

เกมกระดานมหาสนุก

FUNNY BOARD



เฉลิมชัย อัครเดชวุฒิ  
ธีรวัฒน์ ตออำนาจ  
สุเกติองค์ ภูพัฒน์

เลขหมู่.....  
เลขทะเบียน..... 43031  
วัน, เดือน, ปี..... ๖.๖.๒๕๔๕

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# **FUNNY BOARD**



**CHALERMCHAI    ATSAWADACHWUTH**  
**THERAWAT    TOOUMNAUY**  
**SUKATIONG    PHUPHATANA**

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	เกมกระดานมหาสนุก FUNNY BOARD	
ชื่อนักศึกษา	นายเฉลิมชัย อัสวเดชวุฒิ	41056017
	นายธีรวัฒน์ ตออำนาจ	41056044
	นายสุเกติองค์ ภูพัฒน์	41056120
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	อาจารย์อิสระ บุรินทรมาตย์	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2544

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ผู้ช่วยศาสตราจารย์สุนทร สุชาติเวชภูมิ	
กรรมการ	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
กรรมการและอาจารย์ที่ปรึกษา	อาจารย์อิสระ บุรินทรมาตย์	

(ผู้ช่วยศาสตราจารย์ ไพโรบลย์ พันธรักษ์พงษ์)  
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	เกมกระดานมหาสนุก	
ชื่อนักศึกษา	นายเฉลิมชัย อัสวเดชวุฒิ	41056017
	นายธีรวัฒน์ ตออำนาจ	41056044
	นายสุเกศดิองค์ ภูพัฒน์	41056120
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2544	
อาจารย์ที่ปรึกษา	อาจารย์อิสระ บุรินทรมาตย์	

### บทคัดย่อ

ในปัจจุบัน คอมพิวเตอร์ได้เข้ามามีบทบาทต่อชีวิตของคนเราในหลายๆด้าน ความบันเทิงก็เป็นด้านหนึ่ง ซึ่งมีการนำคอมพิวเตอร์มาใช้กันอย่างแพร่หลาย เกมคอมพิวเตอร์เป็นความบันเทิงอย่างหนึ่งซึ่งได้รับความนิยมอย่างมาก โดยเฉพาะในกลุ่มเด็ก เนื่องจากมันเป็นสื่อผสมซึ่งสามารถโต้ตอบกับผู้เล่นได้ ทำให้ผู้เล่นได้มีส่วนร่วมมากกว่าสื่อบันเทิงแบบอื่นๆ แต่เกมคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันมักจะทำให้แต่ความสนุกสนานเท่านั้น ทั้งที่การสอดแทรกความรู้และการฝึกฝนทักษะต่างๆลงไปกับเกมจะทำให้ผู้เล่นได้รับประโยชน์จากการเล่นมากขึ้น

เกมกระดานมหาสนุกจึงถูกพัฒนาขึ้น โดยมีจุดมุ่งหมายที่จะพัฒนาให้เป็นเกมสำหรับเด็กที่ให้ความสนใจสนุกสนานและความรู้กับการฝึกฝนทักษะต่างๆด้วย เกมกระดานมหาสนุกมีพื้นฐานวิธีการเล่นเหมือนเกมกระดานทั่วไป คือ ทอยลูกเต๋า,เดินตามแต้มที่ได้แล้วทำตามคำสั่งในช่องที่เดินไปหยุด ผู้เล่นที่ท่าคะแนนได้ถึงเป้าหมายที่กำหนดก่อนก็จะเป็นผู้ชนะ โดยคะแนนส่วนใหญ่จะได้ออกจากการเล่นเกมน้อย ซึ่งในเกมย่อยต่างๆจะมีการสอดแทรกความรู้และการฝึกทักษะต่างๆให้ผู้เล่นไปพร้อมกับความสนุกสนาน ตัวเกมนำเสนอด้วยภาพการ์ตูน 2D Animation โดยใช้ DirectX ควบคุมการแสดงผล ใช้ Microsoft Visual Basic ในการพัฒนาโปรแกรม

<b>Special Project Title</b>	Funny Board		
<b>Students</b>	Mr. Chalermchai Atsawadachwuth	41056017	
	Mr. Therawat Toomnauy	41056044	
	Mr. Sukationg Phuphatana	41056120	
<b>Degree</b>	Bachelor's Degree of Science		
<b>Department</b>	Mathematics and Computer Science, Faculty of Science		
<b>Programme</b>	Computer Science		
<b>Academic Year</b>	2001		
<b>Special Project Advisor</b>	Lecturer Isara Burintramart		

## ABSTRACT

Currently , computer is playing an important role in human life. In many fields , the computer technology is applied to enhance the performance of products and activities. One of them is applied to an entertainment area. Computer game has gained more popularity among the children. With its interactive multimedia capability , it enables user to have more participation to the game than any other kind of entertainment. But most of them only provide just amusement , not including other knowledges.

Funny Board is developed and aimed to be a child computer game which provides both amusement and knowledge at the same time. Funny Board is based on other ordinary board games. It includes dice tossing , moving by the point from dice tossing and following the instructions given. The player who proceed to the target point first will win the game. Most of the points come from subgames which are blended with knowledges and skills as well as they are enjoyable. Game is presented in 2D animation using DirectX and it is developed by Microsoft Visual Basic.

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องเกมกระดานมหาสนุกสามารถเสร็จลุล่วงไปไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ อาจารย์อิสระ นรินทรมาตย์ อาจารย์ผู้รับผิดชอบปัญหาฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่าง ๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ และขอขอบคุณ ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล ที่ช่วยให้คำแนะนำ รวมทั้งข้อมูลต่าง ๆ เกี่ยวกับปัญหาพิเศษ

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อน ๆ ทุกคนที่ให้ความช่วยเหลือต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ  
มีนาคม 2545



