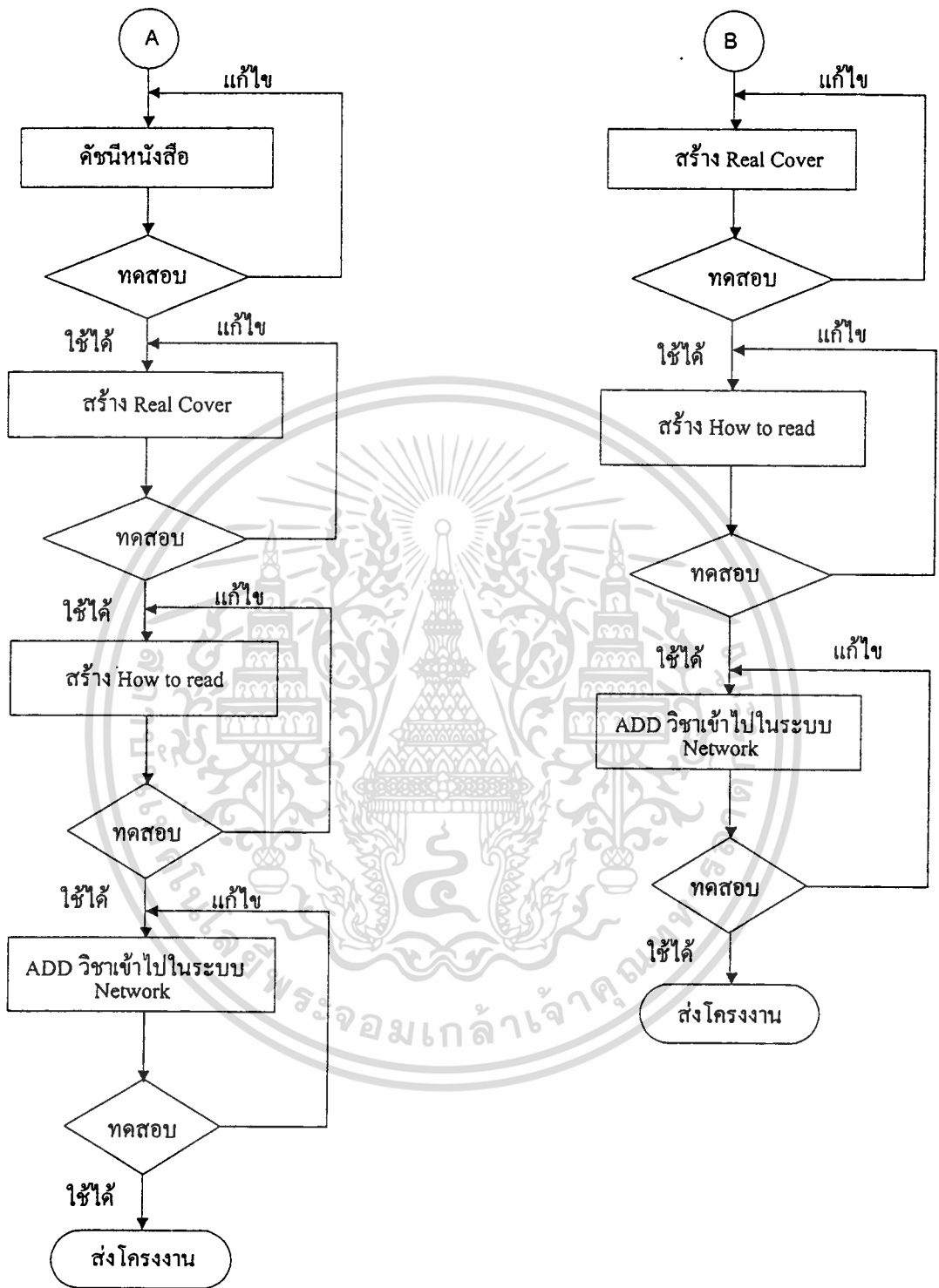
จากผังการทำงานของโปรแกรมที่กำหนดขึ้น ได้ออกแบบและสร้างส่วนต่าง ๆ ของผังการทำงาน โดยมีลำดับขั้นตอนการออกแบบและสร้างโปรแกรม ดังผังการทำงานในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ผังขั้นตอนการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

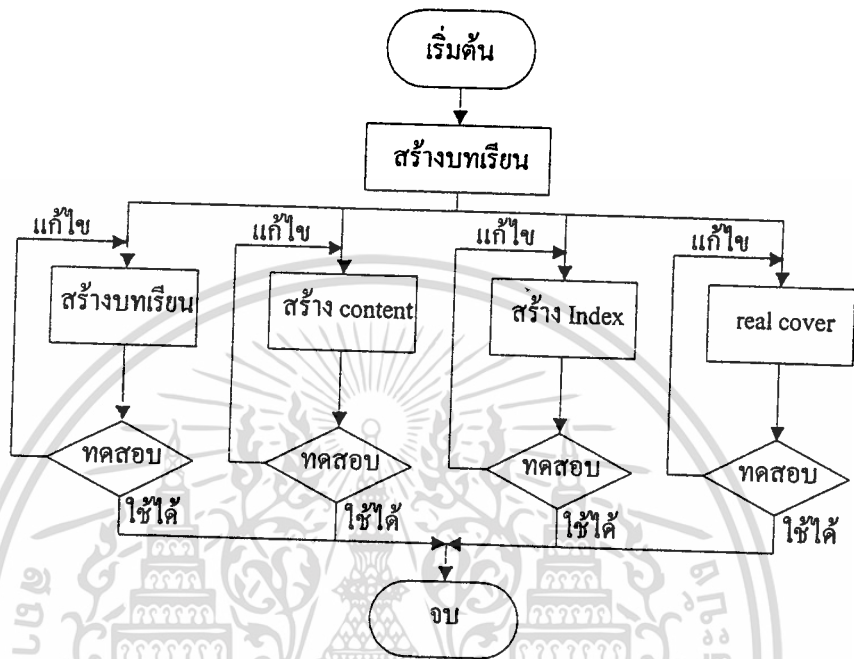


รูปที่ 3.2 (ต่อ) ผังขั้นตอนการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการสร้างบทเรียน

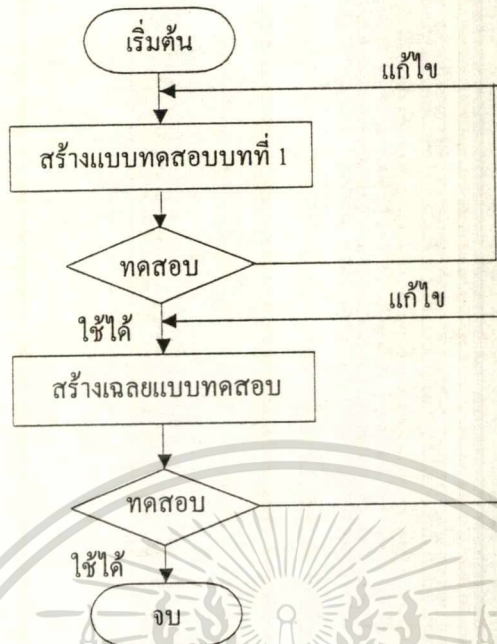
การสร้างบทเรียนด้วยโปรแกรม HURESS มีขั้นตอนการสร้างดังรูปที่ 3.3



รูปที่ 3.3 แผนผังการสร้างบทเรียน

3.4 ขั้นตอนการสร้างแบบฝึกหัด

การสร้างแบบฝึกหัดด้วยโปรแกรม HURESS มีขั้นตอนการสร้างดังรูปที่ 3.4

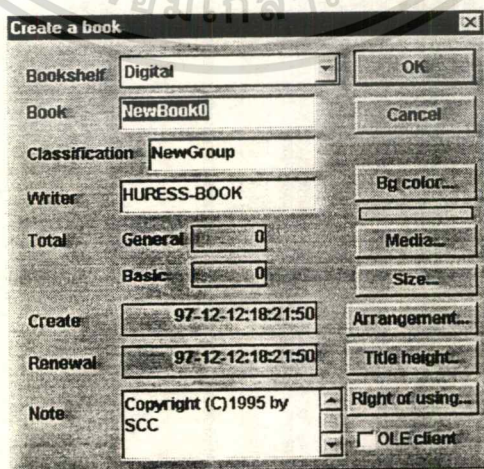


รูปที่ 3.4 แผนผังการสร้างแบบฝึกหัด

3.5 วิธีการสร้างส่วนของบทเรียนด้วยโปรแกรม HURESS

3.5.1 การสร้างหนังสือ

คลิกเมาส์ไปที่ Create and borrow a new book บนเมนู Bookshelf หน้าจอจะแสดง Dialog Box ดังรูปที่ 3.5



รูปที่ 3.5 เมนูการสร้างหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

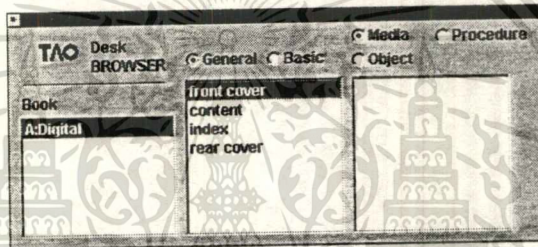
ทำการเปลี่ยนชื่อจาก NewBook0 เป็น Digital พร้อมกำหนดขนาดและพื้น Background การกำหนดรายละเอียดต่าง ๆ นั้นสามารถทำได้บนเมนู Dialog Box ทั้งหมด

3.5.2 การสร้าง Background (Basic page)

การสร้าง Background เป็นการสร้างหน้ารูปแบบพื้นฐานของส่วนต่างๆในหนังสือ ซึ่งทำได้ดังขั้นตอนต่อไปนี้

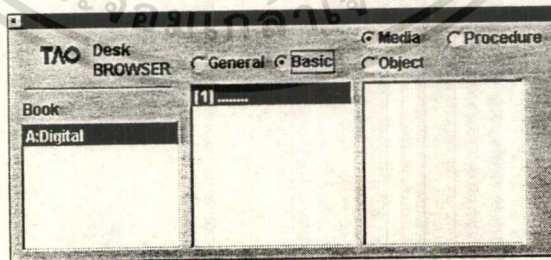
- 1.คลิกเมาส์ไปที่ Display Desk Browser บนเมนู Desk จะปรากฏ Dialog Box ดังรูปที่

3.6



รูปที่ 3.6 TAO Desk Browser

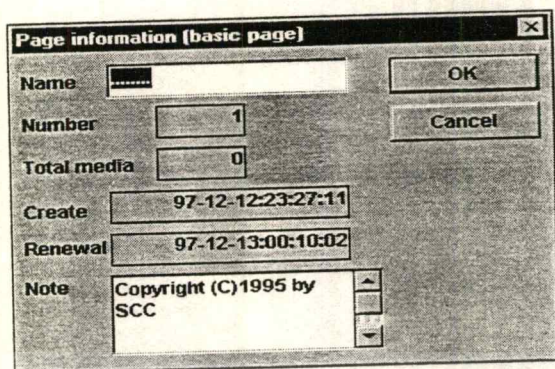
- 2.คลิกเมาส์ไปที่ Basic บน Desk Browser จากนั้นคลิกเมาส์ไปที่เมนู Book แล้วเลื่อนเมาส์ไปที่ Create a Page จะปรากฏผลดังรูปที่ 3.7



รูปที่ 3.7 การสร้างหน้า Background

3. ดับเบิ้ลคลิกที่ “([1].....)” บน Desk Browser เพื่อกำหนดชื่อของ Basic Page จะปรากฏผลดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

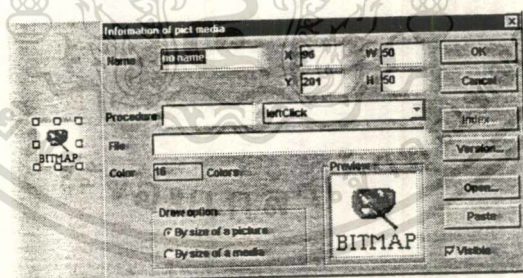


รูปที่ 3.8 การกำหนดชื่อหน้า Background

หลังจากนั้น สามารถนำมีเดีย (media) บน Operation of Media Selection Bar มาทำการสร้างรูปแบบที่ต้องการ

4. คลิกเมาส์ไปที่ Pict media บน Operation of Media Selection Bar แล้วคลิกขวาลงบนหน้าหนังสือ

5. ดับเบิ้ลคลิกที่ Pict media ที่นำมาวางบนหน้าหนังสือเพื่อเลือกรูปภาพที่จะใช้เป็น Background และกำหนดรายละเอียดต่างๆ ซึ่งจะปรากฏเป็น Dialog Box ดังรูปที่ 3.9



รูปที่ 3.9 การกำหนดรายละเอียดต่างๆของ Pict media

6. หลังจากกำหนดรายละเอียดต่างๆ แล้ว จะได้ Background ดังรูปที่ 3.10



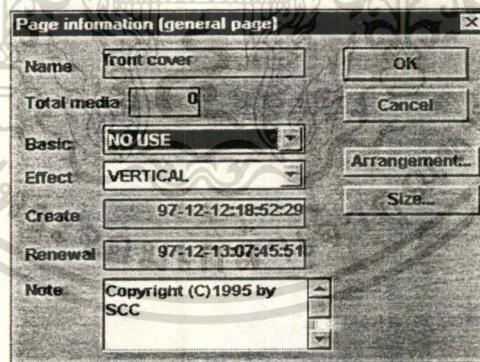
รูปที่ 3.10 หน้า Background ของหนังสือ

3.5.3 การสร้าง Front Cover

การสร้าง Front Cover เป็นการทำหน้าปกของหนังสือ ซึ่งทำได้ดังขั้นตอนต่อไปนี้

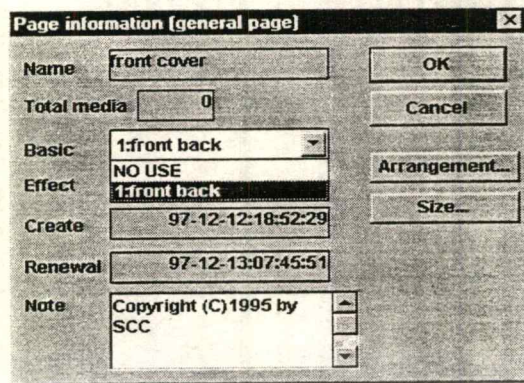
1. คลิกเมาส์ที่ General บน Desk Browser
2. ดับเบิลคลิกที่ Front Cover เพื่อทำการกำหนดรายละเอียดต่างๆ จะปรากฏ Dialog

Box เป็นดังรูปที่ 3.11



รูปที่ 3.11 การกำหนดรายละเอียดของ Front Cover

3. เลือก Background ที่สร้างไว้สำหรับหน้า Front Cover โดยคลิกเมาส์ไปที่ Basic บน Dialog Box แล้วทำการเลือกหน้า Background ที่ต้องการ ดังรูปที่ 3.12



รูปที่ 3.12 การเลือก Background ของหน้า Front Cover

หลังจากทำการกำหนดรายละเอียดและเลือก Background ที่ต้องการแล้วจะปรากฏผล
ดังรูปที่ 3.13

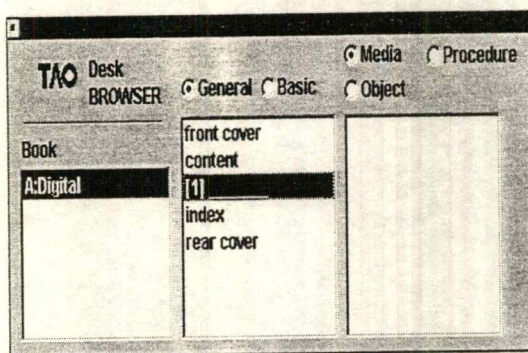


รูปที่ 3.13 หน้า Front Cover ของหนังสือ

3.5.4 การสร้างบทเรียน

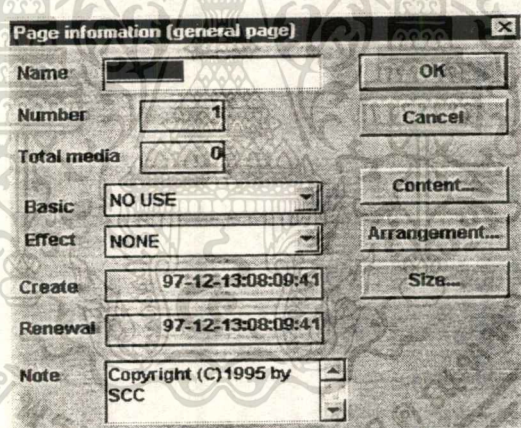
ในการสร้างบทเรียน ต้องทำการสร้างหน้าหนังสือขึ้นมาก่อน ซึ่งสามารถสร้างทำได้
ดังขั้นตอนต่อไปนี้

1. คลิกเมาส์ไปที่เมนู Book แล้วคลิกที่ Create a Page จะปรากฏผลดังรูปที่ 3.14



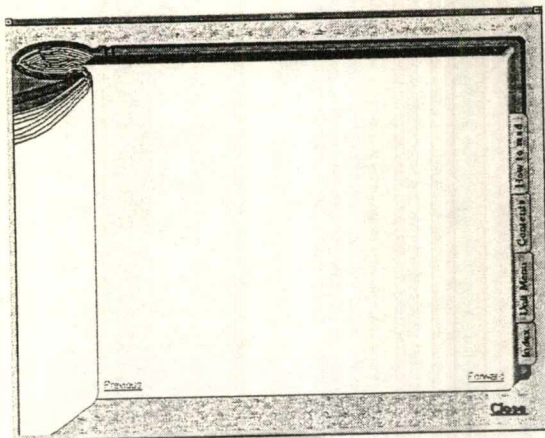
รูปที่ 3.14 การสร้างหน้าของหนังสือ

2. ดับเบิลคลิกที่ [1] ของ Desk Browser เพื่อเลือก Background ของบทเรียน และกำหนดรายละเอียดต่างๆของหน้า จะปรากฏ Dialog Box ดังรูปที่ 3.15



รูปที่ 3.15 การกำหนดรายละเอียดของหน้าบทเรียน

หลังจากกำหนดรายละเอียดต่างๆของหน้าบทเรียนแล้ว จะได้ผลดังรูปที่ 3.16



รูปที่ 3.16 หน้าของบทเรียน

เมื่อได้หน้าของบทเรียนแล้ว ถ้าต้องการเพิ่มหน้าของบทเรียนสามารถทำได้โดยการสร้างหน้าขึ้นมาใหม่ หรือทำการคัดลอกหน้าของบทเรียนที่ทำไว้แล้วโดยดับเบิ้ลคลิกขวาที่ชื่อของหน้านั้นๆบน Desk Browser

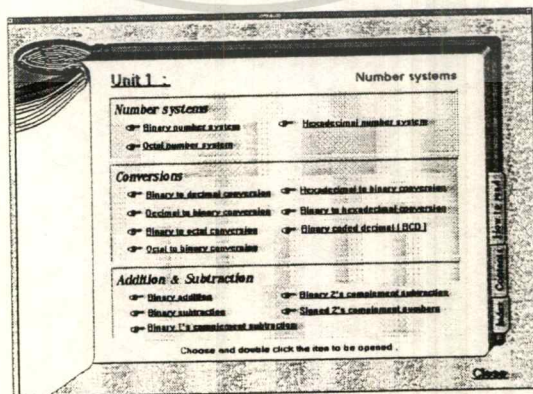
3.5.5 การสร้าง Unit menu

การสร้าง Unit menu ของแต่ละบทสามารถทำได้ดังขั้นตอนต่อไปนี้

1. ทำการสร้างหรือคัดลอกหน้าของหนังสือเพื่อนำไปสร้าง Unit menu ของแต่ละบท
2. เลือก Text media เพื่อเขียนหัวข้อของเนื้อหาในบทเรียนนั้นๆ โดยสามารถเลือกใช้

มีเดียตัวอื่นๆมาประกอบเพื่อให้เกิดความเหมาะสมสวยงามตามต้องการ

ผลที่ได้จากการสร้างเป็นดังรูปที่ 3.17



รูปที่ 3.17 หน้า Unit menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 การสร้าง Main menu

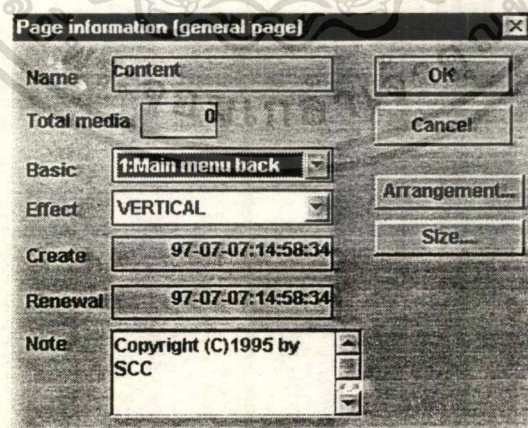
การสร้าง Main menu สามารถทำได้ดังขั้นตอนต่อไปนี้

1. คลิกเมาส์ไปที่ Content บน Desk Browser และคลิกเมาส์ไปที่ Content Media จากนั้นคลิกขวาวาง media ลงบนหน้า Content
2. ดับเบิลคลิกเมาส์ที่ content บน Desk Browser จะปรากฏ Dialog Box ดังรูปที่ 3.18



รูปที่ 3.18 Dialog Box ของ Main menu

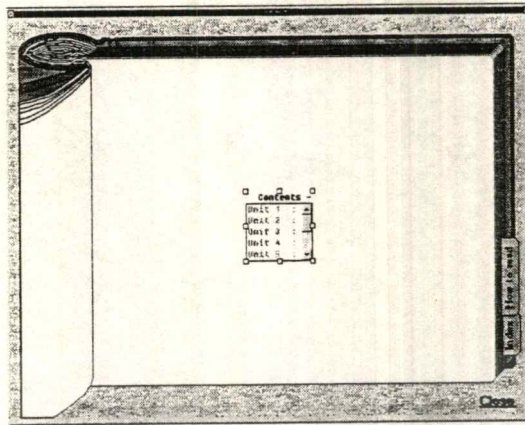
3. เลือก Background ที่สร้างไว้สำหรับหน้า Main menu โดยคลิกเมาส์ไปที่ Basic บน Dialog Box แล้วทำการเลือกหน้า Background ที่ต้องการ ดังรูปที่ 3.19



รูปที่ 3.19 การเลือก Background และกำหนดรายละเอียดต่างๆ ของ Main menu

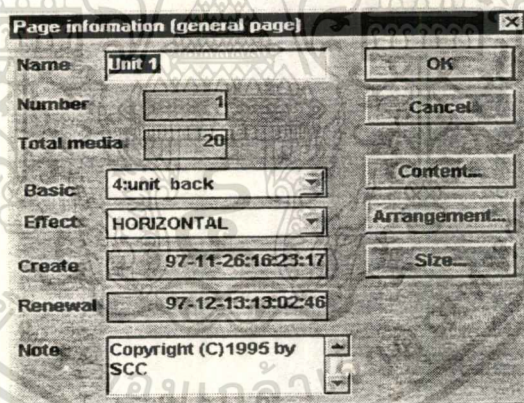
หลังจากกำหนดรายละเอียดต่างๆแล้วจะได้ผลดังรูปที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการวิจัยเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 หน้าของ Main menu

