

เกมต่อภาพบนคอมพิวเตอร์

JIGSAW ON COMPUTER



ธนวัฒน์ จิรจิตติกาล
ยวุธลักษณ์ แซ่ฮ่วย
สมศักดิ์ ยิ่งสุทธิพันธ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่.....
เลขทะเบียน 39673
วัน, เดือน, ปี 19 ส.ย. 2544

ปีการศึกษา 2543

.b.....
.i.....

JIGSAW ON COMPUTER

THANAWAT JIRATTIKARN

YUVALUK SAENGUE

SOMSAK YINGSUTTIPHAN

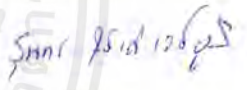
**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2000**

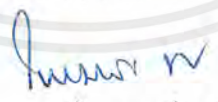
หัวข้อปัญหาพิเศษ เกมต่อภาพบนคอมพิวเตอร์
 JIGSAW ON COMPUTER

ชื่อนักศึกษา นายธนวัฒน์ จิรจิตติกาล 40056034
 นางสาวยูวลักษณ์ แซ่ฮ่วย 40056062
 นายสมศักดิ์ ยิ่งสุทธิพันธุ์ 40056091

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
 สาขาวิชา วิทยาการคอมพิวเตอร์
 ปีการศึกษา 2543
 อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2543

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผู้ช่วยศาสตราจารย์สุนทร สุชาติเวชภูมิ	  
กรรมการ	อาจารย์อิสระ บุรินทร์รามาศย์	
กรรมการและอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	


 (ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)
 หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	เกมต่อภาพบนคอมพิวเตอร์	
ชื่อนักศึกษา	นายธนวัฒน์ จิรจิตติกาล	40056034
	นางสาวยุวลักษณ์ แซ่ฮ่วย	40056062
	นายสมศักดิ์ ยิ่งสุทธิพันธุ์	40056091
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2543	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	

บทคัดย่อ

ปัจจุบันในวงการอุตสาหกรรมการผลิตโปรแกรมเกมคอมพิวเตอร์ ได้มีการปรับตัวอย่างรวดเร็ว และมีผู้ผลิตมากมายในตลาด เพราะผู้ผลิตต่างๆได้นำเทคโนโลยีขั้นสูงเข้ามาช่วยในการพัฒนาคุณลักษณะและเทคนิคในเกม ดังนั้นทำให้คณะศึกษาได้เลือกศึกษาโครงการเกี่ยวกับการสร้างโปรแกรมเกมสำหรับคอมพิวเตอร์ ซึ่งในมุมมองของทีมงาน โปรแกรมเกมจะไม่ได้วัดกันด้วยแง่ธุรกิจ หรือยอดขาย หรือความนิยมของผู้เล่นเพียงอย่างเดียว แต่จะต้องวัดที่ความท้าทาย ความแปลกใหม่ และการตอบสนองระหว่างเกมกับผู้เล่น ซึ่งคณะผู้ศึกษาได้ทำการศึกษาวิจัยเทคนิคต่างๆจากโปรแกรมอื่นๆในท้องตลาด

พร้อมกันนี้คณะผู้ศึกษาจะได้นำการเสนอกระบวนการพัฒนาที่เป็นลักษณะแนวความคิดเฉพาะแบบ โดยเกมจะจำลองสถานการณ์คล้ายกับการต่อตัวจิกซอว์โดยทั่วไป ซึ่งผู้เล่นจะต้องพยายามทำการต่อตัวจิกซอว์นั้นจนกว่าจะสมบูรณ์ ซึ่งจะทำให้ผู้เล่นได้ฝึกไหวพริบและปฏิภาณตลอดเวลา

Special Project Title	JIGSAW ON COMPUTER	
Students	Mister Thanawat Jiratthitikarn	40056034
	Miss Yuvaluk Saengue	40056062
	Mister Somsak YingsutthiPhan	40056091
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Sciences , Faculty of Science	
Programme	Computer Sciences	
Academic Year	2000	
Special Project Advisor	Assistant Professor Teerawat Prakobphon	

ABSTRACT

The game software industry had fast development and more competitors in the market. Because they use high technology for develop feature and technique in game. With This reason, it had more challenge for us for choose project about build game software. With our focus, The game software did not performance by business or sales volume or popular game only. But we will performance with the new challenge, new style, and interactive of game that we have research technique from other products in market.

Our game software project will be create new challenge, new style, and interactive between game and player with difference from others. We will present with the special development system in our unique concept. Game will be simulate reality from the real jigsaw. Player will try to complete each of jigsaw like the format picture . With this reason, Player will be practice their strategy every times.

กิตติกรรมประกาศ

กราบขอบพระคุณบุพพการี ผู้อุปการะ ผู้ปกครอง ที่เลี้ยงดู อุ้มชู จนพบเราพบกับวันแห่งความสำเร็จเช่นวันนี้

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล ซึ่งเป็นอาจารย์ที่ปรึกษาปัญหาพิเศษที่มอบความเป็นกันเอง และความเอาใจใส่ ทำให้พวกเรามีกำลังใจและไม่ท้อแท้

ขอขอบพระคุณอาจารย์ทุกท่านที่มอบความเมตตา และให้คำปรึกษาด้วยดีตลอดมา



คณะผู้จัดทำ
มีนาคม 2543

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 สมมติฐานของการศึกษา	1
1.4 ขอบเขตการศึกษา	1
1.5 ขั้นตอนการศึกษา	2
1.5.1 ศึกษาเกี่ยวกับระบบการสร้างเกม	2
1.5.2 ศึกษาโปรแกรมและเครื่องมือต่างๆ	2
1.5.3 เก็บรวบรวมเอกสารและข้อมูลต่างๆ	2
1.5.4 วิเคราะห์และออกแบบระบบเกม	2
1.5.5 พัฒนาโปรแกรม	2
1.5.6 การทดสอบโปรแกรมและปรับปรุงโปรแกรม	2
1.6 ข้อตกลงเบื้องต้น	2
1.7 ข้อยกเว้นของการศึกษา	3
1.8 คำจำกัดความที่ใช้ในการศึกษา	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	4
2.1 โปรแกรม Microsoft Visual Basic 6.0	4
2.1.1 ความเป็นมา	4
2.1.2 หลักการของ Microsoft Visual Basic 6.0	4

2.1.3 ส่วนประกอบของ Microsoft Visual Basic 6.0	5
2.2 Control ต่างๆที่ใช้ในการทำงาน	6
2.2.1 ฟอรัม	6
2.2.2 เมนู	7
2.3 การเขียนโปรแกรมบน Windows	10
2.3.1 Win 32 Libraries	11
2.3.2 ตัวแปรและการประกาศตัวแปร	11
2.3.3 ค่าคงที่และการประกาศค่าคงที่	12
2.3.4 ชนิดของข้อมูลที่นำมาใช้	12
2.3.5 คำสั่งที่ใช้ในการควบคุมการทำงาน	13
2.3.6 หน่วยวัดในระบบกราฟฟิคของ Microsoft Visual Basic 6.0	15
2.4 การจัดการ Mouse	17
2.4.1 การจัดการอีเวนต์ของ Mouse	17
2.4.2 การตรวจสอบสถานะต่างๆของ Mouse	17
2.4.3 การตรวจสอบการกดปุ่ม Mouse	17
2.4.4 การตรวจสอบสถานะของปุ่ม <Shift> ,<Ctrl> และ <Alt> ...	18
2.4.5 อีเวนต์ที่เกี่ยวกับ Drag and Drop	19
2.5 Windows API	20
2.5.1 ข้อดีในการใช้ DLL	21
2.5.2 ส่วนประกอบของ Windows API	21
2.5.3 วิธีการใช้งาน โปรแกรม API Text Viewer	22
2.5.4 การเรียกใช้ Windows API	22
2.5.5 คุณสมบัติของ Control ที่ต้องการเรียกใช้ใน Windows API	24
2.5.6 การเรียกใช้ข้อมูลเมื่อใช้ Windows API	24
2.6 วงจรเกม	25
2.7 โปรแกรม Photo shop 5.5	26
2.7.1 ประวัติความเป็นมา	26
2.7.2 ความสามารถของโปรแกรม Photo shop	26
2.7.3 ส่วนประกอบของโปรแกรม Photo shop	26
บทที่ 3 ขั้นตอนการดำเนินและวิจัยงาน	28
3.1 ขั้นตอนการออกแบบ	29

3.1.1 การออกแบบระบบเกม	29
3.1.2 การออกแบบระบบเกมที่ใช้ติดต่อกับผู้เล่น	31
3.1.3 การออกแบบวิธีการโต้ตอบของฝ่ายคอมพิวเตอร์	32
3.2 ขั้นตอนการเขียนโปรแกรม	33
3.2.1 ขั้นตอนการแสดงรายละเอียดของตัวเกม	34
3.2.2 ส่วนแสดงการเก็บรูปแบบ	35
3.2.3 ส่วนหลักที่ควบคุมลักษณะการเล่นของตัวเกม	38
3.2.4 ส่วนที่ใช้ในการใส่ชื่อผู้เล่นที่ติด 10 อันดับแรกในแต่ละระดับ	48
3.2.5 ส่วนที่ใช้แสดงผู้ที่ได้คะแนนอันดับสูงสุด 10 อันดับแรก	49
3.3 การทำไฟล์เพื่อติดตั้งโปรแกรม	49
3.3.1 การ Packaging	49
3.3.2 การ Deployment	50
บทที่ 4 ผลการทดลองและวิเคราะห์ปัญหา.....	51
4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็น	51
4.2 ขั้นตอนการทดสอบการรันและการประมวลผลภายใต้ระบบที่กำหนด	52
4.3 ขั้นตอนการตรวจสอบการใช้คำสั่งบนเมนูบาร์และการทำงานตามคำสั่งต่างๆ	53
4.4 ขั้นตอนการทดสอบการหาข้อผิดพลาดของโปรแกรม	54
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	55
5.1 ขั้นตอนการออกแบบ	55
5.2 ขั้นตอนการเขียนโปรแกรม	55
5.3 ขั้นตอนการตกแต่งหน้าจอและรายละเอียดต่างๆ	55
ภาคผนวก ก. คู่มือการเล่น	57
ภาคผนวก ข. คุณสมบัติของโปรแกรม	65
บรรณานุกรม	70

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงคำสั่งของฟอร์มใน Microsoft Visual Basic 6.0	6
2.2 แสดงเมธอดและอีเวนต์ในฟอร์ม	7
2.3 แสดงคุณสมบัติของเมนู	7
2.4 แสดงคุณสมบัติที่สำคัญของ PictureBox	9
2.5 แสดงชนิดของข้อมูลและขอบเขตของตัวแปร	12
2.6 แสดงหน่วยวัดของระบบพิกัดของ Control	16
2.7 แสดงค่าคงที่ของอีเวนต์การกดปุ่มเมาส์	18
2.8 แสดงค่าของการกด <Shift>, <Ctrl> และ <Alt>	18
2.9 แสดงอีเวนต์เกี่ยวกับ Drag and Drop	19
2.10 แสดงค่าพารามิเตอร์ที่ใช้ในการ Drag and Drop แบบ Manual	20
2.11 แสดงการประกาศพารามิเตอร์ต่างๆ	23
2.12 แสดงคุณสมบัติของคอนโทรลที่ต้องการเรียกใช้ใน Windows API	24
2.13 แสดงชนิดข้อมูลของ Windows API เปรียบเทียบกับชนิดข้อมูลใน VB 6.0	24
4.1 แสดงการใช้คำสั่งบนเมนูบาร์และผลการทำงานตามคำสั่ง.....	53

สารบัญภาพ

ภาพที่	หน้า
2.1 แสดงหน้าจอของ Microsoft Visual Basic 6.0	5
2.2 แสดงส่วนประกอบต่างๆของเมนู	8
2.3 แสดงระบบการส่ง Message ใน Windows	10
2.4 แสดง Properties ของฟอร์มซึ่งแสดงหน่วยวัดในระบบกราฟฟิก	15
2.5 แสดงการกำหนดหน่วยวัดเอง	16
2.6 แสดงการตรวจสอบการกดปุ่มเมาส์	17
2.7 แสดงการตรวจสอบสถานะของปุ่ม <Shift> , <Ctrl> และ<Alt>	18
2.8 แสดงหน้าจอหลักของโปรแกรม API Text Viewer	21
2.9 แสดงวงจรเกม	25
2.10 ส่วนประกอบของ Adobe Photo shop 5.5	27
3.1 แสดงขั้นตอนที่กว้างแผน	28
3.2 แสดงรูปเมาส์ที่ใช้ในขณะที่เล่นเกม	31
3.3 แสดงรูปเมาส์ขณะทำการเลือกตัวต่อจิกซอร์ว	31
3.4 แสดงส่วนติดต่อกับผู้เล่น	32
3.5 แสดงขั้นตอนการเขียนโปรแกรม	33
3.6 แสดงการเปลี่ยน Tab ของ SSTab	35
3.7 แสดงการทำงานการบันทึกข้อมูลรูปภาพลงแฟ้มข้อมูลรูปภาพ	38
3.8 แสดงการเล่นไฟล์เพลงที่ใช้ในเกม	40
3.9 แสดงการทำงานของฟังก์ชัน Helper	43
3.10 แสดงการทำงานของการสุ่มตำแหน่งที่วาง	44
3.11 แสดงการตรวจสอบตำแหน่งที่วางตัวต่อจิกซอร์ว	45
3.12 แสดงตารางแทนตำแหน่งจิกซอร์ว 48 ชิ้น	48
4.1 แสดงภาพหน้าจอก่อนเข้าสู่เกม	52
4.2 แสดงหน้าจอการเริ่มเล่นเกม	52

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันอุตสาหกรรมเกมได้มีการพัฒนาและเติบโตอย่างรวดเร็วจากการคาดการณ์ในอนาคตอุตสาหกรรมเกมจะเป็นอุตสาหกรรมที่มีการแข่งขันทางด้านการตลาดที่มีมูลค่าสูงแต่ในปัจจุบันยังไม่ได้มีการพัฒนาอุตสาหกรรมเกมที่เกี่ยวข้องกับวัฒนธรรม และประเพณีไทยของเรา ดังนั้นเราจึงควรที่จะทำการพัฒนาอุตสาหกรรมเกมที่เกี่ยวข้องกับวัฒนธรรม และประเพณีให้เจริญมากยิ่งขึ้น เพราะฉะนั้นการศึกษาวิธี และหลักในการสร้างระบบเกมจึงเป็นที่ที่น่าสนใจ โดยที่ผู้ศึกษาจะได้รับความรู้เกี่ยวกับวิธีการสร้างระบบเกม วิธีการติดต่อกับ ฮาร์ดแวร์ อินพุต เอาท์พุต รวมถึงวิธีการใช้ เครื่องมือต่างๆ ในการสร้างรูปภาพ ภาพเคลื่อนไหว อีกทั้งยังสามารถเขียน โปรแกรมเกม เพื่อตอบโต้กับผู้เล่นได้ในหลายรูปแบบอีกด้วย โดยหวังว่าจะเป็นประโยชน์ต่อการนำไปศึกษาต่อเพื่อเป็นแนวทางเบื้องต้นสำหรับผู้สนใจการพัฒนาเกม และยังเป็นแรงผลักดันให้เกิดอุตสาหกรรมการผลิตเกมในประเทศไทย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อศึกษาระบบและวิธีการสร้างเกม
- 1.2.2 เพื่อเป็นแนวทางผลักดันให้เกิดอุตสาหกรรมการผลิตเกมในประเทศไทย
- 1.2.3 เพื่อศึกษาการเขียนโปรแกรมบน Windows ด้วยเทคโนโลยีการแสดงผลภาพและประมวลผลภาพโดยใช้โปรแกรม Microsoft Visual Basic 6.0
- 1.2.4 เพื่อเป็นการเผยแพร่วัฒนธรรมและประเพณีของไทย

1.3 สมมติฐานของการศึกษา

- 1.3.1 เกมสามารถโต้ตอบกับผู้เล่นได้หลายรูปแบบ
- 1.3.2 เป็นประโยชน์ต่อผู้ที่นำไปศึกษาเป็นแนวทางต่อไป
- 1.3.3 สามารถเขียน โปรแกรม แสดงภาพเคลื่อนไหว และโต้ตอบกับผู้เล่นได้

1.4 ขอบเขตการศึกษา

- 1.4.1 การพัฒนาโปรแกรมทำได้โดยโปรแกรม Microsoft Visual Basic 6.0
- 1.4.2 ใช้ Windows API เป็น Library สำคัญในการพัฒนา

1.4.3 ศึกษาวิธีการวาดภาพด้วย Paint Picture ซึ่งเป็นการวาดภาพที่มีประสิทธิภาพของ Visual Basic 6.0

1.4.4 ศึกษาวิธีการแต่งรูปภาพกราฟฟิกโดยใช้โปรแกรมตกแต่งภาพ Adobe Photoshop 5.5

1.5 ขั้นตอนการศึกษา

1.5.1 ศึกษาเกี่ยวกับระบบการสร้างเกม

เป็นขั้นตอนการศึกษา วิธี และหลักการที่ใช้ในการสร้างเกมรวมทั้งเทคนิคต่างๆที่จะนำมาเขียน และพัฒนาระบบเกม รวมถึงวิธีการวาดภาพและการออกแบบเกม

1.5.2 ศึกษาโปรแกรมและเครื่องมือต่างๆ

เป็นการศึกษาโปรแกรมและเครื่องมือ ที่จะนำมาใช้และพัฒนาในการเขียนโปรแกรม โดย โปรแกรมและเครื่องมือที่จะนำมาใช้ในการพัฒนาจะเน้น Microsoft Visual Basic6.0 , Windows API และโปรแกรมทางด้านกราฟฟิก เพื่อนำมาใช้ในการสร้างภาพ โดยจะเน้นที่โปรแกรม Paintshop3.0 , Adobe Photo shop 5.5 เป็นต้น

1.5.3 เก็บรวบรวมเอกสารและข้อมูลต่างๆ

เป็นการนำเอาเอกสารและข้อมูลที่เกี่ยวข้องที่ใช้ในการพัฒนาโครงการมารวบรวมและใช้ ประกอบการทำงาน โดยส่วนมากจะเป็นการรวบรวมจากหนังสือ ตำรา และเว็บไซต์ที่เกี่ยวข้องรวมถึงข้อมูลที่ได้จากผู้ที่มีความรู้ในแต่ละด้าน

1.5.4 วิเคราะห์และออกแบบระบบเกม

วิเคราะห์และออกแบบระบบงานโดยแบ่งออกเป็นส่วนๆ เช่น ส่วนรับข้อมูล ส่วนประมวลผล ส่วนแสดงผล และออกแบบระบบเพื่อให้ทำงานได้เกิดประสิทธิภาพมากที่สุด

1.5.5 พัฒนาโปรแกรม

เป็นขั้นตอนการเขียนโปรแกรมตามขั้นตอนที่ได้ออกแบบไว้ในขั้นตอนการวิเคราะห์และออกแบบระบบ

1.5.6 การทดสอบโปรแกรมและปรับปรุงโปรแกรม

เป็นขั้นตอนการทดสอบโปรแกรมและบอกถึงความสามารถทั้งหมดที่เป็นไปได้ของโปรแกรม

1.6 ข้อตกลงเบื้องต้น

1.6.1 โปรแกรมรันเฉพาะระบบปฏิบัติการ Windows ตั้งแต่ Version 95 ขึ้นไป

1.6.2 ต้องการคอมพิวเตอร์ที่มี Processor ตั้งแต่ระดับ Pentium ความเร็ว 166 MHz ขึ้นไป

1.6.3 มีเนื้อที่ในหน่วยความจำอย่างน้อย 50 MB.

1.6.4 ต้องมี Hardware input ประเภท Mouse และ Keyboard

1.7 ข้อจำกัดของการศึกษา

1.7.1 ระยะเวลาที่ทำการศึกษา

1.7.2 ความรู้ความเข้าใจที่ใช้ในการศึกษา

1.8 คำจำกัดความที่ใช้ในการศึกษา

1.8.1 Driver ตัวขับโปรแกรมควบคุมอุปกรณ์ หรือโปรแกรมอื่นๆ

1.8.2 API ย่อมาจาก Application Programming Interface เป็นโปรแกรมย่อยที่เป็นมาตรฐานของระบบปฏิบัติการ Windows ที่ Application จะต้องเรียกใช้เพื่อให้ระบบปฏิบัติการบางอย่างใช้งาน

1.8.3 GDI ย่อมาจาก Graphic Device Interface หมายถึงส่วนการติดต่อกับอุปกรณ์การแสดงผลทางด้าน Graphic

1.8.4 Compiler หมายถึงตัวแปลงโปรแกรม

1.8.5 Bitmap คือภาพที่ประกอบจากจุดเล็กๆจำนวนมาก

1.8.6 File Binary คือไฟล์ใดไฟล์หนึ่งที่มีนามสกุล fix เป็นไฟล์แบบ Binary มีไว้สำหรับเก็บข้อมูลเกี่ยวกับภาพต่างๆที่เราได้ใช้งานบนฟอรัมนั้น

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 โปรแกรมไมโครซอฟท์ Visual Basic

2.1.1 ความเป็นมา

ในปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์ได้ก้าวหน้าอย่างรวดเร็วจากอดีตที่ผ่านมาทางด้านฮาร์ดแวร์และ ซอฟต์แวร์ โดยฮาร์ดแวร์นั้นเป็นอุปกรณ์ที่ประกอบเป็นคอมพิวเตอร์ เช่น CPU , ฮาร์ดดิสก์ เป็นต้น ส่วนซอฟต์แวร์ นั้นเป็นชุดคำสั่งที่ทำให้คอมพิวเตอร์ทำงานได้ ซึ่งมีการเปลี่ยนแปลงอย่างรวดเร็วมาก และในปัจจุบันมีเครื่องมือต่างๆที่ช่วยในการพัฒนาโปรแกรมอย่างมากมาย ซึ่งหนึ่งในเครื่องมือต่างๆนี้ก็มีเครื่องมืออยู่อย่างหนึ่งที่มีการใช้งานกันอย่างแพร่หลายเรียกว่า Microsoft Visual Basic 6.0 ที่ช่วยให้การพัฒนา Software ต่างๆ เป็นไปได้อย่างสะดวกรวดเร็ว รวมทั้งยังสามารถพัฒนา Software ได้หลายอย่างด้วยกัน ตั้งแต่โปรแกรมธรรมดาทั่วไป โปรแกรมเกี่ยวกับฐานข้อมูล หรือโปรแกรมทางอินเทอร์เน็ต และ โปรแกรมเกมต่างๆ เป็นต้น

ก่อนทำความรู้จักกับ Microsoft Visual Basic 6.0 เราต้องทราบความหมายของการเขียนโปรแกรมก่อน ซึ่งการเขียนโปรแกรมก็คือการสั่งให้คอมพิวเตอร์ทำงานตามที่เรต้องการ เช่น โปรแกรมเกมโดยทั่วไป เป็นโปรแกรมที่สั่งให้คอมพิวเตอร์ได้ตอบกับการกดแป้นคีย์บอร์ดและการ Click Mouse จากผู้เล่นเกม เป็นต้น

สำหรับ Microsoft Visual Basic 6.0 เป็นเครื่องมือในการสร้างโปรแกรมบนระบบปฏิบัติการ Windows ที่ใช้งานง่าย โดยการสร้างโปรแกรมใน Microsoft Visual Basic 6.0 นั้นจะเป็นการเลือกเครื่องมือต่างๆมาออกแบบหน้าจอของโปรแกรมที่เราจะสร้าง ซึ่งเราเรียกการเขียนโปรแกรมลักษณะนี้ว่า Visual Programming การเขียนโปรแกรมแบบนี้เราจะไม่จำเป็นต้องเขียนคำสั่งต่างๆมากนักก็สามารถสร้างโปรแกรมได้อย่างรวดเร็ว

2.1.2 หลักการของ Microsoft Visual basic 6.0

สำหรับในหัวข้อนี้เราจะกล่าวถึงพื้นฐานเกี่ยวกับหลักการที่ควรสร้างในการพัฒนาโปรแกรมด้วย Microsoft Visual Basic 6.0 โดยมีสิ่งที่เราควรรู้จัก คือ Control คุณสมบัติ และการเขียนโปรแกรมแบบ Event-Driven สำหรับหลักการต่างๆเหล่านี้จะเป็นสิ่งใหม่สำหรับผู้เริ่มเขียนโปรแกรมบน Windows จึงควรที่จะศึกษาถึงสิ่งเหล่านี้ให้ละเอียดเนื่องจากมีความสำคัญอย่างมากในการใช้ Microsoft Visual Basic 6.0 ซึ่งหลักการที่สำคัญนั้นได้ถูกแบ่งออกเป็น 2 ขั้นตอนหลักได้แก่

2.1.2.1 การออกแบบหน้าจอของโปรแกรม ซึ่งเป็นส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้เรียกว่า User Interface

2.1.2.2 การเขียนโปรแกรมซึ่งใน Microsoft Visual Basic 6.0 เป็นการกำหนดคุณสมบัติของ Control บนฟอร์มให้เหมาะสม และการเขียนคำสั่งตอบสนองต่อ Event

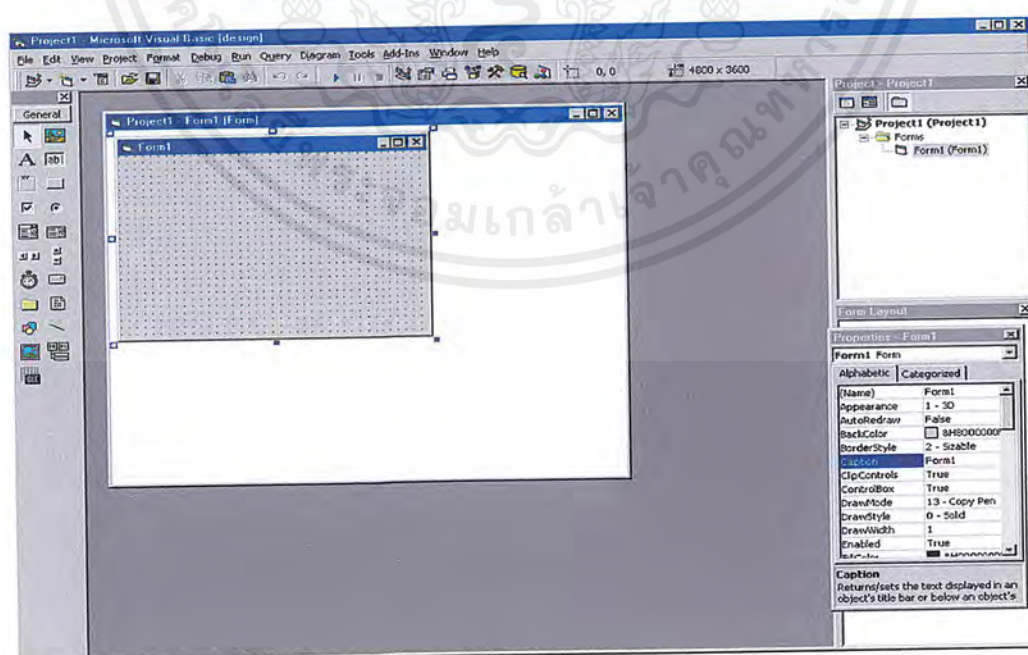
Control เป็นเครื่องมือในการออกแบบหน้าจอโปรแกรมซึ่งเครื่องมือต่างๆเหล่านี้ Microsoft Visual Basic 6.0 ได้เตรียมไว้ใน Tool box แล้ว โดยเราเลือก Control ที่ตรงกับจุดประสงค์ในการใช้งานและนำมาวางบนฟอร์มที่ปรากฏอยู่