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญตาราง .....	VII
สารบัญรูป .....	VIII
<b>บทที่ 1 บทนำ</b> .....	<b>1</b>
1.1 ความสำคัญและที่มาของปัญหา .....	1
1.2 วัตถุประสงค์ .....	1
1.3 ขอบเขตของปัญหา .....	2
1.4 ขั้นตอนในการดำเนินงาน .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	2
<b>บทที่ 2 ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง</b> .....	<b>3</b>
2.1 หลักการและทฤษฎีทั่วไปเกี่ยวกับ DirectX .....	3
2.2 DirectDraw .....	4
2.2.1 สถาปัตยกรรมและการทำงานทั่วไปของ DirectDraw .....	4
2.2.2 Object ที่สำคัญใน DirectDraw .....	5
2.2.3 Cooperative level และ Display mode .....	6
2.2.4 Surface .....	6
2.2.5 Blitting และ Flipping .....	6
2.2.6 Transparent Blitting และ Color Keys .....	8
2.2.7 Palettes .....	10
2.2.8 Clippers .....	10
2.3 DirectSound .....	11
2.3.1 สถาปัตยกรรมและการทำงานทั่วไปของ DirectSound .....	11
2.3.2 Object ที่สำคัญใน DirectSound .....	12
2.3.3 RIFF , WAV Files และ Data Structure .....	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.3.4 Play Streaming Sound Buffer .....	14
2.4 Direct Music .....	14
2.4.1 การทำงานทั่วไปของ DirectMusic .....	14
2.4.2 Object ที่สำคัญของ DirectMusic .....	15
2.4.3 MIDI (Musical Instrument Digital Interface) .....	15
<b>บทที่ 3 ขั้นตอนการดำเนินงานวิจัย .....</b>	<b>17</b>
3.1 ขั้นตอนการออกแบบ .....	18
3.1.1 การออกแบบกติกาและระบบการเล่น .....	18
3.1.2 การออกแบบภาพ .....	19
3.1.3 การออกแบบ Interface ที่ใช้ติดต่อและตอบโต้กับผู้เล่น .....	19
3.1.4 การออกแบบการจัดเก็บข้อมูล .....	21
3.1.5 การออกแบบวิธีการเขียน โปรแกรมและกำหนด Hardware , Software ที่จะใช้ .....	21
3.2 การสร้างภาพGraphic .....	21
3.2.1 การสร้างรูปภาพ 2 มิติ .....	21
3.2.1.1 Pen .....	21
3.2.1.2 Adobe Photoshop .....	22
3.2.2 การสร้างรูปภาพ 3 มิติ .....	22
3.2.2.1 3D Studio Max .....	23
3.2.2.2 การสร้างการเคลื่อนไหวให้กับตัวละคร .....	23
3.2.3 ภาพที่ใช้ภายในเกมกระดานมหาสนุก .....	24
3.2.3.1 ภาพตัวละคร .....	24
3.2.3.2 ภาพแผนที่และทางเดิน .....	25
3.2.3.3 ภาพสิ่งของประกอบจากแผนที่ .....	26
3.3 ขั้นตอนการเขียนโปรแกรม .....	26
3.3.1 ส่วนแสดงผลภาพ โดยใช้ DirectDraw .....	28
3.3.2 ส่วนแสดงผลเสียงดนตรีในรูปแบบ Midi โดยใช้ DirectMusic .....	30
3.3.3 ส่วนแสดงผลเสียงในรูปแบบ Wave File Format โดยใช้ DirectSound ....	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.3.4 ส่วนการรับ input จากผู้เล่น .....	32
3.3.5 ส่วนควบคุมการเล่นเกมหลัก .....	32
<b>บทที่ 4 ผลการทดลองและการวิเคราะห์ปัญหา .....</b>	<b>44</b>
4.1 ขั้นตอนการทดสอบการติดตั้ง โปรแกรมและซอฟต์แวร์ที่จำเป็น .....	44
4.2 ขั้นตอนการทดสอบการรันและประมวลผลภายใต้ระบบที่กำหนด .....	45
4.3 ขั้นตอนการทดสอบการทำงานในส่วนของเกมหลัก .....	47
4.4 ขั้นตอนการทดสอบการทำงานของเกมย่อย .....	48
4.4.1 เกม Magic Equation .....	48
4.4.2 เกม Magic Vocab .....	48
4.4.3 เกม Magic Memory .....	49
4.4.4 เกม Magic Image .....	50
4.5 ขั้นตอนการทดสอบการ Save และ Load เกม .....	50
4.6 ขั้นตอนการทดสอบการหาข้อผิดพลาดของโปรแกรม .....	50
4.7 ปัญหาและการวิเคราะห์ .....	51
<b>บทที่ 5 สรุปผลการศึกษาและข้อเสนอแนะ .....</b>	<b>52</b>
5.1 สรุปการศึกษา .....	52
5.1.1 การศึกษารวบรวมข้อมูล .....	52
5.1.2 การวิเคราะห์และการออกแบบเกม .....	52
5.1.3 การสร้างตัวละครและภาพต่างๆ .....	52
5.1.4 การพัฒนาโปรแกรม .....	53
5.2 ข้อเสนอแนะ .....	53
<b>ภาคผนวก .....</b>	<b>54</b>
ภาคผนวก ก. การติดตั้งและใช้งานโปรแกรม .....	55
ภาคผนวก ข. กติกาการเล่น .....	70
<b>บรรณานุกรม .....</b>	<b>72</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ของเกมกระดานมหาสนุก .....	20
4.1 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกมหลัก .....	47
4.2 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Equation .....	48
4.3 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Vocab .....	49
4.4 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Memory .....	49
4.5 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Image .....	50



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 แสดงสถาปัตยกรรมของ DirectDraw .....	4
2.2 แสดงการทำ Flipping ระหว่าง 2 buffer .....	7
2.3 แสดงการทำ Flipping ระหว่าง 3 buffer .....	8
2.4 แสดง Source Surface และ Destination Surface ที่มีการกำหนด Color Key ก่อนการ blitting .....	9
2.5 (ซ้าย) Surface ที่ได้หลังการ BLIT โดยใช้ Source Color Key อย่างเดียว .....	10
2.5 (ขวา) Surface ที่ได้หลังการ BLIT โดยใช้ทั้ง Source Color Key และ Destination Color Key .....	10
2.6 แสดงสถาปัตยกรรมของ DirectSound .....	11
2.7 RIFF File Structure .....	13
2.8 Wave File Structure .....	14
3.1 รูปต้นมะพร้าวที่ใช้ pen ในการวาดเป็นภาพ 2 มิติ .....	22
3.2 รูปภาพที่สร้างจาก โปรแกรม 3D Studio Max .....	23
3.3 แสดงจังหวะการเดินของตัวละครในแนวเฉียงขึ้นทางด้านซ้าย .....	24
3.4 แสดงจังหวะการเดินของตัวละครในแนวเฉียงขึ้นทางด้านขวา .....	24
3.5 แสดงจังหวะการเดินของตัวละครในแนวเฉียงลงทางด้านซ้าย .....	24
3.6 แสดงจังหวะการเดินของตัวละครในแนวเฉียงลงทางด้านขวา .....	25
3.7 แสดงจังหวะต่าง ๆ ในขณะยืนของตัวละคร .....	25
3.8 ภาพของเส้นทางที่มีการกำหนดมาก่อน .....	25
3.9 ภาพของแผนที่หลังจากมีการรวมกับภาพของเส้นทาง .....	26
3.10 ภาพสุนัขทั้ง 4 จังหวะที่นำมาประกอบฉาก .....	26
3.11 แสดงการเลือก References จากเมนู Project .....	27
3.12 แสดงการเลือก DirectX 7 จากหน้าจอ References .....	27
3.13 แผนผังการทำงานในส่วนของ title .....	33
3.14 แผนผังการทำงานในส่วนของ Character Select .....	34
3.15 แผนผังการทำงานในส่วนของ Mini Game .....	35
3.16 แผนผังการทำงานในส่วนของ Help .....	36
3.17 แผนผังการทำงานในส่วนของ Save Game.....	37
3.18 แผนผังการทำงานในส่วนของ Load Game .....	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 แผนผังการทำงานในส่วนของเกมหลัก .....	39
3.20 แผนผังการทำงานของเกมย่อย Magic Image .....	40
3.21 แผนผังการทำงานของเกมย่อย Magic Equation .....	41
3.22 แผนผังการทำงานของเกมย่อย Magic Memory .....	42
3.23 แผนผังการทำงานของเกมย่อย Magic Vocab .....	43
4.1 หน้าจอ Title ของโปรแกรม .....	45
4.2 แผนผังแสดงลำดับการแสดงผลหน้าจอของโปรแกรมเกมกระดานมหาสนุก .....	46
4.3 หน้าจอแผนที่ในการเล่นเกมนหลัก .....	47
4.4 หน้าจอการเล่นเกมน Magic Equation .....	48
4.5 หน้าจอการเล่นเกมน Magic Vocab .....	48
4.6 หน้าจอการเล่นเกมน Magic Memory .....	49
4.7 หน้าจอการเล่นเกมน Magic Image .....	50
ก-1 หน้าจอแสดงคำเตือนเกี่ยวกับลิขสิทธิ์ .....	55
ก-2 หน้าจอเลือก directory ที่ต้องการติดตั้งเกมกระดานมหาสนุก .....	56
ก-3 หน้าจอขณะกำลังติดตั้งเกมกระดานมหาสนุก .....	57
ก-4 หน้าจอเมื่อติดตั้งเกมกระดานมหาสนุกเสร็จเรียบร้อยแล้ว .....	58
ก-5 การเลือกเล่นเกมกระดานมหาสนุก .....	59
ก-6 หน้าจอยืนยันการถอดเกมกระดานมหาสนุกออกจากระบบ .....	60
ก-7 หน้าจอเมื่อทำการถอดเกมกระดานมหาสนุกเสร็จสิ้น .....	60
ก-8 หน้าจอเริ่มต้นเกม .....	61
ก-9 หน้าจอเลือกจำนวนผู้เล่น .....	62
ก-10 หน้าจอเลือกตัวละคร .....	62
ก-11 หน้าจอเลือกแผนที่ .....	63
ก-12 หน้าจอเริ่มต้นเกมหลัก .....	63
ก-13 หน้าจอการเล่นเกมนหลัก .....	64
ก-14 หน้าจอ Save Game .....	65
ก-15 หน้าจอ Load Game .....	65
ก-16 หน้าจอ Mini Game .....	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
ก-17 หน้าจอเกมช้อย Magic Vocab .....	66
ก-18 หน้าจอเกมช้อย Magic Equation .....	67
ก-19 หน้าจอเกมช้อย Magic Memory .....	67
ก-20 หน้าจอเกมช้อย Magic Image .....	68
ก-21 หน้าจอ Help หลัก .....	68
ก-22 ตัวอย่างหน้าจอ Help ช้อย .....	69