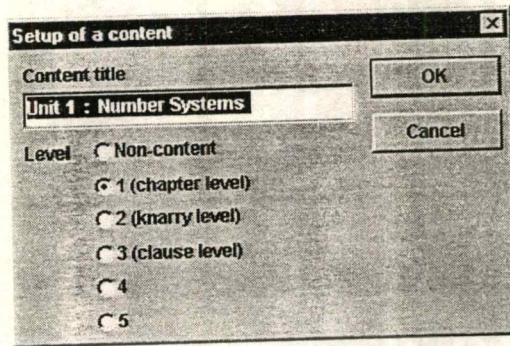
4. กลับไปที่หน้าส่วนของ Unit menu แล้วดับเบิลคลิกที่ชื่อของ Unit menu จะปรากฏ Dialog Box ดังรูปที่ 3.21



รูปที่ 3.21 Dialog Box ของ Main menu

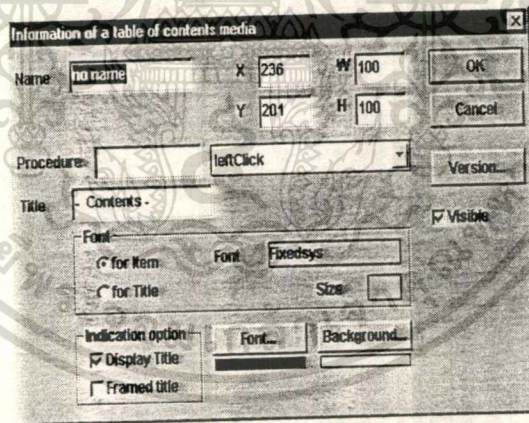
5.คลิกที่ Content บน Dialog Box ของ Main menu จะปรากฏ Dialog Box ดังรูปที่

3.22



รูปที่ 3.22 Dialog Box ของการทำ Main menu

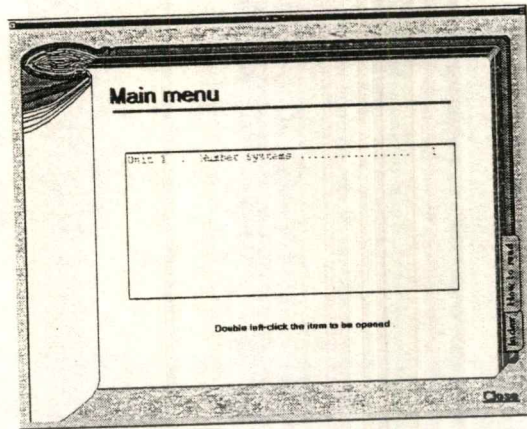
6. เขียนหัวข้อของ Main menu ที่ต้องการและเลือกรูปแบบการแสดงผลของ Main menu
7. ดับเบิลคลิกที่ Content media เพื่อกำหนดรายละเอียดต่างๆ ของ Main menu ซึ่งจะปรากฏ Dialog Box ดังรูปที่ 3.23



รูปที่ 3.23 Dialog Box ของ Content media

8. เมื่อกำหนดรายละเอียดต่างๆของ Main menu แล้ว ผลที่ได้จะเป็นดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



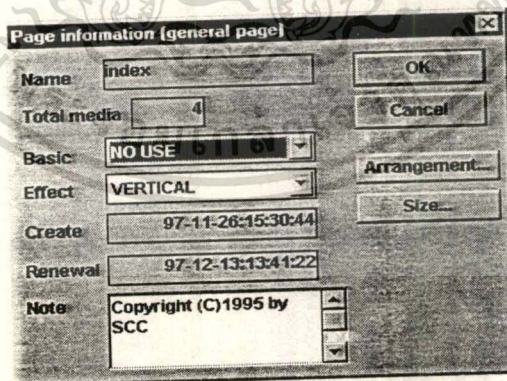
รูปที่ 3.24 หน้าของ Main menu

3.5.7 การสร้าง Index

การสร้างหน้า Index สามารถทำได้ตามขั้นตอนต่อไปนี้

1. คลิกเมาส์ไปที่ Index บน Desk Browser และคลิกเมาส์ไปที่ Index Media จากนั้นคลิกขวาวาง media ลงบนหน้า Index
2. ดับเบิ้ลคลิกเมาส์ไปที่ Index บน Desk Browser จะปรากฏ Dialog Box ดังรูปที่

3.25

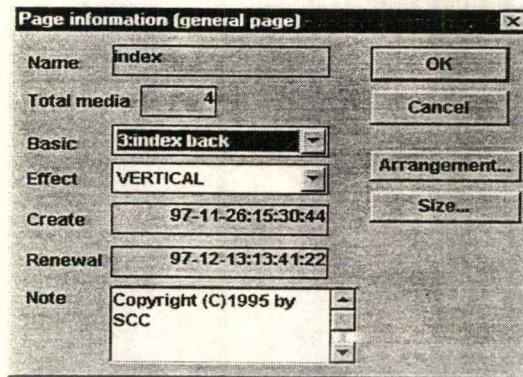


รูปที่ 3.25 Dialog Box ของ Index

3. เลือก Background ที่สร้างไว้สำหรับหน้า Index โดยคลิกเมาส์ไปที่ Basic บน

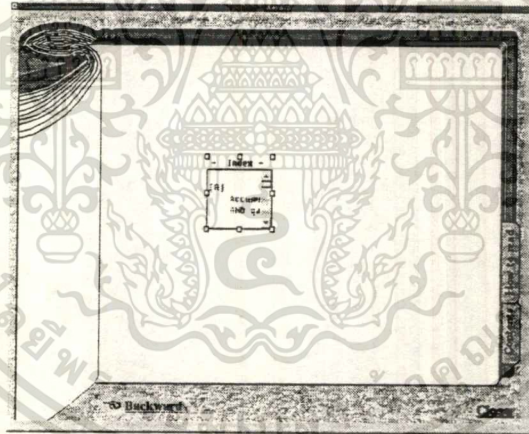
Dialog Box แล้วทำการเลือกหน้า Background ที่ต้องการ ดังรูปที่ 3.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 การเลือก Background และกำหนดรายละเอียดต่างๆ ของ Main menu

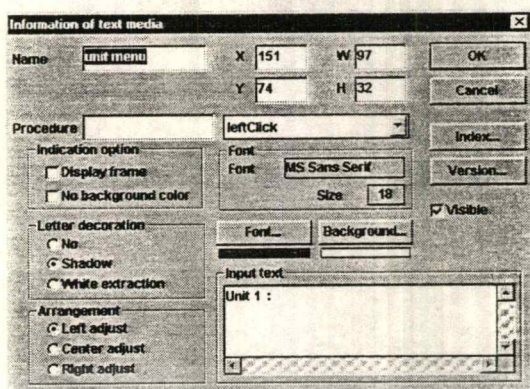
หลังจากกำหนดรายละเอียดต่างๆแล้วจะได้ผลดังรูปที่ 3.27



รูปที่ 3.27 หน้าของ Index

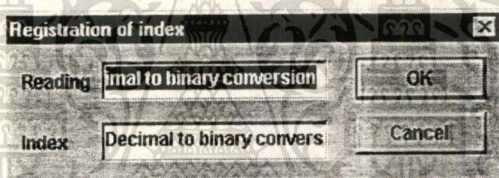
4. กลับไปที่หน้าส่วนของเนื้อหา แล้วดับเบิลคลิกที่ชื่อของหัวข้อแต่ละหัวข้อ จะปรากฏ Dialog Box ดังรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 Dialog Box ของ Main menu

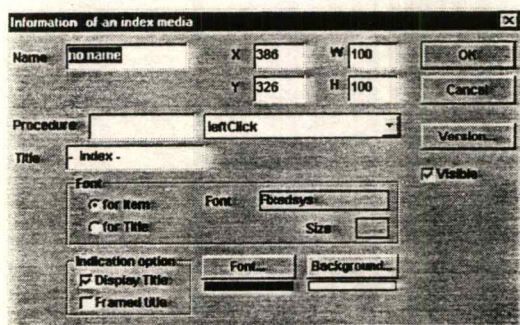
5.คลิกที่ Index บน Dialog Box ของ Index จะปรากฏ Dialog Box ดังรูปที่ 3.29



รูปที่ 3.29 Dialog Box ของการทำ Index

6. เขียนหัวข้อของ Index ที่ต้องการและเลือกรูปแบบการแสดงผลของ Index

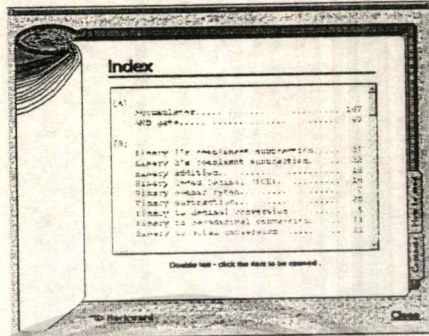
7. ดับเบิ้ลคลิกที่ Index media เพื่อกำหนดรายละเอียดต่างๆ ของ Index ซึ่งจะปรากฏ Dialog Box ดังรูปที่ 3.30



รูปที่ 3.30 Dialog Box ของ Index media

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.30 Dialog Box ของ Index media นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อกำหนดรายละเอียดต่างๆของ Index แล้ว ผลที่ได้จะเป็นดังรูปที่ 3.31

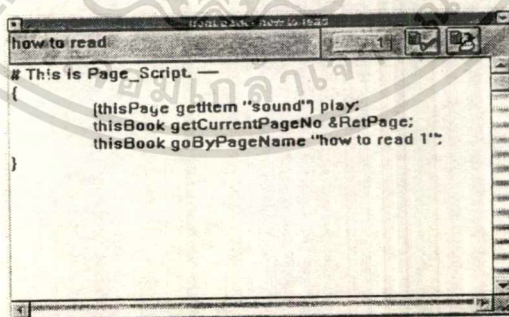


รูปที่ 3.31 หน้าของ Index

3.5.8 การสร้างลำดับการทำงานของหนังสือ

ลำดับการทำงานของหนังสือ จะใช้งานในส่วนของมีเดีย Hypertext การที่จะให้มีเดียแต่ละตัวทำงานนั้น จะต้องเขียน โปรแกรมย่อยให้กับมีเดีย Hypertext แต่ละตัว ซึ่งในหนังสือจะประกอบด้วย มีเดีย Hypertext ดังต่อไปนี้

มีเดีย [How to read] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.32

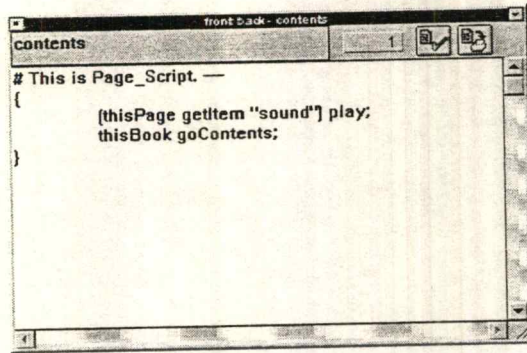


รูปที่ 3.32 โปรแกรมย่อยของมีเดีย How to read

มีเดีย [Contents หรือ Main menu] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่

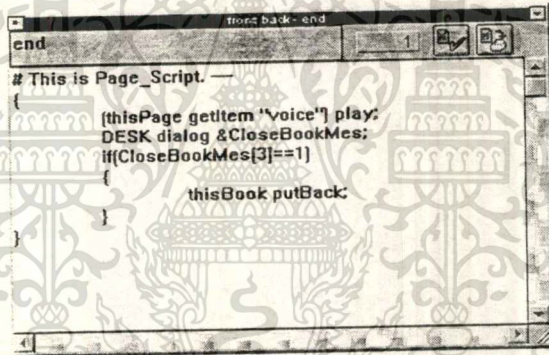
3.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



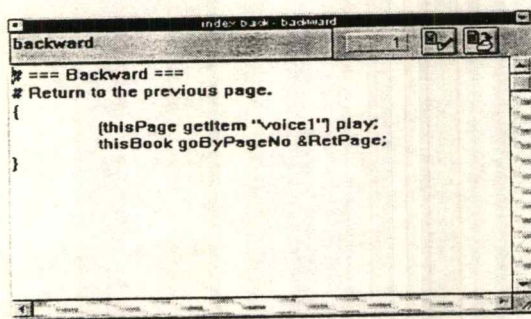
รูปที่ 3.33 โปรแกรมย่อยของมีเดีย Contents หรือ Main menu

มีเดีย [Close] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.34



รูปที่ 3.34 โปรแกรมย่อยของมีเดีย [Close]

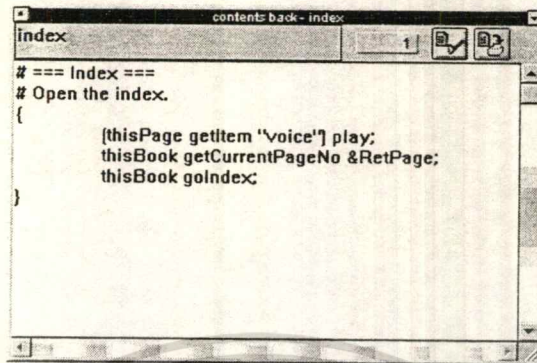
มีเดีย [Backward] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.35



รูปที่ 3.35 โปรแกรมย่อยของมีเดีย [Backward]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น มิใช่เพื่อเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีเดีย [Index] โปรแกรมย่อยการทำงานของมีเดียเป็นดังรูปที่ 3.36

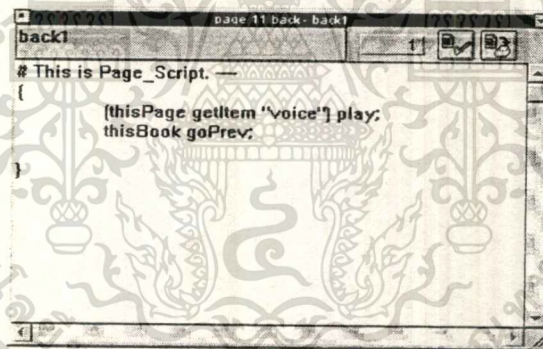


```

index
# === Index ===
# Open the index.
{
    [thisPage getItem 'voice'] play;
    thisBook getCurrentPageNo &RetPage;
    thisBook goIndex;
}
  
```

รูปที่ 3.36 โปรแกรมย่อยของมีเดีย [Index]

มีเดีย [Previous] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.37

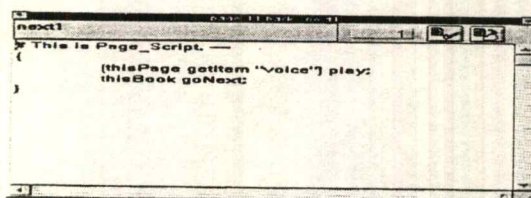


```

back1
# This is Page_Script. ---
{
    [thisPage getItem 'voice'] play;
    thisBook goPrev;
}
  
```

รูปที่ 3.37 โปรแกรมย่อยของมีเดีย [Previous]

มีเดีย [Forward] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.38



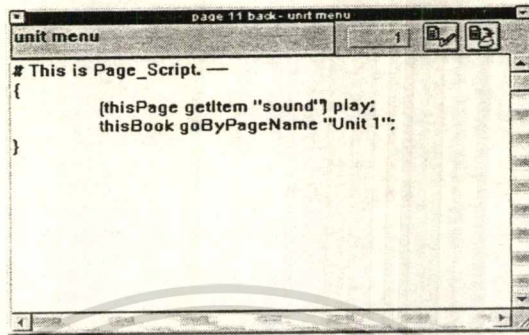
```

next1
# This is Page_Script. ---
{
    [thisPage getItem 'voice'] play;
    thisBook goNext;
}
  
```

รูปที่ 3.38 โปรแกรมย่อยของมีเดีย [Forward]

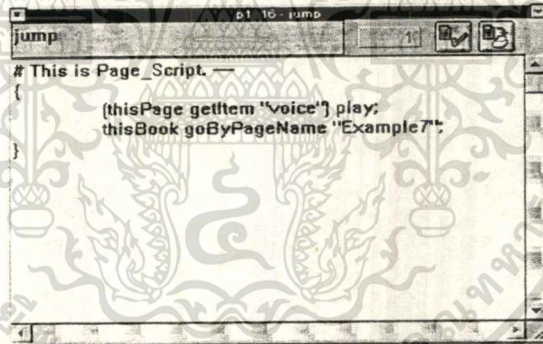
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีเดีย [Unit menu] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.39



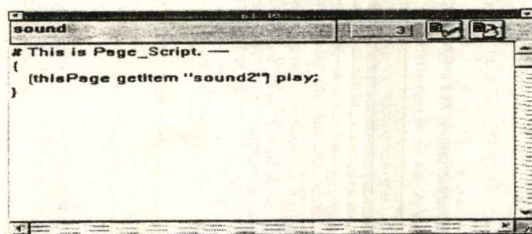
รูปที่ 3.39 โปรแกรมย่อยของมีเดีย [Unit menu]

มีเดีย [Example] โปรแกรมย่อยการทำงานของมีเดีย เป็นดังรูปที่ 3.40



รูปที่ 3.40 โปรแกรมย่อยของมีเดีย [Example]

โปรแกรมย่อยของการใช้ มีเดียเสียง [Voice media] เป็นไปดังรูปที่ 3.41

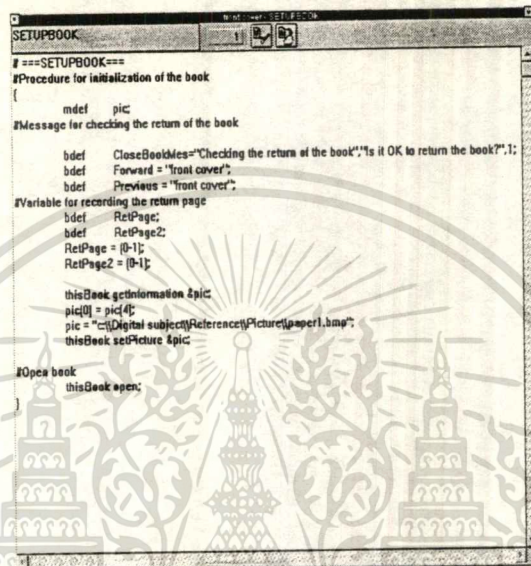


รูปที่ 3.41 โปรแกรมย่อยของ [Voice media]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.9 การเขียนโปรแกรมเพื่อ SETUP การทำงานของหนังสือ

หนังสือจะทำงานได้นั้น ต้องเขียนโปรแกรมย่อย SETUPBOOK ให้กับหนังสือที่หน้า front cover การเขียนโปรแกรม SETUPBOOK เป็นดังรูปที่ 3.42

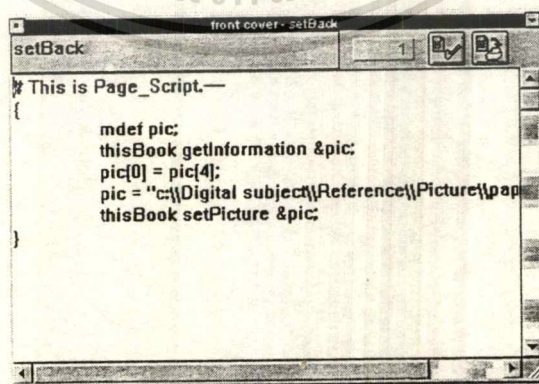


```

SETUPBOOK
#==SETUPBOOK==
#Procedure for initialization of the book
{
  mdef pic
  #Message for checking the return of the book
  bdef CloseBookMes="Checking the return of the book! Is it OK to return the book?":1;
  bdef Forward = "front cover";
  bdef Previous = "front cover";
  #Variable for recording the return page
  bdef RetPage;
  bdef RetPage2;
  RetPage = [0-1];
  RetPage2 = [0-1];
  thisBook getInformation &pic;
  pic[0] = pic[4];
  pic = "c:\\Digital subject\\Reference\\Picture\\paper1.bmp";
  thisBook setPicture &pic;
  #Open book
  thisBook open;
}
  
```

รูปที่ 3.42 โปรแกรมย่อยของ SETUPBOOK

นอกจากโปรแกรม SETUPBOOK แล้ว ยังต้องมี โปรแกรมย่อยของการเซ็ท Background ด้วย หนังสือจึงจะทำงานได้สมบูรณ์ โปรแกรมย่อยของการเซ็ท Background เป็นดังรูปที่ 3.43



```

setBack
# This is Page_Script.—
{
  mdef pic;
  thisBook getInformation &pic;
  pic[0] = pic[4];
  pic = "c:\\Digital subject\\Reference\\Picture\\pap
  thisBook setPicture &pic;
}
  
```

รูปที่ 3.43 โปรแกรมย่อยของการเซ็ท Background

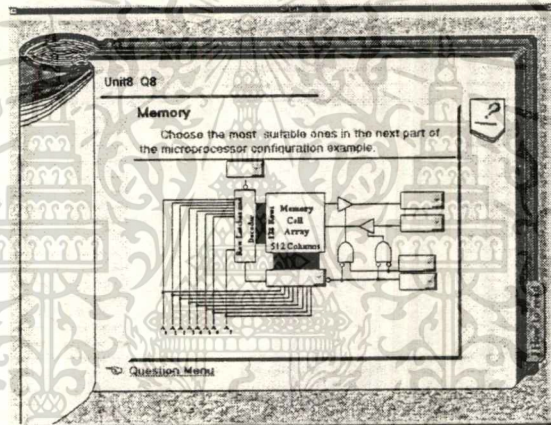
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วิธีการสร้างส่วนของแบบฝึกหัดด้วยโปรแกรม HURESS

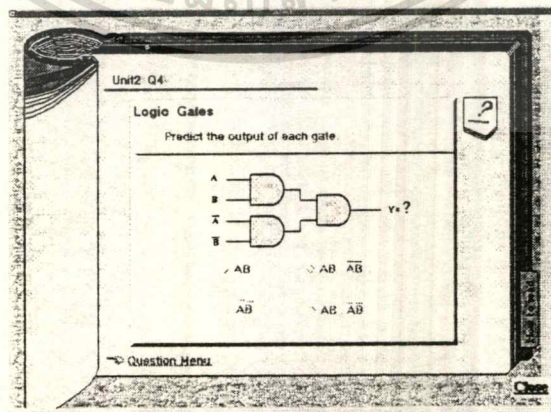
การสร้างส่วนของแบบฝึกหัดจะต้องสร้างหนังสือขึ้นมาสองเล่มด้วยกัน เล่มหนึ่งเป็นส่วนของแบบฝึกหัดและอีกเล่มหนึ่งเป็นส่วนเฉลย โดยทั้งสองเล่มจะมีขั้นตอนในการสร้าง Background, Front Cover, Unit menu, Main menu เหมือนกับการสร้างบทเรียน

3.6.1 การสร้างแบบฝึกหัด

การสร้างแบบฝึกหัดสามารถใช้มีเดียใน Menu selection Bar เพื่อสร้างส่วนต่างๆ โดยใช้ Text media ส่วนการเลือกคำตอบจะใช้ Combo media สำหรับการตอบแบบเลือกตอบและใช้ Radio media สำหรับคำตอบแบบ choice ดังแสดงในรูปที่ 3.44 และรูปที่ 3.45 ตามลำดับ



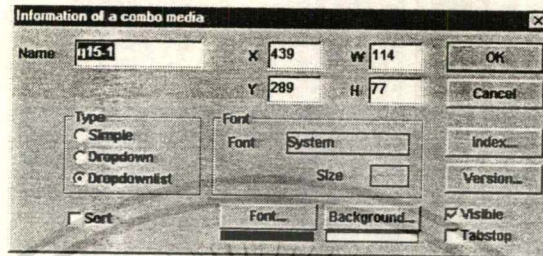
รูปที่ 3.44 การใช้ Combo media สร้างแบบฝึกหัด