คุณสมบัติ คือ ลักษณะต่างๆของ Control ที่ถูกนำมาวางบนฟอร์ม ที่เราสามารถกำหนดได้ เช่น รูปแบบของ Control หรือการกำหนดความกว้าง ความยาวของ Control รวมทั้งการแสดงผล Control เป็นต้น

การเขียนโปรแกรมแบบ Event-Driven คือการใช้คำสั่งกำหนดให้ Control ตอบสนองต่อเหตุการณ์บางอย่างที่เกิดขึ้น (เรียกว่า Event) เมื่อโปรแกรมที่สร้างถูกนำมาใช้งาน เช่น ถ้าเราต้องการให้โปรแกรมเกิดการตอบสนองเมื่อมีการ Click Mouse ในการเคลื่อนย้ายตัวเกมจากที่หนึ่งไปยังอีกที่หนึ่ง

2.1.3 ส่วนประกอบของ Microsoft Visual Basic 6.0

เมื่อทำการรันโปรแกรม Microsoft Visual Basic 6.0 ขึ้นมาโดยทั่วไปแล้วเราจะเลือก Standard.exe ซึ่งเป็นการใช้ Microsoft Visual Basic 6.0 ในการเขียนโปรแกรมที่รันบน Windows ทั่วไป โดยมีชื่อของส่วนประกอบต่างๆที่เราจำเป็นต้องทราบในตอนนี้ ดังรูปที่ 2.1



รูปที่ 2.1 แสดงหน้าจอของ Microsoft Visual Basic 6.0

รายละเอียดของส่วนประกอบต่างๆของหน้าจอข้างต้น มีดังต่อไปนี้

2.1.3.1 Menubar เก็บคำสั่งที่เราสามารถใช้งานได้ทั้งหมดใน Microsoft Visual Basic 6.0 ประกอบไปด้วยเมนูทำงานกับ Files, View และ Windows

2.1.3.2 Toolbar ประกอบด้วยปุ่มคำสั่งต่างๆที่ช่วยให้เราใช้งานคำสั่งของ Microsoft Visual Basic 6.0 ได้รวดเร็วยิ่งขึ้น

2.1.3.3 Toolbox เป็นที่แสดงเครื่องมือต่างๆ ที่เราเรียกว่า Control ซึ่งเป็นเครื่องมือที่เราสามารถเลือกไปวางลงบนฟอร์มได้ เพื่อออกแบบหน้าจอของโปรแกรม(เรียกว่าส่วนติดต่อกับผู้ใช้ หรือ User Interface)

2.1.3.4 Project Explorer เป็นหน้าต่างที่แสดงโมดูลต่างๆ ที่มีอยู่ในโปรเจกต์ทั้งหมด เราสามารถกดปุ่ม <CTRL> + <R> เพื่อให้แสดงหน้าต่างนี้ได้

2.1.3.5 Properties Window เป็นหน้าต่างที่แสดงคุณสมบัติของ Control ที่เลือกอยู่ในขณะนั้น สามารถกดปุ่ม <F4> เพื่อให้แสดงหน้าต่างนี้ได้

2.1.3.6 Form ฟอร์มที่ใช้ในการออกแบบ เป็นหน้าต่างที่ใช้ในการออกแบบหน้าจอโปรแกรม

2.2 Controlต่างๆที่ใช้ในการทำงาน

2.2.1 ฟอร์ม(Form)

ฟอร์มจะเป็นเครื่องมือที่เราจะต้องทำงานด้วยบ่อยมาก โดยที่ฟอร์มจะเป็นหน้าต่างที่ผู้ใช้ติดต่อทำงานด้วยผ่านทาง Control ต่างๆที่เราวางบนฟอร์มดังนั้นในหัวข้อนี้จะต้องศึกษาคุณสมบัติเมธอดและอีเวนต์ของฟอร์มที่จำเป็นก่อน

ในขั้นตอนแรก เราจะต้องกำหนดคุณสมบัติของฟอร์ม ซึ่งอาจจะผ่านทาง Properties Window ในขณะออกแบบ หรือกำหนดผ่านทางโปรแกรมในขณะรันก็ได้

2.2.1.1 คุณสมบัติที่สำคัญของฟอร์ม

ฟอร์มมีคุณสมบัติต่างๆ มากมาย ในที่นี้จะขอยกตัวอย่างบางคำสั่งเท่านั้น เช่น

ตารางที่ 2.1 แสดงคำสั่งของฟอร์มใน Microsoft Visual Basic 6.0

ชื่อคุณสมบัติ	คำอธิบาย
Name *	กำหนดชื่อของฟอร์มที่เราใช้อ้างอิงถึงในโปรแกรม
BorderStyle	กำหนดลักษณะในการเปลี่ยนขนาดของฟอร์มเป็นอย่างไร
Caption *	กำหนดข้อความที่แสดงบนไตเติลบาร์ของ

	ฟอร์ม
Moveable	กำหนดว่าฟอร์มจะสามารถเคลื่อนย้ายได้หรือไม่
Enable *	กำหนดให้ฟอร์มสามารถตอบสนองต่อ event ได้หรือไม่

2.2.1.2 เมฆอดและอีเอนต์ที่สำคัญของฟอร์ม

ฟอร์มมีเมฆอด และอีเอนต์ที่เราารู้จัก ดังจะยกตัวอย่างบางส่วน ได้แก่

ตารางที่ 2.2 แสดงเมฆอดและอีเอนต์ในฟอร์ม

ชื่อเมฆอด ,อีเอนต์	คำอธิบาย
Load	เป็นอีเอนต์ที่เกิดเมื่อเรียกฟอร์มขึ้นมาครั้งแรก
Active	เป็นอีเอนต์ที่เกิดขึ้นเมื่อฟอร์มนั้นเป็นฟอร์มที่เราทำงานในขณะนั้น
Click	เป็นอีเอนต์ที่เกิดเมื่อมีการ Click mouse บนฟอร์ม
Hide	เป็นเมฆอดที่ใช้ซ่อนฟอร์มไว้
Print	เป็นเมฆอดที่ใช้ในการลบสิ่งที่วาด หรือพิมพ์บนฟอร์ม

2.2.2 เมนู(Menu)

เมนูเป็นเอกลักษณ์อย่างหนึ่งของโปรแกรมบนWindowsเพราะเป็นที่เก็บคำสั่งต่างๆที่โปรแกรมมีไว้ให้ผู้ใช้ได้ตอบด้วย

2.2.2.1 คุณสมบัติและอีเอนต์ของเมนู

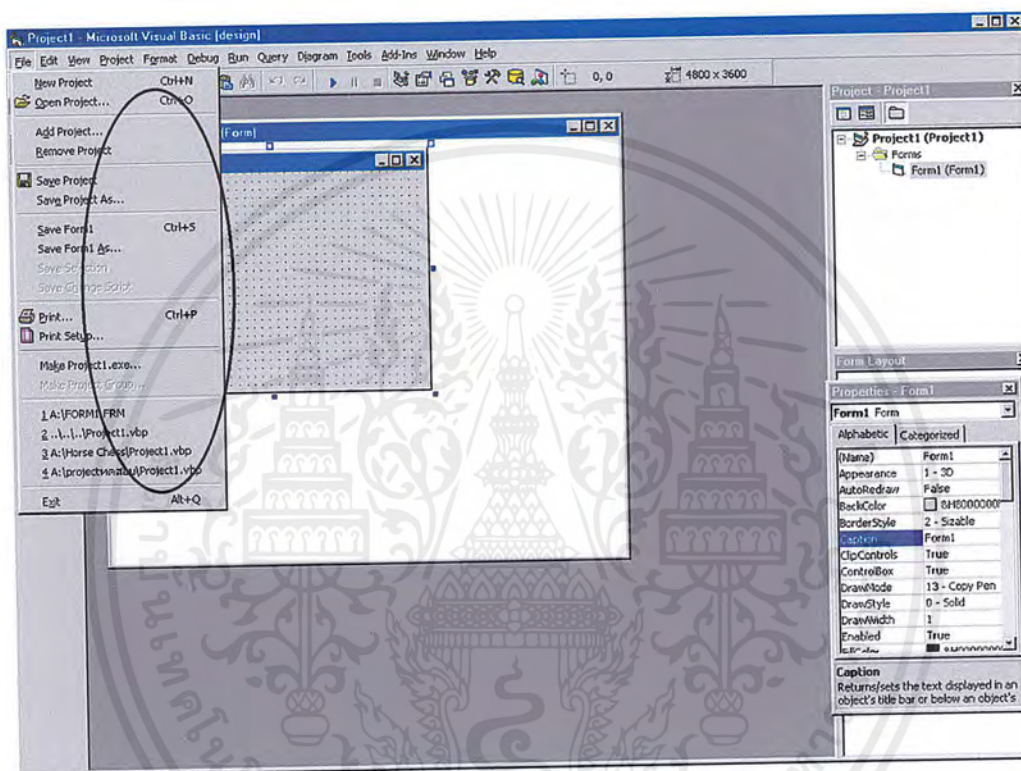
เมนูเป็น Control รูปแบบหนึ่ง ซึ่งมีคุณสมบัติที่สำคัญดังต่อไปนี้

ตารางที่ 2.3 แสดงคุณสมบัติของเมนู

คุณสมบัติ	คำอธิบาย
Name	เป็นชื่อของเมนู
Caption	กำหนดข้อความที่จะให้แสดงบนเมนู
Visible	กำหนดว่าจะให้เมื่อนั้นมองเห็นตอนรันได้หรือไม่

	ไม่
Enabled	กำหนดว่าจะให้เมื่อนั้นตอบสนองต่อผู้ใช้ได้หรือไม่
Checked	ใช้ตรวจสอบว่าเมื่อนั้นถูกเช็คหรือไม่

ซึ่งส่วนประกอบต่างๆของเมนูแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 แสดงส่วนประกอบต่างๆของเมนู

2.2.2.2 Menu Editors

เราสามารถใช้ Menu Editor สร้างเมนูที่จะนำมาใช้ในโปรแกรมได้โดยไม่ต้องเขียนคำสั่งไปจัดการเลย โดย Menu Editor สามารถจัดการในเรื่องต่างๆเกี่ยวกับเมนู เช่น การเพิ่ม ลบ และแก้ไขรายการที่ปรากฏในเมนู โดยเมนูที่สร้างจะมีลักษณะเหมือน Control ทั่วไปที่สามารถกำหนดคุณสมบัติของเมนูผ่านทาง Properties Window และตอบสนองต่อ Event ของเมนูได้

1) Control ที่ใช้ในการแสดง Graphic

Control ที่ใช้แสดงภาพกราฟฟิกใน Microsoft Visual Basic 6.0 สามารถแสดงภาพกราฟฟิกจากไฟล์ภาพ JPG , GIF, BMP , ICO, WMF และ DIB ซึ่งมี Control ที่ใช้แสดงกราฟฟิกอยู่ 2 Control คือ Control PictureBox และ Control Image

2) Control PictureBox

เป็น Control ที่ใช้ในการวาดรูปทรงกราฟฟิกในขณะรัน โปรแกรม โดยใช้เมธอดทางด้านกราฟฟิกที่ได้กล่าวมา หรือแสดงรูปภาพจากไฟล์ที่มีนามสกุล JPG , GIF , BMP, DIB และICO โดยจะใช้คำสั่ง LoadPicture โหลดภาพเข้า Control PictureBox ในขณะรัน โปรแกรมซึ่งมีรูปแบบ ดังนี้

```
LoadPicture (<ไคเรททอรีของไฟล์รูปภาพ>)
```

ตารางที่ 2.4 แสดงคุณสมบัติที่สำคัญของ PictureBox

คุณสมบัติ	อธิบาย
AutoRedraw	ถ้าคุณสมบัตินี้ถูกกำหนดเป็น True จะทำให้ Microsoft Visual Basic6.0 วาดกราฟฟิกขึ้นมาใหม่โดยอัตโนมัติ ทุกครั้งที่จำเป็นต้องวาดใหม่ โดยที่ Microsoft Visual Basic6.0 จะกันเนื้อที่หน่วยความจำไว้ในการวาดกราฟฟิกใหม่ แต่ถ้าคุณสมบัตินี้ถูกกำหนดเป็น False จะไม่มีการวาดกราฟฟิกใหม่ เราต้องสั่งเองผ่านอ็อบเจกต์ Paint
AutoSize	ถ้าคุณสมบัตินี้ถูกกำหนดให้เป็น True ก็จะมีการปรับขนาดของ Control ตามขนาดของรูปภาพที่โหลดเข้ามาโดยอัตโนมัติ แต่ถ้าเป็น False Control จะมีขนาดตามที่เรากำหนดไว้
Picture	เป็นคุณสมบัติที่กำหนดไฟล์รูปภาพที่เราโหลดเข้ามาแสดงใน Control ซึ่งกำหนดให้ตั้งแต่ตอนออกแบบโปรแกรม หรือกำหนดในขณะรันโปรแกรมก็ได้ โดยใช้ขณะรันโปรแกรมก็ได้ โดยใช้คำสั่ง LoadPicture

3) Control Image

Control Image ทำหน้าที่แสดงรูปภาพและวาดกราฟฟิกด้วยเมธอดทางด้านกราฟฟิกเหมือนกับ Control PictureBox ดังนี้ คือ

- 3.1) ใช้ทรัพยากรของระบบคอมพิวเตอร์น้อยกว่า
- 3.2) ไม่สามารถเป็น Container ของ Control อื่นๆได้
- 3.3) ไม่สามารถใช้ติดต่อผ่านทาง DDE ได้

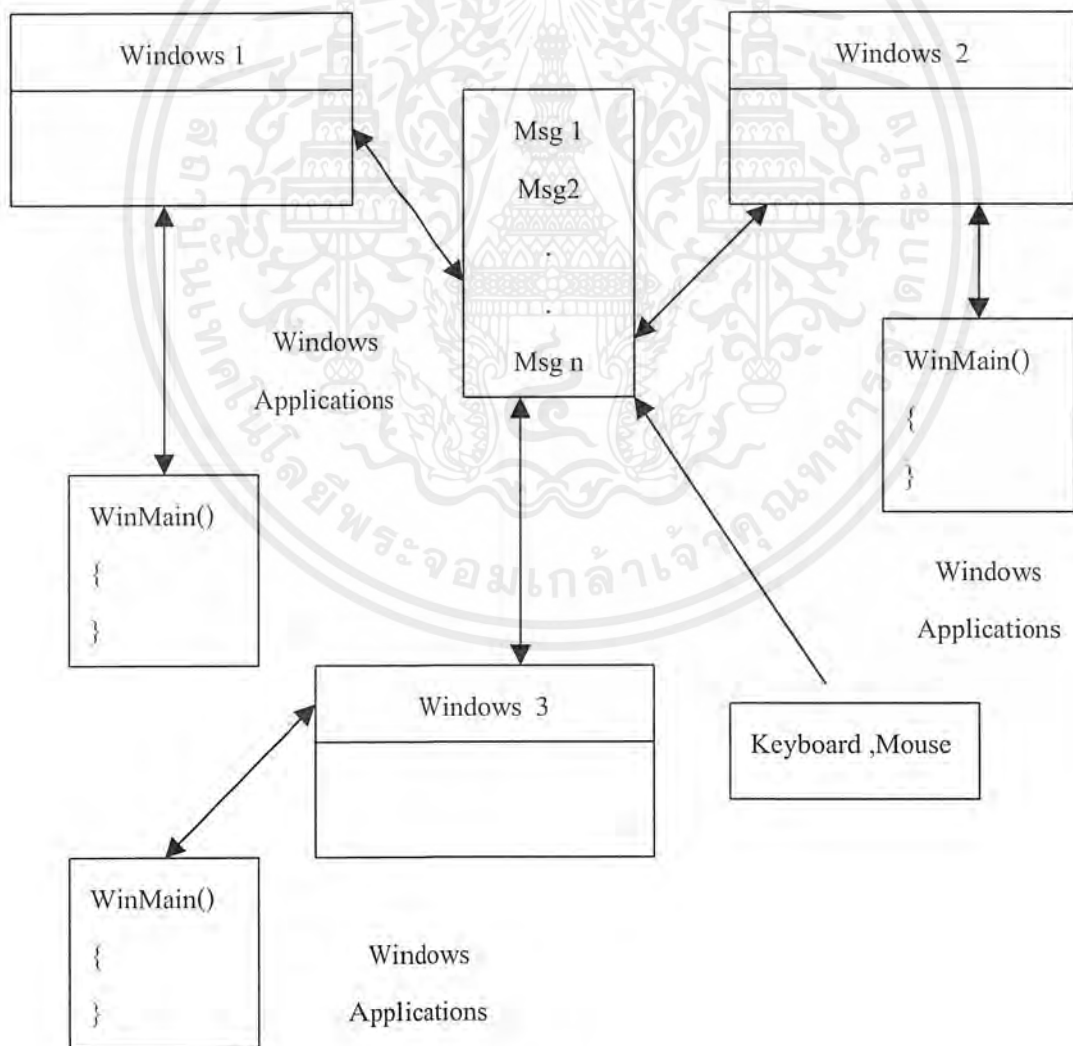
คุณสมบัติที่สำคัญของ Control Image

คุณสมบัติ Stretch

ถ้าเรากำหนดคุณสมบัตินี้ให้มีค่าเป็น True รูปภาพที่แสดงใน Control จะถูกปรับให้มีขนาดเท่ากับขนาดของ Control

2.3 การเขียนโปรแกรมบน Windows

การเขียนโปรแกรม Windows อาจจะเป็นเรื่องที่ยากหรือยาก ขึ้นอยู่กับขนาดของโปรแกรมที่ต้องการทำ การเขียน Word Processor อาจต้องการความรู้ทางการเขียนโปรแกรมที่มากพอสมควร แต่สำหรับการเขียนโปรแกรมโดยใช้ Direct X ต้องการเพียงเล็กน้อยเท่านั้น ในส่วนนี้จะอธิบายถึงการเขียนโปรแกรมบน Windows และพื้นฐานที่ควรจะต้องรู้สำหรับการเขียนโปรแกรมบน Windows เพื่อนำไปใช้ในการเขียนเกม โดยรูป 2.3 จะแสดงระบบการส่ง Message ใน Windows



รูปที่ 2.3 แสดงระบบการส่ง Message ใน Windows

Windows เป็นระบบปฏิบัติการแบบ Event-based โดย Message เป็นเสมือนสัญญาณที่จะบอกให้ Windows ต้องทำอะไร ทำอย่างไร หรือเกิดอะไร ตัวอย่าง

Windows event เมื่อผู้ใช้ต้องการเปลี่ยนขนาดของ Window Application Windows ก็จะทำการส่ง Message ไปยัง Application ตัวนั้น ว่าจะเปลี่ยนขนาดอย่างไร

Keyboard event เมื่อผู้ใช้กดแป้นพิมพ์บน Keyboard message ของตัวแป้นพิมพ์ที่กด ก็จะถูกส่งไปยัง Application เพื่อทำการประมวลผลข้อมูลนั้น และ แสดงผลข้อมูลนั้นออกมา เช่น แสดงเมนูย่อยจากเมนูบาร์หลัก

Drawing event ถ้า Windows ตัวใหม่ถูกแสดงออกมาทับตัวเก่าที่แสดงอยู่ หน้าจอจะต้องทำการวาดใหม่อีกครั้ง โดย Repaint message จะกระทำในส่วนนี้

2.3.1 Win32 Libraries

เมื่อเราสร้าง Windows Application ขึ้นมา Compiler จะเป็นตัวจัดการเกี่ยวกับส่วนของ Libraries ให้เองทั้งหมด ยกเว้นเสียแต่ Libraries ของ Application ที่เราต้องการไม่ได้ถูกตั้งค่าไว้ตั้งแต่เริ่มต้น เช่น เมื่อเราต้องการสร้าง Application เกี่ยวกับ Multimedia ซึ่งต้องการ Winmm.Lib Library แต่ใน Directory ของ LIB ในตัวของ Compiler ไม่มี หรือไม่ได้ถูกกำหนดไว้ ผู้ใช้ก็ต้องเพิ่มหรือกำหนดเข้าไปเอง

เมื่อสร้าง Application ที่ใช้ Windows API ผู้ใช้ก็ต้องเพิ่ม Libraries ของ Windows API เข้าไปใน Application นั้นด้วย

2.3.2 ตัวแปรและการประกาศตัวแปร

ตัวแปรใน Microsoft Visual Basic 6.0 มีหน้าที่เก็บข้อมูลในการทำงานของโปรแกรมไว้เป็นการชั่วคราว ตัวแปรที่เราทำการกำหนดขึ้นมาจะต้องประกอบด้วยชื่อ และชนิดของข้อมูล (Data Type) ที่ตัวแปรเก็บได้ สำหรับการประกาศตัวแปรนั้นเราจะประกาศด้วยคำสั่ง Dim จะตามด้วยชื่อของตัวแปร คำว่า As และตามด้วยชนิดของข้อมูล (Type) แล้วแต่ตัวแปรที่เราทำการประกาศไว้ นั้นต้องการใช้ชนิดข้อมูลอะไร สำหรับการประกาศตัวแปรนี้จะเหมือนกับการลงทะเบียนให้ Microsoft Visual Basic 6.0 รู้ว่า เราต้องการให้มีตัวแปรชื่อนี้เพื่อนำมาใช้งานในโปรแกรมต่อไป ซึ่งต่อไปนี้จะเห็นรูปแบบที่ใช้ในการประกาศตัวแปร

Dim <ชื่อของตัวแปร> As <ชนิดของข้อมูล>

ตัวอย่าง เช่น

Dim SelectPic As Integer

สำหรับการตั้งชื่อตัวแปร เราจะต้องตั้งชื่อตามกฎต่อไปนี้

2.3.2.1 ชื่อตัวแปรต้องไม่ซ้ำกันในขอบเขตเดียวกัน

2.3.2.2 ชื่อต้องไม่ซ้ำกับ Key Word (คำสงวน) ของ Microsoft Visual Basic6.0 เช่น คำว่า Dim เป็นต้น

2.3.2.3 ความยาวต้องไม่เกิน 255 ตัวอักษร

2.3.2.4 จะต้องเริ่มต้นด้วยตัวอักษร A-Z , a-z

ขอบเขตของการประกาศตัวแปร คือ การกำหนดให้ตัวแปรนั้นสามารถมองเห็นได้ในส่วนใดบ้างในโปรแกรม เช่น ต้องการให้เข้าถึงตัวแปรได้เฉพาะในโปรแกรมย่อยนั้น หรือให้ทุกโปรแกรมย่อย ในโมดูลนั้นเห็นทั้งหมด หรือจะให้เห็นทั้ง Project เป็นต้น สำหรับคำสั่งที่ใช้ในการประกาศมีรูปแบบดังนี้

[Private|Public] <ชื่อตัวแปร> [As Type]

2.3.2.5 ชนิดของการประกาศตัวแปรที่สำคัญมีอยู่ 2 แบบ

1) Implicit Declaration เป็นการนำตัวแปรมาใช้โดยไม่มีการประกาศค่า

2) Explicit Declaration เป็นการประกาศตัวแปรโดยใช้คำสั่ง Dim ในการประกาศตัวแปรเหมือนที่กล่าวมาแล้ว

2.3.3 ค่าคงที่และการประกาศค่าคงที่

ค่าคงที่มีไว้สำหรับเก็บค่าที่ไม่มีการเปลี่ยนแปลงตลอดทั้งโปรแกรมซึ่งถ้าในโปรแกรมมีการใช้คำสั่งที่เปลี่ยนค่าในค่าคงที่ Microsoft Visual Basic6.0 จะแสดงข้อผิดพลาดขึ้นมา เพราะเราเปลี่ยนค่าที่ประกาศในค่าคงที่ไม่ได้

สำหรับคำสั่งที่ใช้ในการประกาศค่าคงที่มีรูปแบบ ดังนี้

[Private | Public] Const <ชื่อค่าคงที่> [As Type] = <ค่าที่กำหนดให้>

โดยเราจะใช้ชื่อค่าคงที่ที่กำหนด ในการอ้างอิงถึงค่าคงที่นี้ในคำสั่งโปรแกรม และเรายังสามารถกำหนดชนิดข้อมูลของค่าคงที่ได้ด้วย

2.3.4 ชนิดของข้อมูลที่นำมาใช้

ตารางที่ 2.5 แสดงชนิดของข้อมูลและขอบเขตของตัวแปร

ชนิดของข้อมูล	ขนาด(ไบต์)	คำอธิบาย
Integer	2	มีค่าตั้งแต่ -32767 ถึง 32768
Boolean	1	มีค่า True (แทนค่าได้ด้วยค่า -1) กับ False (แทนได้ด้วยค่า 0)
Object		เป็นข้อมูลชนิดคอนโทรล
Variant		เป็นชนิดข้อมูลที่สามารถใช้แทนชนิดข้อมูลอื่นๆ ได้

2.3.5 คำสั่งที่ใช้ในการควบคุมการทำงาน

2.3.5.1 คำสั่งในการเลือกเส้นทางการทำงาน (Decision Structures)

คำสั่งที่ใช้ในการเลือกเส้นทางการทำงานของโปรแกรม จะทำให้เราสามารถที่จะเลือกเส้นทางการทำงานของโปรแกรมของเราได้ตามเงื่อนไขที่อยู่ในโปรแกรมของเรา คำสั่งประเภทนี้แบ่งออกได้ ดังนี้

1) คำสั่ง If-Then-Else เป็นคำสั่งที่มีรูปแบบ ดังนี้

If <นิพจน์ที่เป็น Boolean> Then

 คำสั่งสำหรับนิพจน์ที่มีค่าเป็น True

Else

 คำสั่งสำหรับนิพจน์ที่มีค่าเป็น False

End If

คำสั่งข้างต้น จะเป็นคำสั่งที่ตรวจสอบเงื่อนไขของคำสั่งหลังคำว่า If ถ้าเงื่อนไขเป็น True จะทำตามคำสั่งที่อยู่หลังคำว่า Then แต่ถ้าเงื่อนไขเป็น False จะทำตามคำสั่งหลังคำว่า Else จนถึงคำว่า End If

ตัวอย่าง เช่น

if I > 0 Then

Else

 Debug.Print "Less than 0 or equal"

End if

2) คำสั่ง Select-Case

เป็นคำสั่งที่ใช้ในการเลือกเส้นทางการทำงานของโปรแกรมจากค่าของนิพจน์ที่กำหนดเป็นคำสั่งที่มีรูปแบบต่อไปนี้

Select Case <นิพจน์>

 Case <ค่าของนิพจน์กลุ่มแรก>

 ‘ คำสั่งที่ทำงานเมื่อนิพจน์ตรงกับค่าในกลุ่มแรก

 Case <ค่าของนิพจน์กลุ่มที่สอง>

 ‘ คำสั่งที่ทำงานเมื่อนิพจน์ตรงกับค่าในกลุ่มที่สอง

Case Else

 ‘ คำสั่งเมื่อไม่ตรงกับกรณีไหนเลย

End Select

คำสั่ง Select-Case นั้นจะทำงานตามค่าที่อยู่หลังคำว่า Select Case ถ้าค่าตรงกับค่าที่อยู่หลังคำว่า Case ใด ก็จะทำงานตามคำสั่งที่อยู่หลัง Case นั้น แต่ถ้าไม่ตรงกับค่าใดเลยจะทำตามคำสั่งที่อยู่หลังคำว่า Case Else จนถึงคำว่า End Select ตัวอย่างการใช้งาน เช่น

```
Select Case Number
```

```
Case 1 To 3
```

```
    Debug.Print "Between 1 And 3"
```

```
Case 4,5,6,7,8
```

```
    Debug.Print "Between 4 And 8"
```

```
Case Else
```

```
    Debug.Print "Not positive number"
```

```
End Select
```