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของปัญหา

เกมเป็นสื่อที่เปิดโอกาสให้ผู้ใช้งานได้มีการตอบโต้กับตัวสื่อ ซึ่งจะช่วยให้ผู้เล่นเข้าสู่เนื้อหาและความบันเทิงที่ต้องการนำเสนอได้ง่าย ด้วยเหตุนี้เกมจึงเป็นสื่อที่เหมาะสมในการเสริมสร้างทักษะต่าง ๆ ให้แก่เด็กด้วยภาพที่สวยงาม ดนตรีที่น่ามาประกอบ และการเล่นที่สนุกสนานทำให้เด็กผลิตเฟลีนมีสมาธิสามารถรับสิ่งต่าง ๆ ที่เกมนำเสนอได้ดี เกมกระดานเป็นรูปแบบที่เด็กคุ้นเคยและสามารถทำความเข้าใจได้รวดเร็ว ด้วยวิธีการเล่นที่ง่ายโดยทยอยลูกเต๋าแล้วเดินไปตามแต้มที่ได้ตกที่ช่องใดก็ต้องทำตามคำสั่งในช่องนั้น ๆ การเสริมสร้างทักษะให้กับเด็กผ่านทางเกมกระดานนั้นสามารถทำได้ผ่านทางเกมย่อยซึ่งจะเป็นคำสั่งที่สั่งให้ปฏิบัติตามเมื่อเดินไปถึงช่องนั้น การนำเสนอจะเป็นภาพการ์ตูนสีสันสดใสแบบที่เด็ก ๆ ชอบ มีดนตรีประกอบเพิ่มความน่าสนใจและความสนุกสนาน ด้านการตอบโต้กับผู้เล่นออกแบบให้ตอบโต้ผ่านทางเมาส์ทั้งหมด เพื่อให้เด็กเล่นได้ง่าย

ในแง่ของการพัฒนา โปรแกรมเกมเป็นโปรแกรมที่ต้องใช้ความรู้ความสามารถหลายด้านในการพัฒนา ทั้งในด้านการออกแบบใช้ความคิดสร้างสรรค์ในการกำหนดกฎกติการูปแบบการเล่นให้สนุกสนานและได้สาระไปพร้อมกัน ด้านการพัฒนาโปรแกรมมัลติมีเดียซึ่งจะต้องมีความรู้ในการพัฒนาโปรแกรมที่เน้นด้านการแสดงผลทั้งภาพกราฟฟิโคนิเมชันและเสียง ด้านอัลกอริทึมที่ต้องใช้ในโปรแกรมก็มีความหลากหลายตามรูปแบบของเกม ดังนั้นโปรแกรมเกมจึงควรได้รับการศึกษาและพัฒนาต่อไป

### 1.2 วัตถุประสงค์

- 1) เพื่อฝึกฝนทักษะและความสามารถในการพัฒนาโปรแกรมมัลติมีเดีย
- 2) เพื่อศึกษาและพัฒนาโปรแกรมที่ใช้การแสดงผลภาพผ่าน DirectX
- 3) เพื่อให้ผู้เล่นได้รับการฝึกฝนทักษะต่าง ๆ ไปพร้อมกับความสนุกสนานจากการเล่นเกม

### 1.3 ขอบเขตของปัญหา

พัฒนาโปรแกรมเกมมัลติมีเดียซึ่งเป็นเกมกระดานที่มีจำนวนผู้เล่นได้ตั้งแต่ 2 ถึง 4 คน มีฉากให้เลือกอย่างน้อย 2 ฉากและในแต่ละฉากจะมีเกมย่อยให้เล่นเสริมทักษะ สามารถบันทึกเกมระหว่างที่เล่นอยู่ได้เพื่อนำกลับมาเล่นต่อในภายหลัง การแสดงผลใช้ภาพการ์ตูน 2 มิติซึ่งได้รับการเรนเดอร์มาจากวัตถุ 3 มิติและมีดนตรีประกอบ

### 1.4 ขั้นตอนในการดำเนินงาน

- 1) ศึกษาการใช้งาน Visual Basic 6 ทั้งทางรูปแบบภาษาและอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้
- 2) ศึกษาการใช้งาน DirectX ในส่วนของ DirectDraw , DirectMusic และ DirectSound
- 3) ออกแบบกติกา วิธีการเล่นและส่วนประกอบต่าง ๆ ของเกม
- 4) ออกแบบหน้าจอและการตอบโต้กับผู้ใช้
- 5) สร้างรูปภาพที่จะใช้ในเกม
- 6) ออกแบบการทำงานต่าง ๆ ภายในโปรแกรม
- 7) ออกแบบโครงสร้างไฟล์ต่าง ๆ ที่ต้องใช้
- 8) พัฒนาโปรแกรมในส่วนต่าง ๆ และนำมาประกอบกัน
- 9) ทดสอบและแก้ไขข้อผิดพลาดต่าง ๆ ในโปรแกรม
- 10) สรุปผลและเขียนรายงาน

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้ในการใช้งานคอมพิวเตอร์ภาษา Visual Basic ที่ทำงานร่วมกับ DirectX
- 2) ได้รับความรู้เกี่ยวกับการแสดงผลภาพมัลติมีเดียของ DirectX
- 3) ผู้เล่นได้รับประโยชน์จากการฝึกทักษะต่าง ๆ ที่สอดแทรกไว้ในเกม และความสนุกสนานจากการเล่นเกม

## บทที่ 2

# ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

### 2.1 หลักการและทฤษฎีทั่วไปเกี่ยวกับ DirectX

DirectX เป็นเครื่องมือที่ใช้สำหรับสร้างงานมัลติมีเดียบนเครื่องคอมพิวเตอร์พีซีที่มีระบบปฏิบัติการ window ซึ่งถูกผลิตขึ้นโดยบริษัทไมโครซอฟต์ DirectX จะช่วยให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมมัลติมีเดียต่าง ๆ ได้โดยไม่ต้องคำนึงถึงเรื่องการติดต่อกับฮาร์ดแวร์ต่าง ๆ ไม่ว่าจะเป็นการ์ดจอ การ์ดเสียง หรือแม้แต่อุปกรณ์อินพุต เช่น คีย์บอร์ด เมาส์ gamepad ฯลฯ ซึ่งในอดีตการพัฒนาซอฟต์แวร์มัลติมีเดียจะต้องมีความรู้และคำนึงถึงฮาร์ดแวร์ต่าง ๆ เหล่านี้อย่างมาก DirectX จะทำหน้าที่เป็นตัวกลางช่วยในการติดต่อกับฮาร์ดแวร์ต่าง ๆ จึงทำให้ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับฮาร์ดแวร์เลย นอกจากนี้การติดต่อกับฮาร์ดแวร์ของ DirectX นั้นจะกระทำโดยตรง ไม่ต้องผ่านระบบปฏิบัติการ ทำให้ซอฟต์แวร์มีประสิทธิภาพการทำงานที่เร็วขึ้น ซึ่งเหมาะสมกับซอฟต์แวร์ด้านมัลติมีเดียที่จะต้องทำการติดต่อกับฮาร์ดแวร์ โดยเฉพาะฮาร์ดแวร์ในการแสดงผลที่มีรับ-ส่งข้อมูลเป็นจำนวนมาก DirectX จึงเป็นเครื่องมือที่เหมาะสมอย่างยิ่งในการสร้างซอฟต์แวร์มัลติมีเดียบน window ใน DirectX จะประกอบด้วย component ในการทำงานต่าง ๆ ดังนี้