รูปที่ 3.45 การใช้ Radio media สร้างแบบฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

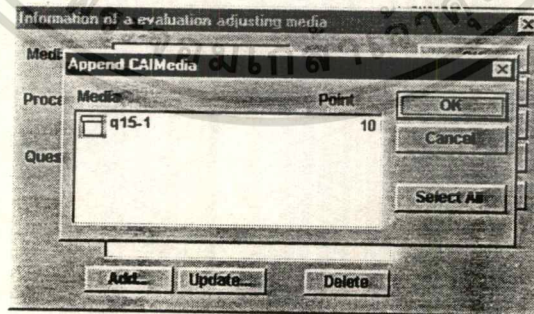
Combo media ที่ใช้ จะใช้แบบ Dropdownlist โดยสามารถเลือกรูปแบบได้โดยการ
ดับเบิลคลิกที่ตัว Combo media ที่นำมาวางบนหน้าหนังสือแล้วตั้งชื่อพร้อมเลือกรูปแบบการ
แสดงผลดังรูปที่ 3.46



รูปที่ 3.46 การเลือกรูปแบบแสดงผล

โดยการแสดงตัวเลือกจะต้องส่งผ่าน Evaluation revised media โดยมีขั้นตอนดังนี้

1. เลือก Evaluation revised media บน Menu selection Bar แล้วนำมาวางบนหน้าหนังสือ
2. ดับเบิลคลิกที่ Evaluation revised media เพื่อตั้งชื่อเลือกโปรแกรมใช้งานและเลือกชื่อ Combo media หรือ Radio media ตัวที่เราต้องการส่งผ่านคำตอบไปให้มัน โดยการเลือก Add ดังรูปที่ 3.47



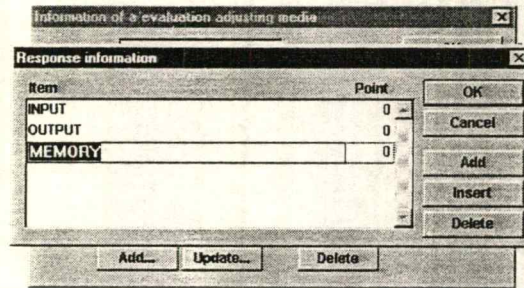
รูปที่ 3.47 การเลือก Combo media ผ่าน Evaluation revised media

3. ดับเบิลคลิกที่ชื่อของ Combo media หรือ Radio media ตัวที่เราต้องการสื่อผ่านตัว

เลือกให้มัน จะปรากฏ Dialog Box เพื่อ Add ตัวเลือกเข้าไป ดังรูปที่ 3.48

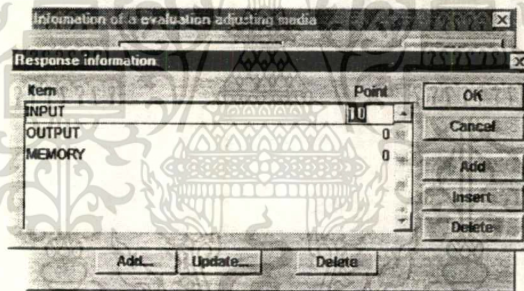
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า เฉพาะผู้จัดทำเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.48 การ Add ตัวเลือกไปยัง Combo Media หรือ Radio Media

4. ตัวเลือกตัวใดเป็นคำตอบที่ถูกต้อง ให้ใส่ค่า Point เท่ากับ 10 เพื่อใช้ส่งผ่านค่าในโปรแกรม ส่วนคำตอบที่ผิดใส่ค่า Point เท่ากับ 0 ดังรูปที่ 3.49

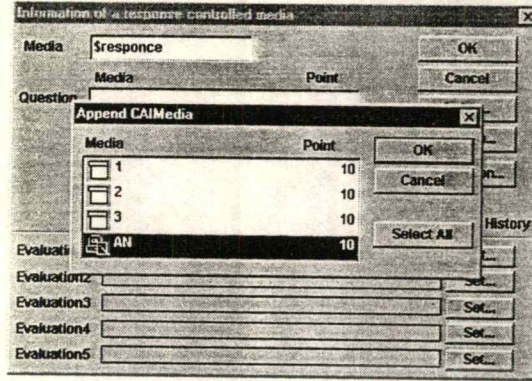


รูปที่ 3.49 การใส่ค่า Point ให้กับตัวเลือก

หลังจากนั้น คลิกที่ Visible เพื่อซ่อนการแสดงของ Evaluation Revised Media

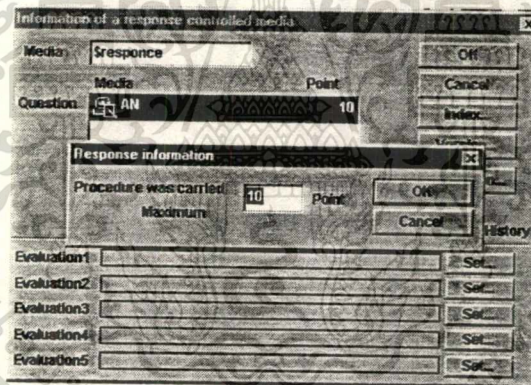
5. จากนั้นเลือก Response Controlled Media เพื่อส่งผ่านค่าจาก Evaluation Revised Media ไปยังหนังสืออีกเล่มหนึ่ง ซึ่งเป็นส่วนของการเฉลย จะนำค่าไปเช็คในโปรแกรมเพื่อเลือกการแสดงผลหน้าของการเฉลยคำตอบออกมา ซึ่งจะมีทั้งหมด 28 รูปแบบ แบ่งเป็น คำตอบถูก 14 รูปแบบและคำตอบผิด 14 รูปแบบ เมื่อแสดงว่าคำตอบถูกหรือผิดแล้ว จะมีโปรแกรมย่อย เรียกการแสดงผลของคำอธิบายคำตอบที่ถูกต้องขึ้นมา เราจะต้องตั้งชื่อให้กับ Response Controlled Media และเลือก Evaluation Revised Media ที่เราส่งผ่านค่าตัวเลือกให้กับมัน โดยการ Add เข้าไป ดังรูปที่ 3.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



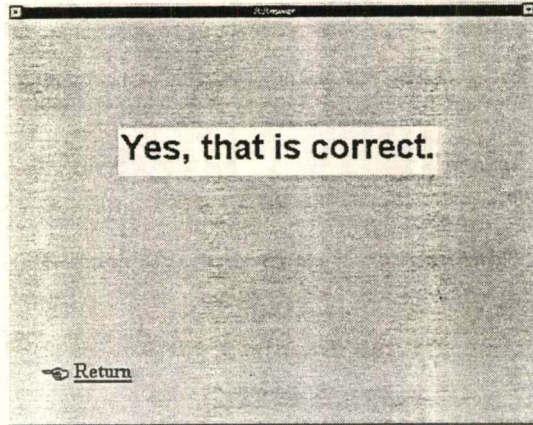
รูปที่ 3.50 การ Add Evaluation revised media ใน Response controlled media

6. ดับเบิลคลิกที่ชื่อของ Evaluation revised media จะปรากฏ Dialog Box เพื่อทำการใส่ค่า point ให้ใส่ค่าเท่ากับ 10 ดังรูปที่ 3.51

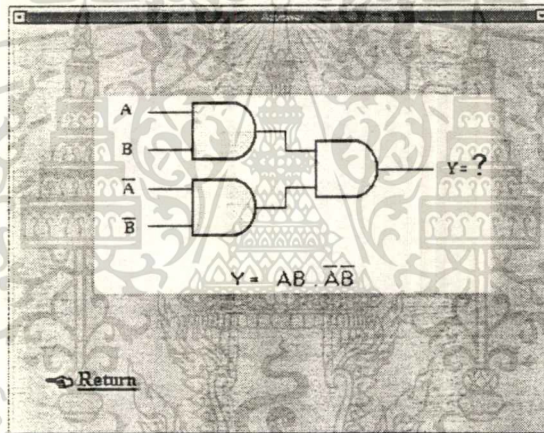


รูปที่ 3.51 การใส่ค่า point ให้กับ Evaluation revised media

7. ไปยังหนังสืออีกเล่ม แล้วทำการสร้างแบบของการเฉลยและคำอธิบายที่ถูกต้อง ดังตัวอย่างรูปที่ 3.52 และรูปที่ 3.53 ตามลำดับค่าที่ส่งผ่านจากโปรแกรม ของหนังสือแบบฝึกหัด จะดีกว่าจะให้แสดงเฉลยรูปแบบใดและคำอธิบายที่ถูกต้อง ของคำถามข้อใดออกมา



รูปที่ 3.52 หน้าเฉลย

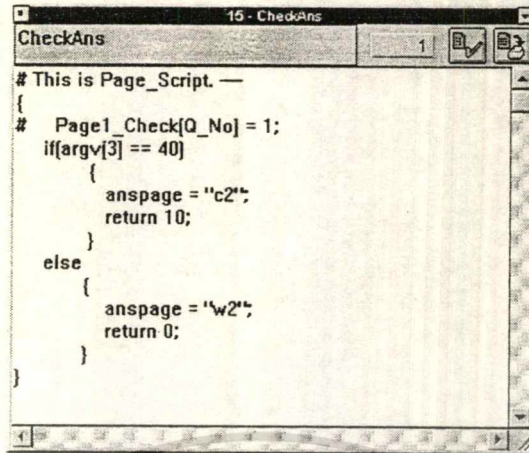


รูปที่ 3.53 หน้าคำอธิบายที่ถูกต้อง

3.6.2 โปรแกรมในส่วนของการสร้างแบบฝึกหัด

โปรแกรมในส่วนของการสร้างแบบฝึกหัดจะประกอบด้วย โปรแกรมในการเช็คคำตอบและเลือกแสดงคำตอบ, โปรแกรมเซตอัฟเลือกหน้าอธิบายคำตอบที่ถูกต้อง, โปรแกรมเซตอัฟหนังสือและโปรแกรมเรียกใช้หนังสือเฉลย ส่วนโปรแกรมอื่นๆสามารถได้กล่าวไว้ในส่วนของการสร้างบทเรียนแล้ว

1. โปรแกรมเช็คคำตอบและเลือกแสดงคำตอบเป็นดังรูปที่ 3.54



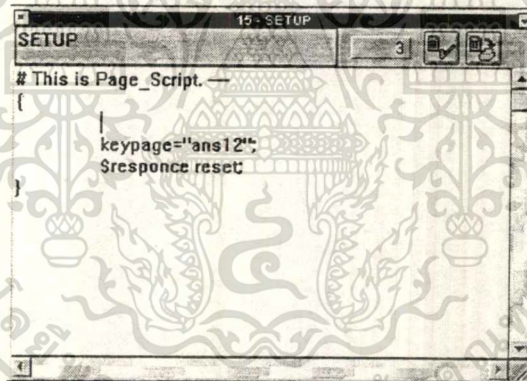
```

CheckAns
# This is Page_Script. —
{
#   Page1_Check[Q_No] = 1;
  if[argv[3] == 40]
    {
      anspage = "c2";
      return 10;
    }
  else
    {
      anspage = "w2";
      return 0;
    }
}

```

รูปที่ 3.54 โปรแกรมเช็คคำตอบและเลือกแสดงคำตอบ

2. โปรแกรมเซตอัพเลือกหน้าอธิบายคำตอบที่ถูกต้องเป็นดังรูปที่ 3.55



```

SETUP
# This is Page_Script. —
{
  keypage="ans12";
  $response reset;
}

```

รูปที่ 3.55 โปรแกรมเซตอัพเลือกหน้าอธิบายคำตอบที่ถูกต้อง

3. โปรแกรมเซตอัปหนังสือเป็นดังรูปที่ 3.56

```

SETUPBOOK
==== SETUPBOOK ====
# Procedure for initialization of the book
(
  gdef Common;
  mdef pic;

  # Message for checking the return of the book
  bdef CloseBookMes="Checking the return of book", "Is it OK to

  # Variable for recording the return page
  bdef MenuPageName;
  bdef anspage="w1";
  bdef keypage="ans1";
  bdef RetPage;
  bdef RetPage2;
  RetPage = (0-1);
  RetPage2 = (0-1);

  thisBook getInformation &pic;
  pic[0] = pic[4];
  pic = "c:\\Huresse\\Library\\Sample\\welcome\\Test\\mesbg.bmp";
  thisBook setPicture &pic;

  # Open book
  thisBook open;
)

```

รูปที่ 3.56 โปรแกรมเซตอัปหนังสือ

4. โปรแกรมเรียกใช้หนังสือเฉลยเป็นดังรูปที่ 3.57

```

answer
# This is Page_Script.
{
  sresponce evaluate;
}

```

รูปที่ 3.57 โปรแกรมเรียกใช้หนังสือเฉลย

บทที่ 4

การทดลอง และผลการทดลอง

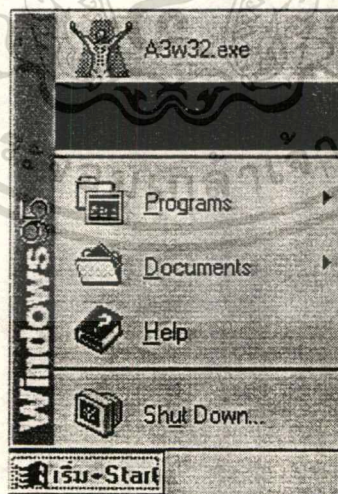
การใช้งานและทดสอบโปรแกรมช่วยสอนวิชาดิจิตอลนั้น จะทำการทดสอบการใช้โปรแกรมในแต่ละส่วนตามเมนูหลักของหนังสือ โดยในส่วนนี้จะแบ่งการทดสอบโปรแกรมออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของเนื้อหาและส่วนของแบบฝึกหัด

จากนั้นได้มีการทดสอบการใช้งานได้จริงในรูปแบบการเชื่อมโยงในระบบเครือข่าย ในส่วนนี้ได้ทดลองนำโปรแกรม HURESS มาทำการติดตั้งในระบบ Network อื่น ๆ และเพิ่มวิชาที่สร้างเข้าไปในระบบ เพื่อทดสอบโปรแกรม

4.1 การเข้าสู่โปรแกรม HURESS

1. เปิดเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server
2. เปิดเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client
3. นำเมาส์ไปคลิกที่ปุ่ม Start เลือกเมนู Programs แล้วรันโปรแกรม HURESS BOOK

ดังรูปที่ 4.1



รูปที่ 4.1 การเข้าสู่โปรแกรม HURESS

4. ป้อน User name เพื่อเข้าสู่โปรแกรม ดังรูปที่ 4.2

รูปที่ 4.2 การป้อน User name

5. เมื่อเข้าสู่โปรแกรม Huress แล้วจะปรากฏรายชื่อของวิชา ในแต่ละวิชาจะมีทั้งส่วนของบทเรียนและแบบทดสอบ ซึ่งเราสามารถเลือกได้โดยเลือกคลิกที่ Evaluation Material หรือ Reference Material ดังรูปที่ 4.3

รูปที่ 4.3 การเลือกวิชาเรียน

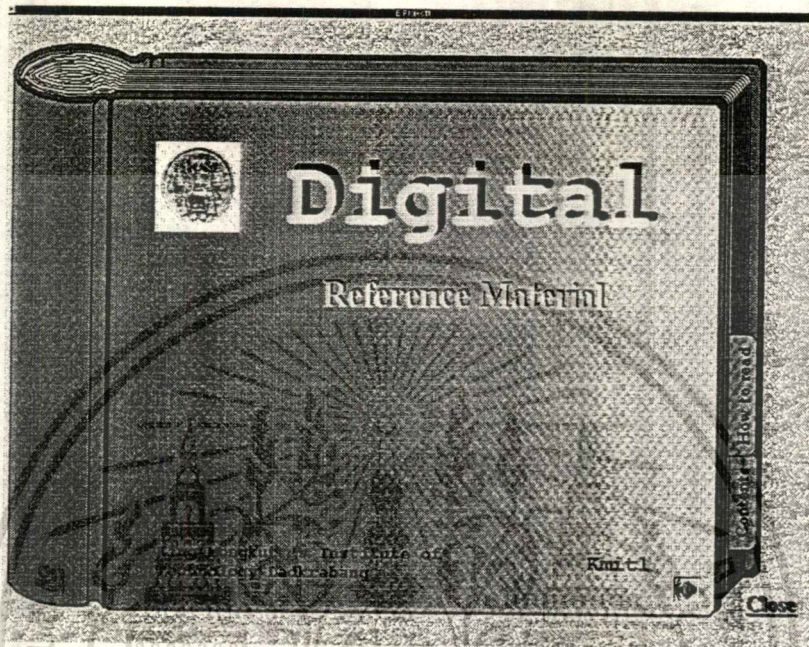
4.2 การใช้งานของโปรแกรม

4.2.1 ส่วนของเนื้อหา

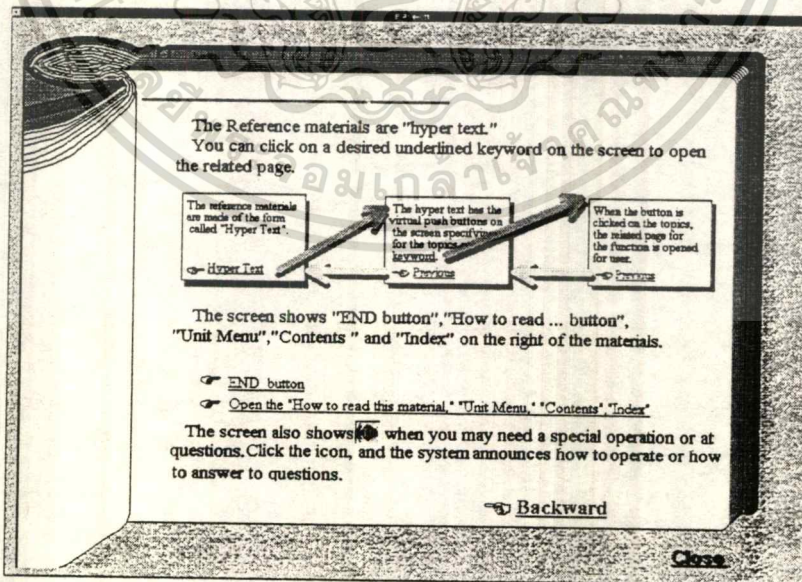
1. เมื่อเข้าสู่เนื้อหาจะปรากฏส่วนของ Front Cover ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลื่อนเมาส์ไปคลิกที่ How to read (ซึ่งเป็นข้อความแบบ hypertext) จะปรากฏหน้าจอดังรูปที่ 4.5 ซึ่งเป็นคำอธิบายการใช้งานโปรแกรม



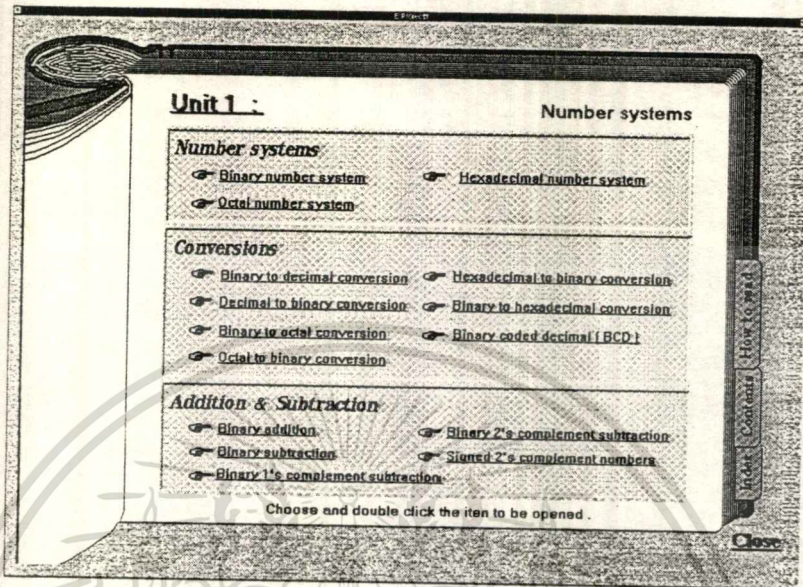
รูปที่ 4.4 Front Cover



รูปที่ 4.5 How to read

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

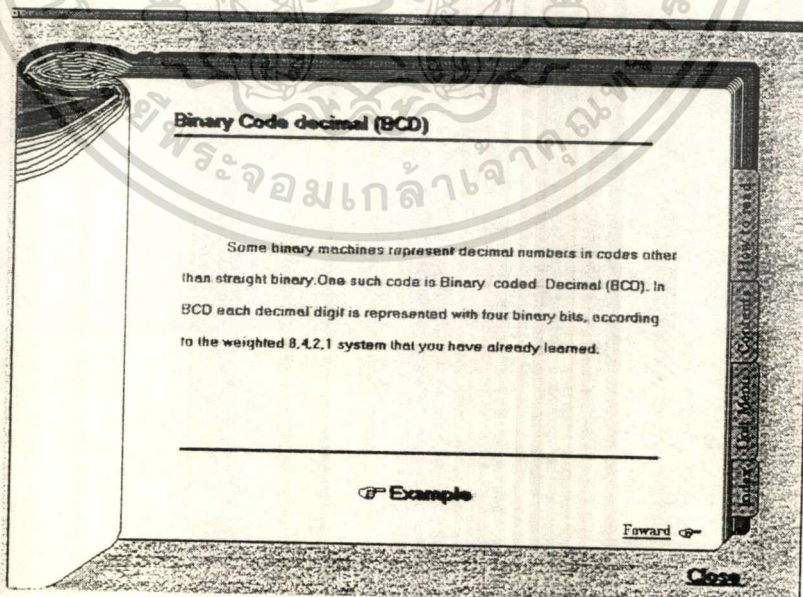
3. เลื่อนเมาส์ไปคลิกที่ Unit Menu จะแสดงรายชื่อของบทเรียนขึ้นมาดังรูปที่ 4.6



รูปที่ 4.6 Unit menu

4. เลื่อนเมาส์ไปคลิกยังบทเรียนที่ต้องการจะแสดงเนื้อหาของบทเรียนนั้นๆ ดังรูปที่

4.7



รูปที่ 4.7 เนื้อหาของบทเรียนและการไปหน้าตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

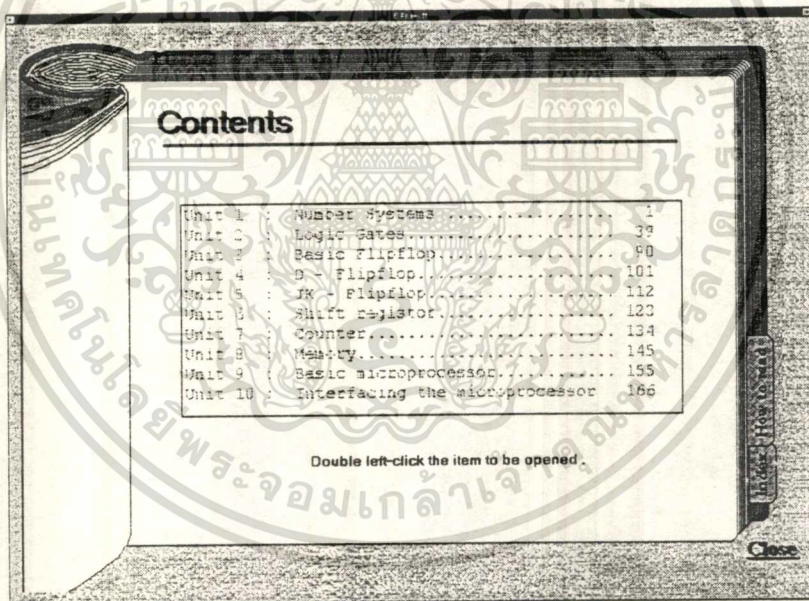
5. จากรูปที่ 4.7 สามารถกลับไปข้างหน้า Unit Menu, Contant, Idex และ How to Read ได้โดยเลื่อนเมาส์ไปคลิกที่ปุ่มนั้น