2.3.5.2 คำสั่งที่ใช้ในการทำงานซ้ำ

1) คำสั่ง For-Next

คำสั่งนี้จะใช้ในการทำงานซ้ำเป็นจำนวนครั้งที่แน่นอน โดยมีรูปแบบ ดังนี้

```
For<ชื่อตัวแปร>=<ค่าเริ่มต้น>To<ค่าสุดท้าย>[Step<ค่าที่เพิ่มขึ้นของตัวแปร>]
```

```
    ‘ชุดคำสั่ง
```

```
    [Exit For]
```

```
Next<ชื่อตัวแปร>
```

การทำงานของคำสั่งจะทำงานโดยมีตัวแปรหนึ่งนับจำนวนครั้งการทำซ้ำ กำหนดค่าเริ่มต้นให้ตัวแปรนี้ และเมื่อทำงานมาถึงบรรทัด Next <ชื่อตัวแปร> ก็จะทำการเพิ่มค่าตัวแปรนั้นขึ้นไปเท่ากับค่าที่อยู่หลังคำว่า Step ที่อยู่ตอนต้นของคำสั่ง For ... Next และทำซ้ำจนกว่าค่าของตัวแปรจะมีค่ามากกว่าค่าสุดท้ายที่กำหนดไว้ ถ้ามีคำสั่ง Exit For จะทำให้การทำงานหลุดออกจาก Loop ทันที

ตัวอย่างการใช้งานคำสั่ง เช่น

```
For I=0 to 10 Step 2
```

```
    Print I
```

```
Next I
```

2) คำสั่ง Do-Loop

เป็นคำสั่งที่ใช้ในการทำซ้ำ โดยจำนวนครั้งในการทำซ้ำขึ้นอยู่กับเงื่อนไขหลังคำว่า While หรือ Until รูปแบบของคำสั่งจะเป็นดังนี้

```
Do While<Condition>
```

```
    ‘ชุดคำสั่ง
```

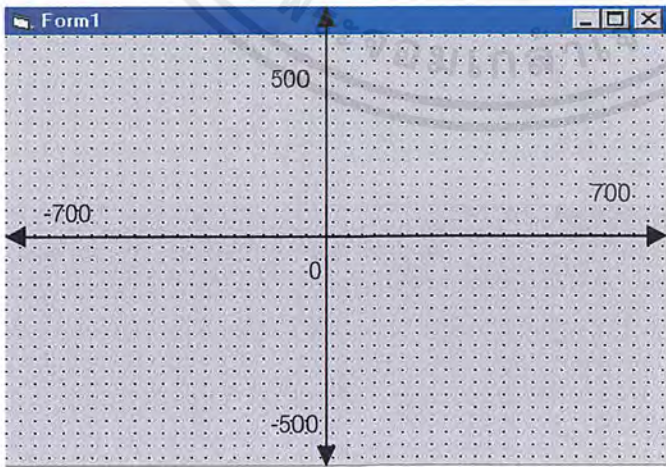
```
    [Exit Do]
```


ScaleMode เป็นคุณสมบัติที่มีอยู่ใน Form Control Image และ Control PictureBox ซึ่งจะใช้กำหนดหน่วยวัดของระบบพิกัดของ Control โดยกำหนดค่าได้ดังตารางต่อไปนี้

ตารางที่ 2.6 แสดงหน่วยวัดของระบบพิกัดของ Control

ค่าของ ScaleMode	หน่วยวัด
0	ผู้ใช้กำหนด
1	Twip(1440 Twip = 1Inch)
2	Point (72 Point = 1 Inch)
3	Pixel (ขึ้นกับความละเอียดที่กำหนดใน Display Properties)
4	Character (6 อักษรต่อนี้วในแนวตั้ง 12 อักษรต่อนี้วในแนวนอน)
5	Inch
6	Millimeter
7	Centimeter

การกำหนดหน่วยวัดเอง
บางครั้งเราอาจจะต้องการกำหนดขนาดของพิกเซลขึ้นมาเองเพื่อความสะดวกในการเขียนโปรแกรม โดยถึงแม้เราจะเปลี่ยนขนาดของฟอร์มไปแล้ว แต่ขนาดของพิกัดที่เรากำหนดขนาดนั้นจะไม่เปลี่ยนแปลงตามไปด้วย ตัวอย่างเช่นเราอาจต้องการให้ขนาดของฟอร์มที่มีพิกัดแบบ (700,500)(-700,500) ดังรูปที่ 2.5



รูปที่ 2.5 แสดงการกำหนดหน่วยวัดเอง

2.4 การจัดการ Mouse

2.4.1 การจัดการอีเวนต์ของเมาส์

ในหัวข้อนี้เราจะศึกษาวิธีการควบคุมการทำงานของเมาส์โดยใช้อีเวนต์ต่างๆ ดังต่อไปนี้

Mouse Up เกิดเมื่อมีการปล่อยปุ่มเมาส์ไม่ว่าปุ่มใดก็ตาม

Mouse Down เกิดเมื่อมีการกดปุ่มเมาส์ไม่ว่าปุ่มใดก็ตาม

MouseMove เกิดเมื่อมีการเคลื่อนปุ่มเมาส์ไปยังตำแหน่งใดๆภายในหน้าจอโปรแกรมของเรา

อีเวนต์ต่างๆเหล่านี้จะเกิดขึ้น เมื่อตัวชี้เมาส์อยู่ภายในฟอร์ม หรือ Control และจำมีลำดับในการเกิดอีเวนต์เรียงกันไปดังนี้ คือ เมื่อเราเริ่มต้นกดปุ่มเมาส์จะเกิดอีเวนต์ MouseDown เมื่อเราย้ายตัวชี้เมาส์ก็จะเกิดอีเวนต์ Mousemove และเมื่อเราปล่อยปุ่มเมาส์ที่กดไว้ก็จะเกิดอีเวนต์ MouseUp

2.4.2 การตรวจสอบสถานะต่างๆของเมาส์

ในอีเวนต์ของเมาส์ทั้ง 3 อีเวนต์จะมีพารามิเตอร์ที่ส่งเข้ามา เพื่อใช้ตรวจสอบสถานะต่างๆของเมาส์ ซึ่งพารามิเตอร์ต่างๆ มีรายละเอียดดังต่อไปนี้

Button ใช้ตรวจสอบว่าผู้ใช้ได้กดเมาส์ปุ่มใดอยู่ในขณะนั้น

Shift ใช้ตรวจสอบสถานะของการกดปุ่ม Shift, Ctrl, และ Alt

x,y ใช้ตรวจสอบตำแหน่งของตัวชี้เมาส์ว่าอยู่ที่ตำแหน่งใด

2.4.3 การตรวจสอบการกดปุ่มเมาส์

สามารถตรวจสอบการกดปุ่มเมาส์ด้วยพารามิเตอร์ Button ซึ่งความสัมพันธ์ระหว่างปุ่มเมาส์ที่ถูกกด กับบิต 3 บิตท้ายของพารามิเตอร์ Button แสดงได้ดังรูปต่อไปนี้



รูปที่ 2.6 แสดงการตรวจสอบการกดปุ่มเมาส์

นอกจากวิธีการตรวจสอบการกดปุ่มเมาส์ตามที่ได้กล่าวมา เรายังสามารถใช้ค่าคงที่ที่ Microsoft Visual Basic 6.0 ได้เตรียมไว้สำหรับการกดปุ่มเมาส์ได้ ดังตารางต่อไปนี้

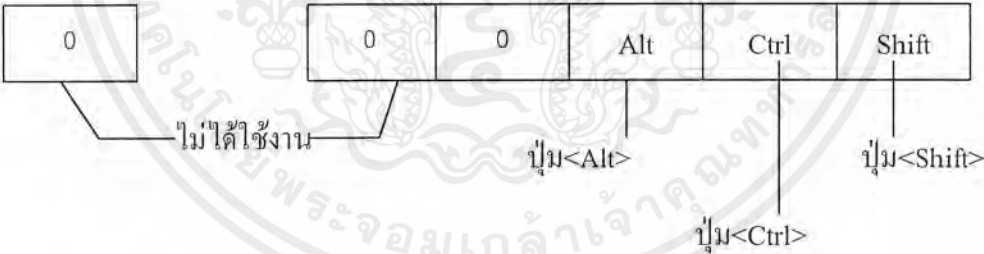
ตารางที่ 2.7 แสดงค่าคงที่ของอีเวนต์การกดปุ่มเมาส์

ค่า 3 บิตท้าย	ค่าเลขฐาน10	ค่าคงที่ใน Microsoft Visual Basic 6.0	คำอธิบาย
001	1	VbLeftButton	เมาส์ปุ่มซ้ายถูกกด
010	2	VbRightButton	เมาส์ปุ่มขวาถูกกด
100	4	VbMiddleButton	เมาส์ปุ่มกลางถูกกด

นอกจากนี้ ถ้ามีการกดปุ่มเมาส์หลายๆแบบหลายๆปุ่มพร้อมกัน เราจะสามารถตรวจสอบได้โดยการรวมค่าของปุ่มต่างๆเข้าด้วยกัน เช่น ถ้ามีการกดปุ่มเมาส์ซ้ายและขวาพร้อมกัน ค่าของพารามิเตอร์ Button จะเป็น 011(001+010) หรือ3(1+2) หรือ VbRightButton+VbLeftButton ซึ่งเราสามารถตรวจสอบการกดปุ่มร่วมกันแบบอื่นๆได้ในทำนองเดียวกัน

2.4.4 การตรวจสอบสถานะของปุ่ม <Shift>, <Ctrl> และ<Alt>

สามารถตรวจสอบการกดปุ่ม <Shift>, <Ctrl> และ<Alt> ที่เกิดพร้อมกับอีเวนต์ที่เกี่ยวกับการทำงานของเมาส์ทั้ง 3 ได้ โดยการตรวจสอบค่าพารามิเตอร์ Shift ที่ส่งเข้ามาซึ่งความสัมพันธ์ระหว่างการกดปุ่ม <Shift>, <Ctrl> และ<Alt> กับบิต 3 บิตล่างของพารามิเตอร์ Shift แสดงได้ดังรูปที่ 2.7



รูปที่ 2.7 แสดงการตรวจสอบสถานะของปุ่ม <Shift> , <Ctrl> และ<Alt>

จากรูปข้างต้น เมื่อไม่มีการกดปุ่มใดใน 3 ปุ่มคีย์บอร์ด บิตทั้ง 3 จะมีค่าเป็น 0 แต่ถ้ามีการกดปุ่ม <Shift> ค่าของพารามิเตอร์ Shift จะเท่ากับ 1 ถ้ามีการกดปุ่ม <Ctrl> ค่าของShift จะเท่ากับ 2 และถ้ามีการกดปุ่ม <Alt> ค่าของ Shift จะเท่ากับ 4 หรือเราอาจใช้ค่าคงที่ในตารางเหล่านี้แทนได้

ตารางที่ 2.8 แสดงค่าของการกด <Shift>, <Ctrl> และ <Alt>

ค่า3บิตท้าย	ค่าเลขฐาน10	ค่าคงที่ในMicrosoft Visual Basic 6.0	คำอธิบาย
001	1	VbShiftMask	ปุ่ม<Shift>ถูกกด
010	2	VbCtrlMask	ปุ่ม<Ctrl>ถูกกด
100	4	VbAltMask	ปุ่ม<Alt>ถูกกด

นอกจากนี้ ถ้ามีการกดปุ่มเมาส์หลายๆแบบหลายๆปุ่มพร้อมกัน เราจะสามารถตรวจสอบได้โดยการรวมค่าของปุ่มต่างๆเข้าด้วยกัน เช่น ถ้ามีการกดปุ่ม <Shift> และ<Ctrl> ค่าของ Shiftจะเป็น 011 (001+010)หรือ 3(1+2) หรือ VbCtrlMask+VbAltMask ซึ่งการกดปุ่มพร้อมกันแบบนี้เราสามารถตรวจสอบได้ในทำนองเดียวกัน

2.4.5 อีเวนต์ที่เกี่ยวข้องกับ Drag and Drop

Microsoft Visual Basic 6.0 สนับสนุนอีเวนต์เกี่ยวกับการ Drag and Drop นี้โดยเมื่อเรากดปุ่มเมาส์และลาก Controlใดไป เราจะเรียกว่า Dragging และเมื่อเราปล่อยปุ่มเมาส์จากการ Drag เราจะเรียกว่า Dropping ซึ่งก่อนที่จะเราจะสร้างโปรแกรมที่มีการใช้เมาส์ Drag and Drop ได้เราจะต้องรู้จักกับสิ่งต่างๆ ดังต่อไปนี้

ตารางที่ 2.9 แสดงอีเวนต์เกี่ยวกับ Drag and Drop

ชื่อ Member	คำอธิบาย
DragMode	เป็นคุณสมบัติที่กำหนดว่า เป็นการ Drag mouse แบบ Automatic หรือ Manual
DragIcon	เป็นคุณสมบัติที่บอกว่าเมื่อมีการ Drag เมาส์แล้ว จะแสดงตัวชี้เมาส์เป็นรูปแบบใด
DragDrop	เป็นอีเวนต์ที่เกิดขึ้น เมื่อคอนโทรลถูก Drop ไปวางบนคอนโทรลที่เกิดอีเวนต์นี้
DragOver	เป็นอีเวนต์ที่เกิดขึ้น เมื่อคอนโทรลถูก Drag mouse มาอยู่บนคอนโทรลที่เกิดอีเวนต์นี้
Drag	เป็นเมธอดที่ใช้เริ่มต้น และสิ้นสุดการ Drag

โดยปกติคุณสมบัติ DragMode จะมีค่าเป็น 1- Manual ซึ่งจะให้เราจัดการเกี่ยวกับการ Drag mouse ด้วยเมธอด Drag เอง แต่เราสามารถกำหนดให้มีการใช้เมาส์ลากคอนโทรลแบบอัตโนมัติได้โดยกำหนดให้คุณสมบัตินี้มีค่าเป็น 0-Automatic

2.4.5.1 การจัดการ Drag and Drop แบบ Manual

ดังที่ได้ทราบมาแล้วว่า ถ้าคุณสมบัติ DragMode ของคอนโทรลมีค่าเป็น 0-Automatic เราจะสามารถใช้เมาส์ลากคอนโทรลได้ตลอดเวลา แต่ถ้าเรากำหนดคุณสมบัติ DragMode ของ Control เป็น 1-Manual เราจะสามารถควบคุมการใช้เมาส์คอนโทรลได้เอง โดยใช้เมธอด Drag ซึ่งมีรูปแบบการใช้งานดังต่อไปนี้

<ชื่อคอนโทรล>.Drag Action
โดยพารามิเตอร์ Action มีค่าที่กำหนดไว้ดังตารางต่อไปนี้

ตารางที่ 2.10 แสดงค่าพารามิเตอร์ที่ใช้ในการ Drag and Drop แบบ Manual

ค่าที่ส่งเป็นพารามิเตอร์	ค่าคงที่ใน Microsoft Visual Basic 6.0	คำอธิบาย
0	VbCancel	ยกเลิกการ Drag โดยที่จะไม่ทำให้เกิด Event DragDrop
1	VbBeginDrag	เริ่มต้นการ Drag
2	VbEndDrag	สิ้นสุดการ Drag ซึ่งจะทำให้เกิด Event DragDrop

2.4.5.2 การตอบสนองต่อการ Drop

เราสามารถใช้อีเวนต์ DragDrop ควบคุมการปล่อยเมาส์ได้ ซึ่งอีเวนต์ DragDrop จะเกิดเมื่อมีการใช้เมาส์ลากคอนโทรลแล้วปล่อยปุ่มเมาส์ที่เรากดอยู่ โดยจะเกิดกับคอนโทรลที่อยู่ ณ ตำแหน่งที่เราปล่อยปุ่มเมาส์ หรือเกิดจากการเรียกเมธอด Drag ด้วยพารามิเตอร์ VbEndDrag ซึ่งจะทำให้เกิดอีเวนต์ DragDrop กับคอนโทรลที่อยู่ ณ ตำแหน่งเมาส์ในขณะนั้น

สำหรับอีเวนต์ DragDrop มีพารามิเตอร์ 2 ตัวซึ่งมีรายละเอียด ดังต่อไปนี้

Source เป็นข้อมูลชนิด Control ที่อ้างอิงถึง Control ที่ถูกเราใช้เมาส์ลากและปล่อยซึ่งเราสามารถจะใช้ TypeOf หรือฟังก์ชัน TypeName ตรวจสอบ Source ว่าเป็น Control ชนิดใดได้ และนำคุณสมบัติของ Control กับ เมธอดกลับมาใช้ได้ด้วย

x,y เป็นตำแหน่งของตัวชี้เมาส์บนฟอร์มในขณะเกิดอีเวนต์

2.5 Windows API

ในระบบปฏิบัติการ Windows จะมีคุณสมบัติหนึ่งที่เรียกว่า Dynamic Link Library (เรียกว่าไดนามิกลิงก์ไลบรารี)ที่แทนด้วยคำย่อว่า DLL โดยใน Windows จะเก็บไลบรารีนี้ไว้ในไฟล์ที่มีส่วนขยายเป็น .dll DLL เป็นไฟล์ที่เก็บโปรแกรมย่อยและเก็บฟังก์ชัน ซึ่งจะถูกโหลดเข้าหน่วยความจำ

คอนรันโปรแกรมที่มีการเรียกใช้ไฟล์ DLL ไฟล์นั้น และจะมีการทำงานตามโปรแกรมย่อย หรือ ฟังก์ชันในไฟล์ DLL นั้นแล้วจึงส่งค่ากลับมายังโปรแกรมหลักที่เรียกใช้ DLL

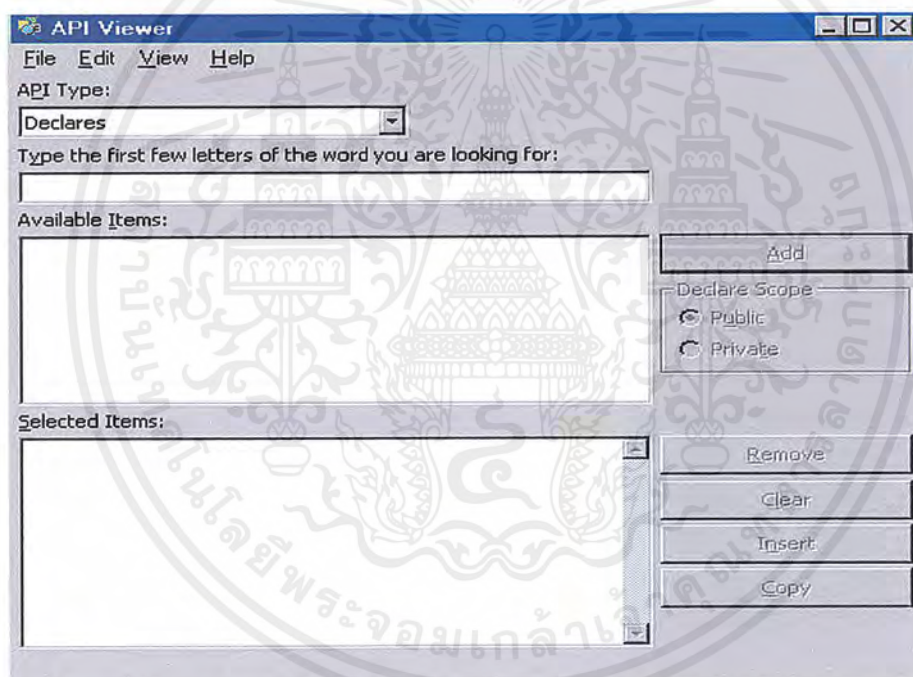
2.5.1 ข้อดีในการใช้ DLL คือ

2.5.1.1 ช่วยลดขนาดของโปรแกรมที่เราสร้าง โดยเตรียมฟังก์ชันและโปรแกรมย่อยอยู่ในไฟล์ที่สามารถใช้ร่วมกันในโปรแกรมอื่นๆได้

2.5.1.2 ทำให้การทำงานของโปรแกรมเร็วขึ้น เนื่องจากไม่ต้องโหลดไฟล์ทำงานที่มีขนาดใหญ่ แต่โหลดเฉพาะไฟล์ DLL ที่ต้องใช้งานเท่านั้น

2.5.1.3 ทำให้โปรแกรมทำงานบางอย่างที่ Microsoft Visual Basic 6.0 ไม่สามารถทำได้ หรือทำได้แต่ไม่ประสิทธิภาพเท่าที่ควร

2.5.2 ส่วนประกอบของ Windows API แสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 แสดงหน้าจอหลักของโปรแกรม API Text Viewer

หน้าจอโปรแกรม API Text Viewer ประกอบไปด้วยส่วนต่างๆ ดังต่อไปนี้

2.5.2.1 API Type สำหรับแสดงกลุ่มของคำประกาศ ซึ่งแบ่งออกเป็น 3 กลุ่มคือ

Constants เป็นข้อความประกาศค่าคงที่ที่ถูกอ้างอิงในฟังก์ชัน Windows API

Declares เป็นข้อความประกาศฟังก์ชันต่างๆ ของ Windows API

Type เป็นข้อความประกาศชนิดข้อมูลเฉพาะที่ถูกอ้างอิงในฟังก์ชัน Windows API

2.5.2.2 หน้าต่างแสดงรายชื่อ สำหรับแสดงรายชื่อการประกาศของกลุ่ม API ที่เราสามารถเลือกได้

2.5.2.3 หน้าต่างแสดงข้อความที่ถูกเลือก สำหรับแสดงรายชื่อของข้อความที่เราได้เลือกเอาไว้แล้ว

2.5.2.4 ปุ่มสั่งงาน Add สำหรับเพิ่มคำประกาศจากรายชื่อที่ถูกเลือกในหน้าต่างแสดงรายชื่อ ไปไว้ในหน้าต่าง

2.5.2.5 ปุ่มสั่งงาน Remove สำหรับลบคำประกาศล่าสุดที่แสดงในหน้าต่างแสดงข้อความที่ถูกเลือกออกไป

2.5.2.6 ปุ่มสั่งงาน Copy สำหรับ Copy คำประกาศทั้งหมดที่อยู่ในหน้าต่างแสดงข้อความที่ถูกเลือกไปยัง Clipboard

2.5.3 วิธีการใช้งานโปรแกรม API Text Viewer ก็ประกอบไปด้วยขั้นตอนดังต่อไปนี้

2.5.3.1 เลือกกลุ่มของคำประกาศ API ที่เราต้องการระหว่าง 3 กลุ่ม

2.5.3.2 เลือกคำประกาศที่เราต้องการจากหน้าต่างแสดงรายละเอียด ถ้าหากไม่อยู่ในหน้าต่างที่แสดงไว้ในปัจจุบัน ก็สามารถใช้แถบเลื่อนแนวนอนเพื่อเลื่อนดูรายชื่อถัดไป หรือ Click Mouse ในหน้าต่างแสดงรายชื่อแล้วพิมพ์ชื่อต้นของฟังก์ชันเพื่อค้นหาอย่างรวดเร็ว

2.5.3.3 Click Mouse เลือกรายชื่อที่ต้องการ แล้ว Click สั่งงาน Add

2.5.3.4 ทำซ้ำตั้งแต่ขั้นตอนที่ 2.5.3.1 ถึง 2.5.3.3 จนได้คำประกาศครบถ้วนตามต้องการ

2.5.3.5 Click ปุ่มสั่งงาน Copy เพื่อ Copy ทั้งหมดไว้ใน Clipboard

2.5.3.6 ใช้คำสั่ง Paste เพื่อวางข้อความจาก Clipboard ไปยังที่ๆเราต้องการ ในที่นี้คือส่วนประกาศของโมดูลฟอร์ม หรือในโมดูลมาตรฐานของ Project โปรแกรมของเรา

2.5.3.7 ถ้าต้องการการเปลี่ยนแปลงคำประกาศที่เลือกเอาไว้แล้วในหน้าต่างแสดงข้อความที่ถูกเลือกสามารถ Click ปุ่มสั่งงาน Remove เพื่อลบข้อความล่าสุดที่เข้ามาในหน้าต่างได้

โปรแกรม API Text Viewer นี้สามารถเปิดไฟล์แบบข้อความ (Text File) หรือแบบ Database เพื่อใช้ในการหาข้อมูลได้ ในการใช้งานโปรแกรมนี้นี้เป็นครั้งแรกจะพบว่ามียูเอไอแบบข้อความเท่านั้นที่เราเลือกเปิด เมื่อใช้งานไปสักครู่โปรแกรมอาจจะถามเราว่าต้องการเปลี่ยนข้อมูลแบบข้อความให้เป็น Database หรือไม่ ก็ควรที่เราจะตอบตกลงเนื่องจากจะทำให้การค้นหารายชื่อคำประกาศของโปรแกรมนี้นี้เป็นไปอย่างรวดเร็วยิ่งขึ้น โปรแกรมก็จะทำการเปลี่ยนรูปแบบข้อมูลให้อัตโนมัติ ซึ่งอาจจะกินเวลาสักเล็กน้อย แต่ในครั้งต่อไปเราก็สามารถเลือกที่จะเปิดไฟล์ Database ได้แทนไฟล์แบบข้อความ (Text File) โปรแกรมจะสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่า

2.5.4 การเรียกใช้ Windows API

2.5.4.1 ขั้นตอนการเรียกใช้งาน Windows API ดังนี้

- 1) ใช้คำสั่ง Declare ประกาศฟังก์ชันและโปรแกรมย่อยที่จะใช้ในโปรแกรมของเรา
- 2) เรียกใช้ฟังก์ชัน หรือโปรแกรมย่อยที่ได้ประกาศไว้แล้ว

คำสั่ง Declare จะเป็นการบอกว่าฟังก์ชัน หรือโปรแกรมย่อยนั้นจะหาได้จากไฟล์ชื่ออะไร ซึ่งจะมีรูปแบบ ดังต่อไปนี้

สำหรับโปรแกรมย่อย

```
Declare Sub <Name> Lib "LibName" [Alias "AliasName"] (<พารามิเตอร์ที่ส่งไป>)
```

สำหรับฟังก์ชัน

```
Declare Function <Name> Lib "LibName" [Alias "AliasName"] (<พารามิเตอร์ที่ส่งไป>) [As Type]
```

จากรูปแบบข้างต้น การประกาศต้องมีการกำหนดค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้

ตารางที่ 2.11 แสดงการประกาศพารามิเตอร์ต่างๆ

Name	เป็นชื่อของฟังก์ชัน และ โปรแกรมย่อยที่จะเรียกใช้ใน โปรแกรมของเรา
LibName	เป็นชื่อของไฟล์ DLL ที่มีฟังก์ชัน หรือ โปรแกรมย่อยนี้อยู่
Alias	เป็นการบอกว่าสามารถใช้ชื่อ AliasName แทนชื่อของฟังก์ชัน หรือ โปรแกรมย่อยนั้นได้ ส่วนใหญ่จะใช้เมื่อชื่อของฟังก์ชันซ้ำกับคำสั่งใน Microsoft Visual Basic 6.0

ตัวอย่างการใช้งาน เช่น

```
Declare Function SetWindowPos Lib "User32" ( ByVal h As Long, ByVal hb As Long, ByVal x As Long, ByVal y As Long,ByVal cx As Long, ByVal cy As Long, Byval f As Long) As Long
```

2.5.4.2 การส่งพารามิเตอร์ในการใช้งาน Windows API

ในการใช้งาน DLL เราจำเป็นต้องทราบว่า ต้องส่งพารามิเตอร์แบบส่งค่าไปอย่างเดียว (By Value) หรือส่งแบบอ้างอิง (By Reference) ซึ่งจะมีรายละเอียดในการส่ง ดังต่อไปนี้