- 1) DirectDraw เป็นส่วนที่ทำหน้าที่ในการแสดงผลภาพ 2 มิติต่าง ๆ จะติดต่อกับโดยตรงกับ video hardware ภาพที่จะนำมาแสดงนั้นอาจถูกจัดเก็บอยู่บน video memory หรือ system memory ก็ได้ ใน DirectDraw จะมี function ใช้งานมากมาย ไม่ว่าจะเป็นการทำ flipping , blitting และการใช้ clipper , video buffer , palettes ฯลฯ ซึ่งจะอำนวยความสะดวกให้กับการจัดการแสดงผลภาพ 2 มิติ
- 2) DirectSound ทำหน้าที่ติดต่อกับ sound hardware ต่าง ๆ ในการแสดงเสียง และจัดการกับเสียงในแบบต่าง ๆ เช่น การผสมเสียงจาก 2 buffer เป็นต้น และสามารถให้แสดงเสียงแบบ 3 มิติได้อีกด้วย
- 3) DirectMusic ทำหน้าที่จัดการกับดนตรีต่าง ๆ เช่น การเริ่มเล่นและปิดเสียงดนตรี หรือควบคุมความเร็วในการเล่นเสียงดนตรีต่าง ๆ เป็นต้น
- 4) DirectInput ทำหน้าที่จัดการกับอุปกรณ์อินพุตต่าง ๆ เช่น คีย์บอร์ด , เมาส์ , joy stick รวมถึงอุปกรณ์ force feed-back ต่าง ๆ ด้วย
- 5) Direct3D ทำหน้าที่จัดการการแสดงผลภาพ 3 มิติต่าง ๆ โดยส่วนนี้จำเป็นต้องใช้ฮาร์ดแวร์ที่สนับสนุน Direct3D ด้วย เพราะฮาร์ดแวร์แสดงผล 3 มิตินั้นยังไม่มีมาตรฐานวิธีการสร้างและการทำงานต่าง ๆ ภายในขึ้นอยู่กับผู้ผลิต ซึ่งจำเป็นต้องใช้ driver เฉพาะของผู้ผลิตนั้นๆ

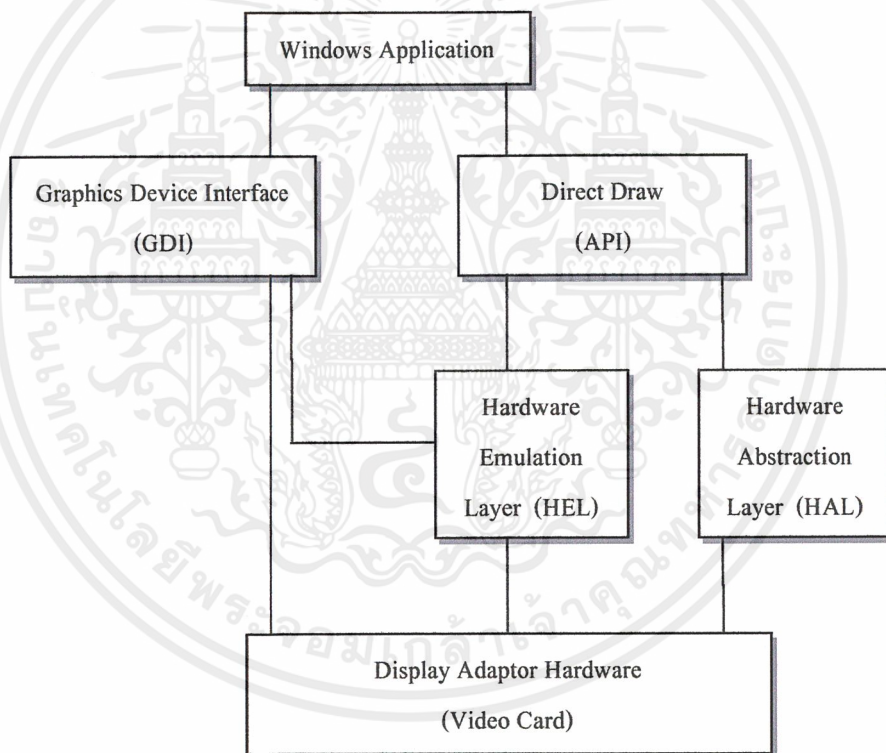
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) DirectPlay ทำหน้าที่จัดการการติดต่อกันผ่านเครือข่าย โดยรองรับ protocol TCP/IP , IPX และรวมถึงการติดต่อผ่าน serial port และ modem ด้วย
- 7) DirectSetup ทำหน้าที่จัดการสร้างตัว setup หรือ install program โดยกำหนดการ Install DX runtime ไปพร้อมกัน ในขั้นตอนการลงโปรแกรมได้ และยังสร้าง function Autoplay สำหรับ CD ได้ด้วย

## 2.2 DirectDraw

### 2.2.1 สถาปัตยกรรมและการทำงานทั่วไปของ DirectDraw

โครงสร้างการทำงานของ DirectDraw จะเป็นดังรูป



รูปที่ 2.1 แสดงสถาปัตยกรรมของ DirectDraw

จากรูปที่ 2.1 จะพบว่า application ต่าง ๆ บน window จะติดต่อกับ Display Adaptor Hardware (Video Card) ได้ 2 ทาง คือ (1) ผ่าน Graphics Device Interface (GDI) หรือ (2) ผ่าน Direct Draw (API)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ Graphics Device Interface (GDI) จะต้องทำการเรียกใช้ผ่าน window ทำให้การติดต่อกับ Display Adaptor Hardware (Video Card) ทำได้ช้ากว่าการใช้ Direct Draw (API) ซึ่งจะติดต่อโดยตรงโดยไม่ต้องผ่าน window

เมื่อ application เรียกใช้ DirectDraw มันจะตัดสินใจว่าจะทำงานตามที่ application สั่งอย่างไร ถ้าคำสั่งนั้น graphics hardware สามารถทำงานได้ คำสั่งนั้นจะถูกส่งต่อไปให้ Hardware Abstraction Layer (HAL) เพื่อสั่งงานให้ graphics hardware ทำตามคำสั่งนั้น Hardware Abstraction Layer (HAL) ก็คือ driver เฉพาะของอุปกรณ์ที่ผู้ผลิตสร้างขึ้นนั่นเอง โดย driver เหล่านี้จะถูกเขียนให้สนับสนุนการทำงานของ DirectX เพราะฉะนั้นเมื่อมีการติดตั้ง DirectX version ใหม่จึงต้อง update driver ของอุปกรณ์ให้เข้ากันด้วย

แต่ถ้าคำสั่งนั้น graphics hardware ไม่สามารถทำงานได้จะถูกส่งต่อไปให้ Hardware Emulation Layer (HEL) ซึ่งเป็นส่วนที่ DirectDraw จะทำการจำลองการทำงานตามคำสั่งนั้น ขึ้นบนซอฟต์แวร์ก่อนแล้วนำผลที่ได้ส่งให้ graphics hardware ทำการแสดงผล ทั้ง HAL และ HEL ต่างก็ถูกซ่อนจากผู้พัฒนา กล่าวคือ ผู้พัฒนาจะไม่ทราบรายละเอียดภายในของมันเลย แต่จะเรียกใช้ผ่านทาง DirectDraw Interface เท่านั้น

### 2.2.2 Object ที่สำคัญใน DirectDraw

ใน DirectDraw ประกอบด้วย object ที่ทำงานต่าง ๆ กันไป โดยแต่ละ object ก็จะมี method เป็นของตัวเอง ซึ่งใน DirectDraw มี object ที่สำคัญดังนี้