6. การกลับไปข้างหน้าเดิมได้โดยเลื่อนเมาส์ไปคลิกที่ "Previous" ส่วนการไปหน้าถัดไปให้คลิกที่ "Forward"

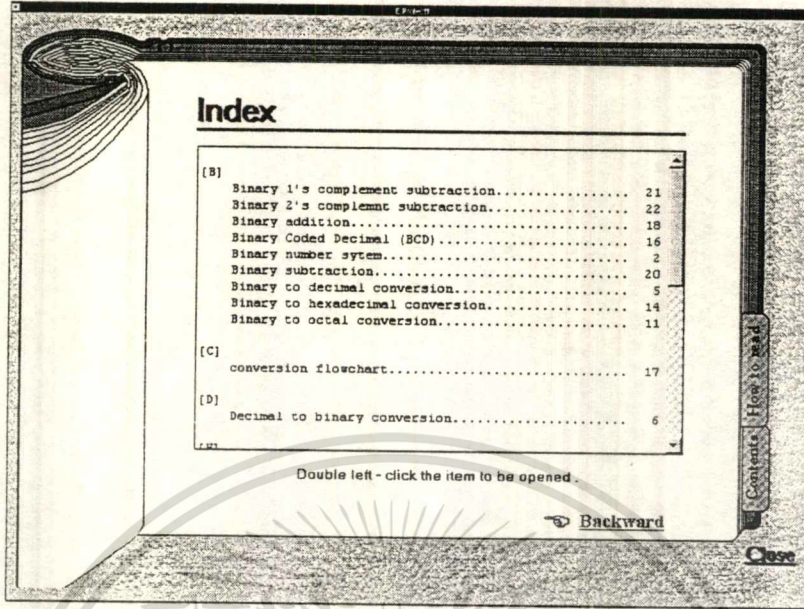
7. เมื่อต้องการดูตัวอย่าง สามารถทำได้โดยการเลื่อนเมาส์ไปคลิกที่ "Example" ดังรูปที่ 4.7

8. เมื่อทำการเลื่อนเมาส์ไปคลิกที่ "Content" จะปรากฏหน้าจอดังรูปที่ 4.8 และการใช้งานทำได้โดยการดับเบิลคลิกบทเรียนที่ต้องการ

9. เมื่อทำการเลื่อนเมาส์ไปคลิกที่ "Index" จะปรากฏหน้าจอดังรูปที่ 4.9 และการใช้งานทำได้โดยการดับเบิลคลิกคำที่ต้องการเพื่อที่จะไปหน้าที่มีรายละเอียดของคำนั้น

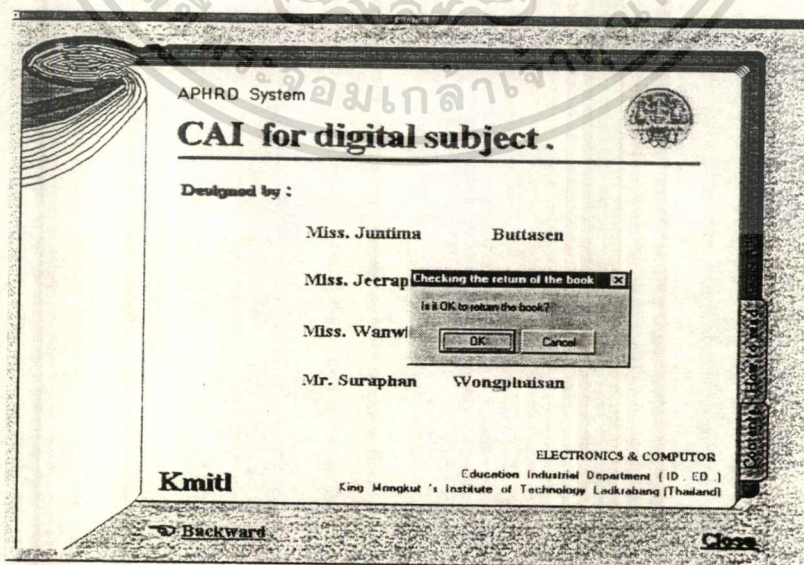


รูปที่ 4.8 การใช้งานหน้าสารบัญ



รูปที่ 4.9 การใช้งานหน้าต่างนี้

10. การออกจากโปรแกรมทำได้โดยการเลื่อนเมาส์ไปคลิกที่ " Close " ที่อยู่ตรงมุมล่างด้านขวาดังรูปที่ 4.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.10 การออกจากโปรแกรม อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

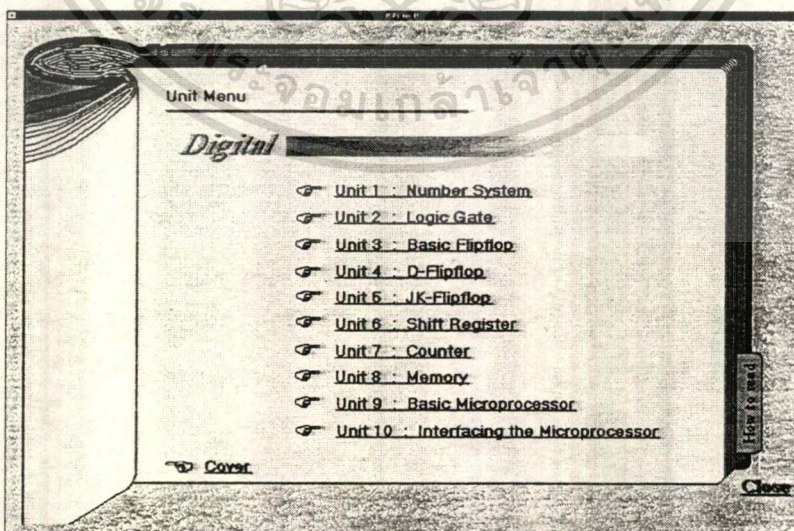
4.2.2 ส่วนของแบบทดสอบ

1. เมื่อเข้าสู่แบบทดสอบจะปรากฏหน้าจอหลัก ดังรูปที่ 4.11



รูปที่ 4.11 หน้าจอหลักของส่วนแบบทดสอบ

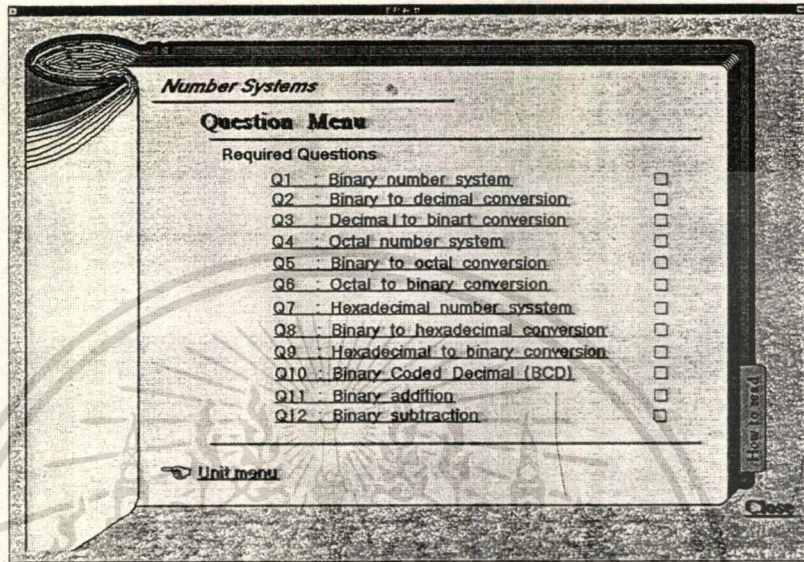
2. คลิกที่ปุ่ม Question Menu เพื่อเข้าสู่หน้าจอหลักหัวข้อแบบทดสอบดังรูปที่ 4.12



รูปที่ 4.12 หน้าจอหลักของหัวข้อแบบทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกบทเรียนที่จะทำการทดสอบจะเข้าสู่หน้าจอ Question Menu ของบทเรียนนั้นๆ ดังรูปที่ 4.13

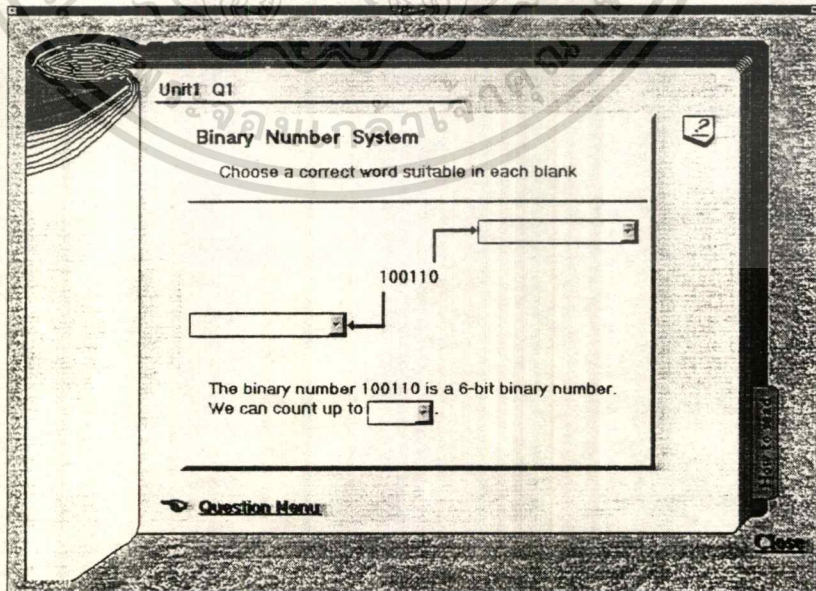


รูปที่ 4.13 หน้าจอแบบทดสอบหลักของบทเรียน

4. เลือกหัวข้อที่จะทำการทดสอบจะแสดงคำถามของหัวข้อที่ต้องการทดสอบดังรูป


ที่ 4.14

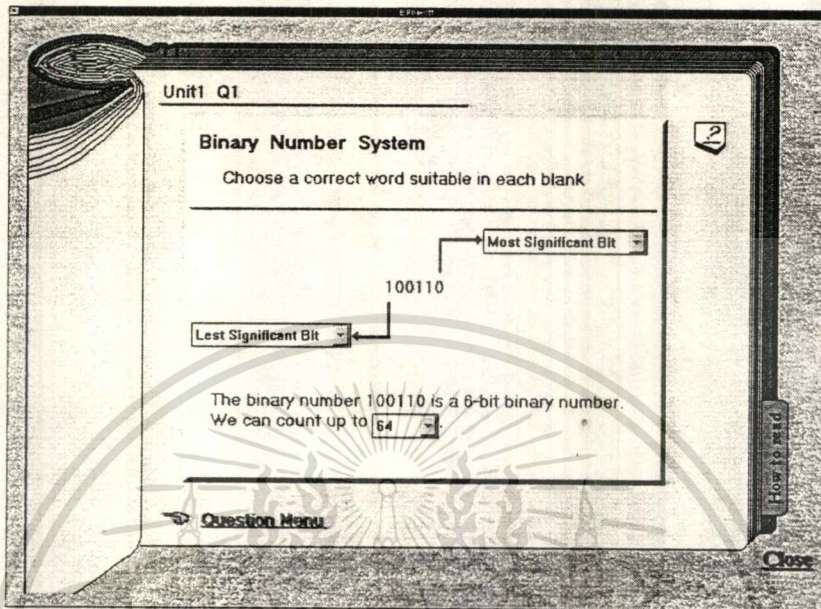
5.




รูปที่ 4.14 คำถามหัวข้อที่ต้องการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ห้ามเผยแพร่ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

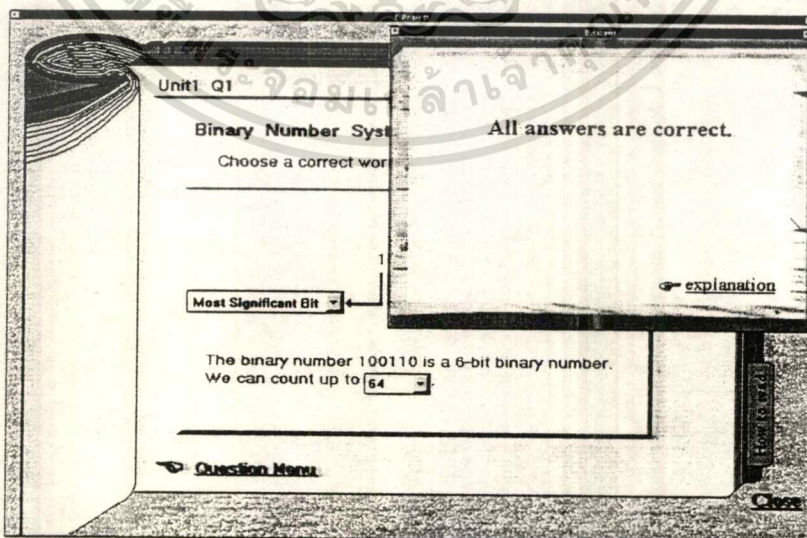
5. ทำการเลือกคำตอบโดยคลิกที่  ซึ่งจะแสดงผลดังรูปที่ 4.15



รูปที่ 4.15 คำตอบที่ถูกเลือก

6. เมื่อได้เลือกคำตอบเรียบร้อยแล้วให้คลิกที่  เพื่อตรวจสอบคำตอบ

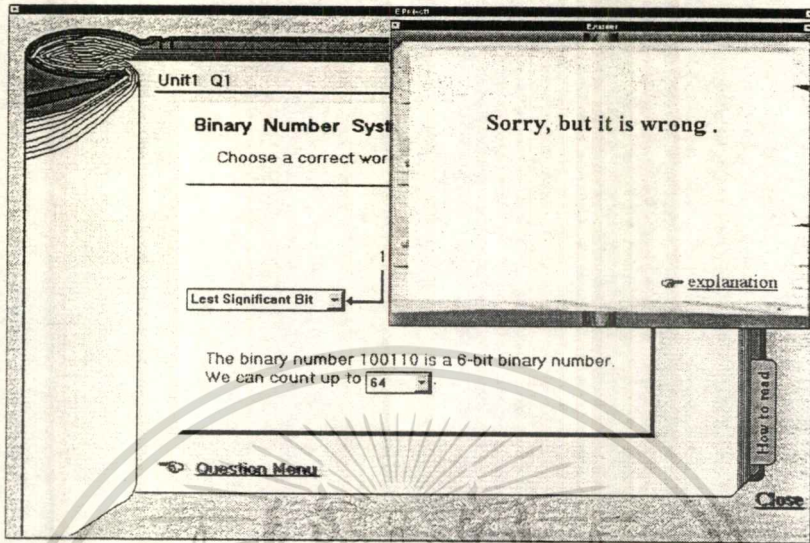
7. หากตอบถูกจะปรากฏหน้าจอดังรูปที่ 4.16



รูปที่ 4.16 คำตอบที่ถูกต้อง

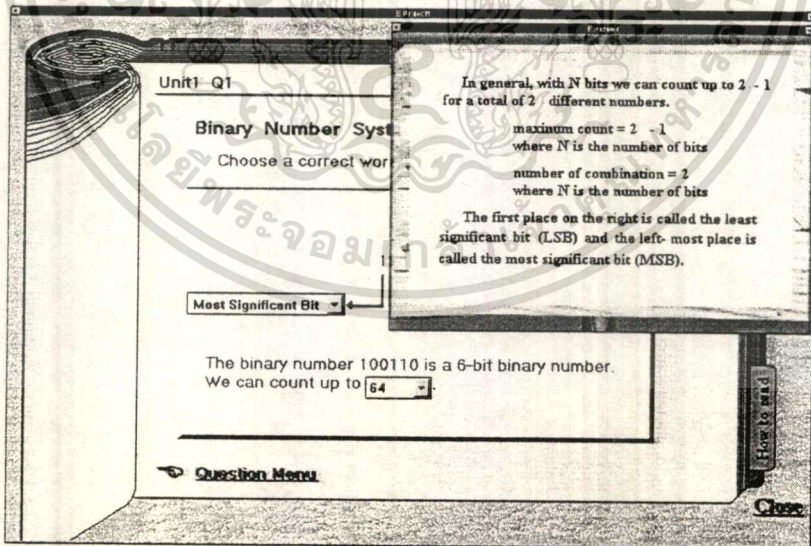
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. หากตอบผิดจะปรากฏหน้าจอดังรูปที่ 4.17



รูปที่ 4.17 คำตอบที่ผิด

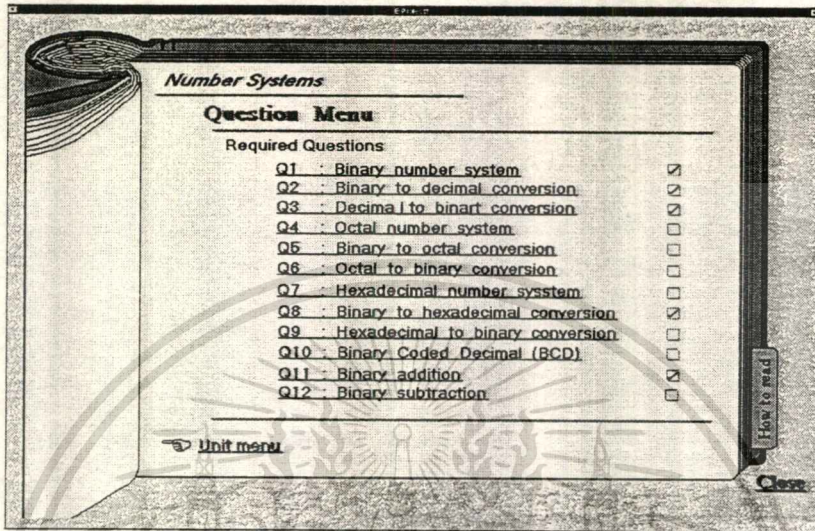
9. กดปุ่ม Explanation เพื่อดูคำอธิบายเพิ่มเติมดังรูปที่ 4.18



รูปที่ 4.18 คำอธิบายคำตอบเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อกด "Question menu" จะกลับไปสู่หน้าจอแบบทดสอบหลักของบทเรียน พร้อมทั้งทำการ Check (ทำเครื่องหมาย) ว่าได้ทำข้อนั้นๆ ไปแล้ว ดังผลในรูปที่ 4.19



รูปที่ 4.19 หน้า Question menu หลังจากทำแบบทดสอบ

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

ปริญญานิพนธ์ฉบับนี้ เสนอการสร้างบทเรียนโปรแกรมช่วยสอนวิชา คณิตศาสตร์ ซึ่งมีเนื้อหาบรรยายเป็นภาษาอังกฤษเน้นความเป็นสากล ทั้งนี้เพื่อพัฒนาเนื้อหาวิชาให้สอดคล้องกับระบบของโครงการวิจัยร่วมระหว่างไทย ญี่ปุ่น และจีน ในการพัฒนาทรัพยากรมนุษย์ด้านโทรคมนาคมในภูมิภาคเอเชียแปซิฟิกและเพื่อพัฒนาระบบเครือข่ายการสอนทางไกลด้วยสื่อระบบสองทาง ผ่านดาวเทียมระหว่างประเทศที่มีประสิทธิภาพมากกว่าวิธีปกติ ที่ใช้สอนระหว่างมนุษย์ต่อมนุษย์โดยตรง พร้อมสื่อประสมผ่านดาวเทียม โดยใช้เครื่องมือการพัฒนา CAI (AUTHORING TOOLS) ชื่อ โปรแกรม HURESS ซึ่งบทเรียน โปรแกรมคอมพิวเตอร์ช่วยสอนนี้สามารถนำไปใช้เป็นบทเรียนจริงหรือนำไปใช้เป็นบทเรียนเสริมวิชาคณิตศาสตร์ ซึ่งผู้เรียนสามารถเรียนรู้ได้ตามความเข้าใจของตนเอง ทำให้การเรียนรู้มีประสิทธิภาพมากขึ้นเนื่องจากข้อได้เปรียบในเรื่องการให้ข้อมูลป้อนกลับ ในบทเรียนจะมีแบบทดสอบเพื่อทดสอบความรู้ความเข้าใจของผู้เรียน โดยที่ผลการทดสอบของนักเรียนแต่ละคนจะปรากฏบนจอคอมพิวเตอร์ของผู้สอนด้วย ซึ่งผู้สอนจะเป็นคนเช็คได้ว่า นักเรียนคนไหนตอบผิดหรือถูก

ในส่วนของบทเรียน โปรแกรมนี้จะมีเนื้อหาที่เป็นใจความสำคัญของหัวข้อต่างๆ มีรูปภาพประกอบคำอธิบาย ในแต่ละบทเรียนจะมีคู่มือที่เป็นเนื้อหาอธิบายหัวข้อต่างๆ ให้กับอาจารย์ผู้สอนใช้ประกอบการบรรยาย การเปิดโอกาสให้ผู้เรียนได้มีส่วนร่วมในการเรียนการสอนและยังลดปัญหาความแตกต่างระหว่างบุคคล โครงการนี้ได้รับความสนใจจากนักศึกษาและประชาชนทั่วไป และจะเป็นประโยชน์อย่างยิ่งต่อการพัฒนาการศึกษา และช่วยแก้ปัญหาการขาดแคลนอาจารย์ สร้างความเสมอภาคทางการศึกษา สามารถติดต่อกับมหาวิทยาลัยในต่างประเทศทั่วโลกได้ อันจะนำไปสู่การพัฒนาทางวิชาการต่อไป และการสร้างบทเรียนคอมพิวเตอร์ช่วยสอนนี้นอกจากจะสร้างขึ้นโดยโปรแกรม HURESS แล้วยังใช้โปรแกรมช่วยสร้างภาพอื่นๆ ทำให้สามารถช่วยออกแบบบทเรียน โปรแกรมคอมพิวเตอร์ช่วยสอนได้เป็นอย่างดี แต่อย่างไรก็ตามการทำงานทุกอย่างก็จะต้องมีข้อบกพร่อง และข้อที่จะต้องเพิ่มเติม ซึ่ง

ผู้จัดทำได้เสนอแนะไว้เพื่อที่จะเป็นประโยชน์สำหรับผู้ที่มาโปรแกรมช่วยสอนวิชาจิตวิทยาไปพัฒนาต่อไป

5.1 ปัญหาที่พบในการทำโครงการ

5.1.1 ในระยะแรกต้องเสียเวลากับการศึกษาการใช้โปรแกรม HURESS เนื่องจากคู่มือการใช้โปรแกรมเป็นภาษาอังกฤษทั้งหมด และไม่มีพื้นฐานในการใช้โปรแกรมมาก่อน

5.1.2 โปรแกรม HURESS มีข้อจำกัดในการทำงาน ทำให้เกิดความล่าช้าในการทำงาน

5.1.3 การทำ Media แต่ละตัวต้องทำงานเสร็จสิ้นก่อนแล้วจึงจะสามารถทำกับ Media ตัวอื่นได้ไม่เช่นนั้นจะทำให้ข้อมูลถูกทำลายได้

5.1.4 เนื้อหาจำเป็นต้องเป็นภาษาอังกฤษเพื่อความเป็นสากลและให้สอดคล้องกับโครงการฯ ทำให้ต้องเสียเวลาในการเรียบเรียงเนื้อหาให้เหมาะสมกับบทเรียน

5.1.5 ในส่วนของ Text Media ไม่สามารถทำตัวอักษรห้อยได้

5.1.6 เนื่องจากระบบที่ใช้เป็น Network station ทำให้สามารถลิ้งค์ข้อมูลถึงกันได้และทำการลบข้อมูลของอีกเครื่องหนึ่งได้โดยไม่ตั้งใจ ทำให้ต้องเสียเวลาในการสร้างงานใหม่ทั้งหมด

5.1.7 ข้อมูลโดยส่วนมาก จะเป็นรูปภาพซึ่งต้องใช้เนื้อที่ในการจัดเก็บมากไม่สามารถใช้แผ่นดิสเก็ตธรรมดาจัดเก็บได้หมดจึงต้องจัดเก็บไว้ในฮาร์ดดิสก์ของระบบและบางครั้งข้อมูลก็หายไปโดยไม่ทราบที่มาของสาเหตุ

5.1.8 เนื่องจากเป็นโครงการวิจัยร่วมระหว่างประเทศไทยและญี่ปุ่น ซึ่งบางครั้งต้องมีการหยุดการทำโครงการเพื่อให้เจ้าหน้าที่ทางญี่ปุ่นมาเช็คระบบ

5.2 การแก้ปัญหา

5.2.1 ต้องพยายามศึกษาการใช้โปรแกรมอย่างต่อเนื่องและจริงจัง

5.2.2 ทำ Media ทีละตัว

5.2.3 ปรึกษาอาจารย์ภาควิชาภาษาอังกฤษ

5.2.4 ใช้ Text Media วางซ้อนกัน

5.3 ข้อเสนอแนะ

5.3.1 นำโปรแกรมที่ใช้สร้าง CAI อื่นๆ มาใช้ร่วมกับโครงการฯ เช่นเดียวกับโปรแกรม HURESS ได้

5.3.2 นำโปรแกรม HURESS มา INSTALL ลงใช้กับเครื่อง PC และสามารถใช้งานจริงได้

5.4 ประโยชน์ที่ได้รับจากโครงการ

5.4.1 ทำให้มีความสามารถในการใช้โปรแกรมสร้างสื่อประสม HURESS

5.4.2 ได้โปรแกรมช่วยสอนวิชาคิติดอล

5.4.3 ทำให้การเรียนการสอนสะดวกยิ่งขึ้น

5.4.4 ใช้เป็นแบบของการศึกษาโปรแกรมสร้างสื่อประสม HURESS

5.4.5 ได้แนวทางสำหรับการพัฒนาวิชาอื่นๆ เพื่อร่วมโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นต่อไปในอนาคต

5.5 แนวทางการพัฒนา

5.5.1 เพิ่มเทคนิคให้สวยงามเพื่อดึงดูดความสนใจของผู้ใช้

5.5.2 เพิ่ม Media ในการนำเสนอให้ผู้เรียนสนใจและใช้สะดวกมากขึ้น

5.5.3 เพิ่มบทเรียนในวิชาอื่นๆ ได้

5.5.4 สามารถนำโปรแกรมมาใช้สอนจริงได้



ภาคผนวก ก

รูปแบบการใช้งานของฟังก์ชันต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งานของฟังก์ชันต่าง ๆ

This manual deal with MEDIA-ADK which operates on Borland C++ ver4.52 object Windows (OWL)2.5 and provides media development kit involving C++ class library designed to develop Huress-Book AS-IN media.