1) การส่งค่าแบบไปอย่างเดียว

จะเป็นการส่งค่าของพารามิเตอร์ไปยังฟังก์ชันหรือโปรแกรมย่อยที่เรียกใช้เท่านั้น โดยที่ฟังก์ชันหรือโปรแกรมย่อยนั้นไม่สามารถที่จะทำการเปลี่ยนแปลงค่าของตัวแปรที่ส่งเป็นพารามิเตอร์นั้นได้เลย

ตัวอย่าง เช่น

```
Declare Function MyFunction Lib "MyDLL" (ByVal MyVar As Long) As Long
```

2) การส่งค่าแบบอ้างอิง

จะเป็นการส่งตำแหน่งที่อยู่ในหน่วยความจำ (Address) ของตัวแปรที่เป็นพารามิเตอร์ไปแทน ทำให้ฟังก์ชันหรือโปรแกรมย่อยที่เรียกใช้สามารถแก้ไขค่าของตัวแปรนั้นได้

ตัวอย่าง เช่น

```
Declare Function MyFunction Lib "MyDLL" (ByVal MyVar As String) As Long
```

2.5.5 คุณสมบัติของคอนโทรลที่ต้องการในการเรียกใช้ Windows API

ตารางที่ 2.12 แสดงคุณสมบัติของคอนโทรลที่ต้องการเรียกใช้ใน Windows API

คุณสมบัติ	คำอธิบาย
HWND	โดยทั่วไปถ้าเราต้องการแก้ไขฟอร์ม หรือคอนโทรลด้วย Windows API จะต้องส่งค่าของแฮนเดิล (Windows Handle) ที่มีอยู่ในแต่ละฟอร์ม และคอนโทรลนั้นเป็นพารามิเตอร์ไปด้วย โดยใช้คุณสมบัติ hWnd ของคอนโทรลตัวนั้นส่งค่าพารามิเตอร์ที่เป็นแฮนเดิลไป
HDC	เป็นคุณสมบัติที่เป็นแฮนเดิลของดีไวซ์คอนเท็กซ์ (Device Context) ของ Objects ต่างๆ เช่น ฟอร์ม และคอนโทรลต่างๆ ใช้เพื่อวาดกราฟฟิก

2.5.6 การเปลี่ยนชนิดของข้อมูลเมื่อใช้ Windows API

ในการเรียกใช้ฟังก์ชัน หรือโปรแกรมย่อยในไฟล์ DLL ของ Windows จะเหมือนกับการเรียกใช้ฟังก์ชัน หรือโปรแกรมย่อยโดยทั่วไปยกเว้นการส่งพารามิเตอร์ เนื่องจากไฟล์ DLLs ไม่ได้สร้างจาก Microsoft Visual Basic 6.0 แต่จะถูกเขียนขึ้นมาโดยภาษา C , C++ ซึ่งมีชนิดของข้อมูลที่แตกต่างจาก Microsoft Visual Basic 6.0 ดังนั้นการใช้งานจึงจำเป็นต้องระวังในเรื่องของชนิดข้อมูลที่ใช้ในการส่งค่าพารามิเตอร์ ซึ่งเราจะเปลี่ยนชนิดข้อมูลที่ใช้ในภาษา C (ใน Windows API) ไปเป็นชนิดข้อมูลใน Microsoft Visual Basic 6.0 ได้โดยใช้ตารางต่อไปนี้

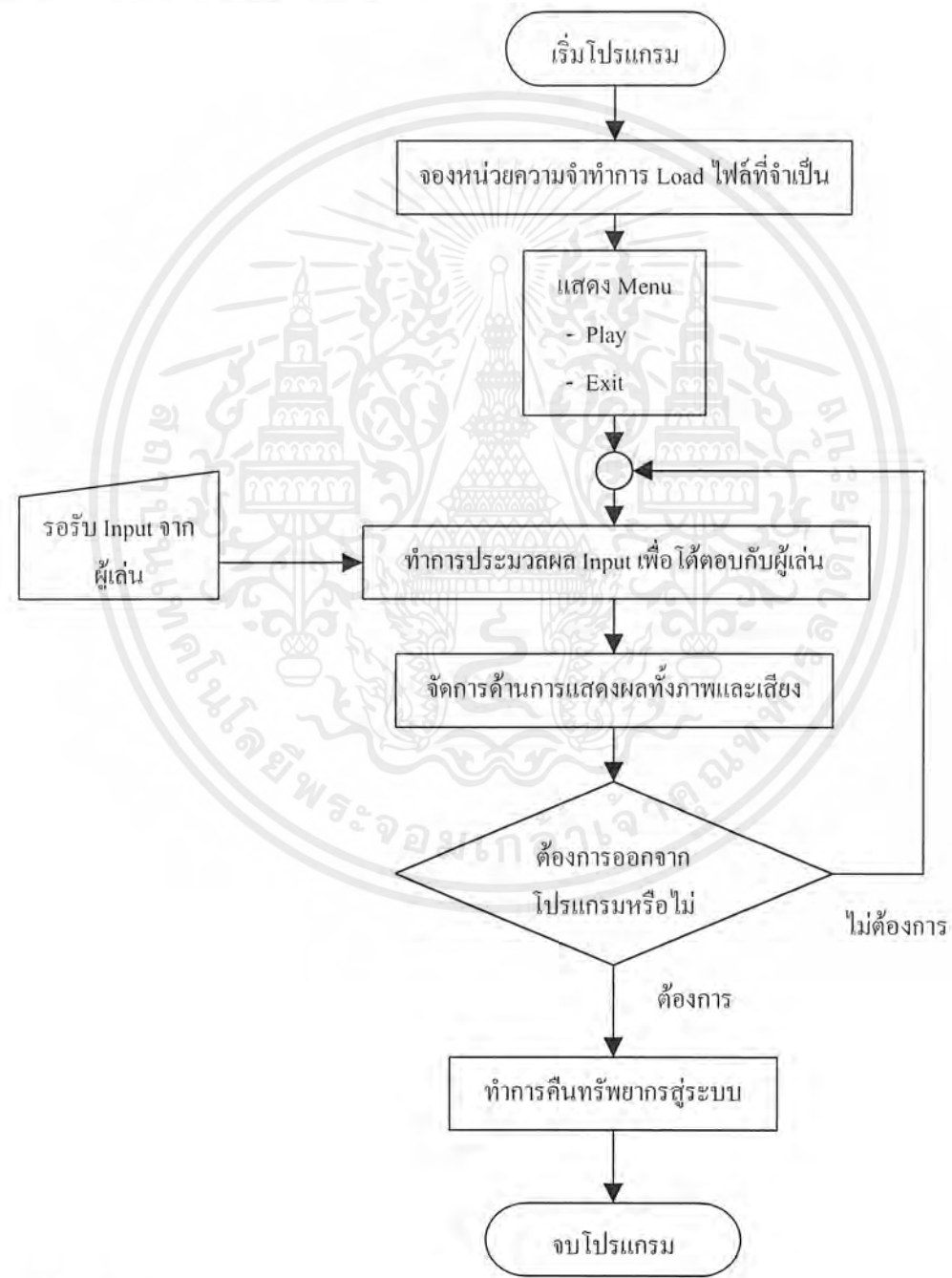
ตารางที่ 2.13 แสดงชนิดข้อมูลของ Windows API เปรียบเทียบกับชนิดข้อมูลใน Microsoft Visual Basic 6.0

ชนิดข้อมูลของ Windows API	ชนิดข้อมูลใน Microsoft Visual Basic 6.0
Int	ByVal <ชื่อตัวแปร> As Long
Float	ByVal <ชื่อตัวแปร> As Single
Double	ByVal <ชื่อตัวแปร> As Double
Pointer (ชี้ไปยัง String)	ByVal <ชื่อตัวแปร> As String
Pointer 32-bit	ByVal <ชื่อตัวแปร> As Long
Null pointer	VbNullString
Void Pointer	<ชื่อตัวแปร> As Any

BOOL	ByVal <ชื่อตัวแปร> As Boolean
------	-------------------------------

2.6 วงจรเกม

หลักการโดยทั่วไปจะทำการสร้างวงจรของเกม(Game Loop) ขึ้นมาเพื่อรองรับ Input จากผู้เล่นประมวลผล แล้วทำการแสดงผลและกลับไปรองรับ Input ประมวลผลและแสดงผลเป็นเช่นนี้เรื่อยๆจนกว่าผู้เล่นจะทำให้เงื่อนไขในการวนลูปเป็นจริง เป็นต้นว่าในปัญหาพิเศษนี้จะต้องบรรลุป่าหมาย หรืออาจคปุ่ม Exit ดังรูปที่ 2.9



รูปที่ 2.9 แสดงวงจรเกม

2.7 โปรแกรม Photoshop 5.5

2.7.1 ประวัติความเป็นมา

โปรแกรม Photo shop นับได้ว่าเป็นโปรแกรมวาดและตกแต่งภาพที่ได้รับความนิยมเป็นอย่างมาก ผลิตโดยบริษัท Adobe System Incorporated ประเทศสหรัฐอเมริกา ซึ่งเป็นบริษัทที่ผลิตโปรแกรมกราฟิกที่มีชื่อเสียง บริษัทนี้เป็นเจ้าของโปรแกรมด้านการจัดการสิ่งพิมพ์ (Desktop Publishing) หรือที่นิยมเรียกย่อๆว่า DTP หลายโปรแกรม เช่น Adobe Page maker และ Adobe Illustrator ซึ่งเป็นโปรแกรมที่ได้รับความนิยมสูงเช่นกัน

ในขณะนี้โปรแกรม Photo shop เวอร์ชัน 5.5 นับเป็นเวอร์ชันล่าสุดมีความสามารถเพิ่มเติมจากรุ่นเก่าหลายประการ อีกทั้งยังสามารถทำงานได้เร็วกว่ารุ่นเก่า จึงได้รับความนิยมอย่างแพร่หลาย

2.7.2 ความสามารถของโปรแกรม Photo shop

เรามักพบว่าโปรแกรม Photo shop ถูกนำไปประยุกต์ใช้งานประเภทต่างๆดังนี้

2.7.2.1 งานตกแต่งภาพถ่าย (Retouching)

คือ การนำภาพมาตกแต่งให้แตกต่างไปจากภาพต้นฉบับ ยกตัวอย่างเช่น การนำใบหน้าของคนหนึ่งไปใส่ไว้กับร่างกายของคนหนึ่ง และการนำภาพเก่าๆมาแต่งเป็นภาพใหม่ เป็นต้น

2.7.2.2 งานสร้างภาพแบบกราฟิก (Graphics)

ภาพกราฟิกคือภาพที่ถูกวาดขึ้นมา โดยอาจจะวาดในคอมพิวเตอร์หรือวาดในกระดาษแล้วสแกนเข้ามาใน

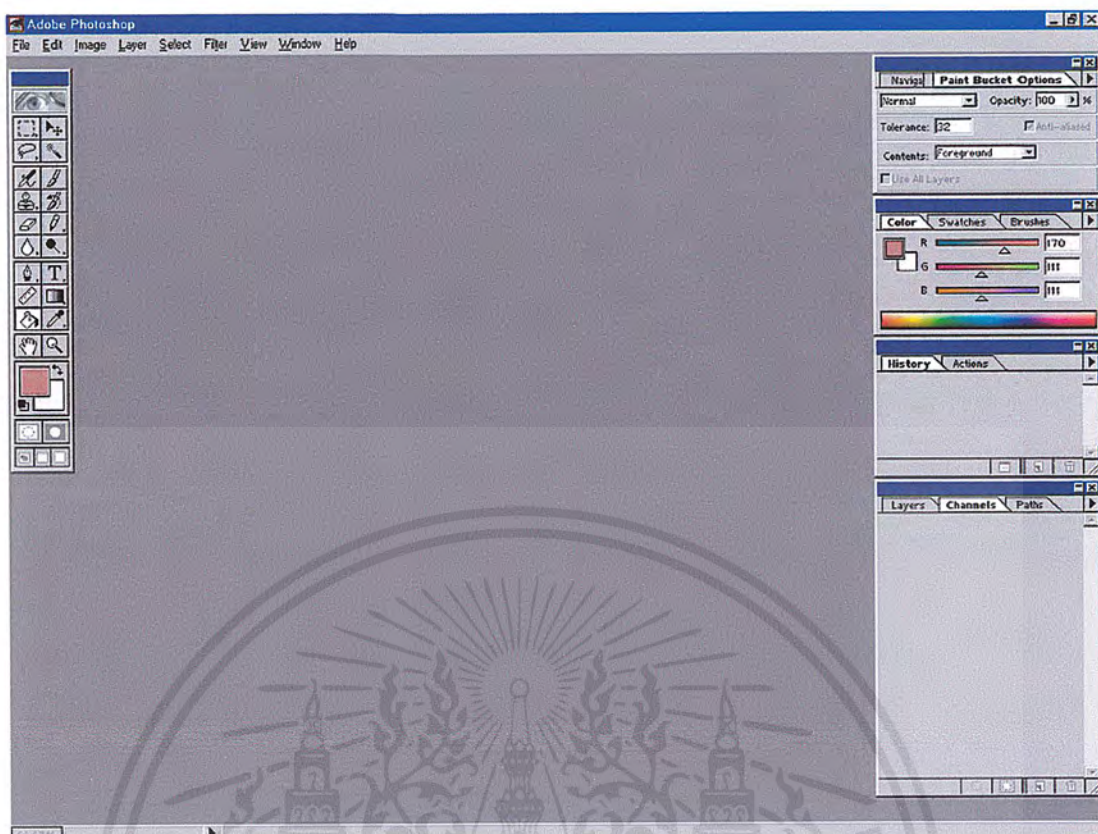
คอมพิวเตอร์ ภาพกราฟิกเหล่านี้สามารถวาดและตกแต่งให้สวยงามได้ด้วย Photo shop เช่น นำภาพจากโปรแกรมด้าน 3 มิติมาช่วยในการตกแต่งทำปกหนังสือและทำโฮมเพจในอินเทอร์เน็ต ด้วยโปรแกรม Photo shop เป็นต้น

2.7.2.3 งานสร้างอักษร

การสร้างอักษรสวยๆสามารถทำได้ง่ายๆใน Photo shop ทั้งอักษรที่สวดลายแปลกๆ และอักษร 3 มิติ

จากความสามารถต่าง ๆ มากมายเช่นนี้ เราจึงพบว่าโปรแกรม Photo shop มีความสำคัญเป็นอย่างยิ่งสำหรับการทำงานในปัจจุบัน

2.7.3 ส่วนประกอบของโปรแกรม Photoshop



รูปที่ 2.10 ส่วนประกอบของ Adobe Photo shop 5.5

ตัวชี้เมาส์ (Mouse Pointer) ใช้แทนตำแหน่งของเมาส์ที่จะใช้ชี้สิ่งต่างๆ ในโปรแกรม

เมนู (Menu) เป็นที่รวบรวมการใช้งานทุกคำสั่ง

กล่องเครื่องมือ (Tool Box) เป็นเครื่องมือที่ใช้ในการวาดภาพและตกแต่งภาพ

พาเลท (Palette) คือกรอบที่ใช้ปรับเปลี่ยนคุณสมบัติที่จะช่วยในการวาดและตกแต่งภาพควบคู่กับเครื่องมือในกล่องเครื่องมือ

แถบสถานะ (Status Bar) ใช้แสดงข้อความที่เป็นประโยชน์ในขณะที่ทำงาน ถ้าไม่พบกับแถบสถานะเลือกเมนู Window > Show Status Bar เพื่อเปิดแถบสถานะ

บทที่ 3

ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนทั้งหมดในงานวิจัยจะเกิดขึ้น ไม่ได้หากขาดการออกแบบ และการวางแผนที่ดีและมีเป้าหมายชัดเจน การดำเนินงานทั้งหมดจะถูกกระทำภายใต้แผนที่วางเอาไว้อย่างรัดกุมเพื่อความสำเร็จตามเป้าหมายที่วางเอาไว้ในแต่ละช่วง ซึ่งแผนงานทั้งหมดมีดังนี้

ขั้นตอนการออกแบบ ซึ่งอาจจะแบ่งเป็นส่วนย่อยได้อีกหลายส่วนงาน

- การออกแบบระบบเกม
- การออกแบบ Interface ที่ใช้ติดต่อกับผู้เล่น
- การออกแบบวิธีการโต้ตอบของฝ่ายคอมพิวเตอร์

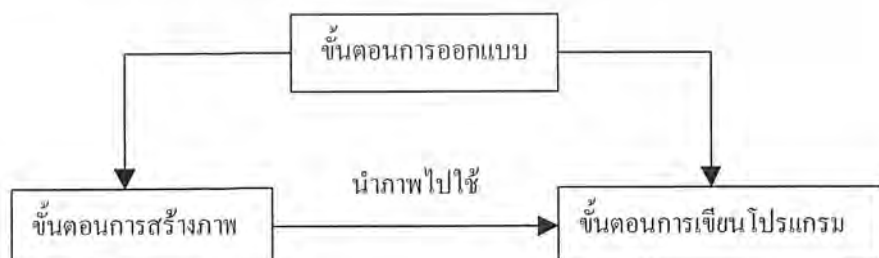
ในขั้นตอนการออกแบบนี้จะใช้เวลานานพอสมควร เพื่อให้ได้รูปแบบที่เป็นเอกลักษณ์และสอดคล้องกับเป้าหมายซึ่งเน้นรูปแบบที่แปลกใหม่

ขั้นตอนการเขียนโปรแกรม

ขั้นตอนนี้ใช้เวลานานที่สุดในตลอดระยะเวลาทำวิจัย เพราะจำเป็นที่จะต้องศึกษารูปแบบวิธีการเขียนโปรแกรมที่เป็นลักษณะ Realtime และการเขียนโปรแกรมแบบเชิงวัตถุ เพื่อให้ได้โปรแกรมที่กระชับรัดและทำงานได้ถูกต้องเข้าใจง่าย

ขั้นตอนการสร้างฉาก และการสร้างรูปภาพต่างๆที่ใช้ในเกม

ขั้นตอนนี้ใช้ระยะเวลาสั้นๆ โดยทำการสร้างฉากที่ใช้ในการเล่นเกมน และรูปภาพต่างๆซึ่งในที่นี้เราจะรวมถึงรูปที่เป็น Logo ตอนเริ่มเล่นเกมด้วย โดยข้อมูลเหล่านี้จะรวมเก็บไว้เป็นแหล่งข้อมูลต่างๆที่ใช้ในเกม เราสามารถสรุปขั้นตอนการวางแผนเป็นแผนภาพได้ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการวางแผน

3.1 ขั้นตอนการออกแบบ

ขั้นตอนการออกแบบมีรายละเอียดดังนี้

3.1.1 การออกแบบระบบเกม

การออกแบบระบบเกมเป็นส่วนที่สำคัญที่สุดในขั้นตอนการออกแบบทั้งหมด เพราะระบบเกมคือทุกสิ่งทุกอย่างตั้งแต่แนวความคิดของเกม การดำเนินเรื่องของเกม จนถึงการดำเนินเกม โดยการออกแบบมีรายละเอียดต่างๆ ดังนี้

- การออกแบบทั่วไป
- การออกแบบคำสั่งต่างๆ และการรับข้อมูล
- การออกแบบรูปภาพและตัวต่อจิกซอร์ที่ใช้ในการเล่นเกมน

3.1.1.1 การออกแบบทั่วไป

การออกแบบทั่วไปเป็นการออกแบบที่เป็นรูปลักษณะของเกม โดยมีสิ่งต่างๆ ดังนี้

- รูปแบบของฉากที่ใช้ในการเล่นเกมน
- การออกแบบตัวต่อจิกซอร์ที่ใช้ในการเล่นเกมน
- เกมมีลักษณะเป็น Realtime คือ สามารถที่จะทำการเล่นและดำเนินการได้เลย
- มุมมองที่ใช้ในการเป็นมุมมองแบบ 2 มิติ
- Interface ที่ใช้ในการติดต่อกับผู้เล่น

3.1.1.2 การออกแบบคำสั่งต่างๆและการรับข้อมูล

เป็นการออกแบบคำสั่งที่ใช้ในการควบคุม และเล่นเกมต่างๆ รวมทั้งการออกแบบรูปแบบการรับข้อมูลเกี่ยวกับตัวเกม เช่น การเลือกรูปภาพที่จะเล่น เป็นต้น ซึ่งโดยทั่วไปสามารถใช้เมาส์ปุ่มซ้ายอย่างเดียวนแล้วทำการ Click หรือ Double Click ด้วยเมาส์ปุ่มซ้าย โปรแกรมจะทำการตรวจสอบข้อมูล หรือคำสั่งที่ทำการรับเข้ามาแล้วทำการเลือกการทำงานที่เหมาะสมกับคำสั่งนั้นๆเพื่อที่จะให้เกมของเราทำการดำเนินต่อไปได้โดยไม่ติดขัดหรือเกิดจุดบกพร่องต่างๆขึ้น

ตัวเกมมีลักษณะคำสั่งที่ใช้ดำเนินการที่สำคัญ ดังนี้

รูปแบบของคำสั่งต่างๆที่มีอยู่ในตัวเกมนั้นถูกจัดกระทำขึ้นโดยใช้ Menu Editor ใน Visual Basic 6.0 ซึ่ง จะได้อธิบาย ดังต่อไปนี้

- ไฟล์
- ภาพ
- เสี่ยงขณะเล่น
- ระดับการเล่น
- รายละเอียด

โดยที่แต่ละคำสั่งมีรายละเอียดปลีกย่อยดังต่อไปนี้

ไฟล์

เป็นคำสั่งที่ใช้ในการควบคุมการทำงานที่เป็นส่วนที่สำคัญของตัวเกม โดยที่ในส่วนของการควบคุมตัวเกมทั้งหมดนั้นยังมีคำสั่งย่อยลงไปอีก ดังนี้

- เริ่มเกมใหม่ เป็นส่วนที่ควบคุมเกี่ยวกับการเริ่มเล่นเกมใหม่ และใช้ในการ Clear ค่าเริ่มต้นของข้อมูลทั้งหมดเพื่อใช้ในการเล่นเกมต่อไป
- คละจีนส่วน เป็นส่วนที่ใช้สำหรับการสุ่ม (Random) ให้ตัวต่อจิกซอวันนั้นขึ้นมาแสดงอยู่บนฉากที่ใช้ในการเล่น
- ตัวช่วย เป็นส่วนที่ให้ผู้เล่นสามารถที่จะทำการเลือกที่ใช้ตัวช่วยในการเล่นเพื่อคอยช่วยให้ผู้เล่นสามารถที่จะเริ่มเล่นได้อย่างมีหลักการ หากเริ่มต้น ไม่ถูกต้อง
- เฉลย เป็นส่วนที่ใช้ในการเฉลยคำตอบในการของภาพตัวต่อจิกซอวันนั้นๆ ก่อนออกจากเกมสเป็นส่วนที่ใช้ในการออกจากเกมเมื่อไม่ต้องการเล่นเกมต่อไป

ภาพ

เป็นส่วนที่ทำการควบคุมเกี่ยวกับการทำงานเกี่ยวกับรูปภาพทั้งหมดที่เกี่ยวกับตัวเกม ซึ่งอาจจะมีคำสั่งย่อยๆในส่วนนี้ดังต่อไปนี้

- เปลี่ยนรูป เป็นส่วนที่ใช้สำหรับการเปลี่ยนรูปภาพที่ใช้ในการเล่น เพราะว่ายากเกินไปหรือต้องการที่จะเปลี่ยนรูปภาพในการเล่น
- เพิ่มภาพ เป็นส่วนที่ใช้ในการเพิ่มรูปภาพที่เราต้องการจะเล่นลงเก็บไว้ในตัวเกมเพื่อใช้ในการเล่นต่อไป หรือต้องการเล่นภาพตัวนั้นๆแทน

เสียงขณะเล่น

เป็นส่วนที่ใช้ในการในการควบคุมเกี่ยวกับการเล่นไฟล์เพลงต่างที่ใช้ในตัวเกมซึ่งจะเป็น Midi File ทั้งหมด โดยหากเราต้องการฟังเสียงในขณะที่ทำการเล่นเราก็ทำการเปิดที่ตัวคำสั่งนี้ด้วยการ Click Mouse ที่ปุ่มซ้ายมืออีกทั้งยังมีเสียงที่แสดงถึงการวางตำแหน่งว่าผิด หรือถูกด้วย

ระดับการเล่น

เป็นส่วนที่ใช้ให้ผู้เล่นทำการเลือกระดับการเล่นตามความสามารถของผู้เล่น ซึ่งมีทั้งหมด 3 ระดับ คือ ระดับง่าย ระดับปานกลาง ระดับยาก ซึ่งแต่ละระดับจะมีจำนวนตัวต่อจิกซอวันตามนี้ คือ 16 , 32 ,48 ขึ้นตามลำดับ

รายละเอียด

เป็นส่วนที่อธิบายเกี่ยวกับตัวเกม และรายละเอียดต่างๆเกี่ยวกับตัวเกม รวมถึงวิธีการให้คะแนนที่ใช้ในการเล่นอีกด้วย

3.1.1.3 การออกแบบรูปภาพและตัวต่อจิกซอร์ที่ใช้ในการเล่นเกม

เป็นการออกแบบเกี่ยวกับรูปภาพที่ใช้ในการเล่นและตัวต่อจิกซอร์ที่ใช้ในการเล่นจะมีรายละเอียด ดังต่อไปนี้

- การออกแบบรูปภาพ เป็นส่วนที่ใช้ในการออกแบบรูปภาพโดยรูปภาพที่ได้นั้นจะถูกทำการเก็บลงใน Control Image เพื่อนำไปใช้ในการเล่นตามขนาดของรูปภาพ หากมีขนาดเกิน 3600x56 ไบต์ก็ทำการเก็บลงใน Control Image ที่มีขนาด 300x400 Pixel
- การออกแบบตัวต่อจิกซอร์ เป็นการออกแบบตัวต่อจิกซอร์ที่ใช้ในการเล่นซึ่งในที่นี้นั้นเราจะใช้ตัวต่อจิกซอร์ที่มีรูปร่างเป็นสี่เหลี่ยมจัตุรัสขนาด 50x50 Pixel เป็นตัวต่อจิกซอร์ซึ่งใช้ในการเล่น ซึ่งจะถูเก็บโดย Control Picture ตามระดับที่ผู้เล่นได้เลือกไว้
- การออกแบบรูปเมาส์ที่ใช้ในการเล่น เป็นการออกแบบรูปเมาส์ที่ใช้ในขณะที่เล่นซึ่งมีลักษณะดังรูปที่ 3.2 และรูปที่ 3.3