- 1) DirectDraw Object เป็น object หลักของ DirectDraw มี method ที่ใช้ในการตั้งค่าต่าง ๆ เพื่อเตรียมตัวแสดงผล เช่น การตั้งค่าให้แสดงผลแบบ window หรือ full screen และเลือกโหมดการแสดงผล เป็นต้น นอกจากนี้ยังมี method ที่ใช้สร้าง object อื่น ๆ ต่อไป
- 2) DirectDrawSurface Object เป็น object ที่จัดการกับ surface ซึ่ง surface ก็คือ พื้นที่ใน video หรือ system memory ที่ใช้สำหรับเก็บข้อมูลภาพ เพื่อเตรียมนำภาพไปแสดงผล (จะอธิบายในหัวข้อต่อไป)
- 3) DirectDrawPalette Object เป็น object ที่ทำหน้าที่จัดการตารางสี (palette) สำหรับ display mode ที่ใช้ระบบตารางสีมีจำนวนสีได้ตั้งแต่ 16-256 สี ซึ่งในตารางสีใช้สีแบบ RGB โดย palette จะถูกผูกติดอยู่กับ surface
- 4) DirectDrawClipper Object เป็น object ที่ทำงานเกี่ยวกับการคลิปภาพ เมื่อมีการนำ surface มาซ้อนทับกัน

- 5) DirectDrawVideoPort Object เป็น object ที่ใช้แสดง port และรายละเอียดของ video hardware ที่ติดตั้งอยู่ และสามารถเข้าถึง frame buffer โดยไม่ต้องผ่าน CPU หรือ PCI bus

### 2.2.3 Cooperative level และ Display mode

ก่อนที่เราจะใช้คำสั่งต่าง ๆ ใน DirectX แสดงผลภาพได้นั้น เราจะต้อง set Cooperative level และ Display mode ก่อน การ set Cooperative level ก็คือ การบอก DirectDraw ว่าเราจะทำงานกับ DirectDraw ในลักษณะใด เช่น window mode หรือ fullscreen mode สามารถ interrupt ด้วยการกดปุ่ม Ctrl-Alt-Del ได้หรือไม่ และสามารถที่จะเปลี่ยนขนาดของ window (ใน window mode) ได้หรือไม่ เป็นต้น หลังจาก set Cooperative level แล้วจะต้อง set Display mode เป็นลำดับถัดมา เพื่อระบุ screen resolution , color depths และ refresh rate ที่ต้องการ โดยจะต้องระบุความกว้าง ความยาวของ resolution เป็นหน่วย pixel และจะใช้สีแบบใด 8 , 6 , 24 หรือ 32 บิต หรือจะใช้ตารางสี (palette) ทั้งนี้ video hardware ที่ใช้จะต้องสนับสนุนกับ mode ที่เลือกด้วย มิฉะนั้นจะเกิด error Display mode มาตรฐานที่ใช้กันคือ 640x480x8 bpp

### 2.2.4 Surface

surface เป็นพื้นที่บน memory ที่ใช้สำหรับเก็บข้อมูลภาพ bitmap เพื่อเตรียมสำหรับการนำขึ้นมาแสดงผลบนจอภาพต่อไป โดยพื้นที่นี้อาจอยู่บน video memory หรือ system memory ก็ได้ขึ้นอยู่กับค่า Flags ที่ส่งให้ตอนสร้าง surface แบ่งออกเป็น 2 ชนิด คือ Primary surface และ Off-screen surface

Primary surface นั้นเป็น surface หลักที่จะแสดงผลออกสู่จอภาพ โดยจะต้องมีขนาดเท่ากับ Display mode ที่เลือกไว้ ภาพใด ๆ ก็ตามที่ต้องการแสดงผลออกจอภาพต้องนำมาวาดไว้บน Primary surface เสมอ แต่การนำภาพมาวาดลงบน Primary surface โดยตรงจะทำให้ภาพที่ได้กระพริบไม่ราบรื่น จึงต้องใช้การทำ flipping มาช่วย (ซึ่งจะกล่าวถึงในหัวข้อต่อไป)

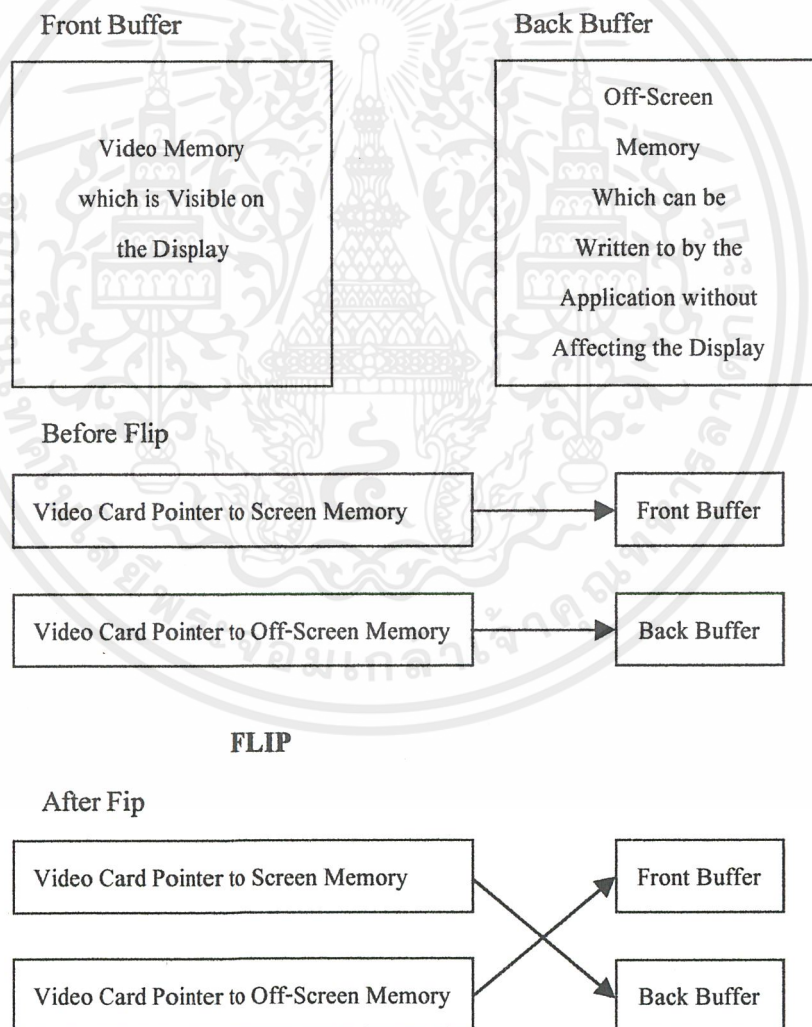
Off-screen surface คือ surface ที่ใช้สำหรับเก็บข้อมูลภาพต่าง ๆ เพื่อ Blitting ลงใน buffer หรือ surface อื่นเพื่อเตรียมแสดงผลต่อไป

### 2.2.5 Blitting และ Flipping

Blitting คือ การนำภาพต่าง ๆ จาก source ไปวาดลงบนบริเวณ destination ที่กำหนด โดยอาจมีการย่อ-ขยายภาพให้พอดีกับขนาด destination ที่กำหนด และสามารถทำการ transparent สีบางสีได้ด้วย การทำ Blitting สามารถทำได้หลายครั้ง โดยภาพที่ทำ Blitting จะเรียงซ้อนทับกันเป็นชั้น ๆ

Flipping เป็นการสลับที่กันของ buffer โดยก่อนการทำ flipping จะต้องมีการสร้าง buffer ขึ้นมาอย่างน้อย 2 buffer คือ Front buffer กับ Back buffer และอาจมี Third buffer หรือไม่มีก็ได้ Front buffer คือ buffer ที่จะแสดงภาพออกที่หน้าจอ หรือก็คือ Primary surface นั่นเอง Back buffer กับ Third buffer เป็นที่พักของภาพที่รอจะแสดงผลบนหน้าจอ buffer ทั้ง 3 จะมีขนาดเท่ากับ Display mode ที่เลือกไว้