In case you desire to develop Huress-Book applications but if you are not satisfied with the functions offered by the standard media, application developers can create new media programs with unique functions. In order to create new media programs, you need have a basic knowledge on:

1. Borland C++ ver 4.52 OWL 2.5 Programming
2. Outline of Huress-Book and its operation
3. Outline of media on Huress-Book

This manual discuss media programming methods assuming that users are familiar with the above-mentioned software. Besides, it is essential that user of MEDIA-SDK have proper hardware and software operating on Huress-Book and Borland C++ ver4.52 OWL 2.5

1. The terms and the description format of MEDIA-SDK

In this manual, special terms of the MEDIA-SDK are used beside Huress-Book terms.

DW_OBJECT	The object to be managed by Huress-Book
DW_MESSAGE	The message to be processed by Huress-Book
DW_METHOD	The procedure to be managed by Huress-Book
DW_SELECTOR	The string recognizing DW_MESSAGE
DW_CLASS	The DW_OBJECT of classes to be managed by Huress-Book
Media Class	The DW_OBJECT to manage each media (same type of media)
Media Instance	The DW_OBJECT to create each media

Media Method	The DW_OBJECT included in the media class and media instance
Media Message	The messages for the media class and media instance
Media View	The contents displayed on the screen of the media instance.
It	is a GUI of the media instance.
Book View	The contents displayed on the screen of the book (pages).
Identifier	The file open identifier for the media,etc.
Identification Number	The unique number of media,etc.
Media Information Dialog	The modal dialog to update media information included in the media instance
Media file	The file to hold media information created for each media instance.
Huress-Book Class Name	The name of media class managed in Huress_-Book.
Huress-Book Class Number	The class number of media class managed in Huress-Book.

In this manual, functions and DW_METHOD included in the DW_OBJECT are represented as follows.

[Format]	Calling format
goByPageNo	DW_SELECTOR
stage	Return value
arg	A variable of the arguments
	An argument to represent input
	An argument to represent output
[Comment]	Explanation of METHOD functions

Sample 1 DW_METHOD

goByPageNo

[format] stage=thisBook goByPageNo arg;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ arg[0] Page Number (An integer DW_OBJECT) ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

arg[1] The DW_OBJECT of the page

stage The DW_OBJECT for true or false

[Comment] This message closes the currently opened page of the book, then open the page specified by the page number. If the message is issued from and to the current page, the message return no value since it quits the current page immediately.

Sample 2 Function (A function of the C++ language))

HPK_newBasicObject

[format] HPK_ObjectId HPK_newBasicObject(

HPK_classId xclass //A type of the basic DW_OBJECT

);

[Comment] This function creates the basic DW_OBJECT specified by xclass.

HPK_Integer an integer DW_OBJECT

HPK_Float A floating point DW_OBJECT

HPK_Character A character DW_OBJECT

HPK_String A string DW_OBJECT

HPK_Point A position DW_OBJECT

HPK_Rect A rectangle DW_OBJECT

HPK_Boolean A logical value DW_OBJECT

HPK_Atom An atom DW_OBJECT

[Example] r = HPK_newBasicObjec(HPK_Integer);

An integer DW_OBJECT is created into r.

2. Installation of the MEDIA_SDK

Let's create a parent folder to be used for developing media. This folder is called development root folder temporarily. This folder creates the following folder:

- A folder storing MEDIS-SDK files (\SDK)
- A folder storing sample programs (\SMP)

Copy the files from the Huress-Book folder in to the development folder.

\MEDLASDK\SDK Copy all the files of the file extension. 'H' and '.LIB' to the development folder \SDK.

\MEDLASDK\SMP Copy all the files to the development folder \SMP.

The list of MEDIA-SDK files are:

MEDLASDK\SDK\

- | | |
|-------------|---|
| SDK.H | The main header of the MEDIA-SDK |
| SDK_HPK.H | The header of the Huress-Book interface library |
| SDK_REC.H | The header of the media message dispatch in the MEDIA |
| SDK_CLS.H | The header of the media class in MEDIA-SDK |
| SDK_INS.H | The header of the media instance in MEDIA-SDK |
| SDK.CPP | The main source of the media in MEDIA-SDK |
| SDK_REC.CPP | The source of the media message dispatch in MEDIA-SDK |
| SDK_CLS.CPP | The class source of the media in MEDIA-SDK |
| SDK_INS.CPP | The source of the media instance in MEDIA-SDK |
| SDK_ETC.CPP | The source of the function library in MEDIA-SDK |
| SDK_.LIB | The MEDIA-SDK importing library |
| HPK.LIB | The Huress-Book interface importing library |

MEDLASDK\SMP\

- | | |
|---------|-------------------------------------|
| SMP.IDE | A project of a basic sample program |
| SMP.H | Header of basic sample program |

SMP.CPP	Source of a basic sample program
SMP.RC	A resource of a basic sample program
SMP.RES	A complied source of a basic sample program
SMP.DEF	A definition file of a basic sample program
SMP.A.H	Header of a Type A media sample program
SMP.A.CPP	Source of Type A media sample program
SMP.A.RC	A resource source of a Type A media sample program
SMP.A.RES	A complied resource of a Type A media sample program
SMP.A.DEF	A definition file of a Type A media sample program
SMP.B.H	Header of a Type B media sample program
SMP.B.CPP	Source of Type B media sample program
SMP.B.RC	A resource source of a Type B media sample program
SMP.B.RES	A complied resource of a Type B media sample program
SMP.B.DEF	A definition file of a Type B media sample program

3. Procedure for media programming

The media program is created according to the following procedures.

3.1 Creation of media development folders

Base on the procedure set forth in Section [MEDIA-SDK Installation] , a media development folder id created , which can be named at the option of a developer. Files to be created hereafter should be stored in the particular folder.

3.2 Creation of project files (IDE)

Develop project files by referencing SMP.IDE. Refer to OPTION assigned to SMP.IDE to assign PROJECTOPTION for project files, where you be required to adjust assignment of folders and files dependent on new media programs.

3.3 Creation of module definition file

Develop module definition file by referencing SMP.DEF.

Contents of <SMP.DEF>

LIBRARY SAMPLE

DESCRIPTION 'Huress-Book Media SAMPLE'

EXETYPE WINDOWS

CODE PRELODE MOVEABLE DISCRARBLE

DATA PELOAD MOVEABLE SINGLE

3.4 Creation of media program source files

Create the following resources by referencing Section 4 (Sources Programming Procedures).

3.5 Creation of resource definition file

Create the following resource by referencing SMP.RC:

-Icon for media program identification

Create Icon to be used for media program assignment in the [Book Information]0.

Huress-Book uses the first Icon in the resource a media program identifier Icon.

-Media class identifier bit map

Create the bit map with a specification of 22*2016 color. Where you desire to create multiple media classes with the media program, create a bit map for each media class.

Upon creation of the multiple class, this bitmap should be specified according to Section 4

-Media information dialog

Create a dialogue capable of concurrently updating common media information and individual media information. Section 8 and Section 8.9 explain common media information. Individual media information should be designed by a media developer. Where multiple media class are generated using the same media program dialogue should be creates for each class.

3.6 Compiling

After confirming that the following items have been assigned in the project file, proceed to the compile process.

Source programs	(.CPP.H)
resource file	(.RES)
module definition files	(.DEF)
Huress-Book interface import library	(HPK.LIB)
MEDLASDK import library	(SDK.LIB)

After normal termination of compilation, a new media program. DLL is generated in the media development folder. Copy this in the folder\EXEMEDIA in which Huress-Book is installed

3.7 Introduction

The media program can be used if the BOOK utility is used for editing and the media program id installed in the utility.

4. How to Make Source Programs.

In this chapter, using a simple sample we will explain the basic of creating a media program. For creating higher level media programs, refer to other sample programs and the explanation Chapter 9.

4.1 Definition of the "Include Files" Used

In the source file of the media program, define in order the following items. The file to perform "include" on is <sdk.h>. This contains all the files which need to be "included" for MEDIA-SDK.

4.2 Definitions of the Global Variable

<code>TModule *the Module = NULL;</code>	A media program,(.DLL) A pointer to module
<code>Tmedia Class *mySMPclass;</code>	A pointer to media class
<code>HPK_ObjectId nullObject = {NULL}</code>	An underfined DW_OBJECT
<code>HPK_ObjectId trueObject;</code>	A true logic value DW_OBJECT
<code>HPK_ObjectId falseObject;</code>	A false logical value DW_OBJECT

4.3 Definitions of the DllEntryPoint

```
bool WINAPI DllEntryPoint
{
    if(reason ==DLL_PROCESS_ATTACH){
        HBITMAP bitmap;

        trueObject    =b2oB(TRUE);
        falseObject   =b2oB(False);

        TIndexArray(0,0,0);
```

```

The Module = new Tmodule("SMP",hInstance);
bitmap = LoadBitmap(hInstance, "SMP");
mySMPClass = new TSMPClass("SMPPRG", "SMP",93,bitmap);
}
return TRUE;

```

"SMPPRG" is the name of the media program which will be created here.

"SMP" is the Huress-Book class name of the "media class" which we define here.

"93" is the Huress-Book class number of the "media class" which we here.

4.4 Definition of the Call Back Function for Handling Completion

```

extern "C" {
void FAR PASCAL_expor DetachMediaClass()
{
    delete mySMPClass;
}
}

```

At the time of completing the media program, this function is used by Huress-Book.

4.5 Definition of TMediaClass, a Derivative Class

\from Huress-Book the following BW_MESSAGE (requests) is communicated to "media class."

The requests to "media class" are:

- new Create a new media "instance."
- load Load media information from the media file and create a media "instance."
- pate Load media information from the clipboard and create a media "instance."
- getError Return media class error information.

In "media class," the capability to respond to each of these requests is needed, and TmediaClass of Media-SDK handles this processing on behalf of the "Constructor" and the **make Instance function.**

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For Details on the DW_MESSAGE to "media class," refer to 7, "Desk DW_OBJECT Reference."

] Here-below is a sample definition of the derivative class.

<Definition of class>

```
class TSMPClass      :publicClass
```

```
{;
```

<Definition of member function >

//constructor (override is necessary)

```
TSMPClass::TSMPClass(char*progName,char*className,
                    HPK_ClassId no,HBITMAP paletteBmp)
                    :TMediaClass(progName,className,no paletteBmp)
```

```
{
```

```
}
```

//create a media instance (override is necessary)

```
Tmedia Instance *TSMPClass::makeInstance() // creation of a media instance
```

```
{
```

```
    //create a media instance and return the pointer
```

```
    return new TSMPIInstance();
```

```
}
```

4.6 Definition of TmediaInstance, a Derivative Class

From Huress-Book the following DW_MESSAGE (requests) is communicated to "media instance," and within this class, the capability to respond to each of the requests is needed.

For details about the DW_MESSAGE to "media instance", refer to "7.Desk DW_ONJECT Reference."

Requests to "media instance":

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

save Save the media information.
remove Delete the information (media file).
copy Copy the media information into the clipboard.
point Move the media instance.
size Change the size of the media instance.
draw Enable the media view.
invalid Temporarily select the media.
show show the media.
hide hide the media.
setFlage Set execute mode.
getError Return media error information.
dispose Delete media instance.
info Renew the media information by means if the media information dialog.
asString Return the name of the media instance.
move change the (rectangle) coordinate of the media instance.
setFocus Set focus on media instance.
getPoint Return the coordinates (position) of the media instance.
getSize Return the Size of the media instance.
getRect Return the (rectangle) coordinate of the media instance.

TMediaInstance of MEDIA-SDK handle almost all the above processing on be half of the developer , but it is necessary to override the "member " function in the following sample. Also, in media instance, by defining it own reception-capable DW_METHOD , it can be used from page procedure.

<Definition of a class>

```

class TSMPIInstance: public TMediaInstance
{

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BOOL load();

    BOOL save();

    BOOL information();

public:
    TSMPInstance():TMediaInstance(){}

    BOOL create(short mode ,HPK_ObjectId class,HPK_ObjectId*arg);

    void DMTest1 (TMessageInfo&mi); //original DW_METHOD

    //next line id necessary when original receivable DW_METHOD is offer to media
instance
    DECLARE_METHODE_TABLE(TSMPInstance);

};

<Definition of member functions>
// creation of a media view
BOOL TSMPInstance::create(short mode ,HPK_objectId classId,
                           HPK_OjectId *arg)
{
    //common process to create view
    if(TMediaInstance::create(mode,classId,arg) ==False)
        return FALSE;

    //create a media view here

    //this sample create a button view
    mediaWin = new Tbutton (
        book win,
        -1,
        header. name,
        header.x, headr.y. header.h
    );

    media win->Create();

    return TRUE;
}

```

```

}
//load media information (override is necessary)
BOOL TMP Instance ::load ()
{
    if(TMediaInstance::load()==FALSE) // loading a media common information
        return FALSE;
    // load individual information from media information here
    return TRUE;
}
//save media information (override is necessary)
BOLL TMPInstance::save()
}
// update media informations (override I necessary)
BOOL TSMPInstance::information()
{
    // execute a dialog box to update madia informations (common information and
    //individual information)here
    // use media Win or bookWin for parent window
    Return FALSE; // no information updated
}
// definition of original receivable method
void TSMPInstance::DMTest1(TMessageInfo&mi)
{
    static HPK_ClasId ct[] = {HPK_String};
    //test of parameters
    if(mi.testBasicObject) (ct,1)==False){
        mi.result = null object; // stop the procedure
        return;
    }

```

```

::Message Box(media Win->HWindow, "", "test1",MB_OK);
    mi.result = trueObject;
}
/dispatch table of original receivable W_METHOD
DEFINE_METHOD (TSMPInstance,TMediaInstance)
    METHODENTRY("test1" ,DMTest1 ),
END_METHOD;

```

4.7 Definition of Media View Class (Derivative class of Window)

In general, with media instance the media view must be displayed in a window (HWND) specified by Hures-Book. If this display is done in a derivative class of Twindow, it can be done very simply.

for this reason, define derivative class of Twindow which have the functions set by the developer. Among these derivative classes , you can be Tcontrol, Tbutton , TFrameWindow, ect. but Tdialog is not appropriate.

4.8 Notes on Creating Media programs

4.8.1 About the media view class

with media instance the content (method) of processing must be changed according to the mode communicated from desk DW_OBJECT.

Do the processing in accordance with the selected mode.

(When using Browser mode)

Prohibit renewal of media information.

(When using Edit mode)

Allow renewal of media information.

Ignore all mouse events with media view or alternatively, pass events such as mouse movements or clicks received by the media to the master window (Book View.)

In the basic sample, inside the DW_MESSAGE response function, DMDraw, we switch black

and forth between enabling and disabling the window. นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.2. About dialog box

In the media program, When using dialog box in cases other than the media information dialog box, use the usage start statement (startDialog) and the usage end statement (endDialog) in respect to Desk DW_OBJECT. (Refer to 9.Type A Media Sample.)

4.8.3. When creating dialog box or window

With a constructor such as TWindow, TDialog, and TWindowFrame, always specify its own DLL module (Tmodule).

4.8.4. About window messages

With media view, when using a derivative class of Twindow, there are limitations on window messages sent to Book view, the master window.

WM_COMMAND and WM_SYSCOMMAND may not be sent.

4.8.5. About assignment of Huress-Book class names and Huress-Book class numbers be unique within Huress-Book. These are administered through our development group, so kindly inquire as necessary.

4.8.6. About the MEDIA-SDK source code

The goal of the SDK source code is to provide a hint as to how to create a media program with Huress-Book. Review it thoroughly and understand its outline, including the class layers (overall – specialized construction) and object organization (overall -- component construction.)

Also, when creating a media program, refer to detailed functional capabilities from the parent (super-)class down through successive member functions and data members.

By becoming well versed in MEDIA-SDK, it will be possible to create media programs with programming language environments other than Borland C++ ver4.52 OWL2.5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

However note that in order to guarantee operation of standard media, you may not modify the source code or re-do the SDK-DLL in any way.

4.8.7. About accelerators

When browsing books, media involving input require the use of an accelerator. However, in the book window, a dedicated accelerator is provided, so the (external) accelerator must be reconfigured.

Also, for media, WM_COMMANDs with ID numbers less than 2000 will not be communicated. Use ID numbers than 2000.

4.8.8 About waiving the execute right

A media program may not generate a command to waive the execute right while DW_MESSAGE is being executed. For example, the GetMessage of Windows API would be such a command.

4.8.9 About changing the palette

In cases when the media program changes the system palette of Windows, by all means handle it by using the Huress-Book Interface Library is not used and processing is done only with Windows API, the media will not run normally.

5. Huress-Book Interface Library Reference

Huress-Book Interface Library provides the execution environment for various objects of Huress-Book. In this execution environment, it performs DW_OBJECT management, Variable management and Message management.

In DW_OBJECT management, various load modules on Huress-Book do the direct management (Creation and selection) for related DW_CLASS or DW_INSTANCE. This is just

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a closed loop management in side of he module, therefore it is necessary to create it can notify the DW_OBJECT ,that was created in the load module , to Huress-Book that recognizes the object.

Afterward,with the aid of DW_OBJECT, Huress-Book provides by specifying the object.

Variable management provides the variables for procedure. The variables are able to use for information transfer between DW_OBJECT.

MESSAGE management has the function to send DW_MESSAGE in order to transmit information between DW_OBJECT.

In Huress-Book interface library , there are many C language functions to make possible to create new media programs. MEDIA-SDK class library ,that is description in [8. MEDIA -SDK class library reference] uses this Huress-Book interface library. C language function using in Huress-Book interface library are implement in HPK.DLL.

Types of DW_OBJECT

-Desk DW_OBJECT

Desk DW_OBJECT creates the desk and only one.

-Bookshelf DW_OBJECT

Bookshelf DW_OBJECT create the bookshelf.It is create for each bookshelf connected to the desk.

-Book DW_OBJECT

Book DW_OBJECT create the book. It is when the book is brought in on the desk.

-Page DW_OBJECT

Page DW_OBJECT create the page. It is created for the open page while the book is on the desk.

-Basic DW_OBJECT

Basic DW_OBJECT is needed for the information transfer between DW_OBJECT's and it is divided into the 9 type of DW_OBJECT as follows. The creation of these

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยวิธีการ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DW_OBJECT can be handled by services(function) provided of these DW_OBJECT is done automatically.

integer DW_OBJECT

real number DW_OBJECT

Character DW_OBJECT

Text DW_OBJECT

Placement DW_OBJECT

Rectangular DW_OBJECT

Logic DW_OBJECT

Atom DW_OBJECT For assigning and general purpose dataa domain.

Variable DW_OBJECT For creating variable

-Media DW_OBJECT

Media DW_OBJECT creates the media. There are media DW_OBJECT for each media. This is the MEDIA-SDK to support for create these DW_OBJECT programs.

Text DW_OBJECT

Button DW_OBJECT

HPK Constant

HPK_MAXARG 20 //max. number of the transferring in //and

out DW- OBJECT

HPK_SELLEN 32 //Max. bytes of the DW_SELECTOR

HPK_CLSLEN 32 //32 Max.bytes for Huress-Book class name

HPK_MAXINDEX 500//Max. number of elements of variable

HPK_MEDIAPAL Register partition for palette for media

HPK_Integer 1 //integer

HPK_Float 2 //real number

HPK_Character 3 //character

เอกสารนี้เป็น HPK_String ไว้สำหรับการ 4 ช่างาน //text ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HPK_Point	5	//placement
HPK_Rect	6	//rectangular
HPK_Boolean	7	//logic
HPK_Atom	8	//atom
HPK_Variable	19	//variable(Special Object)
HPK_NULL	0	//NULL object
MBONO	20	// number o class no. o max. basic
ENOCCLASS		undefined class
ECLASSID		illegal class ID
EDUPCLASS		duplicate class
EINSTAANCE		illegal instance
EMESSID		illegal message ID
EDUPVALI		duplicate variable
ENOVAL		undefined variable
EVALI		illegal variable
ENOMEM		lack of memory space
EOTHER		illegal argument etc.
EZERODEV		zero devide
ENOMES		NO Message
EMAXINDEX		exceed max. number of subscript character
typedef short		HPK_Class; // Class number
typedef union HPKOBJ		HPK-objectId; //DW_OBJECT identifier

- HPK_is Null

```
[Form] BOOL HPK_isNull (
    HPK_ObjectId object;
);
```

[Comment] It tests if object is valid.