รูปที่ 3.2 รูปเมาส์ที่ใช้ในขณะที่เล่นเกม



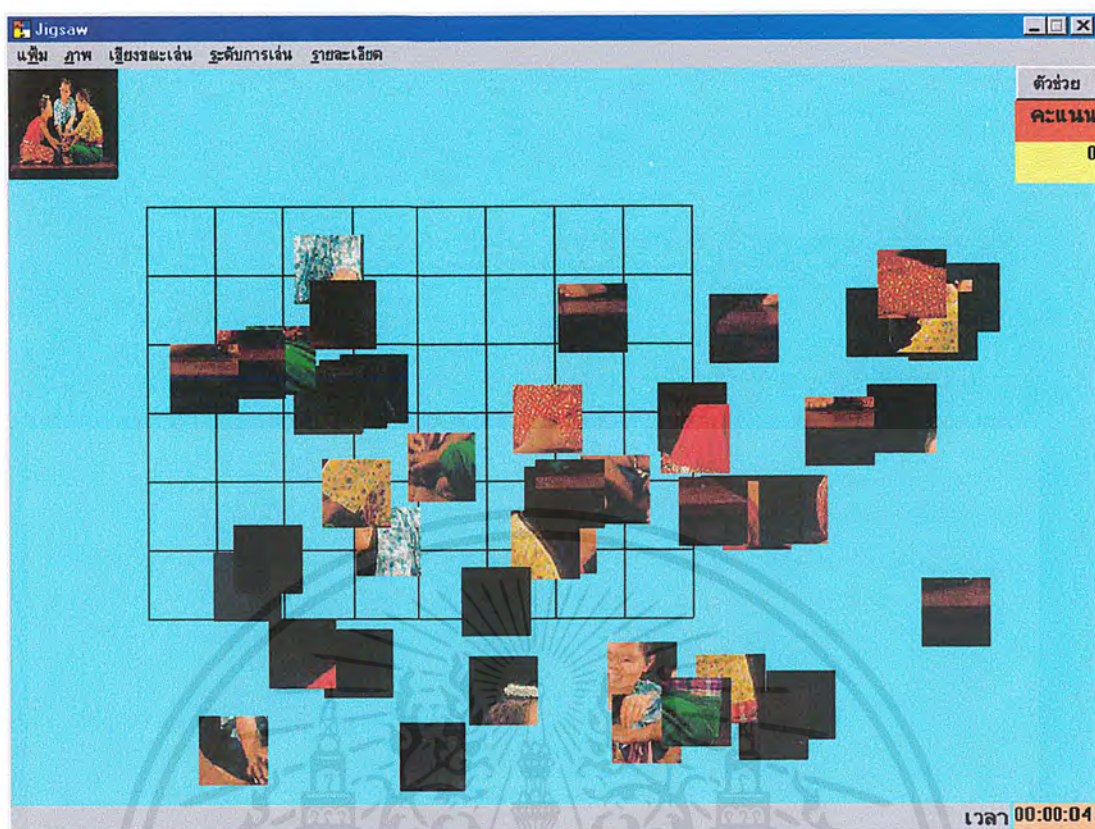
รูปที่ 3.3 รูปเมาส์ขณะทำการเลือกตัวต่อจิกซอร์

ซึ่งรูปเมาส์เหล่านี้จะมีนามสกุลเป็น .ico โดยภาพเมาส์เหล่านี้ทำการสร้างจากโปรแกรม Icon Editor และทำการกำหนดศูนย์กลาง Center ให้ใหม่ด้วย

ส่วนรูปเมาส์ที่ใช้ในการเลือกคำสั่งต่างนั้นจะใช้รูปเมาส์ที่เป็นค่า Default ของ Windows

3.1.2 การออกแบบ Interface ที่ใช้ติดต่อกับผู้เล่น

ในการออกแบบส่วนที่ติดต่อกับผู้เล่นนั้นจะแสดงถึงสถานะของเกมในขณะที่เล่น เช่น ปุ่มคำสั่งต่างๆ ซึ่งหน้าจอส่วนที่ติดต่อกับผู้เล่นมีลักษณะดังรูปที่ 3.4



รูปที่ 3.4 แสดงส่วนติดต่อกับผู้เล่น

จากรูปจะมีส่วนที่สำคัญทั้งหมด 3 ส่วนดังนี้

- กลุ่ม Menu คำสั่ง
- กลุ่มที่ใช้ในการเลือกรูปภาพ
- กลุ่มที่แสดงวันที่ และเวลาขณะเล่นเกม

3.1.2.1 กลุ่ม Menu คำสั่ง

เป็นส่วนที่ใช้ในการแสดงคำสั่งต่างๆที่มีอยู่ในตัวเกมของเรา

3.1.2.2 กลุ่มที่ใช้ในการเลือกรูปภาพ

เป็นส่วนที่ใช้ในการแสดงไฟล์รูปภาพทั้งหมดที่มีเพื่อที่จะให้เราเลือกเพื่อที่จะทำการเล่น

3.1.2.3 กลุ่มที่แสดงวันที่ และเวลาขณะเล่นเกม

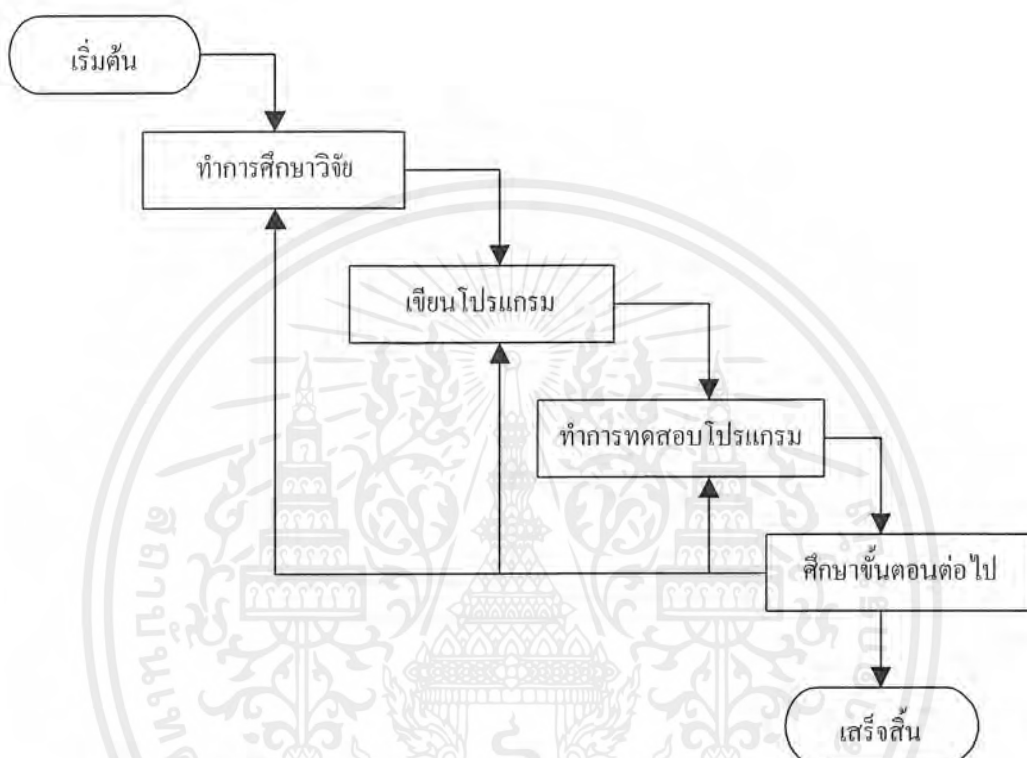
เป็นส่วนที่ใช้แสดงเวลาที่ใช้ในขณะเล่นเกม และคะแนนที่ได้ในการเล่นเกมนั้นๆ

3.1.3 การออกแบบวิธีการโต้ตอบของฝ่ายคอมพิวเตอร์

ส่วนนี้เป็นส่วนที่ใช้ในการโต้ตอบของฝ่ายคอมพิวเตอร์ เช่น เมื่อมีการวางผิดที่ก็จะมีเสียงร้องขึ้นมาว่าวางผิดที่ หรือทำการนำตัวต่อจิกซอว์กลับไปวางไว้ยังตำแหน่งเดิม เป็นต้น

3.2 ขั้นตอนการเขียนโปรแกรม

ส่วนใหญ่แล้วขั้นตอนต่างๆจะมีความเป็นเส้นตรง หมายถึงเมื่อทำงานเสร็จงานหนึ่งก็จะทำงานต่อไปโดยไม่มีการทำงานซ้ำที่งานนั้นใหม่ แต่ขั้นตอนการเขียนโปรแกรมมีความซับซ้อนมากขึ้นและทำการเขียนโปรแกรม และหากยังไม่สำเร็จ ก็จะต้องทำการศึกษาเพิ่มเติมใหม่ แสดงได้ดังรูปที่ 3.5



รูปที่ 3.5 ขั้นตอนการเขียนโปรแกรม

โปรแกรมเกมจะถูกแบ่งเป็นส่วนย่อยๆหลายส่วนดังนี้

- ส่วนการแสดงรายละเอียดของตัวเกม
- ส่วนแสดงการเก็บรูปภาพ
- ส่วนหลักที่ควบคุมลักษณะของการเล่นของตัวเกม
- ส่วนที่ใช้ในการใส่ชื่อผู้เล่นที่คิด 10 อันดับแรกในแต่ละระดับ
- ส่วนที่ใช้แสดงผู้ได้คะแนนสูงสุด 10 อันดับแรกของระดับการเล่นในขณะนั้น

เหตุผลหลักของผู้ทำโครงการแบ่งการทำงานออกเป็นหลายๆส่วน เพื่อต้องการจัดแบ่งหน้าที่การทำงานให้เป็นประเภทเดียวกัน ทำให้ง่ายต่อการตรวจสอบ แก้ไข หรือเพิ่มเติมหลักการ แนวความคิดและรูปแบบของข้อมูลในแต่ละส่วนการทำงาน

3.2.1 ส่วนแสดงรายละเอียดของตัวเกม

ในส่วนนี้เป็นส่วนที่ใช้ในการแสดงรายละเอียดต่างๆของตัวเกม ซึ่งเราจะใช้ Control SSTab โดยเราทำการกำหนดว่าเราจะให้ตัว Control SSTab นั้นมีทั้งหมดกี่ Tab โดยในที่นี้นั้นจะถูกนำไปใช้ใน Form frmHelpGame และ frmHelpScore โดยที่ในแต่ละ Form นั้นมีลักษณะการทำงานที่คล้ายคลึงกัน หรือแทบจะบอกได้ว่าเหมือนกันเพราะฉะนั้นเราจึงขออธิบายการทำงานของตัวโปรแกรมใน Form frmHelpGame แทนซึ่งตัวโปรแกรมนั้นจะอยู่ในไฟล์ frmHelpGame ซึ่งมี Code การทำงานต่อไปนี้

```
Dim HelpStr As String
```

```
//กำหนดข้อความที่ใช้ในการบอกรายละเอียดเกี่ยวกับตัวเกม
```

```
Private Sub Detail_Change(Index As Integer)
```

```
    Detail(Index).Text = HelpStr
```

```
End Sub
```

```
//ฟังก์ชันที่จะทำหน้าที่เปลี่ยนแปลงข้อความเกี่ยวกับรายละเอียดของตัวเกมเมื่อมีการเปลี่ยนแปลงค่า Index ของข้อความ
```

```
Private Sub Form_Load()
```

```
    HelpStr = Detail(0).Text
```

```
End Sub
```

```
//เมื่อทำการ Load Form ขึ้นมาให้ข้อความรายละเอียดเกมเป็นรายละเอียดในส่วนแรกจากทั้งหมด 5 ส่วน
```

```
Private Sub mOK_Click()
```

```
    Unload Me
```

```
End Sub
```

```
//เมื่อมีการกดปุ่มตกลงก็ทำการปิด Form แล้วออกจากตัวโปรแกรมนี้
```

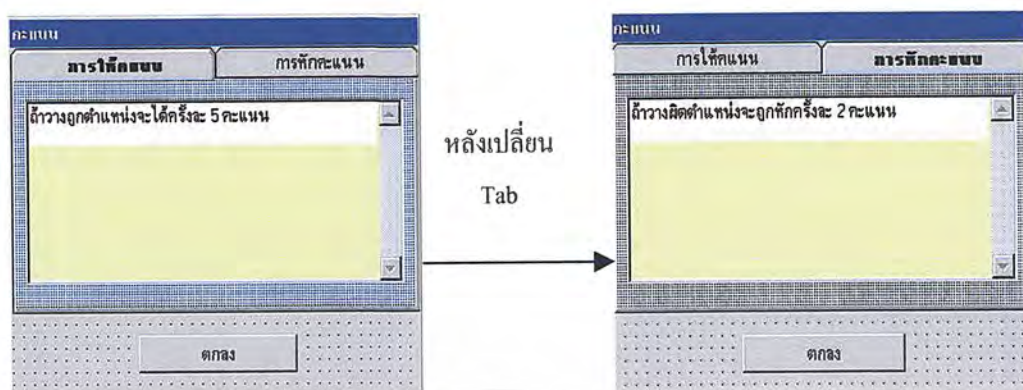
```
Private Sub mMenu_Click(PreviousTab As Integer)
```

```
    HelpStr = Detail(mMenu.Tab).Text
```

```
End Sub
```

```
//ฟังก์ชันที่จะทำงานเมื่อมีการกดที่แถบเมนูต่างๆ โดยจะมีค่าของแถบเมนูที่ทำการกดซึ่งมีค่าเป็นตัวเลข ส่งมาเพื่อทำการเปลี่ยนให้ข้อความรายละเอียดของเกมไปยังแถบเมนูนั้นๆ
```

ซึ่งการทำงานนั้นสามารถแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 แสดงการเปลี่ยน Tab ของ SSTab

3.2.2 ส่วนแสดงการเก็บรูปภาพ

เป็นส่วนที่ใช้ในการเก็บรูปภาพ หรือจะกล่าวอย่างง่ายก็คือการเก็บรูปภาพลงใน Folder ที่ทำการเก็บภาพอยู่ซึ่งในตัวเกมนี้นั้นจะทำการเก็บที่ Folder ใดก็ได้เพื่อสะดวกต่อการที่ไม่ต้องทำการ Check Path เพื่อตรวจสอบรูปซึ่งตัว Code นั้นอยู่ในไฟล์ frmInsPicture ดังนี้

```
Dim TempDrive As String
```

```
//ทำการประกาศชื่อ Drive เป็นข้อมูลประเภท String
```

```
Private Sub DirPicture_Change()
```

```
FilePicture.Path = DirPicture.Path
```

```
End Sub
```

//ฟังก์ชันที่ทำการกำหนดค่าว่าเมื่อมีการเปลี่ยนแปลง Directory ก็จะกำหนดให้ Path ของ FilePicture เป็น Path ของ Directory นั้นๆด้วย

```
Private Sub DrivePicture_Change()
```

```
ret1:
```

```
On Error GoTo err1
```

```
DirPicture.Path = DrivePicture.Drive
```

```
TempDrive = DrivePicture.Drive
```

```
Exit Sub
```

```
err1:
```

```
If MsgBox(UCCase(DrivePicture.Drive) & "\" is not accessible.The device is not ready.", 5, "Error") = 4 Then
```

```
Resume ret1
```

Else

DrivePicture.Drive = TempDrive

End If

End Sub

//ฟังก์ชันนี้จะทำการตรวจสอบว่าเมื่อมีการเปลี่ยนแปลง Drive นั้นมีการทำงานอยู่หรือเปล่า และสามารถที่จะทำการเข้าไปใช้ข้อมูลได้หรือเปล่า ถ้าได้ก็ทำการกำหนดให้ DrivePicture เท่ากับ TempDrive ที่เก็บค่า Drive ที่เปลี่ยนอยู่ แต่ถ้าไม่ได้ก็ทำการแจ้งข้อผิดพลาดที่เกิดขึ้นออกมา เช่น ไม่มี Drive A: อยู่ในเครื่อง เป็นต้น

Private Sub FilePicture_Click()

Set Preview.Picture = LoadPicture(DirPicture.Path & "\" & FilePicture.FileName)

FilePicName.Text = FilePicture.FileName

End Sub

//ฟังก์ชันนี้จะทำงานเมื่อมีการ Click ที่ FilePicture เพื่อทำการเลือกภาพที่จะทำการเล่นซึ่งจะมีภาพตัวอย่างแสดงขึ้นมาเพื่อให้เราดูว่าเราจะทำการเล่นภาพนี้หรือไม่

Private Sub FileType_Click()

Dim FileExtension As String

Select Case FileType.ListIndex

Case 0

FileExtension = "*.bmp;*.gif;*.jpg"

Case 1

FileExtension = "*.bmp"

Case 2

FileExtension = "*.gif"

Case 3

FileExtension = "*.jpg"

End Select

FilePicture.Pattern = FileExtension

End Sub

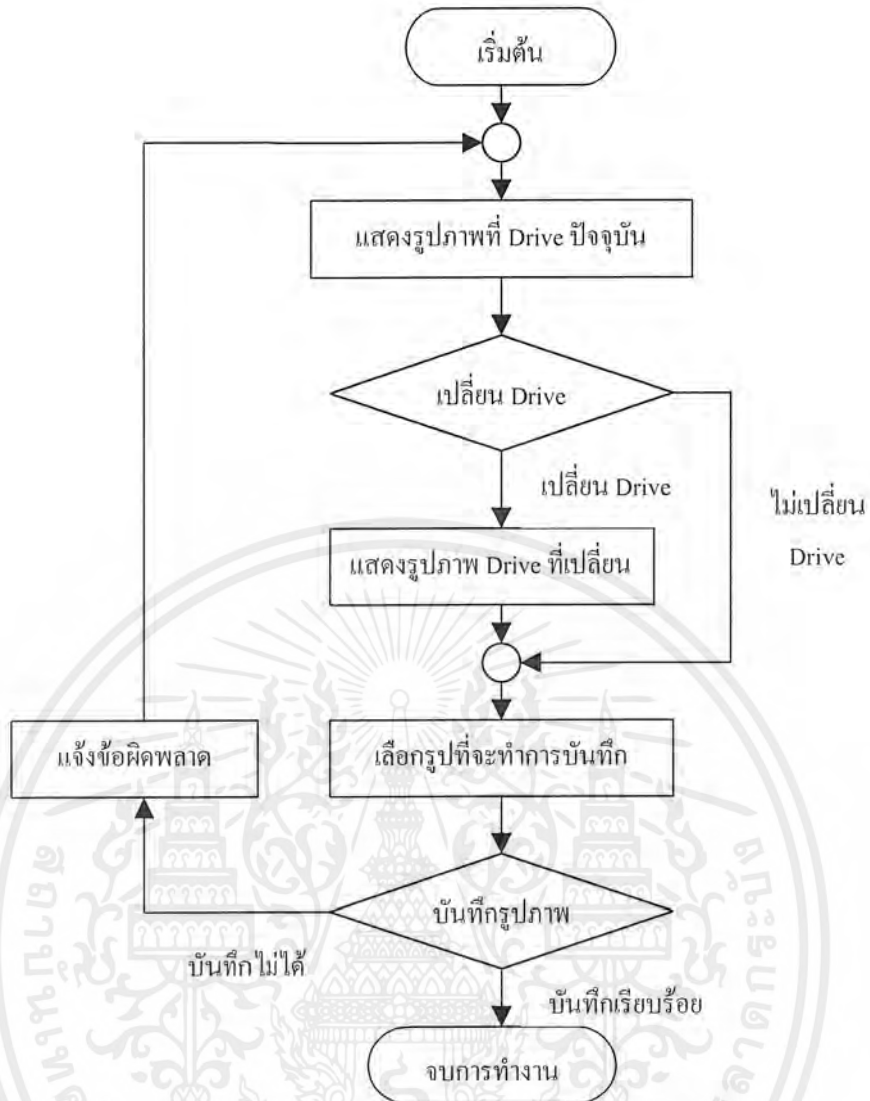
//ฟังก์ชันนี้จะทำการตรวจสอบว่าไฟล์รูปภาพที่เราจะแสดงนั้นเราจะให้แสดงไฟล์รูปภาพชนิดใดได้บ้างซึ่งในที่นี้ก็คือให้แสดงเฉพาะชนิดรูปภาพประเภทไฟล์ Bitmap , Jpeg และ Gif เท่านั้นชนิดอื่นไม่สามารถแสดงได้

Private Sub Form_Load()

```

FileType.Text = "รูปภาพทั้งหมด"
TempDrive = "c:"
End Sub
//เมื่อ Form นั้นถูก Load ขึ้นมาก็จะทำการแสดงรูปภาพทั้งหมดใน Drive C:
Private Sub mCancel_Click()
    Unload Me
End Sub
//ถ้ามีการกดปุ่มนี้ก็จะทำการออกจากตัวโปรแกรม
Private Sub mSave_Click()
    If FilePicName.Text = "" Then
        MsgBox ("โปรดใส่ชื่อไฟล์ที่ช่องชื่อไฟล์")
    Else
        On Error GoTo err
        Call FileCopy(DirPicture.Path & "\" & FilePicture.FileName, App.Path & "images\" &
FilePicture.FileName)
        Unload Me
err:
        MsgBox ("มีไฟล์นี้อยู่แล้ว กรุณาเลือกภาพใหม่")
    End If
End Sub
//ฟังก์ชันนี้จะทำการบันทึกรูปภาพที่เราได้ทำการเลือกเอาไว้ลงในเพิ่มข้อมูลรูปภาพของเรา และ
จะแจ้งข้อผิดพลาดต่างๆด้วยว่าไม่สามารถทำการบันทึกภาพได้เพราะเหตุใด เช่น มีไฟล์รูปภาพตัวนี้
อยู่แล้วจึงไม่สามารถที่จะทำการบันทึกได้โปรดเลือกไฟล์รูปภาพใหม่ เป็นต้น
ซึ่งสามารถที่จะเขียนแผนผังการทำงานได้ดังรูปที่3.7

```



รูปที่ 3.7 ผังแสดงการทำงานของกรบันทึกรูปภาพลงเพิ่มข้อมูลรูปภาพ

3.2.3 ส่วนหลักที่ควบคุมลักษณะการเล่นของตัวเกม

เป็นส่วนที่ทำหน้าที่หลัก และมีความสำคัญมากที่สุดในตัวเกมของเรา เพราะว่าในส่วนนี้นั้นจะทำการรวมฟังก์ชันต่างๆที่ใช้ในการเล่นเกมนั้น ไม่ว่าจะเป็นการคำนวณการสร้างตาราง การตัดรูปภาพที่ใช้เล่นเป็นตัวต่อจิกซอร์ การตรวจสอบตำแหน่งที่วางตัวต่อจิกซอร์ว่าถูกหรือไม่เป็นต้น ซึ่งเราจะได้อธิบายลักษณะของตัว Code ที่สำคัญของแต่ละฟังก์ชันมีดังต่อไปนี้

- ฟังก์ชัน InitNewGame
- ฟังก์ชัน CreateTile
- ฟังก์ชัน FilePicture_Click
- ฟังก์ชัน FilePicture_DblClick
- ฟังก์ชัน Form_Load
- ฟังก์ชัน LoadSound

- ฟังก์ชัน Form_Unload
- ฟังก์ชัน gameTime_Timer
- ฟังก์ชัน Helper
- ฟังก์ชัน LogoTimer_Timer
- ฟังก์ชัน Answer
- ฟังก์ชัน ChoosePicture
- ฟังก์ชัน ChangeTableSize
- ฟังก์ชัน DrawTable
- ฟังก์ชัน TilePositioning
- ฟังก์ชัน ShowTime
- ฟังก์ชัน RedrawTable
- ฟังก์ชัน mEasy_Click , mNormal_Click , mDifficult_Click
- ฟังก์ชัน Form_MouseMove
- ฟังก์ชัน PctTile_MouseDown
- ฟังก์ชัน Form_MouseUp
- การตรวจสอบความถูกต้องของการวางตัวต่อจิกซอร์

หลักการและการทำงานของฟังก์ชัน

3.2.3.1 ฟังก์ชัน InitNewGame

ฟังก์ชันนี้จะทำการกำหนดให้ตัวต่อจิกซอร์สามารถที่จะทำการเล่น หรือเคลื่อนที่ได้ รวมทั้งยังกำหนดค่าของตัวแปรต่างๆที่ใช้ในตอนเริ่มต้นเกมให้ถูกกำหนดค่าตามที่เราได้ตั้งเอาไว้ และจะทำการเรียก ฟังก์ชัน TilePositioning ต่อไป

3.2.3.2 ฟังก์ชัน CreateTile

ฟังก์ชันนี้จะทำการ Copy ขนาดของไฟล์ภาพตัวอย่างมายังตัวต่อจิกซอร์ของเราตามขนาดของตัวต่อจิกซอร์ และระดับของการเล่น เช่น ทำการ Copy ไฟล์ภาพจากภาพตัวอย่างมาลงยังตัว Control PictureBox ที่ทำเป็นตัวต่อจิกซอร์ หรือกล่าวได้ว่าทำการ Copy ขนาดไฟล์ภาพตัวอย่างขนาด 50x50 Pixel จากมุมซ้าย และมุมบน ไปยังตัว Control PictureBox ที่เป็นตัวต่อจิกซอร์ขนาด 50x50 Pixel ที่ได้เตรียมการไว้ จนจบขนาดของภาพตัวอย่าง เป็นต้น

3.2.3.3 ฟังก์ชัน FilePicture_Click

เป็นส่วนที่ทำการให้เราเลือกว่าเราจะทำการเล่นภาพไหนก็ทำการ Click ไปเลือกไปเพื่อทำการเลือกเล่นรูปภาพนั้นๆ

3.2.3.4 ฟังก์ชัน FilePicture_DblClick

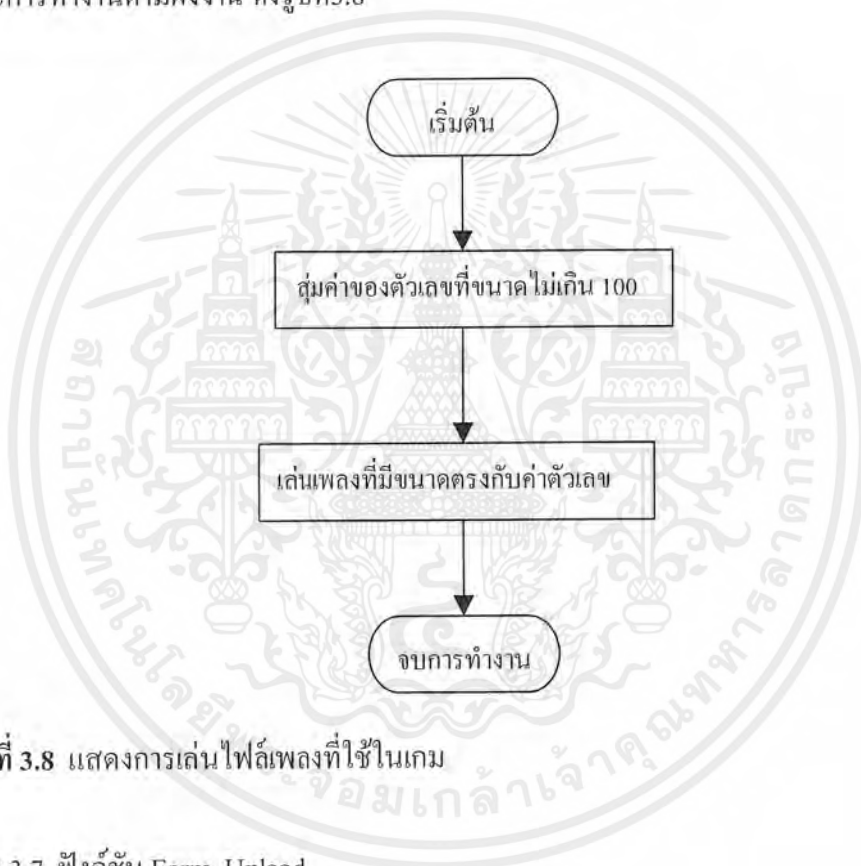
เป็นส่วนที่ทำการเลือกภาพนั้นๆแล้วเพื่อจะทำการเล่นเกม เพราะฉะนั้นเมื่อทำการ Double Click แล้วก็จะทำการ Load ภาพนั้นขึ้นมาและทำการส่งไปทำการตัดเป็นตัวต่อจิกซอร์ว์ และทำการเล่นเกมต่อไป