การวาดหรือ Blitting ลงบน Front buffer โดยตรงจะทำให้ภาพกระพริบ ควรเตรียมภาพที่จะแสดงผลไว้ให้พร้อมบน Back buffer แล้วทำการ Flipping DirectDraw จะสลับภาพระหว่าง Front buffer กับ Back buffer เพื่อนำภาพบน Back buffer มาแสดงบนจอภาพ โดยการสลับในที่นี้ไม่ได้หมายถึง การสลับที่ข้อมูลภาพแต่เป็นการสลับเฉพาะ pointer ที่ชี้พื้นที่ที่เก็บข้อมูลอยู่เท่านั้น ซึ่งทำได้อย่างรวดเร็วและไม่ทำให้เกิดการกระพริบหรือการกระตุกของภาพ

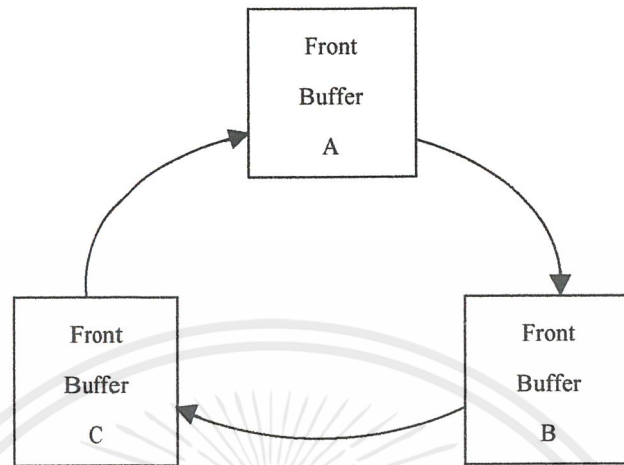


รูปที่ 2.2 แสดงการทำ Flipping ระหว่าง 2 buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

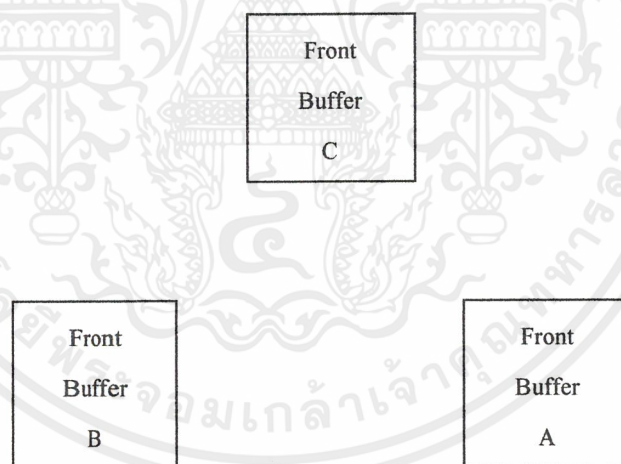
ถ้ามี Third buffer การสลับภาพจะเป็นวงจรดังรูป

Before Flip



FLIP

After Flip



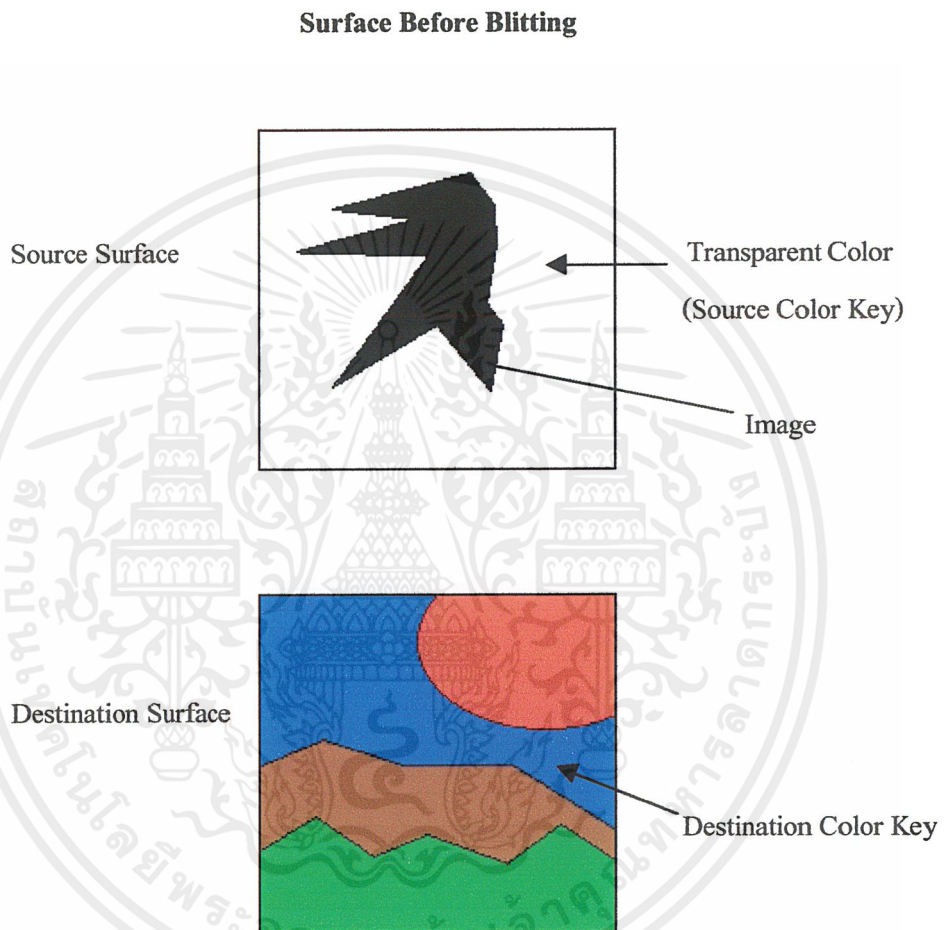
รูปที่ 2.3 แสดงการทำ Flipping ระหว่าง 3 buffer

### 2.2.6 Transparent Blitting และ Color Keys

DirectDraw สามารถทำ Transparent blitting ได้โดยอาศัย Color keys Surface ต่าง ๆ สามารถ set Color keys เป็นสีใดสีหนึ่งหรือช่วงของค่าสีก็ได้ Color keys ที่ถูก set ไว้จะแสดงผลตอน blitting โดยถ้า source มีการทำ set Color keys ไว้ ค่าสีตาม Color keys จะไม่ถูก copy ไปบน destination ทำให้บริเวณสีนั้น โปร่งแสงไป ถ้า destination มีการทำ Color keys ไว้ ค่าสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

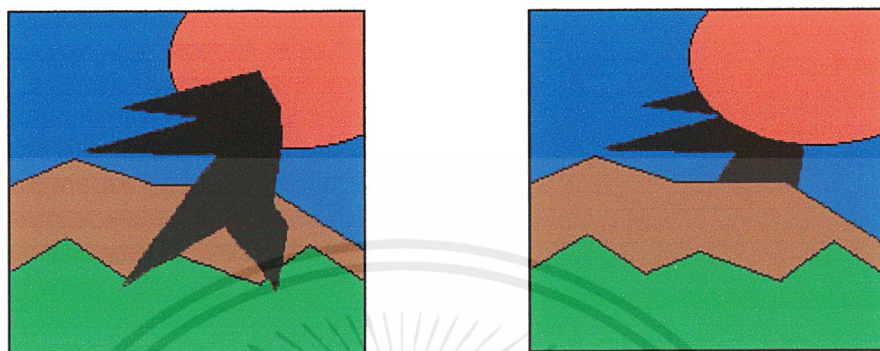
ตาม Color keys เท่านั้นที่จะถูกทับโดยภาพที่ copy มาจาก source การทำ Color keys กับ source มักจะใช้กับตัว sprites ที่มีการเคลื่อนที่ไปบนฉากหลัง ส่วนการทำ Color keys กับ destination จะใช้กับกรณีที่ต้องการปิดบังบางส่วนของตัว sprites