[Example] if(HPK_isNull(result)==TRUE)

เอกสารนี้เป็นเอกสารตัวอย่างที่ใช้เพื่ออธิบายการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

::MessageBOX(NULL,"result
error","MEDIA",MB_OK|MB_SYSTEMMODAL);

```

- HPK_add class

```

[Form]HPK_ObjectId HPK_addClass(
    NULL,                reservation
    void*object,         Internal identified value of DW_CLASS
    char*name,           Huress-Book class name
    HPK_Class xclass    Huress -Book class number
    FARPROC callBack    Call back function at receiving
    DW_MESSAGE
);

```

[Comment] It registers DW_CLASS. Upon completion of registration, a wide area variable is created, and the DW_CLASS id stored in the 1t element <index 0>. Do specify normal null for the object.

[Example] HPK_addClass(NULL,NULL, "MYCLASS", 300, callBack);

- HPK_rmClass

```

[Form] HPK_ObjectId HPK_rmClass(
    HPK_ObjectId classObject
);

```

[Comment] it deletes DW_CLASS specified by class Object.

[Example] HPK_rmClass(SMPCIAS);

- HPK_addInstance

```

[Form] HPK_ObjectId HPK_addInstance(
    HPK_ObjectId xclass,
    void*object
);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Common] It creates HPK_OBJECT and register into Huress-Book .

[Example] HPK_addInstance (300,myInstance);

- HPK_getObject Info

```
[Form] void HPK_getObjectInfo(
    HPK_ObjectId instance,
    HPK_ClassId*xclass,
    void**object
);
```

[Comment] It obtains information of instance.

[Example] HPK_getObjectInfo (myInstance,&myClassId,&myObj);

- HPK_sendMessage

```
[Form] HPK_ObjectId HPK_sendMessage(
    HPK_objectId self,      MESSAGE sends out DW_OBJECT
    HPK_objectId target,    MESSAGE receivers DW_OBJECT
    char *message,          MESSAGE name
    short count,            number of arguments
    HPK_ObjectId obj[]      a row argument DW_OBJECT
);
```

[Comment] It sends DW_MESSAGE from self to target.

Return value is DW_OBJECT from target.

This function is to send is to send out DW_MESSAGE of Huress from C++ language.

It is the function as Huress-Book sending out message from user

[Example] HPK_sendingMessage (mediaObj,ntioObj, "write", arg);

- HPK_defgdef

```
[From] void HPK_degdef(
```

```
char*name
);
```

[Comment] It creates the variable specified by the variable name(name) and register it into the wide area scope. Once the variable is registred, it stays in Huress-Book until HPK_rmgdef () is called.

[Example] HPK_degdef ("my Huress-Book").

- HPK_rmgdef

```
[From] void HPK_rmdef(
char*name
);
```

[Comment] It creates the variable specified by the variable name(name) and register it into the book scope. The variable stays until the Book is returned.

[Example] HPK_degdef ("myBook").

- HPK_findVObj

```
[From] HPK_ObjectId HPK_findVObj(
char*name
);
```

[Comment] It fetches the variable DW_PBJECT specified by the variable name (name). The variable DW_PBJECT that has an object name is retrieved in order of procedure, page, page,book and wide area scope. The first DW_PBJECT found id returned.

[Example] desk = HPK_VObject("DESK");

- HPK_setObjectTo var

```
[form] void HPK_ObjectToVar(
HPK_ObjectId variable,
```

```
HPK_ObjectId Object,
```

short index

);

[Comment] It substitutes DW_OBJECT for the subscript (index) specified by variable

DW_OBJECT (variable).

[Example] HPK_setObjetToVar(variable,object,10);

- HPK_gettObjectForm Var

[Form] HPK_ Object HPK_getObjectFormVar(

HPK_ObjectId variable ,

short index

);

[Comment] it fetches DW_OBJECT of subscript (index) specified by variable DW_OBJECT (variable).

{Notice} There is a maximum value to specify the subscript.

[Example] r = HPK_getObjectFromVar(variable,10);

- HPK_newBasicObject

[From] HPK_ObjectID HPK_newBasicObject(

HPK_ClassID xclass types of basic DW_OBJECT

[Comment] It creates the basic DW_OBJECT specified by xclass.

HPK_Integer Integer DW_OBJECT

HPK_Float Real Number DW_OBJECT

HPK_Character Character DW_OBJECT

HPK_String Text DW_OBJECT

HPK_Point Placement DW_OBJECT

HPK_Rect Rectangle DW_OBJECT

HPK_Boolean Logic DW_OBJECT

HPK_Atom Atom DW_OBJECT

[Example] r = HPK_newBasicObject(HPK_Integer);

เอกสารนี้เป็นเอกสารที่... โปรดให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- HPK_getIValue

```
[From] long HPK_getIValue(
        HPK_ObjectId &object
    );
```

[Comment] It fetches the value of Integer DW_OBJECT(object).

[Example] i = HPK_getIValue(object);

- HPK_getFValue

```
[From] double HPK_getFValue(
        HPK_ObjectId &object
```

[Comment] It fetches the value of real DW_OBJECT(object).

[Example] f = HPK_getFValue(object);

- HPK_getCValue

```
[From] short HPK_getCValue(
        HPK_ObjectId &object
```

[Comment] It fetches the value of character DW_OBJECT(object).

[Example] c = HPK_getCValue(object);

- HPK_getSValue

```
[From] char HPK_getSValue(
        HPK_ObjectId &object
```

[Comment] It fetches the value of text DW_OBJECT(object).

{Notice} The return value is a pointer to text. In case the object is deleted, the entity of text dose not exist anymore once the object has been deleted. Normally make sure to copy the text into your own domain after executing this function.

[Example] s = HPK_getSValue(object);

- HPK_getPValue

```
[From] void HPK_getPValue(
        HPK_ObjectId &object
        short *x,
        short *y
```

);

[Comment] It fetches the value of placement DW_OBJECT(object).

[Example] HPK_getPValue(object,&x,&y);

- HPK_getRValue

[From] void HPK_getRValue(
 HPK_ObjectId &object
 short *x1,
 short *y1,
 short.*x2,
 short *y2
);

[Comment] It fetches the value of rectangle DW_OBJECT(object).

[Example] HPK_getRValue(object,&x1,&y1,&x2,&y2);

- HPK_getBValue

[From] BOOL HPK_getBValue(
 HPK_ObjectId &object
 BOOL b
);

[Comment] It fetches the value of logic DW_OBJECT(object).

{Notice} BOOL is short.

[Example] r = HPK_getBValue(object);

- HPK_getAValue

[From] void HPK_getAValue(
 HPK_ObjectId &object
 long *l,
 void **buffer
);

[Comment] It fetches the value of atom DW_OBJECT(object).

เอกสารนี้เป็นเอกสารที่สงวนไว้ส The value obtained is memory domain address and its size. โยชนด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Example] `HPK_getAValue(object,&len,&buffer);`

- `HPK_setIValue`

```
[From] void HPK_setIValue(
        HPK_ObjectId &object
        long I
    );
```

[Comment] It initializes `DW_OBJECT(object)` with the integer `i`.

[Example] `HPK_setIValue(object,10L);`

- `HPK_setFValue`

```
[From] void HPK_setFValue(
        HPK_ObjectId &object
        double d
    );
```

[Comment] It initializes `DW_OBJECT(object)` with the real number `d`.

[Example] `HPK_setFValue(object,10.5L);`

- `HPK_setCValue`

```
[From] void HPK_setCValue(
        HPK_ObjectId &object
        short c
    );
```

[Comment] It initializes `DW_OBJECT(object)` with the character `c`.

[Example] `HPK_setCValue(object,'C');`

- `HPK_setSValue`

```
[From] void HPK_setSValue(
        HPK_ObjectId &object
        char *s
    );
```

[Comment] It initializes `DW_OBJECT(object)` with the string.

{Notice} It sets a copied text of `s`.

[Example] `HPK_setSValue(object,"string");`

- **HPK_setPValue**

```
[From] void HPK_setPValue(
        HPK_ObjectId &object
        short x,
        short y
    );
```

[Comment] It initializes DW_OBJECT(object) with the placement (x,y).

[Example] `HPK_setPValue(object,10,10);`

- **HPK_setRValue**

```
[From] void HPK_setRValue(
        HPK_ObjectId &object
        short x1,
        short y1,
        short x2,
        short y2
    );
```

[Comment] It initializes DW_OBJECT(object) with the rectangle coordinates (x1,y1,x2,y2).

[Example] `HPK_setRValue(object,10,10,200,200);`

- **HPK_setBValue**

```
[From] void HPK_setBValue(
        HPK_ObjectId &object
        BOOL b
    );
```

[Comment] It initializes DW_OBJECT(object) with the boolean b.

{Notice} BOOL is short.

[Example] `HPK_setBValue(object,TRUE);`

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[From] void HPK_setAValue(
        HPK_ObjectId &object
        long l
    );
```

[Comment] It initializes DW_OBJECT(object) with the long l.

As the results,the data domain of this value is secured.

[Example] HPK_setAValue(object,1000);

- HPK_addPalett

```
[Form] void HPK_addPalette(
        HWND hWnd,           windows handle of media view
        HPALETTE hPalette,   Palette of media instance
        int kind              DW_OBJECT partition
    );
```

[Comment] It registers the media view that changes the palette.If there is no palette in the media instance,the palette can be initialized with NULL.

[Example] HPK_addPalette(HWindow,NULL,HPK_MEDIAPAL);

- HPK_detachPalette

```
[Form] void HPK_detachPalette(
        HWND hWnd,           windows handle of media view
    );
```

[Comment] It deletes the registration of media view that changes the palette.

The registered palette is not deleted by this function.

[Example] HPK_detachPalette(HWindow);

- HPK_realizePalette

```
[Form] BOOL HPK_realizePalette(
        HWND hWnd,           windows handle of media view
        HDC  hDc              devic context of media view
    );
```

[Comment] It changes the palette. To use this function, it is necessary to register previously the media view and palette in the HPK_addPalette. The return value is true when changing the entry of palette and false otherwise.

[Example] HPK_realizePalette(HWindow,hDC);

- HPK_isRelizeBkgnd

[Form] BOOL HPK_isRealizeBkgnd(

 HWND h Wnd windows handle of media view

);

[Comment] To change the background when changing the palette. This function is used when changing the palette without using HPK_realizePalette. To use this function, it is necessary to register previously the media view by HPK_addPalette. The return value is true when the palette must be changed as background, false otherwise.

[Example] HPK_isRealizeBkgnd(HWindow);

- HPK_error

[Form] short HPK_error();

[Comment] It obtains the execution condition of the other HPK functions. It can be used after executing the other HPK functions.

0	Normal
ENOCCLASS	Unidentified class
ECLASSID	Illegal class ID
EDUPCLASS	Duplicated class
EINSTANCE	Illegal instance
EMESSID	Illegal message ID
EDUPVALI	Duplicated variable
ENOVALI	Undefined variable
EVALI	Illegal variable
ENOMEM	Lack of memory
EOTHER	Illegal argument

ENOMES No Message

EMAXINDEX Exceed MAX.number of Subscripts

[Example] r = HPK_setError();

if(r != 0)

::MessageBox(NULL,"HPKERROR","HURESS-

Book",MB_OK | MB_SYSTEMMODAL);

6. Bookshelf DW_OBJECT Reference

The Bookshelf DW_OBJECT provides media program with variable length or fixed length file ad media file for each instance. It can hold the media information in this file. The structure of the file can be constructed freely but with information common with SDK. The Bookshelf DW_OBJECT provides various services in regard of indexing. In this section, users as follows. The ntioObj found in the [Form] is the bookshelf DW_OBJECT . This is mentioned by the desk DW_OBJECT when creating media instance.

•openItemId

[Form] state = ntioObj openItemId arg;

→arg[0] page Identifier (Integer DW_OBJECT)

→arg[1] media Identifier No (Integer DW_OBJECT)

←arg[2] media Identifier (Integer DW_OBJECT)

←state Execution state (Logic DW_OBJECT)

[Comment] It opens the media file.

[Example] send2Com(mediaObj,ntioObj, "openItemId", 3,arg):

• newItem

[Form] state = ntioObj newItemId arg;

→arg[0] page Identifier (Integer DW_OBJECT)

→arg[1] media attribute No (Integer DW_OBJECT)

→arg[2] media name (Text DW_OBJECT)

←arg[3]	media Identifier No	(Integer DW_OBJECT)
←arg[4]	media Identifier	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It create the media file.

[Example] send2Com(mediaObj,ntioObj, "newItemId", 5,arg):

- seek

[Form] state = ntioObj seek arg;

→arg[0]	0	(Integer DW_OBJECT)
→arg[1]	media Identifier	(Integer DW_OBJECT)
→arg[2]	seek Identifier	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It initializes the placement of sataa to read or write.

[Example] send2Com(mediaObj,ntioObj, "seek", 3,arg):

- read

[Form] state = ntioObj read arg;

→arg[0]	0	(Integer DW_OBJECT)
→arg[1]	media Identifier	(Integer DW_OBJECT)
←arg[2]	Atom Identifier	(Atom DW_OBJECT)
←arg[3]	Data length to read out	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It reads out data.

[Example] send2Com(mediaObj,ntioObj, "read", 4,arg):

- write

[Form] state = ntioObj write arg;

→arg[0]	0	(Integer DW_OBJECT)
→arg[1]	media Identifier	(Integer DW_OBJECT)
→arg[2]	Atom for read out	(Atom DW_OBJECT)

→arg[3] Data length to read out (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It write data.

[Example] send2Com(mediaObj,ntioObj, "read", 4,arg):

- removeItem

[Form] state = ntioObj removeItemId arg;

→arg[0] media Identifier (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It deletes the media file.

[Example] send2Com(mediaObj,ntioObj, "removeItem", 1,arg):

- closeItem

[Form] state.= ntioObj closetemId arg;

→arg[0] media Identifier (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It closes the media file.

[Example] send2Com(mediaObj,ntioObj, "closeItem", 1,arg):

- newIndex

[Form] state = newIndex arg;

→arg[0] Book Identifier (Integer DW_OBJECT)
 →arg[1] Index name (Text DW_OBJECT)
 →arg[2] Key Text (Text DW_OBJECT)
 →arg[3] Page Identifier No (Integer DW_OBJECT)
 →arg[4] Media Identifier (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It create the media file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Example] send2Com(mesiaObj,ntioObj, "newIndex", 5,arg);

- removeIndex

[Form] state = removeIndex arg;

→arg[0]	Book Identifier	(Integer DW_OBJECT)
→arg[1]	Index name	(TextDW_OBJECT)
→arg[2]	Key Text	(Text DW_OBJECT)
→arg[3]	Media Identifier	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It deletes theindex.

[Example] send2Com(mesiaObj,ntioObj, "removeIndex", 4,arg);

- renameItem

[Form] state = renameItem arg;

→arg[0]	Media Identifier	(Integer DW_OBJECT)
→arg[1]	New media name	(Text DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It changes the media name.

[Example] send2Com(mesiaObj,ntioObj, "renamItem", 2,arg);

- get Error

[Form] state = getError arg;

←arg[0]	Error Information1	(Integer DW_OBJECT)
←arg[1]	Error Information1	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It gets the error information of bookshelf.

[Example] send2Com(mesiaObj,ntioObj, "getError", 2,arg);

7. Desk DW_OBJECT Reference

There are many function to send an and receive DW_MESSAGE in the Desk DW_OBJECT. This section describes the typical message receiving function that become necessary for making the media program. Also this section describes how to send messages to the media class and the indicated in [Form] in this reference shows the Desk DW_OBJECT.

■ receiving of DW_MESSAGE

• setItem

[Form] state = Desk setItem arg;

→arg[0]	Media program name DDL name except ".DLL"	(Text DW_OBJECT)
→arg[1]	Huress -Book class name of media	(Text DW_OBJECT)
→arg[2]	Media class	
→arg[3]	HBITMAP	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] It notifies the add-in media information to Huress-Book.

[Example] arg[0] = b20("samp"); //DDL name Except "DLL"

```
arg[1] = b20("samp");
arg[2] = HPK_addInstance(mediaId,NULL);
arg[3] = b20((long) paletteBmp);
send2Com(mediaId,deskId "setItem", 4,arg);
```

• startDialog

[Form] state = Desk startDialog arg;

state	Execution state	(Logic DW_OBJECT)
-------	-----------------	--------------------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] It notifies the use of modal dialog to Huress-Book.

[Example] send2Com(mediaId,deskId "startDialog", -1,arg);

- endDialog

[Form] state = Desk endDialog arg;

state Execution state (Logic DW_OBJECT)

[Comment] It notifies the use of modal dialog box to Huress-Book.

[Example] send2Com(mediaId,deskId "endDialog", -1,arg);

■ sent MESSAGE to the media class

- new

[Form] state = media Class new arg;

→arg[0] Book shelf (Bookshelf DW_OBJECT)

→arg[1] HWND of book(page)in which media view is placed.
(Integer DW_OBJECT)

→arg[2] Book Identifier (Integer DW_OBJECT)

→arg[3] Page Identifier (Integer DW_OBJECT)

→arg[4] Page (Page DW_OBJECT)

←arg[5] -

←arg[6] Media (Media DW_OBJECT)

←arg[7] Media name (Media DW_OBJECT)

←arg[8] Media rectangular (Media DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] create the new media instance.

- load

[Form] state = media Class new arg;

→arg[0] Book shelf (Bookshelf DW_OBJECT)

→arg[1] HWND of book(page)in which media view is placed.
(Integer DW_OBJECT)

→arg[2] Book Identifier (Integer DW_OBJECT)

→arg[3]	Page Identifier	(Integer DW_OBJECT
→arg[4]	Page	(Page DW_OBJECT)
←arg[5]	Media Identifier No.	(Integer DW_OBJECT)
←arg[6]	Media	(Media DW_OBJECT)
←arg[7]	Media name	(Text DW_OBJECT)
←arg[8]	Rectangular of Media	(Rectangular DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] Create media instance by loading the media information from media file.

• paste

[Form] state = media Class new arg;

→arg[0]	Book shelf	(Bookshelf DW_OBJECT)
→arg[1]	HWND of book(page)in which media view is placed.	(Integer DW_OBJECT)
→arg[2]	Book Identifier	(Integer DW_OBJECT
→arg[3]	Page Identifier	(Integer DW_OBJECT
→arg[4]	Page	(Page DW_OBJECT)
←arg[5]	-	
←arg[6]	Media	(Media DW_OBJECT)
←arg[7]	Media name	(Text DW_OBJECT)
←arg[8]	Rectangular of Media	(Rectangular DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] Create media instance by loading the media information from the clipboard.

• systemMessage

[Form] state = media Class systemMessage arg;

→arg[0]	WM_XXX prefix message no. of Windows	(Integer DW_OBJECT)
→arg[1]	LPARAM for prefix message of Windows	(Integer DW_OBJECT)
→arg[2]	LPARAM for prefix message of Windows	

(Integer DW_OBJECT)

stage	Execution stage	(Login DW_OBJECT)
-------	-----------------	-------------------

[Comment] Process WM_XXX prefix message sending to an application from Windows.

[Remark] At present there are two WM_XXX prefix message that are sent by this send.

MESSAGE as flows:

WM_PALLETTECHANGED

WM_PALETTEISCHANGING

•getError

[From] stage = media Class getError arg;

←arg[0]	Information 1	(Integer DW_OBJECT)
---------	---------------	-----------------------

←arg[1]	Information 2	(Integer DW_OBJECT)
---------	---------------	-----------------------

←arg[2]	Information 3	(Integer DW_OBJECT)
---------	---------------	-----------------------

stage	Excution stage	(Login DW_OBJECT)
-------	----------------	-------------------

[Comment] Return error information of media class.

Send out message to media instance

• save

[Form] state = media Instance save arg;

arg[0]	Media Identifier No.	(Integer DW_OBJECT)
--------	----------------------	---------------------

state	Execution state	(Logic DW_OBJECT)
-------	-----------------	-------------------

[Comment] Store the media information.

• Remove

[Form] state = media Instance remove;

state	Execution state	(Logic DW_OBJECT)
-------	-----------------	-------------------

[Comment] Delete the media information.

• Copy

[Form] state = mediaInstance copy;

state	Execution state	(Logic DW_OBJECT)
-------	-----------------	-------------------

เอกสาร [Comment] Copy the media information into the clipboard. ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Point

[Form] state = mediaInstance point arg;

arg[0] Destination point coordinates (Placement DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Move the media instance.

- Size

[Form] state = mediaInstance size arg;

arg[0] Width (Integer DW_OBJECT)

arg[1] Height (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[comment] Change the size of media instance.

- Draw

[Form] state = mediaInstance draw arg;

arg[0] Rectangle of draw domain(Rectangle DW_OBJECT)

arg[1] Hight (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Draw the media by canceling a temporary delete.

- Invalidate

[Form] state = mediaInstance invalidate;

state Execution state (Logic DW_OBJECT)

[Comment] Delete temporary the media.

- Show

[Form] state = mediaInstance show;

state Execution state (Logic DW_OBJECT)

[Comment] Display the media.

- Hide

[Form] state = mediaInstance hide;

state Execution state (Logic DW_OBJECT)

[Comment] Hide the media.

- Info

[Form] state = mediaInstance info arg;

arg[0]	-	
arg[1]	Need to change	(Logic DW_OBJECT)
arg[2]	Name after renewal	(Text DW_OBJECT)
arg[3]	Rectangle after renewal	(Rectangle DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] Renew the media information

- setFlag

[Form] state = mediaInstance setFlag arg;

arg[0]	Flag	(Integer DW_OBJECT)
0x00000001	Execution mode	
0x00000002	Index permission mode	
state	Execution state	(Index permission mode)

[Comment] Set Execution mode and Index mode

the 0 th bit	on	edit
	off	peruse
the 1 st bit	on	Index creation permission
	off	Index creation inhibit

- getError

[Form] state = mediaInstance getError arg;

arg[0]	Information1	(Integer DW_OBJECT)
arg[1]	Information2	(Integer DW_OBJECT)
arg[2]	Information3	(Integer DW_OBJECT)
state	Execution state	(Logic DW_OBJECT)

[Comment] Return error information of media class.

- asString

[Form] state = mediaInstance asString;

เอกสารนี้เป็น state ที่ Media name เป็นการใช้งานเพื่อการคี (Text DW_OBJECT) ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] Return name of media instance

- Move

[Form] state = mediaInstance move arg;

arg[0] Coordinates of media (Rectangle DW_OBJECT)
(Rectangle)

state Execution state (Logic DW_OBJECT)

[Comment] Change the coordinates of the media instance

- setFocus

[Form] state = mediaInstance setFocus;

state Executionstate (Logic DW_OBJECT)

[Comment] Set focus to the media instance.

- getPoint

[Form] state = mediaInstance getPoint;

state coordinates (Placement) (Placement DW_OBJECT)

[Comment] Return the coordinates of media instance.

- getSize

[Form] state = mediaInstance getSize;

state size (Placement DW_OBJECT)

[Comment] Return the size of media instance.

- getRect

[Form] state = mediaInstance getRect;

state Coordinate (Rectangle) (Rectangular DW_OBJECT)

[Comment] Return the coordinates of media instance.

8. MEDIA-SDK Class Library Reference

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The MEDIA-SDK class library provide the developers with the common functions required to make the media program by making the class library in C++ language (Borland C++).

The class library consists of the following classes and are used to define the sub-classes.

class for common processing for media class

class for common processing for media instance

class for index registration

class for procedure defining for event and page

class for dispatching DW_MESSAGE to response function specified for each DW_MESSAGE.

Since the source program is available to the developers, try to refer to the function.

8.1 SDK Macro

There are several macros defined for message response function.