3.2.3.5 ฟังก์ชัน Form_Load

เป็นส่วนที่ใช้ในการกำหนดค่าของตัวแปรต่างๆที่ใช้ในการเล่นเกมของตอนเริ่มต้นเล่นเกม

3.2.3.6 ฟังก์ชัน LoadSound

เป็นส่วนที่ทำการสร้างตัวเลขสุ่มขึ้นมาเพื่อที่ใช้ในการเล่นไฟล์เพลงที่เป็นลักษณะ Midi File ในขณะที่เล่นเกมและทำการเปิดไฟล์เพลงที่ใช้เล่นเมื่อวางตัวต่อจิกซอร์ว์ผิดที่ หรือถูกตำแหน่ง ซึ่งมีลักษณะการทำงานตามผังงาน ดังรูปที่3.8



รูปที่ 3.8 แสดงการเล่นไฟล์เพลงที่ใช้ในเกม

3.2.3.7 ฟังก์ชัน Form_Unload

เป็นฟังก์ชันที่ใช้ในการยกเลิกการกำหนดค่าตัวแปรต่างๆเพื่อทำการออกจากตัวโปรแกรมเกมที่เราเล่นไว้ โดยทำการซ่อน Form เอาไว้

3.2.3.8 ฟังก์ชัน gameTime_Timer

เป็นส่วนที่ใช้ในการแสดงเวลาที่ใช้ในการเล่นเกมน ซึ่งเราจะแสดงอยู่ในรูปของรูปแบบเวลา ดังนี้ คือ hh:mm:ss ซึ่งเป็นหน่วย ชั่วโมง นาที และวินาที

3.2.3.9 ฟังก์ชัน Helper

เป็นส่วนที่ใช้เป็นตัวช่วยในการเล่นเกมนั้นเพื่อให้ง่าย หรือรู้แนวทางที่ทำการต่อตัวต่อจิกซอร์ ซึ่งเราจะแบ่งตัวช่วยของเราออกเป็น 3 ระดับ เพื่อให้ง่ายต่อการตรวจสอบ และทำการแก้ไขหากเกิดข้อผิดพลาด ซึ่งสามารถที่จะแสดง Code ดังนี้

```
Private Sub Helper()
```

```
Dim i As Integer
```

```
//กำหนดตัวแปร i เป็นข้อมูลชนิดตัวเลข
```

```
i = 0
```

```
If mEasy.Checked Then
```

```
//ตรวจสอบว่าเป็นระดับง่ายหรือเปล่า
```

```
Do While (i < 16)
```

```
If Erandom(i, 1) = 0 Then
```

```
Erandom(i, 1) = 1
```

```
Eindex = i
```

```
i = 16
```

```
Else
```

```
i = i + 1
```

```
End If
```

```
Loop
```

```
//ตรวจสอบค่าของ Index ที่ใช้เป็นตัวช่วยว่าเป็นตัวที่เท่าไร
```

```
Call Chk_Col_Row
```

```
//ทำการคำนวณว่าอยู่ใน Column หรือ Row ที่เท่าไร
```

```
ElseIf mNormal.Checked Then
```

```
//ตรวจสอบว่าเป็นระดับปานกลางหรือเปล่า
```

```
Do While (i < 32)
```

```
If Nrandom(i, 1) = 0 Then
```

```
Nrandom(i, 1) = 1
```

```
Nindex = i
```

```
i = 32
```

```
Else
```

```
i = i + 1
```

```
End If
```

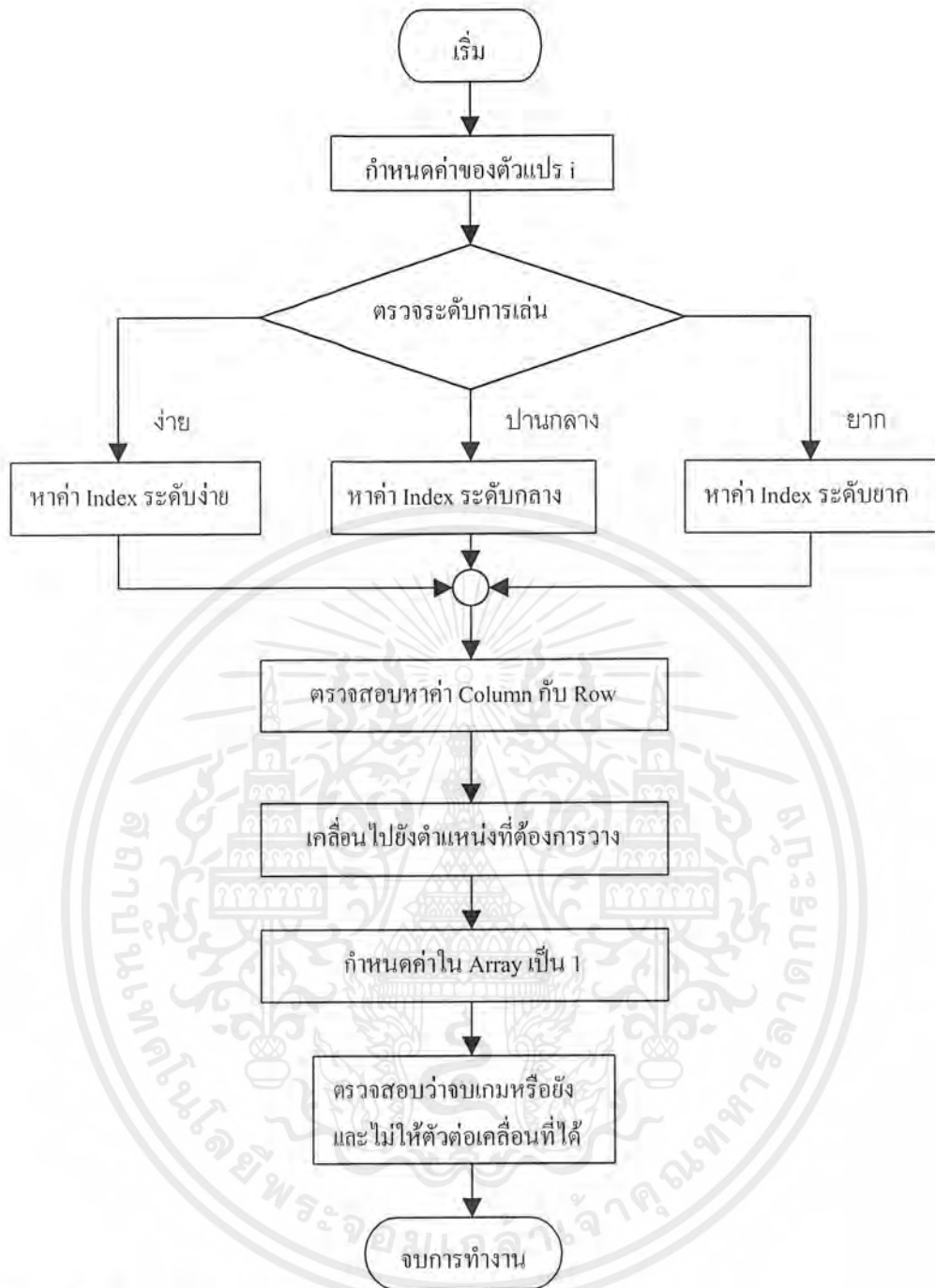
```
Loop
```

```

//ตรวจสอบค่าของ Index ที่ใช้เป็นตัวช่วยว่าเป็นตัวที่เท่าไร
    Call Chk_Col_Row1
//ทำการคำนวณว่าอยู่ใน Column หรือ Row ที่เท่าไร
    Else
//ถ้าเป็นระดับยากจะเข้ามาทำงานที่ส่วนนี้
        Do While (i < 48)
            If Drandom(i, 1) = 0 Then
                Drandom(i, 1) = 1
                Dindex = I
                i = 48
            Else
                i = i + 1
            End If
        Loop
//ตรวจสอบค่าของ Index ที่ใช้เป็นตัวช่วยว่าเป็นตัวที่เท่าไร
        Call Chk_Col_Row2
//ทำการคำนวณว่าอยู่ใน Column หรือ Row ที่เท่าไร
        End If
        Idx = col + (row * SizeWidth)
//เปลี่ยนค่าของ Column กับ Row ไปเป็นค่าของ Index ของตัวจิกซอร์ที่จะย้าย
        PctTile(Idx).Move 100 + (col * PctWidth), 100 + (row * PctHeight)
//เคลื่อนตัวจิกซอร์ไปยังที่ที่ต้องการ
        tileID(Idx) = 1
//ให้ค่าใน Array เป็น 1 เพื่อใช้ตรวจสอบว่าจบเกมหรือยัง
        PctTile(Idx).Enabled = False
//กำหนดค่าของตัวจิกซอร์เป็นค่า False เพื่อไม่ให้ทำการเคลื่อนที่ได้
    End Sub

```

ซึ่งสามารถแสดงการทำงานได้ดังแผนภาพ ดังรูปที่3.9



รูปที่ 3.9 แสดงการทำงานของฟังก์ชัน Helper

3.2.3.10 ฟังก์ชัน LogoTimer_Timer

เป็นฟังก์ชันที่ใช้ในการควบคุมการ Load ภาพ Logo ขึ้นมาแสดง โดยที่เราจะทำการแสดงในภาพนั้นในหน่วยของวินาทีว่านานแค่ไหน

3.2.3.11 ฟังก์ชัน Answer

เป็นฟังก์ชันที่ใช้ในการแสดงคำตอบของการเล่นเกมต่อภาพจิกซอวันนั้นๆ โดยที่จะทำการดึงเอาค่า Index ของตัวต่อจิกซอมาทำการวางลงในตำแหน่งที่ถูกต้อง เมื่อเรานำเอาค่า Index มาแล้ว

เราก็จะได้ทั้งค่าของ Column และค่าของ Row มาทำการคำนวณหาตำแหน่งเพื่อที่จะทำการเคลื่อนไปยังที่ที่เป็นคำตอบในส่วนเฉลย

3.2.3.12 ฟังก์ชัน ChoosePicture

เป็นส่วนของการทำงานที่เราใช้ในการเลือกภาพที่เราต้องจะเล่นเพราะว่าเมื่อเราทำการ Click เลือกภาพแล้วมันก็จะทำการแสดงภาพตัวอย่างที่เราต้องการเล่นออกมา เพื่อที่จะใช้ในการเล่นเกมต่อไป

3.2.3.13 ฟังก์ชัน ChangeTableSize

เป็นส่วนของการทำงานที่เราใช้เพื่อที่เราจะทำการเปลี่ยนแปลงขนาดของตารางที่เราทำการวาดเพื่อที่จะวางตัวต่อจิกซอว์ เมื่อเรามีการเปลี่ยนรูปที่จะเล่นเราก็ต้องทำการวาดตารางใหม่ตามขนาดของรูปใหม่เพื่อที่เราจะได้ทำการวางตัวต่อจิกซอว์ในตำแหน่งที่ถูกต้อง หรือเมื่อมีการเปลี่ยนแปลงระดับความยากที่จะเล่นก็จะคำนวณตารางแล้วทำการวาดตารางขึ้นมาใหม่ เป็นต้น

3.2.3.14 ฟังก์ชัน DrawTable

เป็นส่วนของการทำงานที่ทำการวาดภาพของตารางขึ้นมาเพื่อที่จะทำการวางตัวต่อได้ถูกต้อง

3.2.3.15 ฟังก์ชัน TilePosition

เป็นส่วนที่ควบคุมเกี่ยวกับการทำงานของการสุ่มค่าตำแหน่งของตัวต่อจิกซอว์ของเราเพื่อที่จะทำการแสดงบนหน้าจอของเราในการเล่น เพื่อให้มันละตำแหน่งการวางบนฉากหลังที่เป็นฉากที่เราให้แสดงบนเกมของเรา ซึ่งมีแผนภาพการทำงานดังรูปที่ 3.10



รูปที่ 3.10 แสดงการทำงานของการทำงานการสุ่มตำแหน่งที่วาง

3.2.3.16 ฟังก์ชัน ShowTime

เป็นส่วนที่ใช้ในการแสดงเวลาที่ทำการเล่นเกมว่าได้ดำเนินการเล่นเกมไปนานเท่าไรแล้ว

3.2.3.17 ฟังก์ชัน RedrawTable

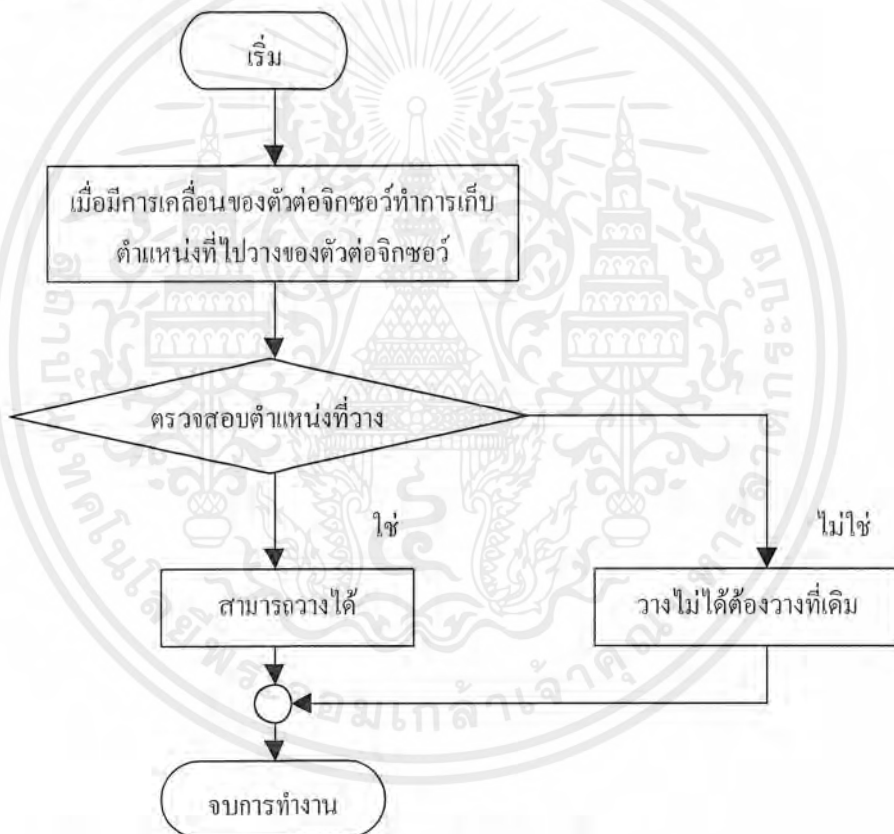
เป็นส่วนของการทำงานที่ทำการวาดภาพของตารางที่จะวางตัวต่อจิกซอว์ใหม่เพื่อที่จะทำการเล่นเกม

3.2.3.18 ฟังก์ชัน mEasy_Click, mNormal_Click, mDifficult_Click

เป็นส่วนที่ใช้ในการเปลี่ยนระดับการเล่นเกมจากระดับที่ง่ายไปยังระดับที่ปานกลาง หรือยากมากขึ้นที่จะทำการเล่น

3.2.3.19 ฟังก์ชัน Form_MouseMove

เป็นส่วนที่ทำการตรวจสอบว่าขณะที่เราทำการเคลื่อนที่ตัวต่อจิกซอว์ไปยังตำแหน่งที่เราต้องการจะวางนั้นมันมีค่าเกินกว่าขอบของฉากที่เรากำหนดหรือไม่ ถ้ามีขนาดเกินกว่าที่เรากำหนดก็จะให้เคลื่อนที่กลับไปยังที่เดิม แต่ถ้าไม่เราก็จะอนุญาตให้มันทำการวางตัวต่อจิกซอว์ที่ตำแหน่งนั้นได้ แสดงได้ ดังรูปที่ 3.11



รูปที่ 3.11 แสดงการตรวจสอบตำแหน่งที่วางตัวต่อจิกซอว์

3.2.3.20 ฟังก์ชัน PetTile_MouseDown

เป็นส่วนที่ใช้ในการทำงานเกี่ยวกับตัวต่อจิกซอว์ว่าตำแหน่งที่วางถูกต้องหรือไม่ ถ้าไม่ถูกต้องต้องทำการเคลื่อนตัวต่อจิกซอว์ไปยังตำแหน่งเดิมก่อนที่ตัวต่อจิกซอว์จะทำการเคลื่อนที่มา เหมือนการนำตัวต่อจิกซอว์ไปวางที่ตำแหน่งเดิมหากวางไม่ถูกต้อง

3.2.3.21 ฟังก์ชัน Form_MouseUp

เป็นส่วนที่ใช้ในการตรวจสอบว่าตำแหน่งที่วางตัวต่อจิกซอว์นั้นเป็นตำแหน่งที่ถูกต้องหรือไม่ โดยทำการตรวจสอบว่าตำแหน่งที่วางตรงกับฐานข้อมูลที่เรามีอยู่หรือไม่ หากตรงกับตำแหน่งที่วาง ตัวต่อจิกซอว์ก็จะทำการกำหนดค่าไม่ให้ตัวต่อจิกซอว์เคลื่อนที่ได้ และทำการตรวจสอบว่าจบเกม หรือยัง รวมทั้งทำการเพิ่มคะแนนหากวางตำแหน่งที่ถูกต้อง

3.2.3.22 การตรวจสอบความถูกต้องของการวางตัวต่อจิกซอว์

การตรวจสอบความถูกต้องของการวางตัวต่อจิกซอว์นั้นเราจะใช้การตรวจสอบตำแหน่งจาก Block ที่ทำการเก็บไว้ในส่วนของฟังก์ชัน `Form_MouseUp` ซึ่งมีวิธีการตรวจสอบดังต่อไปนี้

1. เมื่อทำการ Click ที่ชิ้นส่วนจิกซอว์จะทำการเก็บค่าของ Index ของตัวต่อจิกซอว์ที่ทำการ Click เก็บไว้ในตัวแปร `MoveIdx`

2. ทำการตรวจสอบว่าค่าของ `MoveIdx` มีค่าเป็น `-1` หรือไม่ ถ้ามีค่าเป็น `-1` ให้ทำการออกจากการทำงานของฟังก์ชัน `Form_MouseUp`

3. หลังจากนั้นก็ทำการกำหนดค่าเริ่มต้นของ Pixel X และ Pixel Y เป็นค่า 100 เพื่อเป็นตำแหน่งเริ่มต้นของการตรวจสอบการวางตำแหน่งตัวต่อจิกซอว์

4. ทำการประกาศตัวแปร `Idx`, `mycolumn`, `myrow`

5. นำค่าตำแหน่ง X และ Y ที่ได้จากฟังก์ชัน `Setcapture` มาทำการตรวจสอบตำแหน่งตาม Code ต่อไปนี้

Select Case X

Case `startX To startX + (PetWidth): mycolumn = 0`

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column แรกหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ `mycolumn` เป็น 0

Case `startX + (PetWidth) To startX + (PetWidth * 2): mycolumn = 1`

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่สองถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่สองหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ `mycolumn` เป็น 1

Case `startX + (PetWidth * 2) To startX + (PetWidth * 3): mycolumn = 2`

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่สามถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่สามหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ `mycolumn` เป็น 2

Case `startX + (PetWidth * 3) To startX + (PetWidth * 4): mycolumn = 3`

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่สี่ถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่สี่หรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ `mycolumn` เป็น 3

Case `startX + (PetWidth * 4) To startX + (PetWidth * 5): mycolumn = 4`

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่ห้าถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่หกหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ mycolumn เป็น 4

Case startX + (PctWidth * 5) To startX + (PctWidth * 6): mycolumn = 5

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่หกถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่เจ็ดหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ mycolumn เป็น 5

Case startX + (PctWidth * 6) To startX + (PctWidth * 7): mycolumn = 6

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่เจ็ดถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่แปดหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ mycolumn เป็น 6

Case startX + (PctWidth * 7) To startX + (PctWidth * 8): mycolumn = 7

//ทำการตรวจสอบว่าค่า X ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Column ที่แปดถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Column ที่แปดหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ mycolumn เป็น 7

Case Else: mycolumn = 99

//แต่ถ้าไม่ใช่กรณีใดเลยก็ทำการกำหนดค่าของ mycolumn เป็น 99

End Select

Select Case Y

Case startY To startY + (PctHeight): myrow = 0

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row แรกหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 0

Case startY + (PctHeight) To startY + (PctHeight * 2): myrow = 1

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Row ที่สองถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row ที่สองหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 1

Case startY + (PctHeight * 2) To startY + (PctHeight * 3): myrow = 2

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Row ที่สามถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row ที่สามหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 2

Case startY + (PctHeight * 3) To startY + (PctHeight * 4): myrow = 3

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Row ที่สี่ถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row ที่สี่หรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 3

Case startY + (PctHeight * 4) To startY + (PctHeight * 5): myrow = 4

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Row ที่ห้าถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row ที่ห้าหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 4

Case startY + (PctHeight * 5) To startY + (PctHeight * 6): myrow = 5

//ทำการตรวจสอบว่าค่า Y ที่ได้มีค่าอยู่ในช่วงเริ่มต้นของตัวต่อจิกซอว์ใน Row ที่หกลถึงตำแหน่งสิ้นสุดของตัวต่อจิกซอว์ใน Row ที่หกหรือไม่ ถ้าใช่ก็ทำการกำหนดค่าของ myrow เป็น 5

Case Else: myrow = 99

//แต่ถ้าไม่ใช่กรณีใดเลยก็ทำการกำหนดค่าของ myrow เป็น 99

End Select

6. ทำการคำนวณเพื่อหาค่า Idx จาก Code ต่อไปนี้

Idx = mycolumn + (myrow * SizeWidth)//นำค่า mycolumn และ myrow ที่ได้มาทำการแปลงเป็นค่า Idx ที่จะทำการเคลื่อนที่

7. ทำการตรวจสอบว่าค่า Idx และ ค่า MoveIdx มีค่าตรงกันหรือไม่ถ้ามีค่าตรงกันก็ทำการเคลื่อนที่ไปยังตำแหน่ง X และ Y ที่ได้ทำการคำนวณไว้และทำการกำหนดค่าไม่ให้สามารถเคลื่อนย้ายได้อีกแล้ว แต่ถ้าไม่ตรงก็ทำการเคลื่อนที่กลับไปยังตำแหน่งเดิม

ซึ่งขั้นตอนทั้งหมดสามารถแสดงได้ดังตัวอย่างต่อไปนี้

สมมุติว่าเราทำการเล่นที่ระดับยากเราจะมีจำนวนของตัวต่อจิกซอว์ทั้งหมด 48 ชิ้นซึ่งสามารถแสดงเป็น Block ได้ดังรูปที่3.12

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47

รูปที่ 3.12 รูปแสดงตารางแทนตำแหน่งจิกซอว์ทั้งหมด 48 ชิ้น

เมื่อเราทำการ Click เลือกตัวต่อจิกซอว์มาวางที่ตำแหน่งที่ 9 ซึ่งเป็นตัวต่อจิกซอว์ที่ 10 แล้วฟังก์ชัน SetCapture ก็จะทำให้การดึงค่า X และ Y ที่ได้ออกมา แล้วทำการเก็บค่า MoveIdx ซึ่งในที่นี้เราจะเก็บค่าเป็น 9 ซึ่งตรงกับตำแหน่งที่เราสมมุติไว้ หลังจากนั้นก็จะนำค่า X และ Y ที่ได้ไปทำการตรวจสอบจาก Code ว่าอยู่ใน Column และ Row ที่เท่าไร ซึ่งในที่นี้จะได้ค่า mycolumn และ myrow เป็น 1 และ 1 ตามลำดับจากนั้นก็ทำการคำนวณหาค่า Idx ซึ่งในที่นี้จะได้ค่า Idx = 1+(1*8) = 9 หลังจากนั้นเราก็จะทำการตรวจสอบค่าของ MoveIdx และ Idx ว่าตรงกันหรือไม่ซึ่งในที่นี้ตรงกันจึงทำการเคลื่อนตัวต่อจิกซอว์ไปยังตำแหน่งที่ 9 นั่นก็เป็นอันจบการทำงาน

3.2.4 ส่วนที่ใช้ในการใส่ชื่อผู้เล่นที่ติด 10 อันดับแรกในแต่ละระดับ

เป็นส่วนที่ใช้ในการใส่ชื่อเมื่อผู้เล่นทำคะแนนติด 10 อันดับแรกของการเล่นแต่ละระดับ

3.2.5 ส่วนที่ใช้แสดงผู้ได้คะแนนสูงสุด 10 อันดับแรกของระดับการเล่นในขณะนั้น

เป็นส่วนที่ใช้แสดงรายชื่อของผู้เล่นที่สูงที่สุด 10 อันดับแรกของแต่ละระดับออกมา โดยที่เราจะเรียงตามคะแนนจากมากไปหาน้อย โดยทำการอ่านค่าจากไฟล์ที่ทำการเก็บรายชื่อของผู้เล่นที่ติด 10 อันดับแรกมาแสดง

3.3 การทำไฟล์เพื่อติดตั้งโปรแกรม

เราสามารถที่จะทำไฟล์เพื่อติดตั้งโปรแกรมที่เราสร้างได้โดยใช้โปรแกรม Package and Deployment โดยทำการรันจากเมนู Start ของ Windows โดยเลือก Package and Deployment จากเมนู Visual Studio 6.0 Tools จากนั้นก็จะเข้าสู่ขั้นตอนการ Packaging และ การ Deployment ดังนี้

3.3.1 การ Packaging

มีขั้นตอนดังต่อไปนี้

- 3.3.1.1 เลือก Project ที่จะสร้างไฟล์ Setup เสร็จแล้วให้ Click Mouse ปุ่ม Package
- 3.3.1.2 เลือก Standard Setup package เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.3 เลือก Path ที่เก็บไฟล์ Setup ทั้งหมดที่สร้าง เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.4 เลือก File ที่จะรวมในการ Setup ซึ่งเราสามารถเพิ่ม File เข้าไปได้เองโดยการ Click Mouse ปุ่ม Add เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.5 เลือกว่าจะสร้าง CAB ไฟล์แบบใดเพื่อใช้ในการ Setup ซึ่งในที่นี้ใช้ Setup ผ่านทางแผ่นดิสก์ เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.6 ทำการตั้งชื่อ Title เมื่อโปรแกรมติดตั้งรันขึ้นมา เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.7 กำหนดโครงสร้างให้การเรียกโปรแกรมอยู่ที่เมนู Start ของ Windows เพื่อให้สามารถเข้าถึงโปรแกรมที่ติดตั้งได้ เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.8 กำหนด Path ที่จะทำการติดตั้งโปรแกรม เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.9 ต้องการที่จะ Share ไฟล์หรือไม่ถ้าต้องการก็ทำการ Click ที่ไฟล์ที่ต้องการ Share เสร็จแล้ว Click Mouse ปุ่ม Next
- 3.3.1.10 ปรากฏ Dialog Box ขึ้นมาว่าทำอะไรไปบ้าง เสร็จแล้ว Click Mouse ปุ่ม Finish

3.3.2 การ Deployment

มีขั้นตอนดังต่อไปนี้

- 3.3.2.1 เลือก Project ที่จะถูก Setup เสร็จแล้ว Click ปุ่ม Deploy
- 3.3.2.2 เลือกชื่อ Package ที่สร้างจากการ Packaging เสร็จแล้ว Click Mouse ปุ่ม Next

3.3.2.3 เลือกชนิดของการ Deployment ว่าเป็นแบบใดซึ่งในที่นี้เป็นแบบทั่วไป เสร็จแล้ว Click Mouse ปุ่ม Next

3.3.2.4 เลือก Path ที่จะใช้ติดตั้งโปรแกรม เสร็จแล้ว Click Mouse ปุ่ม Next

3.3.2.5 ปราบกฏ Dialog Box ขึ้นมาว่าทำอะไรไปบ้าง เสร็จแล้ว Click Mouse ปุ่ม Finish