**รูปที่ 2.4** แสดง Source Surface และ Destination Surface ที่มีการกำหนด Color Key ก่อนการ blitting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Resulting Surface after BLIT



รูปที่ 2.5 (ซ้าย) Surface ที่ได้หลังการ BLIT โดยใช้ Source Color Key อย่างเดียว  
(ขวา) Surface ที่ได้หลังการ BLIT โดยใช้ทั้ง Source Color Key และ Destination Color Key

#### 2.2.7 Palettes

Palettes คือ ตารางสีที่ใช้ในระบบ index color ซึ่งสามารถมีจำนวนสีได้ตั้งแต่ 2 , 4 , 16 และ 256 สี ระบบ index color นั้นจะมี array สำหรับเก็บค่าสี RGB โดยแต่ละค่าอาจมีความละเอียดที่ 16 หรือ 24 บิต แล้วในข้อมูลภาพจะเก็บเป็นค่า index จากตาราง Palettes แทนการเก็บค่าสีจริง ซึ่งจะใช้พื้นที่ 8 บิตต่อ 1 จุด DirectDraw สามารถสร้าง Palettes เพื่อเชื่อมกับแต่ละ surface แยกออกจากกันได้

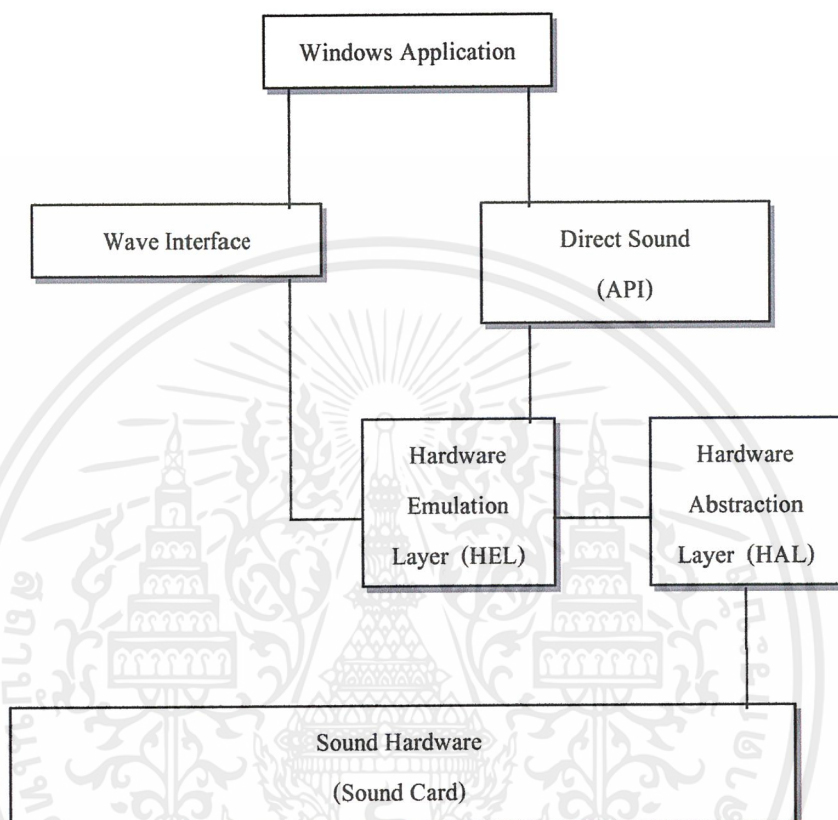
#### 2.2.8 Clippers

Clippers เป็น object ที่ช่วยในการตัดภาพส่วนที่เกินออกจากพื้นที่ที่กำหนดไว้ รวมถึงตัดภาพที่นำมา Blitting บนพื้นที่ที่มี Clipper ด้วย พื้นที่ที่จะทำ Clipper นี้จะเป็นรูปสี่เหลี่ยมใน window mode Clipper จะช่วยในการจำกัดขอบเขตของการวาดภาพไม่ให้เกินออกไปจาก window ที่กำหนด ใน full-screen mode Clipper จะช่วยในการแบ่งหน้าจอออกเป็น ส่วน ๆ ตามที่ต้องการ

## 2.3 DirectSound

### 2.3.1 สถาปัตยกรรมและการทำงานทั่วไปของ DirectSound

โครงสร้างการทำงานของ DirectSound จะเป็นดังรูป



รูปที่ 2.6 แสดงสถาปัตยกรรมของ DirectSound

โครงสร้างการทำงานของ DirectSound มีลักษณะคล้ายกับโครงสร้างการทำงานของ DirectDraw โดยมี Hardware Emulation Layer (HEL) และ Hardware Abstraction Layer (HAL) เหมือนกับ DirectDraw HAL คือ Software driver ของ Sound Hardware ที่บริษัทผู้ผลิตสร้างขึ้นถ้า HAL ไม่สามารถทำงานตามคำสั่งที่ DirectSound ส่งมาได้ HEL ก็ถูกเรียกใช้โดย HEL จะใช้ Window multimedia wavefrom-audio function จำลองเสียงตามคำสั่งจาก DirectSound แล้วส่งข้อมูลเสียงที่ได้ไปให้ Sound Hardware แสดงผล

DirectSound จะใช้ Primary sound buffer ในการ mix เสียงและใช้ Secondary sound buffer สำหรับเตรียมเสียงเพื่อนำไป mix ทันทีที่เสียงถูก mix DirectSound จะส่งข้อมูลใน Primary sound buffer ไปที่ Output Device เพื่อแสดงผลเสียงที่ได้ Primary sound buffer จะถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทันทีเมื่อ Object ของ DirectSound ถูกสร้างขึ้น ส่วน Secondary Sound Buffer ผู้พัฒนาจะต้องสร้างขึ้นเองในโปรแกรมแล้วก็นำข้อมูล Audio ที่ต้องการเล่นไปเก็บไว้เพื่อแสดงผลต่อไป Secondary sound buffer สามารถเป็นได้ทั้งแบบ Static และ Streaming Static sound buffer จะบรรจุข้อมูลเสียงปกติเพียง 1 ข้อมูล ในขณะที่ Streaming sound buffer จะบรรจุส่วนของข้อมูลเสียงส่วนหนึ่ง เพื่อแสดงเสียงต่อเนื่องโดย มันจะถูก Update ข้อมูลตามช่วงเวลาที่กำหนด เพื่อให้การแสดงผลเสียงต่อเนื่องราบรื่น

Direct Sound จะพยายามใส่ sound buffer ลงใน memory ของ Sound Hardware ให้มากที่สุดเท่าที่จะทำได้ เนื่องจาก memory ใน Sound Hardware จะทำงานได้ดีกว่าการใช้ memory ของ system แต่ถ้า memory บน Sound Hardware เต็ม sound buffer ใหม่จะต้องไปอยู่ใน memory ของ system ดังนั้นควรจะมีการกำหนดความสำคัญให้กับ Secondary sound buffer โดยให้ buffer ที่มีความสำคัญใช้งานบ่อยๆอยู่ใน memory ของ Video Hardware ส่วน buffer ที่มีความสำคัญน้อยกว่าก็จะไปอยู่ใน memory ของ system