```
DECLARE_METHOD_TABLE(TMediaClass);
```

```
DECLARE_METHOD_TABLE(cls);
```

```
DEFINE_METHOD(cls)\
```

```
DEFINE_METHODS1(cls,base)\
```

```
METHODENTRY(message,method)
```

8.2 SDK Constant

The following media creation codes are defined.

PROCNEW 0 It creates or is created the new media.

PROCLOAD 1 It creates or is created the media from the media file.

PROCPASTE 2 It creates or is created the media from the clipboard.

8.3 TMessage Info

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

this is the class structure that contains the information regarding DW_MESSAGE generated from Huress-Book. The object is created in TMediaClass::execMessage and transfers to the message response function.

- from (public)

[form] HPK_Object from;

[Comment] It is the DW_OBJECT as the source of DW_MESSAGE.

- self (public)

[Form] HPK_Objct self;

[Comment] It is the DW_OBJECT as the destination of DW_MESSAGE.

- method (public)

[form] char*method;

[Comment] It is the DW_OBJECT as the destination of DW_MESSAGE.

- argCount (public)

[Form] short argCount;

[Comment] This is the number of arguments to be attached to the DW_MESSAGE.

- arg (public)

[Form] HPK_Object *arg;

[Comment] The is a row of arguments to be attached to the DW_MESSAGE.

The entity of argument is in the memory space in the source DW_MESSAGE.

- reslt (public)

[Form] HPK_ObjectId result;

[Coment] This DW_OBJECT is reinitialized with false.

- dispath Object (public)

[From] TMethodHandler *dispatch Object;

[Comment] The object pointer that has message response function is stored.

That is Find(TMessageInfo&info)function of TMethodHandler.

- Entry (public)

[Form] TGenericMethodTableEntry *Entry;

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] This is a table that converts the DW_SELECTOR to the Message response function.

- Constructor (public)

```
[From] TMessageInfo(
    HPK_ObjectId f,
    HPK_ObjectId s,
    char *m,short c,
    HPK_ObjectId *a
);
```

[Comment] It is used in the TmediaClass::execMessage.

- TestVariable (public)

```
[Form] BOOL testVariable(
    HPK_ClassId *xclass,
    short count      Number of arguments
);
```

[Comment] It test arguments to be attached the DW_MESSAGE.

The arguments are defined as variables DW_OBJECT.

The xclass is a row of class number of DW_OBJECT. It skips the test if-1 is included in the row element.

- TestBasicObject (public)

```
[Form] BOOL testBasicObject(
    HPK_ClassId *xclass,
    short count      number of arguments
);
```

[Comment] It tests argument to be attached to the DW_MESSAGE.

The arguments are defined as basic DW_OBJECT.

The xclass is a row of class numbers of DW_OBJECT. It skips the test if-1 is included in the row element.

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Is VariableType (public)

[From] BOOL is VariableType();

[Comment] It tests whether or the argument to be attached are the DW_OBJECT.

8.4 TMethodTableEntry

```
template <class T> class_SDKCLASS TMethodTableEntry {
```

public:

```
    typedef void (T::*XPMF)(TMessageInfo&);
```

```
    char    *message;
```

```
    XPMF Pmf;
```

8.5 TMethodHandler

This is the class to manage the response function of Huress-Book. It has a function to change from the text to the message response function

- Distracter (public)

```
[Form] virtual TMethodHandler(){}
```

[Comment] This is a virtual function.

- Find (public)

```
[Form] virtual BOOL find (TMessageInfo&info) = 0;
```

[Comment] This is a genuine virtual function. It tests whether DW_MESSAGE is receivable or not

- dispatch (public)

```
[Form] virtual void dispatch (TMessageInfo&mi);
```

[Comment] It dispatches to the message response function.

- SearchEntries (protected)

```
[Form] BOOL searchEntries(TGenericMethodTableEntry *entries,
                        TMessageInfo&info);
```

[Comment] It test whether DW_MESSAGE is receivable or not.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.6 TMediaClass

This class is to manage the media. The object is made with one for each media. Users can define the sub-class for each of media by using this class as the parent class. The member functions to override for the sub-class are the constructor and makeInstance function. It is not necessary to modify the other function and member variables.

<<Data Member>>

- Desk (protected)

[Form] HPK_ObjectId deskId;

[Comment] It stores the desk the DW_OBJECT.

- MediaId (protected)

[Form] HPK_ObjectId mediaId;

[Comment] It stores the DW_OBJECT of this media class.

<<Member Function>>

- Consturctor (public)

```
[Form] TMediaClass(
    char *progName,            //Name of media program except .DLL
    char *className,           //Huress-Book class name
    HPK_ClassId classId       //Huress-Book class Number
    HBITMAP paletteBmp       //bit map
);
```

[Comment] It creates the media class by Huress-Book class name className and Huress-Book class number classId. It also sets up bit map (22*20) to be used for media palette. The deletion of bit map is done by Huress-Book

- Destructor (public)

[Form] ~TMediaClass();

[Comment] It delete the media class

- makeInstance (public)

```
[Form] virtual TMediaUnstance *makeInstance(
    short mode,
    HPK_ObjectId classId,
    HPK_ObjectId *arg
);
```

[Comment] It creates the media Instance(TMediaInstance) calling out form DMNew(), DMLoad() and DMPaste(). This function is a virtual function.

- DMNew (public)

```
[Form] void DMNew (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the new media instance.

- DWLoad (public)

```
[Form] void DMLoad (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the media information of media instance by loading from the media file.

- DMPaste (public)

```
[Form] void DMPaste(
    TMessageInfo&mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the media instance from the clipboard.

เอกสาร • DMSystemMessage (public) ารใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[Form] Void DMSystemMessage(
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.
It processes Window's WM_XXX prefix messages.

- DmGetError (public)

```
[Form] void DMGetError(
    TmessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_Object.
It return the error information of media class.

- ExecMessage (public)

```
[Form] HPK_ObjectId execMessage(
    HPK_ObjectId from,
    HPK_ObjectId self,
    char *message,
    short count,
    Hpk_ObjectId obj[]
);
```

[Comment] It dispatches the message from DW_SELECTOR of the Receive DW_MWSSAGE to the response function.

- GetClassId (public)

```
[form] HPK_ClassId getClass();
```

[Comment] It returns class number of the media DW_OBJECT.

```
==<(public)
```

```
[Form] int operator == (const TMediaClass& target) const;
```

```
int operator< (const Tmedia class& target) const;
```

[Comment] It returns the class number of the media DW_OBJECT.

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Form] virtual BOOL is ValidInstance (void *instance);

[Comment] It tests whether the media instance is valid or not.

8.7 Tmedia Index

This is the class to store the index information of media.

<<Data Member>>

- keyName (private)

[Form] char keyName[64];

[Comment] This is a text for the Index key. Categorization takes place by this text.

- IndexName (private)

[Form] char indexName [64];

[Comment] Index name

- saveFlag (private)

[Form] BOOL saveFlag;

[Comment] This is a flag to indicate whether or not index is saved. Normally, this flag is changed automatically by the other classes.

<<Member Function>>

- constructor (public)

[Form] TMediaIndex(

char *key,

char *index

);

[Comment] It creates the index object from the index key name and the index name.

- getKey (public)

[Form] char *getKey();

[Comment] It returns the index key name.

- getIndex (public)

[Form] char *getIndex();

[Comment] It returns the index name.

- setSave (public)

[Form] void setSave();

[Comment] It notifies the necessity of saving the index.

- IsSave (public)

[Form] BOOL issave();

[Comment] It returns TRUE when it is necessary to save the index, FALSE the otherwise.

- ==, <(public)

[Form] int operator == (const TMediaIndex& record) const;

int operator <(const TMediaIndex& record) const;

[Comment] It conducts the comparison arithmetic by the address value of the object.

8.8 TMediaEvent

This is a class to hold the proceder name of the page corresponded to the event and the event name. This is a sub-class of the string class that holds the events name.

<<Data Member>>

- scriptName (private)

[Form] char scriptName[32];

[Comment] This is an array to store the procedure name to be executed.

<<Member Function>>

- Consturctor (public)

[Form] TMediaEvent(

char *event,

char *script = 0

[Comment] It creates the object that stores the event name corresponded to the procedure name.

- `getScriptName` (public)

[Form] `char *getScriptName();`

[Comment] It returns the procedure name.

- `setScriptName` (public)

[Form] `void setscriptName(`

`char* str`

`);`

[Comment] It sets up the procedure name.

- `setscriptName` (public)

[Form] `void setScriptName(`

`string&str`

`);`

[Comment] It sets up the procedure name.

8.9 TMediaInstance

This is the class to create the media instance. The object is created for each media instance. Users can define sub-classes for each for media by using this class as the parent class. In defining the sub-classes, the constructor, load function, save function and information function are overridden. Users can override the other member function and data member as needed.

<<Data Member>>

- `header` (protected)

[Form] `MediaHeader header;`

[Comment] It is the common information of media `DW_OBJECT`.

`Struct MediaHeader {`

`long filler1; reserved)`

short	version	Version of media
char	filler2[7];	(reserved)
char	name[32];	Media name
BOOL	showFlag;	Initial display on/off
short	x;	X coordinate
short	y;	Y coordinate
short	w;	Width
short	h;	Height
char	filler3[31];	(reserved)
short	indexCount	Index number
DWORD	length;	Fixed length of the media
}		

This information is set with the default value in the class.

The x, y, w, h are reinitialized in this class but can be reinitialized in the sub-classes.

Make sure to reflect this change to the media view. The indexCount is managed in this class

- **indexArray** (protected)

[Form] TIndexArray *indexArray;

[Comment] This is array of the index information. Users can only do reference.

To add an index information into the array, use the addIndex function.

To delete an index information in the array, use the delIndex function.

- **EventArray** (protected)

[Form] TEvenArray *evenArray;

[Comment] This is an array to store the procedure name at the occurrence of event.

Users can only do reference.

To add a procedure name into the array, use the setScriptName function.

To obtain a procedure name, use the getscriptName function.

- **ExeMode** (protected)

[Form] BOOL exeMode;

[Comment] It shows the peruse mode if this value is TRUE

It shows the edit mode if it is FALSE.

Users can only do reference.

- IndexMode (protected)

[Form] BOOL indenMode;

[Comment] It shows the permission to create the index if the value is TRUE.

It the value is FALSE, it is not allowed to create the index.

- BookWin (protected)

[Form] Twindow *bookWin;

[Comment] This is a window of OWL to create the in which the media is stored.

- PageObj (protected)

[Form] HPK_Object pageObj;

[Comment] This is DW_OBJECT of page in which the media is stored.

- NtioObj (protected)

[Form] HPK_ObjectId ntioObj;

[Comment] This is DW_OBJECT of the bookshelf in which the media is stored

- mediaObj(protected)

[Form] HPK_BbjectId mediaObj;

[Comment] This is the DW_OBJECT (self Object) of this media.

- bookshelf (protected)

[Form] HPK_Object bookID;

[Commnet] This is the book identifier of the book in which the media is stored.

- pageId (protected)

[Form] HPK_Object pageId;

[Comment] This is the page identifier of the page in the which the media is stored.

- pageNo (protected)

[Form] HPK_ObjectId pageNo;

[Comment] This is the identifier number of the page in which the media is stored.

เอกสาร • **mediaId (protected)** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Do the following when x, y, w h of the header are changed.

```
Arg[8]=b2o(
    header.x,
    header.y,
    header.x+deader.w,
    header.y+header.h
);
```

- DMAString (public)

```
[Form] virtual void DMAString(
    TMessageInfo&mi
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT>

- DMSave (public)

```
[Form] virtual void DMSave(
    TMessageInfo&mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk

- DW_MESSAGE.

It saves the media information to the media file.

- DMRemove (public)

```
[Form] virtual void DMRemove (
    TMessageIn& mi
);
```

[Comment] This is the response function of DW_MESSAGE form the desk.

It delete the media file when DMDispose.

- DMCopy(public)

```
[Form] virtual void DMCopy (
    TMessageInfo& mi
```

```
);
```

[Comment] It reads the data of the media information.

- Write (protected)

```
[Form] long write(
    void *buffer;
```

```
    long size
```

```
);
```

[Comment] It writes the media information data into the media file.

- addIndex (protected)

```
[Form] void addIndex(
    TMediaIndex *mediaIndex
);
```

[Comment] It registers the index.

- delIndex(protected)

```
[Form] void delIndex(
    TmediaIndex *mediaIndex
);
```

[comment] It delete the index.

- get Error (protected)

```
[Form] void gteError();
```

[Comment] It obtains the error information of the bookshelf.

- RunMethod(protected)

```
[Form] virtual BOOL runMethod(
    long status = 0; This is the status of media itself;
);
```

[Comment] It demand the execution of the procedure to the page.

It sets up the following DW_OBJECT with argc and argv for the procedure read out.

argc 2(number of arguments)
 argv[0] Sourec media DW_OBJECT
 argv[1] Text DW_OBJECT of destination media name.

- load (protected)

[Form] virtual BOOL Load();

[Comment] It loads the media information from the media file.

It is necessary to cverride these member function in the sub-classes.

In this case make sure to call initially TMediaInstance::Load().

- save (protected)

[Form] virtual BOOL save();

[Comment] It saves the media information to the media file.

- information (protected)

[Form] virtual BOOL information();

[Comment] It renews the information of the media.

It returns TURE when it is renewed and FALSE when it is abandoned.

- constructor (public)

[Form] virtual ~TMediaInstance();

[Comment] It creates the media instance.

- destructor (public)

[From] virtual ~TMediaInstance();

[Comment] It is called after DMDDispose.

- Create (public)

[Form] BOOL create (short mode,HPK_Object classId,HPK_Object *arg);

[Comment] It create the media view.

It is necessary to override the member function in the sub_classes.

In this case, call TMediaInstance::create(mode,classId,arg);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Form] HPK_ObjectTd mediaId;

[Comment] This is the media identifier of the mrdia.

- MediaNo (protected)

[Form] HPK_ObjectId mediaNo;

[Comment] This is the media number of the media.

- ClassId (protected)

[Form] HPK_ObjectId classId;

[Comment] This is the DW_OBJECT of the media class.

- MediaWin (protected)

[Form] Twindow *mediaWin;

[Comment] This is the OWL window of the media. The value is set by the users.

However, it is not necessary to set up the value when the media does not use OWL window. In this case, users can draw the image of the media directly to the bookWin.

- Invalid (protected)

[Form] BOOL invalid;

[Comment] It shows the media to be invalid.

<<Member function>>

- seek (protected)

[Form] BOOL seek(

 long point

);

[Comment] It renews the write pointer by reading the media information.

- Read (protected)

[Form] long read(

 void *buffer,

 long size

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It copies the media information into the clipboard.

- DMPoint(public)

[Form] virtual void DMPoint (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It move the media.

- DMSize(public)

[Form] virtual void DMSize (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It changes the size of the media.

- DMMove(public)

[Form] virtual void DMMove (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It changes the coordinates and size of the media.

- DMGetPoint(public)

[Form] virtual void DMGetPoint (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It returns the coordinates of the media.

- DMGetSize(public)

[Form] virtual void DMGetSize (

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It returns the size of the media.

- DMGetRect(public)

[Form] virtual void DMGetRect (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It returns the coordinates and size of the media.

- DMDraw(public)

[Form] virtual void DMDraw (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It makes the media instance to be effective.

- DMInvalid(public)

[Form] virtual void DMInvalid (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It makes the media instance to be invalid.

- DMShow(public)

[Form] virtual void DMShow (

TMessageInfo& mi

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It show the media view.

เอกสาร DMHide(public) นี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[Form] virtual void DMHide (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.

It hides the media.

- DMInfo(public)

```
[Form] virtual void DMInfo (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.

It renews the media information.

- DMSetFlag(public)

```
[Form] virtual void DMSetFlag (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.

It sets up the execution mode and index permit mode.

- DMSetFocus(public)

```
[Form] virtual void DMSetFocus (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.

It sets the focus to the media.

- DMGetError(public)

```
[Form] virtual void DMGetError (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.

It returns the media error information.

- DMDDispose(public)

```
[Form] virtual void DMDDispose (
    TMessageInfo& mi
);
```

[Comment] This is the response function of DW_MESSAGE from the desk.
It deletes the media instance.

8. 10. Static Data

This is the static data in the MEDIA-SDK.

HPK_ObjectId	nullObject = {NULL};	undefined DW_OBJECT
HPK_ObjectId	trueObject;	TRUE DW_OBJECT
HPK_ObjectId	falseObject;	FALSE DW_OBJECT
TClassArray	*classArray;	management row of media class
BOOL	rmMediaFlag = FALSE;	felete flag of media instance
long	error1;	error infomation 1
long	error2;	error infomation 2
long	error3;	error infomation 3

8. 11. SDK function Library

Huress-Book provides users with the following function that make easier to use the Huress-Book interface library for creating and setting up DW_OBJECT.

- b20 (No.1)

```
[Form] HPK_ObjectId    b20(long i);
```

[Comment] It creates the integer DW_Object that has integer i.

- b20 (No.2)