บทที่ 4

ผลการทดลองและการวิเคราะห์ปัญหา

การทดลองและการวิเคราะห์ปัญหาจะเป็นการทดสอบขั้นตอนในแต่ละขั้นรวมถึงการสรุปผลการทดลองในแต่ละขั้น เพื่อนำไปวิเคราะห์ถึงปัญหาและแนวทางการพัฒนาเพื่อที่จะสามารถทราบถึงปัญหาที่เกิดขึ้นและวิธีการแก้ไขปัญหานั้นๆ และจะแสดงถึงผลดีผลเสียต่างๆที่เกิดขึ้นจากการทดสอบ

คุณสมบัติของระบบที่นำมาทดสอบ

- ระบบปฏิบัติการ Microsoft Window 98
- เครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลความเร็ว 300 Mhz
- หน่วยความจำหลักขนาด 32 MByte

ขั้นตอนการดำเนินงานทดสอบ

- ทำการติดตั้งตัว โปรแกรมและซอฟต์แวร์ที่จำเป็นต้องใช้
- ทำการรันและประมวลผลภายใต้ระบบที่กำหนด
- ตรวจสอบการใช้คำสั่งบนเมนูบาร์และการทำงานตามคำสั่งต่างๆ
- ตรวจสอบการหาข้อผิดพลาดต่างๆ

ขั้นตอนการประเมินผล

- หลังจากการติดตั้งตัวโปรแกรมแล้วสามารถใช้โปรแกรมได้
 - การประมวลผลของโปรแกรมจะต้องทำความเร็วในระดับที่ยอมรับได้
 - การใช้คำสั่งต่างๆบนเมนูบาร์จะต้องเป็นไปตามรูปแบบที่กำหนด
 - จำนวนข้อผิดพลาดที่เกิดขึ้นจะต้องไม่มีหรือเกิดขึ้นน้อยที่สุด
- ซึ่งขั้นตอนของการดำเนินการทดสอบในแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็น

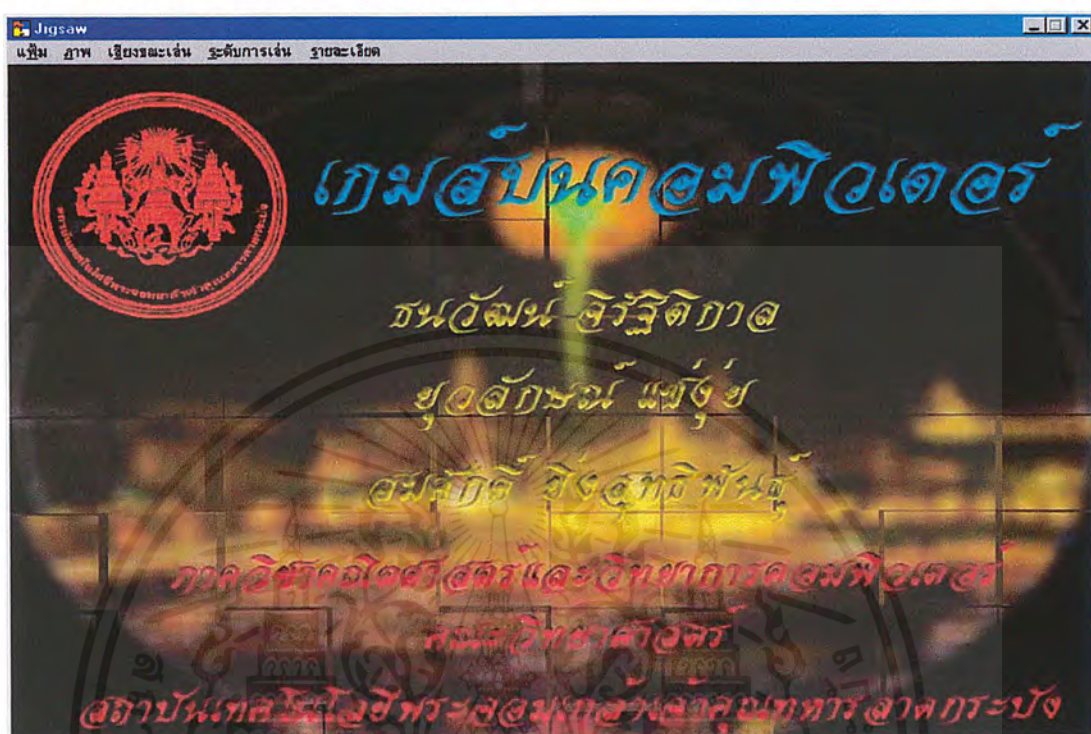
- 4.1.1 ติดตั้งตัวโปรแกรมโดยคลิกที่ไฟล์ “Setup.exe”
- 4.1.2 ตัวโปรแกรมและซอฟต์แวร์จะถูกทำการติดตั้ง
- 4.1.3 ผู้ใช้สามารถทำการรัน โปรแกรมได้โดยคลิกที่ไฟล์ “Jigsaw.exe”

ผลการทดสอบ

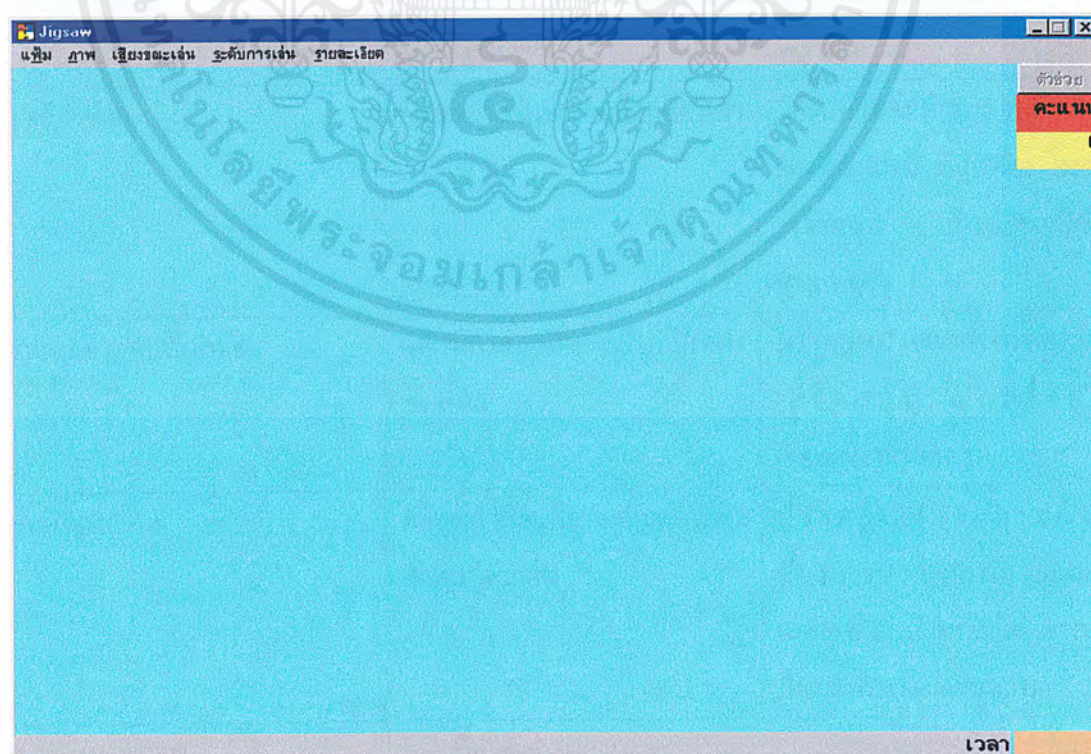
- ติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็นลงบนเครื่องคอมพิวเตอร์ได้
- ผู้ใช้สามารถรันโปรแกรมได้
- เนื้อที่ของฮาร์ดดิสก์ถูกใช้ไปประมาณ 13 MByte

4.2 ขั้นตอนการทดสอบการรันและการประมวลผลภายใต้ระบบที่กำหนด

เมื่อทำการรัน โปรแกรมภายใต้เครื่องคอมพิวเตอร์ที่กำหนดโดยคลิกที่ “Jigsaw.exe” จะปรากฏหน้าจอก่อนเข้าสู่เกมดังรูปที่ 4.1 และจะแสดงหน้าจอการเริ่มเล่นเกมดังรูปที่ 4.2



รูปที่ 4.1 แสดงภาพหน้าจอก่อนเข้าสู่เกม



รูปที่ 4.2 แสดงหน้าจอการเริ่มเล่นเกม

		เลือกภาพเพื่อเริ่มเล่นเกมใหม่ได้
การเพิ่มภาพ	คลิกที่คำสั่งภาพ แล้วคลิกที่คำสั่งเพิ่มภาพ	ปรากฏหน้าจอให้ผู้เล่นสามารถเลือกเพิ่มภาพตามแหล่งที่มาของภาพหรือสามารถยกเลิกการเพิ่มภาพ
การเปิดเสียงขณะเล่น	คลิกที่คำสั่งเสียงขณะเล่น คลิกเปิด	ปรากฏเพลงขณะเล่นเกม
การปิดเสียงขณะเล่น	คลิกที่คำสั่งเสียงขณะเล่น คลิกปิด	เป็นการปิดเพลง
การเลือกระดับการเล่นระดับง่าย	คลิกที่ระดับการเล่น คลิกที่ง่าย	ขนาดของจิกซอร์ว้จะมีขนาดใหญ่โดยจะมีจำนวนชิ้นจิกซอร์ทั้งหมด 16 ชิ้นส่วน
การเลือกระดับการเล่นระดับปานกลาง	คลิกที่ระดับการเล่น คลิกที่ปานกลาง	จิกซอร์ว้จะถูกแบ่งออกเป็นส่วนย่อยๆโดยจะมีจำนวนชิ้นของจิกซอร์ทั้งหมด 32 ชิ้น
การเลือกระดับการเล่นระดับยาก	คลิกที่ระดับการเล่น คลิกที่ยาก	จิกซอร์ว้จะถูกแบ่งออกเป็นส่วนย่อยๆโดยจะมีจำนวนชิ้นของจิกซอร์ทั้งหมด 48 ชิ้น
รายละเอียดต่างๆของเกม	คลิกที่รายละเอียด คลิกที่เกม	ปรากฏรายละเอียดอธิบายเมนูต่างๆของเกม
การให้คะแนน	คลิกที่รายละเอียด คลิกที่คะแนน	ปรากฏหน้าจอเกี่ยวกับการให้คะแนนและการหักคะแนนของเกม

4.4 ขั้นตอนการทดสอบการหาข้อผิดพลาดของโปรแกรม

ในขั้นตอนเพื่อตรวจสอบหาข้อผิดพลาดของโปรแกรมไม่มีรูปแบบที่ชัดเจน แต่จะทดสอบโดยการเล่นเกมหลายๆรอบเพื่อหาข้อผิดพลาดที่เกิดขึ้นและจากการทดสอบหาข้อผิดพลาดนั้นพบว่า ไม่มีข้อผิดพลาดอันใดเกิดขึ้น

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

การทำงานโครงการนี้ได้มีการวางแผนการทำงานในแต่ละขั้นตอนไว้เป็นอย่างดี แต่ถึงแม้ว่าได้มีการวางแผนเป็นอย่างดีก็ยังคงมีปัญหาเกิดกับขั้นตอนการทำงานบางขั้นตอน อาทิเช่น ปัญหาที่เกิดขึ้นจากความล่าช้าในการคิดระเบียบวิธีการโปรแกรมมิ่ง เนื่องจากจำเป็นต้องศึกษาจากบทความภาษาอังกฤษจากหนังสือ และจากโฮมเพจต่างๆ โดยสามารถแบ่งออกเป็นขั้นตอนต่างๆ ได้ 3 ขั้นตอน อันได้แก่

5.1 ขั้นตอนการออกแบบ

ขั้นตอนการออกแบบเป็นขั้นตอนที่รวดเร็วที่สุด เนื่องจากผู้วิจัยได้คิดวิธีการต่างๆ ของการออกแบบ และผู้วิจัยแต่ละคนก็ได้คิดไว้ก่อนแล้ว ซึ่งแนวความคิดที่ได้นั้นเกิดจากประสบการณ์ที่เคยเล่นเกมที่ตนเองชื่นชอบ แต่เกมที่ผู้วิจัยได้เล่นนั้นไม่ได้แสดงออกถึงประเพณีและวัฒนธรรมไทยเท่าที่ควร จึงมีความคิดที่จะผลิตเกมที่มีความเกี่ยวข้องกับวัฒนธรรมและประเพณีไทย ซึ่งผู้วิจัยได้ทำการรวบรวมข้อมูลในการผลิตเกม และทำการสรุปรูปแบบของเกมในงานวิจัย

5.2 ขั้นตอนการเขียนโปรแกรม

ขั้นตอนนี้สามารถทำควบคู่ไปได้กับขั้นตอนการออกแบบหน้าจอและการตกแต่งหน้าจอต่างๆ โดยขั้นตอนนี้จะขั้นตอนที่ใช้เวลานานที่สุดในงานวิจัย เนื่องจากผู้วิจัยต้องทำการศึกษาการเขียนโปรแกรมที่ต้องติดต่อกับฮาร์ดแวร์ อินพุต และเอาต์พุตซึ่งเป็นเรื่องใหม่สำหรับผู้วิจัย เมื่อศึกษาเรียบร้อยแล้วจึงลงมือเขียนโปรแกรมโดยการทำงานในส่วนนี้มีซอฟต์แวร์ที่เกี่ยวข้องที่ใช้ในการพัฒนา คือ Microsoft Visual Basic Version 6.0

5.3 ขั้นตอนการตกแต่งหน้าจอและรายละเอียดต่างๆ

ขั้นตอนนี้เป็นขั้นตอนที่ใช้ระยะเวลาหลังจากการเขียนโปรแกรมไปได้ระยะหนึ่ง เนื่องจากการออกแบบและตกแต่งส่วนประกอบต่างๆ เป็นการทำงานที่ต้องอาศัยความประณีตและละเอียดอ่อน ปัญหาที่เกิดขึ้นเกิดจากการทำงานซอฟต์แวร์กราฟฟิกต่างๆ ซึ่งต้องใช้เวลาศึกษานานพอสมควร เนื่องจากผู้วิจัยไม่เคยมีประสบการณ์ในการใช้ ซอฟต์แวร์กราฟฟิก ขั้นตอนนี้ใช้ Application คือ Adobe Photoshop Version 5.5 ซึ่งเป็นซอฟต์แวร์ที่ใช้สำหรับการตกแต่งภาพทุกชนิด ซึ่งได้นำมาตกแต่งรูป

ภาพต่างๆที่นำมาใช้เป็นรูปภาพของจิกซอว์ และใช้ในการตกแต่งหน้าจอก่อนเข้าสู่เกมให้มีความสวยงามมากยิ่งขึ้น

สรุปแล้วในการทำงานวิจัยเพื่อสร้างเกมบนคอมพิวเตอร์ การทำงานต่างๆ ตลอดจนปัญหาที่เกิดขึ้น ทำให้ผู้วิจัยสามารถทราบถึงแนวทางในการผลิตเกมในตลาดซอฟต์แวร์ไทย



ภาคผนวก ก.

คู่มือการเล่น



คู่มือการเล่นเกม

ความเป็นมา

เนื่องจากปัจจุบันอุตสาหกรรมเกมได้มีการพัฒนาไปอย่างรวดเร็ว แต่เกมที่เกี่ยวข้องกับประเพณีและวัฒนธรรมของไทยนั้นยังไม่ได้มีการพัฒนาไปเท่าที่ควร ดังนั้นเราจึงควรที่จะทำการพัฒนาอุตสาหกรรมเกมที่เกี่ยวข้องกับประเพณี และวัฒนธรรมไทย โดยผู้จัดทำได้นำเอาภาพต่างๆที่เกี่ยวข้องกับประเพณีและวัฒนธรรมไทยอันได้แก่ ภาพการละเล่นของเด็กไทย ภาพของวัดวาอารามและสถานที่ที่สำคัญต่างๆ รวมถึงภาพประเพณีไทยต่างๆมาเป็นภาพในการทำชิ้นส่วนของเกม และยังได้นำเพลงไทยต่างๆเข้ามาประกอบในการเล่นเกมน เพื่อดำรงไว้ซึ่งความเป็นไทยให้กับชนรุ่นหลังสืบต่อไป

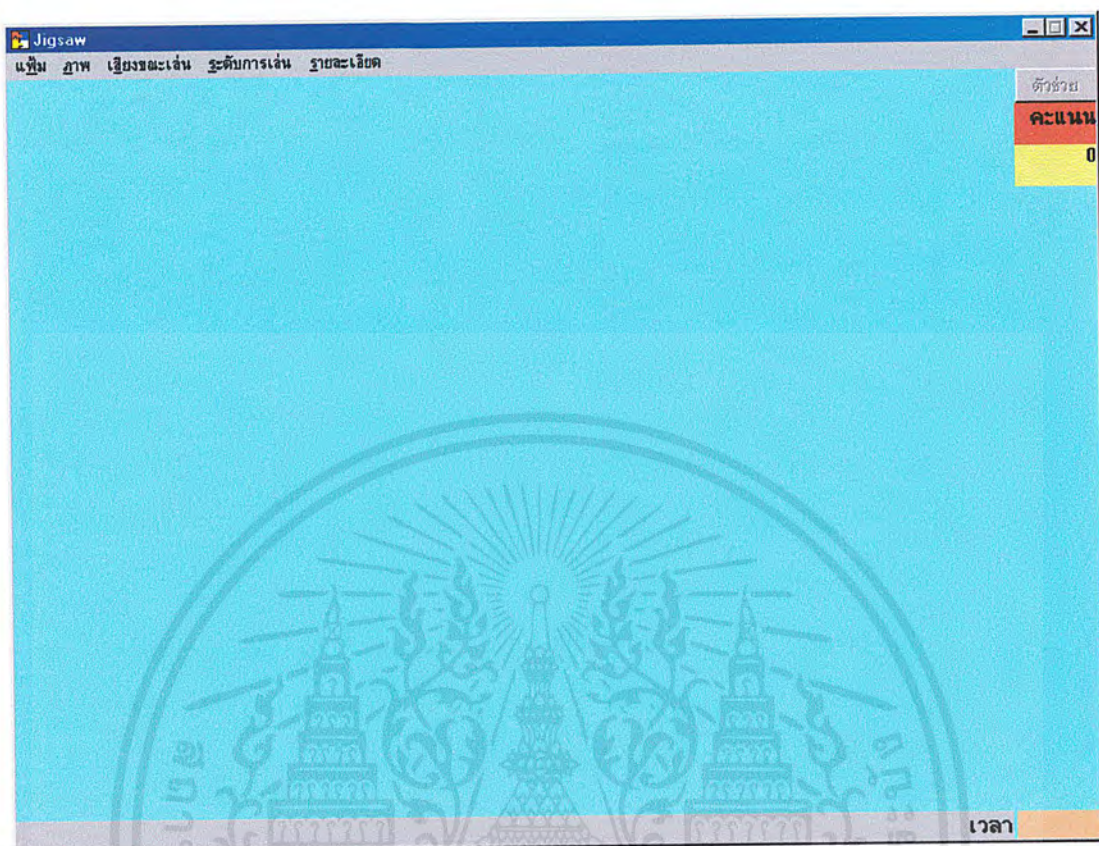
ประสิทธิภาพของระบบที่แนะนำ

- คอมพิวเตอร์ระดับความเร็ว 166 Mhz
- หน่วยความจำขนาด 32 MByte
- อุปกรณ์อินพุตประเภทเมาส์ และคีย์บอร์ด

ขั้นตอนการติดตั้งตัวโปรแกรม

- คลิกที่ไฟล์ “Setup.exe”
- ทำตามขั้นตอน
- เริ่มเล่นด้วยการคลิกที่ไฟล์ชื่อ “Jigsaw.exe”

หน้าจอการเล่น



จากรูปคำสั่งบนเมนูบาร์จะประกอบด้วยคำสั่งทั้งหมด 5 คำสั่งหลักอันได้แก่

- แฟ้ม
- ภาพ
- เสียงขณะเล่น
- ระดับการเล่น
- รายละเอียดของเกม

ซึ่งรายละเอียดของคำสั่งย่อยในแต่ละคำสั่งหลักมีรายละเอียดดังต่อไปนี้

แฟ้ม

ประกอบด้วยคำสั่งย่อยต่างๆ 5 คำสั่งย่อยดังต่อไปนี้

1. เริ่มเกมใหม่

เมื่อผู้เล่นต้องการเริ่มเล่นเกมใหม่ สามารถทำได้โดยคลิกที่คำสั่งแฟ้ม แล้วคลิกที่คำสั่งเริ่มเกมใหม่ โปรแกรมจะแสดงกรอบรายชื่อรูปภาพที่มีทั้งหมดขึ้นมาทางขวามือให้ทำการเลือกรูปภาพที่ต้องการต่อภาพ เมื่อคลิกที่กรอบรายชื่อรูปภาพนั้น 1 ครั้ง โปรแกรมจะแสดงรูปภาพตัวอย่างขึ้นมา

ทางซ้ายมือ และเมื่อดับเบิลคลิกที่กรอบรายชื่อรูปภาพนั้น โปรแกรมก็จะให้ผู้เล่นต่อภาพตามที่ผู้เล่นได้เลือกภาพไว้ โดยผู้เล่นจะต่อภาพตามภาพตัวอย่างที่อยู่ทางซ้ายมือ และกรอบรายชื่อรูปภาพนั้นก็จะหายไป เมื่อผู้เล่นทำการเลือกเมนูข้อนี้อีกครั้ง โปรแกรมจะถามผู้เล่นว่าต้องการเริ่มเล่นเกมใหม่หรือไม่ ถ้าผู้เล่นตอบว่าใช่ โปรแกรมก็จะแสดงกรอบรายชื่อรูปภาพให้ผู้เล่นได้เลือกรูปใหม่อีกครั้ง เพื่อเริ่มเล่นเกมใหม่ต่อไป ดังแสดงได้ดังรูปที่ ก-1



รูปที่ ก-1 แสดงหน้าจอเมื่อผู้เล่นได้ทำการเริ่มเล่นเกมใหม่

2. คละชิ้นส่วน

เมื่อผู้เล่นต้องการที่จะคละชิ้นส่วนของจิกซอว์สามารถทำได้โดยเลือกคำสั่งย่อยคละชิ้นส่วน โปรแกรมจะทำการคละชิ้นส่วนรูปจิกซอว์ทั้งหมด

3. ตัวช่วย

เมื่อผู้เล่นต้องการใช้ตัวช่วยสามารถทำได้โดยคลิกที่คำสั่งย่อยตัวช่วย โปรแกรมจะทำการสุ่มชิ้นส่วนรูปจิกซอว์ แล้วนำมาต่อในตำแหน่งที่ถูกต้องโดยโปรแกรมจะทำการสุ่มจิกซอว์ 1 ตัวมาวางในตำแหน่งที่ถูกต้องต่อการคลิกที่คำสั่งตัวช่วยในแต่ละครั้ง

4. เผลย

เมื่อผู้เล่นต้องการให้โปรแกรมเผลยภาพที่ถูกต้องสามารถทำได้โดยคลิกเลือกคำสั่งย่อยเผลย โปรแกรมจะทำการเผลยการต่อรูปจิกซอร์ทั้งหมดในตำแหน่งที่ถูกต้อง

5. การออกจากเกม

เมื่อผู้เล่นต้องการออกจากเกมสามารถทำได้โดยคลิกที่คำสั่งออกจากเกม โปรแกรมจะถามผู้เล่นว่าต้องการออกจากเกมหรือไม่ ถ้าผู้เล่นต้องการออกจากเกม ให้ตอบว่าใช่

ภาพ

ประกอบด้วยคำสั่งย่อยต่างๆ 2 คำสั่งย่อยดังต่อไปนี้

1. เปลี่ยนรูป

เมื่อผู้เล่นต้องการเปลี่ยนภาพของจิกซอร์สามารถทำได้โดยคลิกเลือกที่คำสั่งเปลี่ยนรูปเมื่อทำการคลิก โปรแกรมจะแสดงกรอบรายชื่อรูปภาพที่มีทั้งหมดขึ้นมาทางขวามือให้ทำการเลือกรูปภาพที่ต้องการต่อภาพ เมื่อคลิกที่กรอบรายชื่อรูปภาพนั้น 1 ครั้ง โปรแกรมจะแสดงรูปภาพตัวอย่างขึ้นมาทางซ้ายมือ และเมื่อดับเบิลคลิกที่กรอบรายชื่อรูปภาพนั้น โปรแกรมก็จะให้ผู้เล่นต่อภาพตามที่คุณเล่นได้เลือกภาพไว้ โดยผู้เล่นจะต่อภาพตามภาพตัวอย่างที่อยู่ทางซ้ายมือ และกรอบรายชื่อรูปภาพนั้นก็จะหายไป

2. การเพิ่มภาพ

เมื่อผู้เล่นต้องการที่จะเพิ่มภาพของมันก็สามารทำได้โดยคลิกที่คำสั่งเพิ่มภาพ โปรแกรมจะแสดงกรอบขึ้นมาให้ผู้เล่นเพิ่มรูปภาพเข้ามาในโปรแกรม โดยผู้เล่นสามารถเลือกแหล่งที่มาของภาพและทำการเพิ่มภาพเข้ามาเพื่อทำการเล่นเกมต่อไป

เสียงขณะเล่น

ประกอบด้วยคำสั่งย่อยต่างๆ 1 เมนูย่อย คือ ปิด/เปิด

ปิด

เมื่อผู้เล่นเลือกคำสั่งนี้ โปรแกรมจะทำการปิดเสียงทั้งหมด

เปิด

เมื่อผู้เล่นเลือกคำสั่งนี้ โปรแกรมจะทำการเปิดเสียง ซึ่งจะเกิด 3 เหตุการณ์ คือ

1. ถ้าไม่มีการลากส่วนจิกซอร์จะมีเสียงเพลงบรรเลง
2. ถ้ามีการลากชิ้นส่วนจิกซอร์ไปวางตำแหน่งที่ถูกต้องจะมีเสียงเตือนขึ้นมา ซึ่งจะเป็นเสียงที่บอกผู้เล่นว่าได้วางจิกซอร์ในตำแหน่งที่ถูกต้องแล้ว

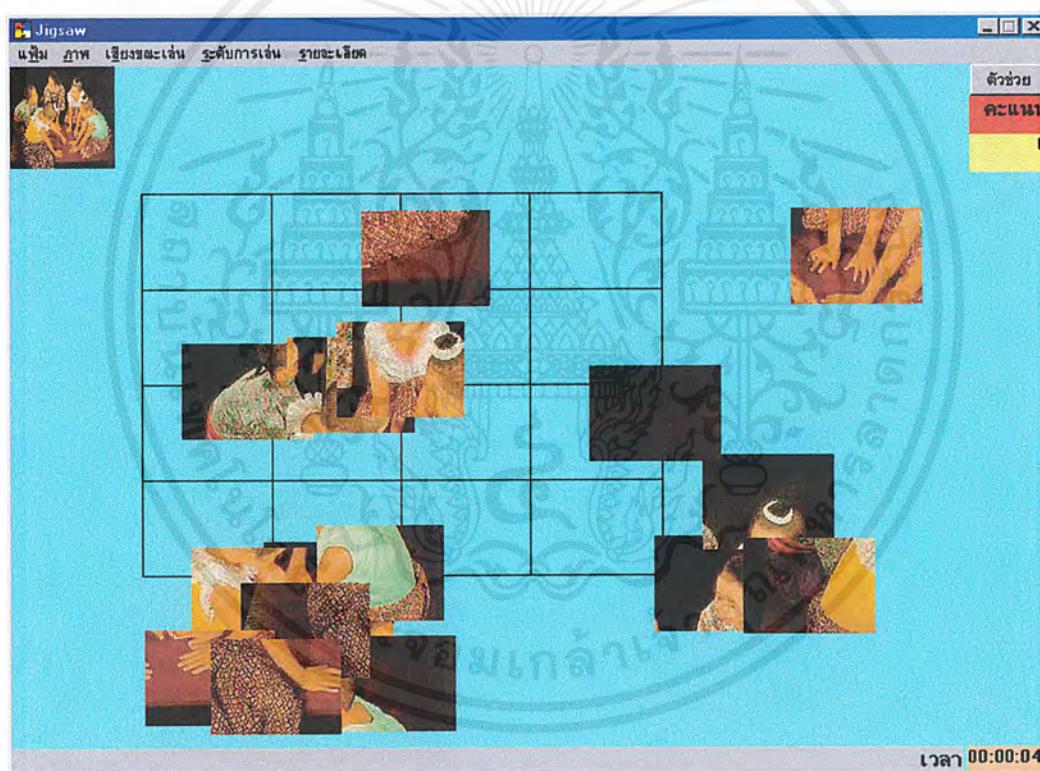
3. ถ้ามีการลากชิ้นส่วนจิกซอว์ไปวางยังตำแหน่งที่ไม่ถูกต้องจะมีเสียงเตือนขึ้นมา ซึ่งจะเป็นเสียงที่บอกผู้เล่นว่าวางจิกซอว์ในตำแหน่งที่ไม่ถูกต้อง และชิ้นส่วนจิกซอว์จะกลับไปอยู่ที่ตำแหน่งเดิม

ระดับการเล่น

ประกอบด้วยคำสั่งย่อยต่างๆ 3 คำสั่งย่อยดังต่อไปนี้

1. ง่าย

เมื่อผู้เล่นเลือกคำสั่งนี้ โปรแกรมจะถามผู้เล่นว่าต้องการจะเปลี่ยนเป็นระดับง่ายใช่หรือไม่ ถ้าผู้เล่นตอบว่าใช่ โปรแกรมจะทำการเปลี่ยนแปลงให้เป็นระดับง่าย โดยกระดานจะมีขนาดเป็น 4×4 และจะมีจำนวนชิ้นของจิกซอว์ทั้งหมด 16 ชิ้น แสดงได้ดังรูปที่ ก-2



รูปที่ ก-2 แสดงภาพของจิกซอว์ในการเล่นระดับง่าย

2. ปานกลาง

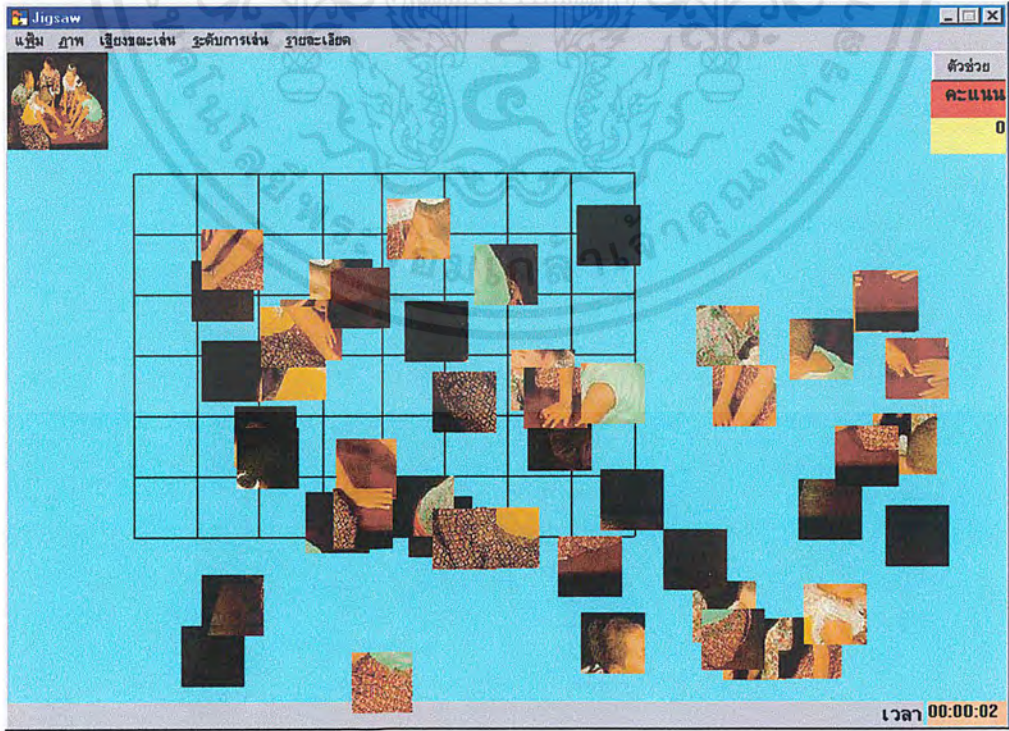
เมื่อผู้เล่นเลือกคำสั่งย่อยนี้ โปรแกรมจะถามผู้เล่นว่าต้องการจะเปลี่ยนเป็นระดับปานกลางใช่หรือไม่ ถ้าผู้เล่นตอบว่าใช่ โปรแกรมจะทำการเปลี่ยนแปลงให้เป็นระดับปานกลาง โดยกระดานจะมีขนาดเป็น 8×4 และจะมีจำนวนจิกซอว์ทั้งหมด 32 ชิ้น แสดงได้ดังรูปที่ ก-3



รูปที่ ก-3 แสดงภาพของจิกซอว์ในการเล่นระดับปานกลาง

3. ยาก

เมื่อผู้เล่นเลือกคำสั่งย่อยนี้ โปรแกรมจะทำการเปลี่ยนแปลงให้เป็นระดับยาก โดยกระดานจะมีขนาดเป็น 8x6 และจะมีจำนวนชิ้นของจิกซอว์ทั้งหมด 48 ชิ้น แสดงได้ดังรูปที่ ก-4



รูปที่ ก-4 แสดงภาพของจิกซอว์ในการเล่นระดับยาก

รายละเอียด

ประกอบด้วยคำสั่งย่อยต่างๆ 2 คำสั่งย่อยดังต่อไปนี้

1. เกม

เมื่อผู้เล่นต้องการรู้รายละเอียดต่างๆเกี่ยวกับเกมนี้สามารถคลิกได้ที่คำสั่งนี้โดยจะประกอบไปด้วยรายละเอียดของคำสั่งต่างๆบนเมนูบาร์รวมถึงคำสั่งย่อยๆต่างๆของแต่ละเมนูหลัก

2. คะแนน

จะบอกถึงรายละเอียดการให้คะแนนและการหักคะแนน โดยหลักเกณฑ์การให้คะแนนมีดังต่อไปนี้

- ถ้าวางจิกซอร์ถูกต้องตำแหน่งจะได้ครั้งละ 5 คะแนน
- ถ้าวางจิกซอร์ผิดตำแหน่งจะถูกหักคะแนนครั้งละ 2 คะแนน

เงื่อนไขการจบเกม

เงื่อนไขการจบเกมจะเกิดขึ้นเมื่อผู้เล่นสามารถต่อจิกซอร์ทุกชิ้นส่วนลงไปยังตำแหน่งที่ถูกต้อง โดยเมื่อจบเกมจะมีหน้าจอบอกว่าชนะในเกมนี้และจะมีหน้าจอให้ใส่ชื่อของผู้เล่น เพื่อทำการเก็บสถิติของผู้เล่น 10 อันดับแรกที่สามารถต่อจิกซอร์ได้สมบูรณ์และเร็วที่สุด

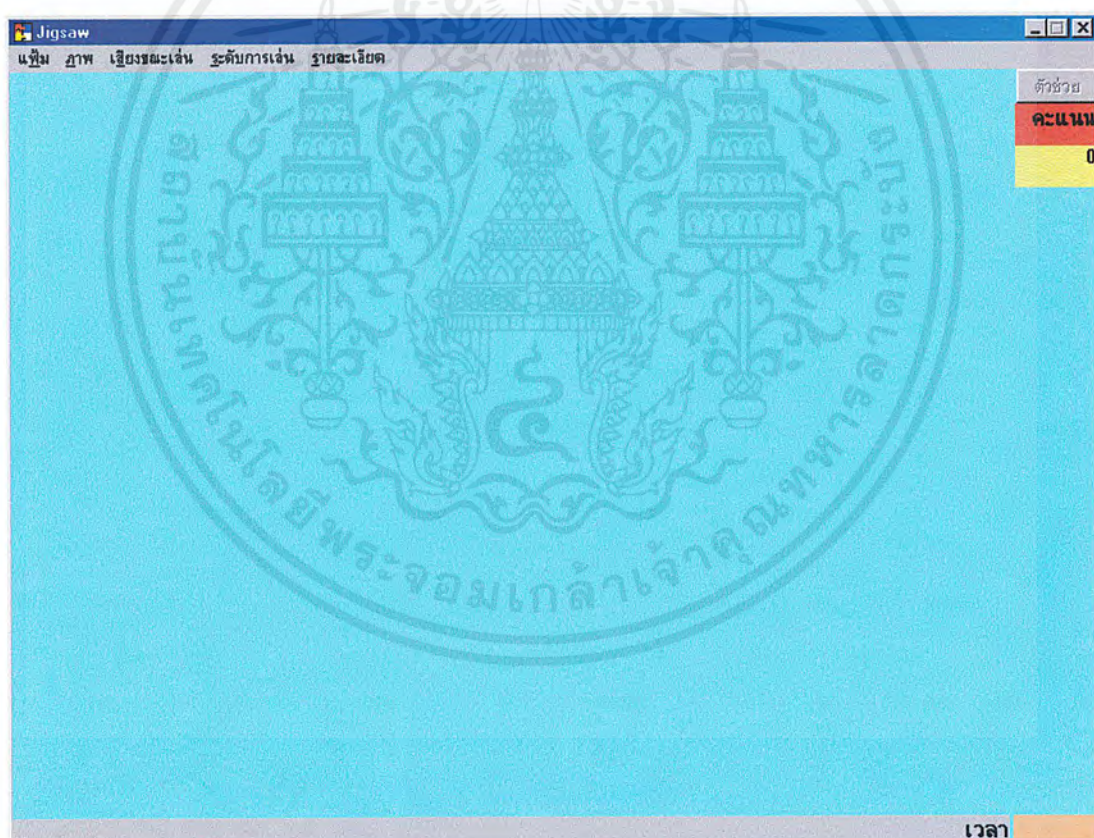
ภาคผนวก ข.

คุณสมบัติของโปรแกรม



คุณสมบัติของโปรแกรม

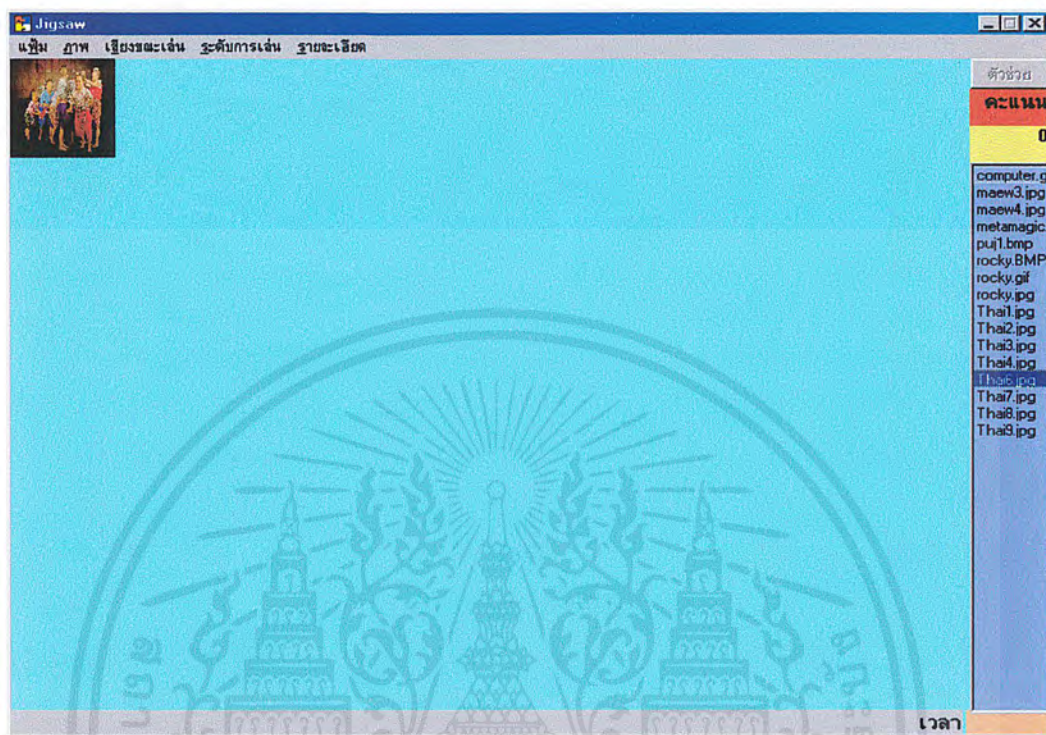
เมื่อผู้เล่นทำการติดตั้งโปรแกรมโดยการคลิกที่ ไฟล์ “ Setup.exe” เรียบร้อยเรียบร้อยแล้ว โปรแกรมเกมจิกซอร์ว่จะถูกติดตั้งโดยจะใช้น้ำหนักของฮาร์ดดิสก์ประมาณ 13 MByte ไฟล์ที่เกี่ยวข้องในการเล่นเกมนั้นทั้งหมดจะถูกติดตั้ง ผู้เล่นสามารถเริ่มเล่นเกมได้โดยการคลิกที่ไฟล์ “ Jigsaw.exe” เมื่อทำการเข้าสู่เกมจะมีเสียงดนตรีไทยบรรเลงประกอบการเล่นเกมเป็นไฟล์เพลงแบบ Midi ซึ่งโปรแกรมจะทำการสุ่มเลือกเล่นเพลง 1 เพลงจากเพลงที่มีอยู่ทั้งหมดและจะทำการบรรเลงเพลงแต่ละเพลงจนจบแล้วจะทำการสุ่มเลือกเพลงใหม่และบรรเลงไปเรื่อยๆจนจบการเล่นเกมนหรือถ้าผู้เล่นไม่ต้องการที่จะฟังเพลงระหว่างการเล่นเกมก็สามารถที่จะกระทำได้โดยการคลิกเลือกที่คำสั่งเสียงขณะเล่นแล้วเลือกที่คำสั่ง “ ปิด ” โปรแกรมจะทำการปิดเสียงเพลง โดยจะปรากฏหน้าจอการเข้าสู่การเล่นเกมนดังรูปที่ ข-1



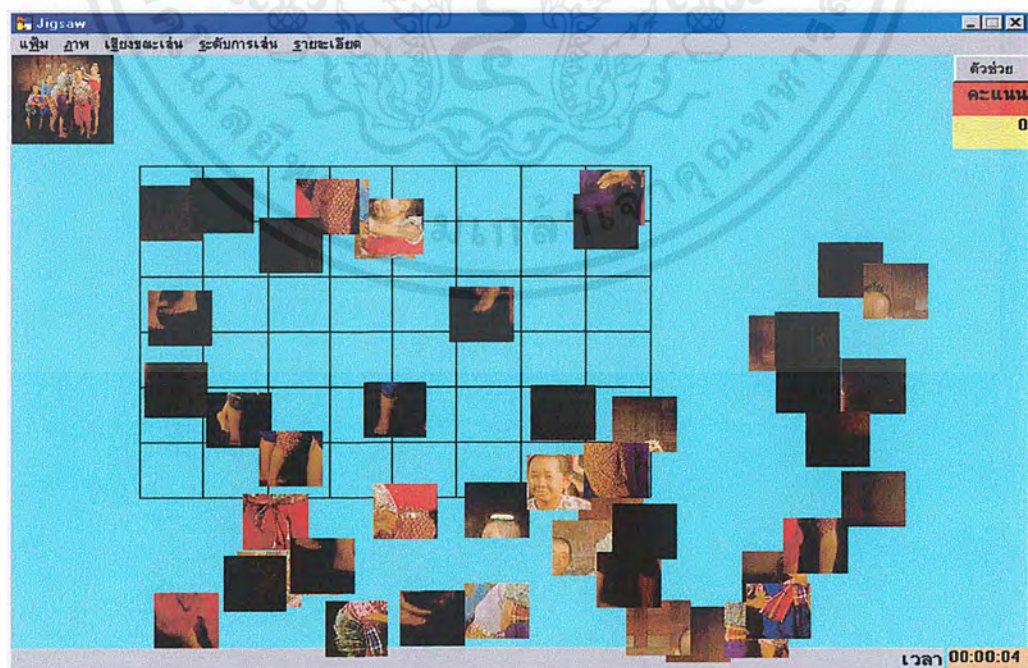
รูปที่ ข-1 แสดงหน้าจอเข้าสู่การเล่นเกม

เมื่อปรากฏหน้าจอเข้าสู่การเล่นเกมนผู้เล่นสามารถเริ่มเล่นเกมได้โดยคลิกที่คำสั่ง “ เริ่มเกมใหม่ ” เพื่อทำการเล่นเกมโดยจะปรากฏรายชื่อของรูปภาพของจิกซอร์ว่ทางด้านขวามือผู้เล่นสามารถเลือกภาพเพื่อใช้ในการเล่นเกมได้โดยการคลิกที่ชื่อภาพ 1 ครั้งจะแสดงภาพตัวอย่างของไฟล์ภาพนั้นๆอยู่

ทางมุมบนด้านซ้ายมีขนาดความกว้าง และความสูง 80 x 80 Pixel ถ้าผู้เล่นเลือกภาพได้แล้วสามารถ
เริ่มเล่นเกมได้โดยดับเบิลคลิกที่ชื่อภาพ โปรแกรมจะทำการตัดภาพที่ผู้เล่นเลือกออกเป็นชิ้นส่วน
ของจิกซอร์ทั้งหมด 48 ชิ้น โดยจะแสดงได้ดังรูปที่ ข-2 และรูปที่ ข-3

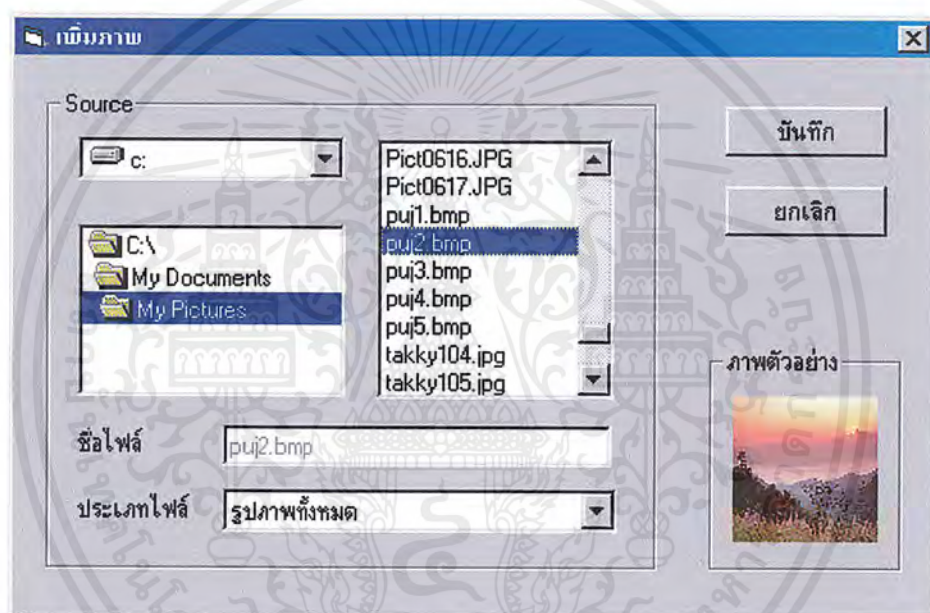


รูปที่ ข-2 แสดงการเลือกภาพเพื่อเล่นเกม



รูปที่ ข-3 แสดงภาพที่ถูกตัดเป็นชิ้นส่วนของจิกซอร์ 48 ชิ้น

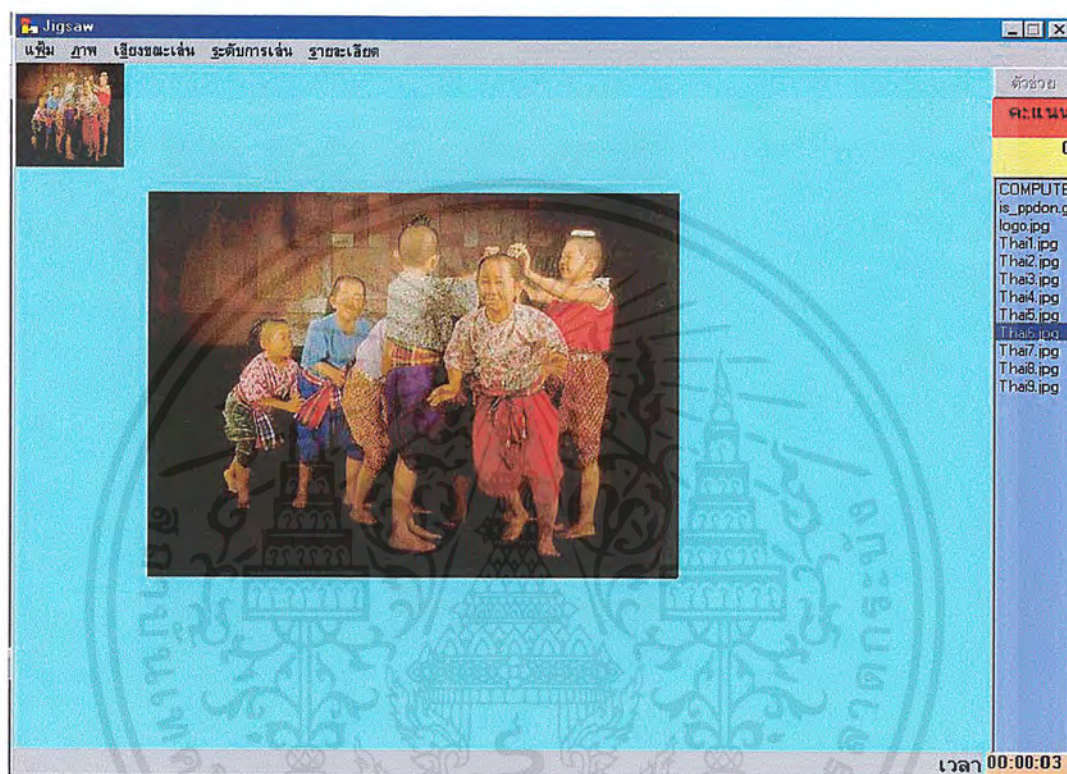
จำนวนจิกซอว์ทั้งหมด 48 ชิ้นเป็นระดับการเล่นในระดับยากถ้าผู้เล่นต้องการที่จะเปลี่ยนเป็นระดับอื่นสามารถทำได้โดยคลิกที่คำสั่ง “ ระดับการเล่น ” และเปลี่ยนการเล่นเป็นระดับอื่นอันได้แก่ระดับง่ายซึ่งมีจำนวนจิกซอว์ทั้งหมด 16 ชิ้น และระดับปานกลางมีจำนวนจิกซอว์ทั้งหมด 32 ชิ้น แต่ถ้าผู้เล่นไม่ชอบภาพที่มีมาให้และต้องการที่จะใช้ภาพที่ผู้เล่นมีอยู่ในการเล่นเกมนั้นก็สามารถที่จะเพิ่มภาพที่ต้องการได้โดยการคลิกเลือกที่คำสั่ง “ เพิ่มภาพ ” จะปรากฏหน้าจอจิกซอว์รูปที่ ข-4 ซึ่งผู้เล่นสามารถเลือกภาพจากแหล่งที่มาของภาพและบันทึกภาพนั้นๆเพื่อใช้ในการเล่นเกมได้ โดยโปรแกรมจะทำการปรับขนาดของภาพให้มีขนาดความกว้างและความสูง 400 X 300 Pixel ถ้าเป็นภาพที่มีอยู่แล้ว โปรแกรมจะแสดงข้อความว่ามีภาพนั้นๆอยู่แล้วและไม่สามารถที่จะบันทึกภาพซ้ำได้



รูปที่ ข-4 หน้าจอแสดงคำสั่งการเพิ่มภาพ

เมื่อทำการเลือกภาพและโปรแกรมทำการตัดชิ้นส่วนเสร็จเรียบร้อยแล้วผู้เล่นสามารถเริ่มเล่นเกมได้โดยการคลิกเลือกที่ชิ้นส่วนจิกซอว์ไปวางยังตาราง ถ้าผู้เล่นวางชิ้นส่วนของจิกซอว์ในตำแหน่งที่ถูกต้อง โปรแกรมจะแสดงเสียงว่าได้วางชิ้นส่วนของจิกซอว์ในตำแหน่งที่ถูกต้อง แต่ถ้าผู้เล่นวางจิกซอว์ในตำแหน่งที่ผิด โปรแกรมจะแสดงเสียงว่าวางจิกซอว์ในตำแหน่งที่ผิดและจะทำการเคลื่อนชิ้นส่วนจิกซอว์ที่วางผิดตำแหน่งนั้นไปวางยังตำแหน่งเดิม โดยผู้เล่นจะต้องทำการประกอบชิ้นส่วนของจิกซอว์ทั้งหมดให้เป็นรูปภาพที่สมบูรณ์ ถ้าผู้เล่นต้องการใช้ตัวช่วยสามารถทำได้โดยคลิกที่คำสั่ง “ ตัวช่วย ” โดยโปรแกรมจะทำการสุ่มชิ้นส่วนของจิกซอว์ที่ไม่อยู่ในตำแหน่งที่ถูกต้องมาวางยังตำแหน่งที่ถูกต้องถ้าผู้เล่นต้องการที่จะดูเฉลยคำตอบก็สามารถทำได้โดยคลิกที่คำสั่ง “ เฉลย ” โปรแกรมจะทำการวางชิ้นส่วนของจิกซอว์ทั้งหมดในตำแหน่งที่ถูกต้องและระหว่าง

การเล่นเกมนั้น โปรแกรมจะทำการจับเวลาที่ใช้ในการเล่นเริ่มตั้งแต่การเริ่มต่อจิกซอว์จนกระทั่ง
ต่อชิ้นส่วนของจิกซอว์เป็นภาพที่สมบูรณ์เมื่อทำการต่อภาพได้สมบูรณ์โปรแกรมจะแสดงข้อความ
ว่าผู้เล่นชนะในเกมนี้ๆ แล้วจะทำการเปรียบเทียบกับสถิติเก่าๆที่ผู้เล่น ได้เคยทำเอาไว้ โดยจะทำการ
การเปรียบเทียบและเก็บสถิติเฉพาะผู้เล่น 10 อันดับแรกที่ต่อจิกซอว์ได้สมบูรณ์และเร็วที่สุดเท่านั้น
โดยรูปที่ ข-5 จะแสดงภาพจิกซอว์ที่ประกอบสมบูรณ์แล้ว



รูปที่ ข-5 แสดงภาพจิกซอว์ที่ประกอบสมบูรณ์แล้ว

บรรณานุกรม

LaMothe , A . 1998. **Windows Programming Game For Dummies**. IDG Books Worldwide, Inc.

ธาริน สัทธรรมขาริ และ ธนัญชัย จ้างักักดี . 2541 . **Microsoft Visual Basic Version 5.0** .

กรุงเทพฯ : ซักเซสมิเดีย.

ธาริน สัทธรรมขาริ และ สุรสิทธิ์ คิวประสพศักดิ์ . 2542 . **คู่มือการเขียนโปรแกรม Microsoft Visual Basic Version 6.0 ฉบับเพื่อการประยุกต์ใช้งาน** . กรุงเทพฯ : ซักเซสมิเดีย.