```
[Form] HPK_ObjectId    b20(double d);
```

[Comment] It creates the real number DW_Object that has floating point d.

เอกสาร • b20 (No.3) ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Form] HPK_ObjectId b20(short c);

[Comment] It creates the character DW_Object that has character c.

- b20 (No.4)

[Form] HPK_ObjectId b20(char *s);

[Comment] It creates the text DW_Object that has text s.

- b20 (No.5)

[Form] HPK_ObjectId b20(short x,short y);

[Comment] It creates the Placement DW_Object that has placement (x,y).

- b20 (No.6)

[Form] HPK_ObjectId b20(POINT p);

[Comment] It creates the Placement DW_Object that has placement p.

- b20 (No.7)

[Form] HPK_ObjectId b20(short x1,short y1,short x2,short y2);

[Comment] It creates the rectangle DW_Object that has rectangle (x1,y1,x2,y2).

- b20 (No.8)

[Form] HPK_ObjectId b20(RECT r);

[Comment] It creates the rectangle DW_Object that has rectangle r.

- b20 (No.9)

[Form] HPK_ObjectId b20(BOOL b);

[Comment] It creates the logic DW_Object that has the logic value b.

- b20 (No.10)

[Form] HPK_ObjectId b20(void *buf,long l);

[Comment] It creates the atom DW_Object of length l and buffer buf.

- 02Rect

[Form] RECT 02Rect(HPK_ObjectId object);

[Comment] It reads the value from the object of the rectangle DW_OBJECT.

- 02Point

[Form] POINT 02Point(HPK_ObjectId object);

[Comment] It reads the value from the object of the placement DW_OBJECT.

เอกสารนี้จัดทำขึ้นเพื่อใช้ภายในหน่วยงานเท่านั้น ไม่สามารถนำออกเผยแพร่ได้
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- send2Com

[From] BOOL send2Com(
 HPK_ObjectId self, SelfDW_OBJECT
 HPK_ObjectId target, Opponent DW_OBJECT
 char *msg, DW_SELECTOR
 short count, number of arguments
 HPK_ObjectId args[], an array of arguments
 BOOL f = TRUE message flag

[Comment] DW_MESSAGE is sent out by assigning variables and substituting the arguments for the variable. The functions are called when f is TRUE. If any error is detected, the error messages are displayed in the system modal dialog box.

Make sure that the number of elements stored in "count" are in the array. These functions handle the transfer arguments to variable. To send out DW_MESSAGE by adding arguments directly, use the HPK_SindMessage() function of the Huress-Book Interface library.

9. How to create Media Using Sample Programs

There are three types of media program on the MEDIA-SDK as follows.

Basic Sample media	(Media class name:SMP)
Type A Media sample	(Media class name:SMP_A)
Type B Media sample	(Media class name:SMP_B)

This section given an overview of the sample program and some advice to follow when creating. Read through this section when using the sample programs.

It is possible to compile the sample of MEDIA_SDK by boland C++ ver.4.52 and execute them by the following procedure.

- Create the folder for the media development as described in [2.MEDIA_SDK installation] and copy the necessary files.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Load the Borland C++ ver.4.52 and open SMP.IDE of \SMP folder.
- Execute compile link. three sets of DLL's become available. (SMP.DLL,SMP_A.DLL,SMP_B.DLL)
- Hurrss-book copies these DLL's to the folder \EXECYMEDIA.
- Load Huress-Book and borrow the book in edit mode.
- Set up media program by book information set up.

9.1. Basic Sample Media

this the media program to make to the media view borrowing entirely from the button class (TButton) of Boland C++ ver.4.52 ObjectWindows2.5. This is a sample program to explain the basic creation of media program,therefore it is not necessarily true to cover the perfect functions.Refer to the other sample for creating the actual media program.

- Function Overview
 - Look like button
 - Does not have the button functions
 - Information dialog box and Index dialog box.are only forms.

- Main class

Tmedia class sub class	TSMPClass
TMedia Instance sub class	TSMPIInstance
Media view	Tbutton
Information Dialog	TInfordlg
Index Dialog	TIndexDlg

- File Construction

Source program	SMP.CPP
Header file	SMP.H
Resource source	SMP.RC

Definition file

SMP.DEF

- The matter to be referenced

Refer to the [4. Creating method of source program] for more details.

9.2. Type A Media Sample

"The type A Media" is a media which have its own windows and to receive the event such as mouse clicking at the time of execution. Here, it is created based on the button class (Tbutton) of objectWindows

- Function element
 - It shows the command button and the media name of the surface.
 - It is possible to call the page procedure by-left clicking at the execution.
 - It is possible to receive own DW_MESSAGE [test1] at the execution. Upon receipt of test1, it show the text transferred as arguments into the dialog box.

- It sets up the arguments.
- It is possible to copy and paste the media.

- Main class

TMediaClass sub class	TSMPCClass
TMediaInstance sub class	SMPInstance
Media view	SButton (created from TButton)
Information Dialog	TInfodlg
Index Dialog	TIndexDlg

- File Construction

Source program	SMP_CPP
Header file	SMP_A.H
Resource source	SMP_A.RC
Definition file	SMP_A.DEF

- The matter to be referenced

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 . Event process at execution

- It defines the number of events, event name table and the event constant.

(refer to the portion of smp_a.cpp)

- It set up an array of event names in the event Array at the time of creation of the newly media instance.

(refer to TSMPInstance::create)

- It set up the procedure of page that is executed in each event at the time renewal of medial information.

(refer to TSMPInstance::information, TInfoDlg)

- When detecting the event in the media view, call runMethod of When detecting the event in the media view, call out runMethod of TMediaInstance.

(refer TSButton::EvLButton Up)

- In case the function of copy and paste of the media are supported, the contents of the eventArray is copied and pasted as well.

(refer to TSMPInstance::DMCopy, TSMPInstance::create)

2 . Receipt of own DW_MEASSAGE at execution

- It declares the use of METHOD TABLE by defining a own DW_MESSAGE process function in the sub-class of TMediaInstance of the header file.

(refer to TSMPInstance class of header file)

- It creates the own DW_MESSAGE process function.

(refer to TSMPInstance::DMTest1)

- It define the MATHOD TABLE.

(refer to the definition macro right after TSMPInstance::DMTest1)

3 . Dialog box execution

- In case the media program uses the dialog box at the execution, it has to send the startDilog and the endDialog to the desk DW_OBJECT before and after the execution.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(refer to TSMPInstance::DMTest1)

4 . Set up for Index

- It transfers the index information of indexArray to the information dialog box.

(refer to TSMPInstance::info2trans.TSMPInstance::information)

- It executes the dialog box to renew the index information.

(refer to TInfoDlg::idIndex. TIndexDlg)

- It set up the renewed index information into the indexArray.

(refer to TSMPInstance::trans2info, TSMPInstance::infomation)

In this sample, it can set up one index for one media only. However it is possible to create multiple indexes for one media since the indexArray of variable length.

5 . copy and Paste of the media

- It define the variable to store the specified clipboard form for own media.

(refer to the head portion of sum_a.cpp)

- It creates the DMCopy, DW_MEASSAGE response function, and copy the media information into the clipboard in the environment.

(refer to TSMPInstance::create)

In this example, the location of paste is shifted below to the right.(The location is controlled by the band variable paste Count)

6 . Avoiding the influence of the background transparent media

There are case when the background disturb the display of type A media if users call the setWindowsPos function of Windows API for Type A media display.

7 . Get the page color

If it is necessary to have a color on the page when transparent media, get the color from the book Win (protect member of TMediaInstance that has a window for the page).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ลอกออกให้วงไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(example)COLORREFbookColor=(COLORREF)bookWin>GetWindowLong(0);

8. Utilization of notice message to parent windows

As the event to the media, there is a case when users want to users the notice message from control to the parent windows. an example is the case of button control, where users want to pick up the notice of BN_CLICKED. In this case, the media view are created as follows:

- Create the media view as a sub-class of TFrameWindows.
- Paste the control that is to be used as client of the media view.

9.3. Type B Media Sample

The "Type B media" is the media that does not have its own windows but draws on the windows of the book. It does not accept the event such as mouse click but it uses smoothly superimpose function for multiple media. Refer to section 2 for type A media and type B media.

•Function Overview

- Draw the picture in some random from in the random color.
- Background of the media is always transparent.
- There is an individual media information for own media (Brightness of RGB and number of apex).
- It treats DB_MESSAGE of show and hide
- It can set up the index
- It is possible to copy and paste the media.

•Main class

TMedia Class sub class

TSMPclass

TMedia Instance sub class

TSMPInstance

Media view

none

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information Dialog	TInfodlg
Index Dialog	TIndexDlg

- File Construction

Source program	SMP_B.CPP
Header file	SMP_B.H
Resource source	SMP_B.RC
Definition file	SMP_B.DEF

- The matters to be referenced

The creation of type B media is basically the same as type A. Here are the description of major differences from type A media sample.

1 . Treatment of media Win

Since type B media does not create its own Windows, make sure to set NULL into media Win When creating media.

(refer to TSMPInstance::create)

2. Display of media

The display of media is done by draw message (DW_MESSAGEAGE).

It is show by rectangle and HDC that are transferred by arguments of draw message.

(refer to TSMPInstance::DMDraw)

3 . Response to show and hide

It is not necessary to describe the most of time for type A media since MEDIA-SDK processes show and hide. However, it need to be overridden for type B media.Asdescribed in the example, the re-write of type B media is always done by draw message. This ir true when changing the display contents by own DW_MESSAGE at execution.

- Override DMshow and DMHide of TMediaInstance

- Set the show Flag of header in DMSHow and DMHide, and make the windows portion of to be invalid. (make a state that the rewrite is necessary.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(refer to TSMPInstance::DMSHow, TSMPInstance::DMHine)

4 . Superimpose of media

To display correctly the superimposed media, the display and deletion of the media should be managed by Huress-Book. When redrawing become necessary for the media , the user should write the program so that rewrite doe not take place but instead informs necessity of redrawing the book Windows.

(refer to TSMPInstance::information. It redraws since the display contents and location of media may change up on the execution of the information dialog box.)

5 . Load and Save pf individual mrdia information

The media have individual media information other than the header of TMediaInstance. (MediaHeader2 struction) To load and save thesmedia, it is necessary to load and save the individual media information as well.

(refer to TSMPInstance::load, TSMPInstance::saveRefer to)

6 . Copy and Paste of media

The media have individual media information other than the header of TMediaInstance. (MediaHeader2 struction) to copy and paste these media, it is necessary to copy and paste the individual media information as well.

(refer to TSMPInstance::dmCopy, TSMPInstance::create)

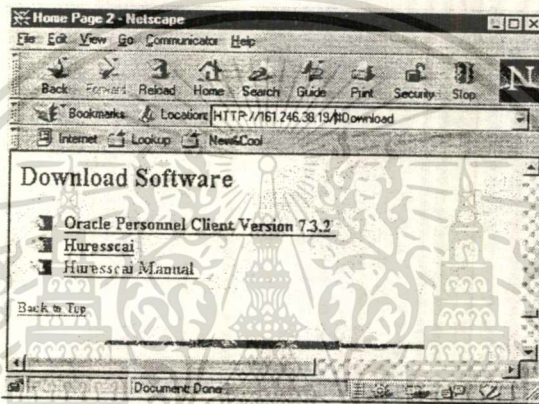


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม HURESS

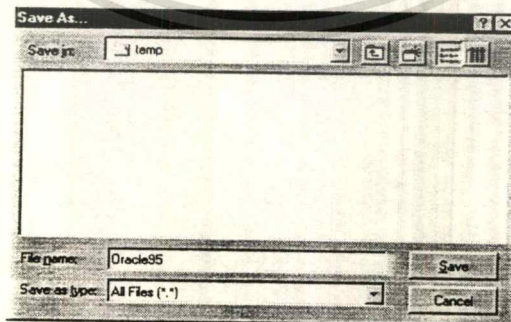
1. การ Download Software

1. เรียกโปรแกรม Netscape หรือ Explorer แล้ว get URL <http://161.246.38.19>
ดังรูปที่ ข.1



รูปที่ ข.1 การ Download Software

2. เลือก Oracle Personnel Client Version 7.3.2 เป็น Software ที่ใช้ติดตั้ง Oracle database ในฝั่ง Client ดังรูปที่ ข.2

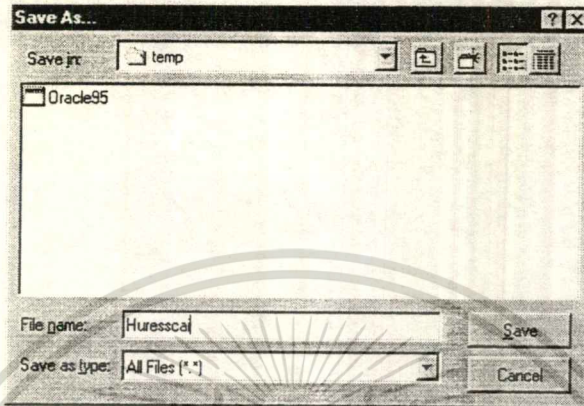


รูปที่ ข.2 ตำแหน่งที่ต้องการเก็บไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

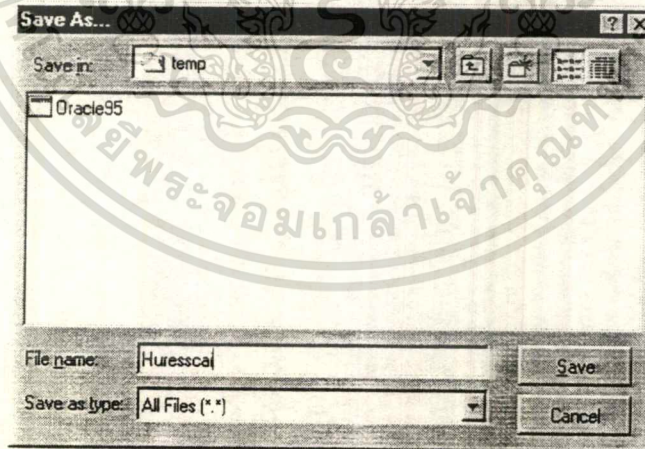
3. เลือก Save และรอนจนกระทั่ง windows dialog ด้านล่างหายไปแสดงว่าเสร็จดังรูปที่

ข.3



รูปที่ ข.3 การบันทึกโปรแกรม

4. เลือก Huresscai เป็น Software ที่ใช้ติดตั้ง HURESS link และการ Setup ค่า parameter ที่สำคัญในการติดต่อกับ Database Server และ Software CAI ดังรูปที่ ข.4

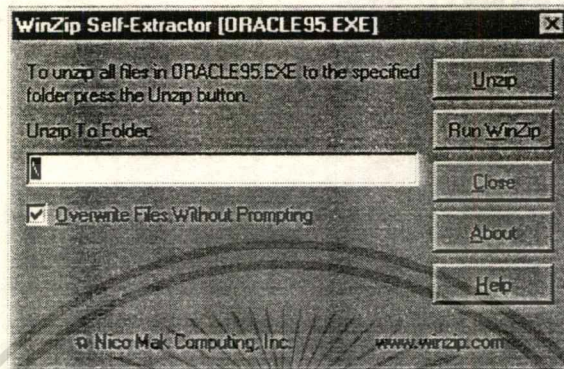


รูปที่ ข.4 ตำแหน่งที่จะเก็บไฟล์

5. เลือก Save และรอนจนกระทั่งเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Extract ไฟล์ Orawin95.exe ซึ่งเป็นกลุ่มของไฟล์ที่ถูก compression โดยใช้ WinZip การ Extract กระทำโดย double click ไฟล์ Orawin95.exe ดังรูปที่ ข.5



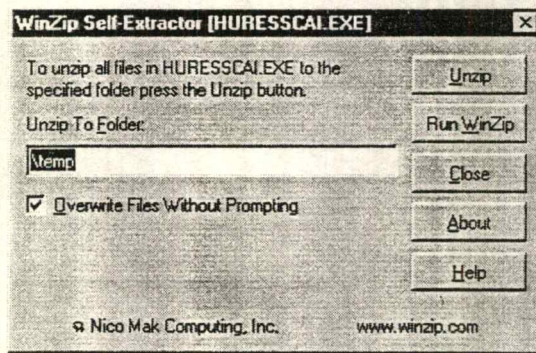
รูปที่ ข.5 การ Extract ไฟล์

จากหน้าจอ ดังรูปที่ ข.5 คลิก Unzip รอจนกระทั่งหน้าจอจะปรากฏ Windows successfully แล้วคลิก OK และคลิก Close บน Windows Self-Extractor



รูปที่ ข.6 ทำการ Extract เสร็จสมบูรณ์

Extract ไฟล์ Huresscai.exe ซึ่งเป็นกลุ่มของไฟล์ที่ถูก compression โดยใช้ WinZip การ Extract กระทำโดย double click ไฟล์ Huresscai.exe ดังรูปที่ ข.7



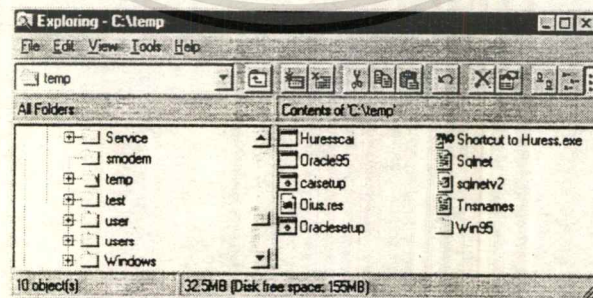
รูปที่ ข.7 การ Extract file

จากหน้าจอคลิก Unzip รอจนกระทั่งหน้าจอจะปรากฏ windows dialog ดังรูปที่ ข.8 จากนั้นคลิก OK แล้วคลิก Close บน Self-Extractor Windows



รูปที่ ข.8 การ Extract เสร็จสมบูรณ์

Subdirectory temp จะปรากฏไฟล์ต่างๆดังรูปที่ ข.9



รูปที่ ข.9 ไฟล์ใน Sub Directory Temp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การติดตั้ง Software

เลือกติดตั้ง Software Oracle ก่อน โดย double click ไฟล์ Oraclesetup ดังรูปที่ ข.10



รูปที่ ข.10 การติดตั้ง ORACLE

Language : English

Company Name : CAI

Oracle Home : C:\ORAWIN95

Installation Options

Application Developer (Complete)

Runtime (Database Only)

Custom

Software Asset Manager

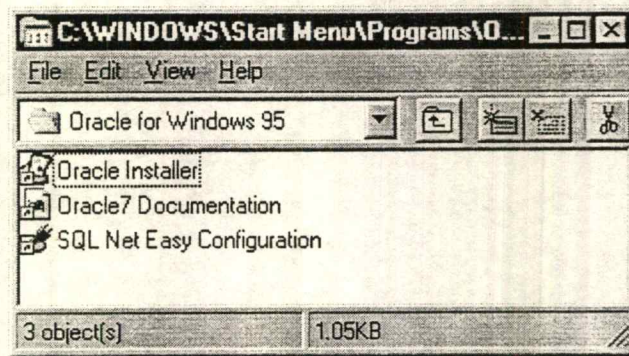
Products available

Oracle Call Interface 7.2.2.3.1

Oracle Installer 3.1.2.1.3

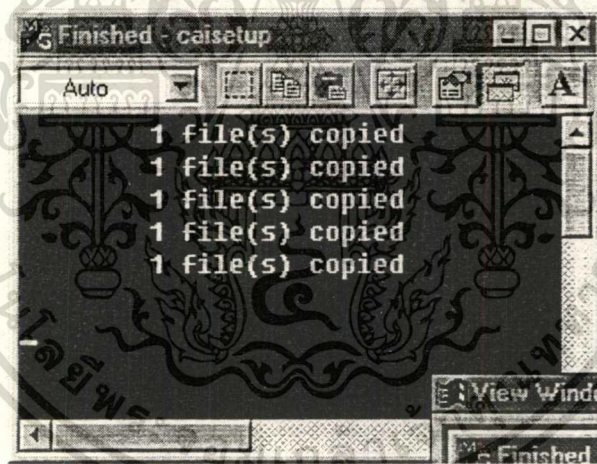
SQL*Net Client 2.2.2.1.0

หลังจากติดตั้ง Oracle เรียบร้อยแล้วจะปรากฏหน้าจอ windows ดังรูปที่ ข.11



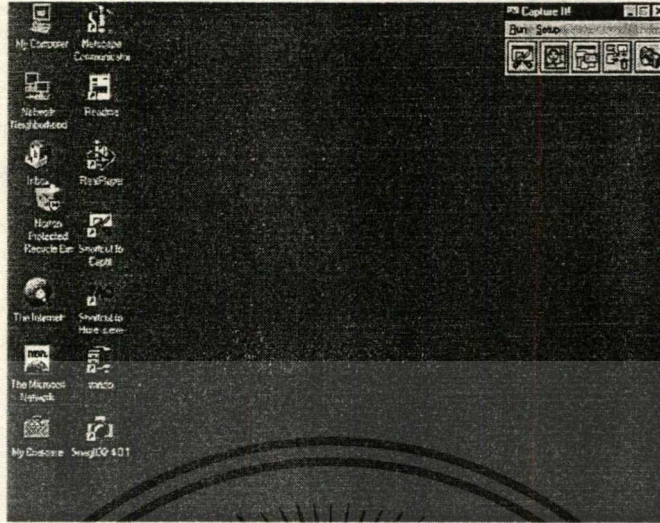
รูปที่ ข.11 การติดตั้ง ORACLE เสร็จสมบูรณ์

การติดตั้ง Software HURESSCAI โดย double click ชื่อไฟล์ caisetup.bat จะปรากฏ หน้าจอ ดังรูปที่ ข.12



รูปที่ ข.12 การติดตั้ง CAI Setup เสร็จสมบูรณ์

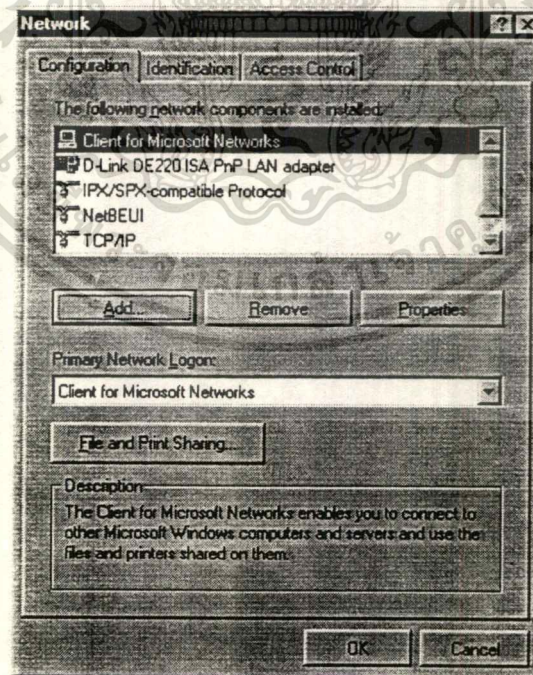
การ move shortcut to HURESS.EXE ไปไว้ยังหน้าจอ desktop ของ windows ดังรูป ที่ ข.13



รูปที่ ข.13 Short Cut ของ HURESS

3. การ Configuration Network Protocol

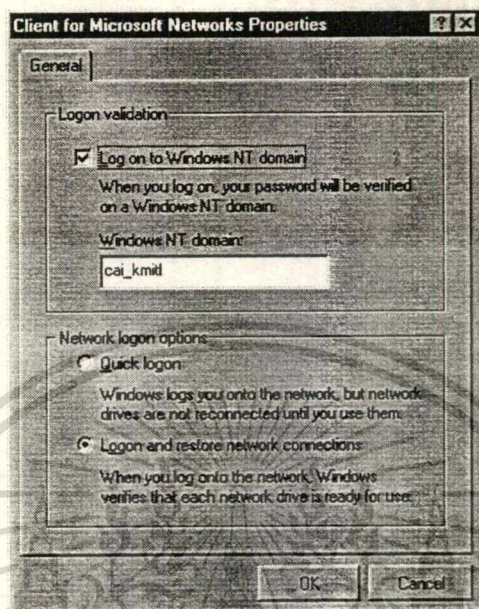
1. กำหนด Domain login จาก Client For Microsoft Networks ดังรูปที่ ข.14



รูปที่ ข.14 Configuration Network

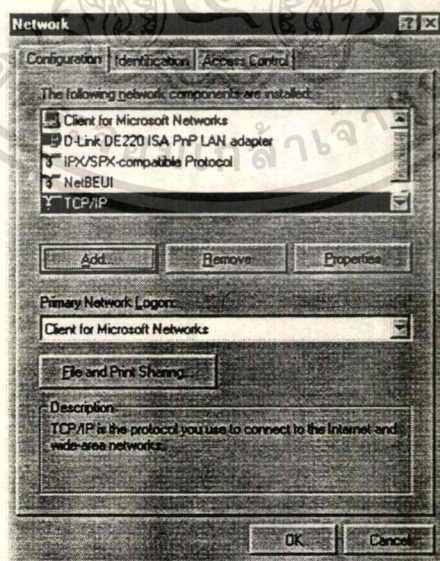
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏดังรูปที่ ข.14 หลังจากนั้นกำหนดค่าตามรูปที่ ข.15



รูปที่ ข.15 Domain Sever

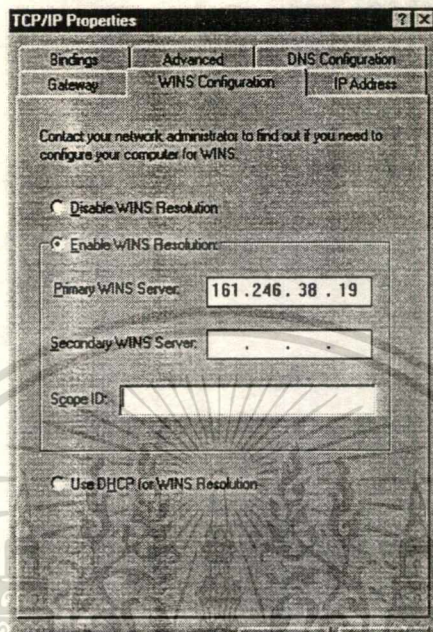
ต่อจากนั้นเลือกรายการ ไปที่ Protocol TCP/IP แล้วคลิกบน Properties ดังรูปที่ ข.16



รูปที่ ข.16 Protocol TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจจะปรากฏปุ่มให้เลือกคลิก mouse ไปที่ WINS ดังรูปที่ ข.17



รูปที่ ข.17 Properties ของ Protocol TCP/IP

ทำการกรอกข้อมูลดังรูปที่ ข.17 โดยให้ใส่ IP Address ของเครื่อง Server แล้ว ทำการ restart เครื่องคอมพิวเตอร์ ต่อจากนั้น login เข้า User ชื่อ student และ password ชื่อ student ชื่อ Domain Name ชื่อ cai_kmitl

บรรณานุกรม

James Bignell and Robert Donovan "Digital Electronics" Delmar Publishers Inc. 1994

Telecommunications Advancement Organization of Japan (TAO) "CAI Materials Making Manual" Software Corporation (SCC). Japan. 1996.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาานิพนธ์	นางสาวจันทิมา บุทเสน
วันเดือนปีเกิด	30 เมษายน 2519
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนาเดิม	92 ถ.สีปศิริ อ.เมือง จ. นครราชสีมา
ที่อยู่ปัจจุบัน	301/85 หมู่บ้านรุ่งอรุณ 2 ถ. ฉลองกรุง เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	-
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนมารีย์วิทยา
มัธยมศึกษาตอนต้น	โรงเรียนมารีย์วิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตภาค ตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คดิพจน์	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวจิราพร สะอาดยิ่ง
วันเดือนปีเกิด	29 พฤษภาคม 2516
สถานที่เกิด	จังหวัดสุรินทร์
ภูมิลำเนาเดิม	1979/2 ถ. สืบศิริ อ.เมือง จ.นครราชสีมา
ที่อยู่ปัจจุบัน	228/45 ถ.อ่อนนุช กรุงเทพฯ
โทรศัพท์	(02) 7390738-41 ต่อ 311
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนลำโรงท่าบวรวิทย์วิทยาเขต
มัธยมศึกษาตอนต้น	โรงเรียนบุรีรัมย์พิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตภาคตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	จงทำวันนี้ให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาววันวิสา วัฒนาศิลป์
วันเดือนปีเกิด	18 พฤษภาคม 2519
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนาเดิม	233/555 ค่ายสุรธรรมพิทักษ์ จ.นครราชสีมา
ที่อยู่ปัจจุบัน	228/45 ถ.อ่อนนุช กรุงเทพฯ
โทรศัพท์	(02) 7390738-41 ต่อ 311
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนรัตโนภาส
มัธยมศึกษาตอนต้น	โรงเรียนสุรธรรมพิทักษ์
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตภาค ตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	ทุนยกเว้นหน่วยกิต ปีการศึกษา 2539
คติพจน์	คนเป็นที่พึ่งแห่งตน

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นายสุรพันธ์ วงษ์ไพศาล
วันเดือนปีเกิด	3 สิงหาคม 2518
สถานที่เกิด	กรุงเทพมหานคร
ภูมิลำเนาเดิม	017/3 ถ. สนามเป้า ต. ท่าพระคู่อ.เมือง จ.ระยอง
ที่อยู่ปัจจุบัน	621/89 บุคลรัตนคอนโค เขตลาดกระบัง กรุงเทพ 10520
โทรศัพท์	01-6217091
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเทศบาลวัดโชดทิมธาราม
มัธยมศึกษา	โรงเรียนระยองวิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคระยอง
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	ทุนยกเว้นหน่วยกิต ปีการศึกษา 2539
คตินิพนธ์	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้