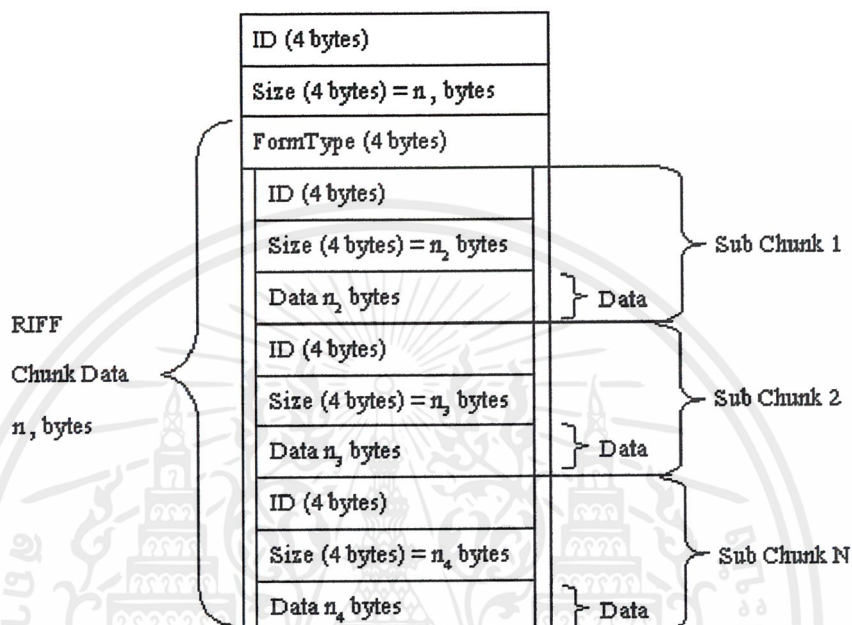
### 2.3.2 Object ที่สำคัญใน DirectSound

ใน DirectSound มีการแบ่งส่วนต่าง ๆ ออกเป็น Object หลาย ๆ Object เพื่อทำหน้าที่ต่าง ๆ กันไป Object ที่สำคัญมีดังนี้

- 1) DirectSound เป็น object หลักที่จำเป็นต้องสร้างก่อนเป็นอันดับแรก ในการใช้งาน DirectSound มี method สำหรับ access อุปกรณ์เสียงเพื่อเรียกใช้งาน function ของมัน , set cooperative level สำหรับการ mix เสียง , set config ของ speaker และอื่นๆ
- 2) DirectSoundBuffer เป็น object ที่ทำหน้าที่สร้าง buffer ทั้ง Primary Sound Buffer (สร้างให้โดยฮาร์ดแวร์) และ Secondary Sound Buffer (สร้างเอง) นอกจากนี้ ยังควบคุมการแสดงผลเสียง , set ค่าต่างๆ เช่น ความดัง , ความถี่ ฯลฯ สำหรับการแสดงผลเสียง
- 3) DirectSound3DListener กับ DirectSound3DBuffer เป็น object ที่ทำงานเกี่ยวกับเสียง 3 มิติ และอุปกรณ์ที่ใช้ (Sound Card , ลำโพง) ต้องสนับสนุนด้วย เสียง 3 มิตินี้จะทำให้ผู้ฟังรู้สึกได้ถึงระยะห่าง และทิศทางของแหล่งกำเนิดเสียง ช่วยให้ได้รับความรู้สึกสมจริง ได้ธรรมชาติในการฟังมากขึ้น
- 4) DirectSoundCapture กับ DirectSoundCaptureBuffer เป็น object ที่ทำงานในการบันทึกเสียงจากอุปกรณ์ภายนอกเข้าสู่ Buffer และสามารถบันทึกลง File หรือสิ่งอื่นๆได้ด้วย

### 2.3.3 RIFF , WAV Files และ Data Structure

DirectSound จะทำงานได้กับข้อมูลเสียงแบบ Pulse Code Modulated (PCM) waveform ซึ่งไฟล์ที่มีรูปแบบดังกล่าวคือ RIFF (Resource Interchange File Format) กับ WAV (wave file) ซึ่งไฟล์ RIFF จะมีโครงสร้างดังนี้



รูปที่ 2.7 RIFF File Structure

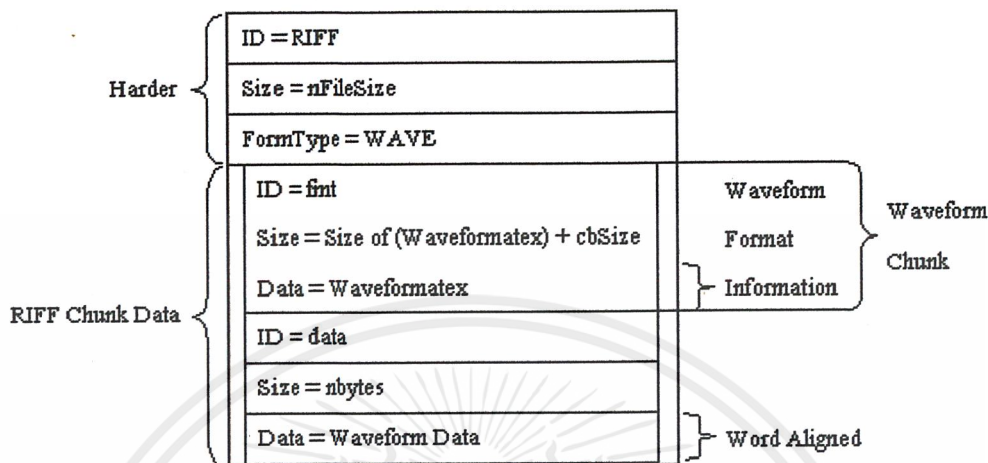
พื้นที่ส่วนบนของ RIFF เป็น 4 byte character ที่เก็บชนิดของไฟล์ ในที่นี้มีค่าเป็น RIFF และไม่สามารถมีค่าเป็น null ได้ถัดมาอีก 4 byte จะเป็น field ที่บอกขนาดของข้อมูลในไฟล์นี้ โดยค่านี้จะเท่ากับขนาดของไฟล์ - 8 byte (ไม่นับ 2 field ที่กล่าวไป) ต่อไปอีก 4 byte เป็น Form Type field ซึ่งจะบอกถึงรูปแบบของข้อมูลที่จัดเก็บ ถ้าเป็น wave file field นี้จะมีค่าเป็น WAVE ต่อไปจากนี้จะเป็นส่วนของข้อมูลเรียกว่า chunk ใน chunk จะถูกแบ่งออกเป็น subchunk ย่อย ๆ ซึ่ง Data ของเสียงจริงๆจะอยู่ภายใน subchunk การทำแบบนี้จะช่วยให้ format ของ file มีความยืดหยุ่นมากขึ้น

ส่วน Windows waveform audio file หรือ wave file จะมีโครงสร้างคล้ายคลึงกับ RIFF มี 3 field แรกเหมือน โครงสร้างของ RIFF และมีการแบ่งออกเป็น chunk และ subchunk เหมือนกันดังรูป 2.8 ในส่วนของ subchunk จะมีข้อมูลที่สำคัญคือ WAVEFORMATEX ซึ่งจะบอกข้อมูลสำคัญของ format ของข้อมูลที่เก็บอยู่คือ Format Tag บอกชนิดของ format ซึ่งใน Direct Sound ใช้ได้เฉพาะ Format ที่ไม่มีการบีบอัดข้อมูล , Channel ถ้าเป็น 1 คือ mono 2 คือ stereo ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample persec บอก samplerate ของ Data , BitPer Sample บอกจำนวน bit ที่ใช้ในการเก็บข้อมูลจากการ Sample 1 ครั้ง

โครงสร้างของไฟล์ WAV จะเป็นดังรูป



รูปที่ 2.8 Wave File Structure

### 2.3.4 Play Streaming Sound Buffer

การแสดงผล Streaming Sound Buffer จะต้องทำอย่างต่อเนื่องโดยจะต้องมีการกำหนดขนาดของ Buffer ก่อน และเตรียมข้อมูลเข้า Buffer เมื่อมีข้อมูลเข้าสู่ Buffer เพียงพอจึงเริ่มแสดงผลเสียงออกไป ในขณะที่เดียวกันก็ยังคงต้องอ่านข้อมูลเข้าสู่ Buffer ต่อไป โดยใช้ Pointer 2 ตัว ตัวหนึ่งชี้ที่จุดที่ต้องอ่านข้อมูลต่อไป ส่วนอีกตัวชี้ที่จุดที่กำลังเล่นอยู่

## 2.4 DirectMusic

### 2.4.1 การทำงานทั่วไปของ DirectMusic

DirectMusic เพิ่งถูกเพิ่มเข้ามาใน DirectX version 6 มีหน้าที่จัดการกับเสียงดนตรีต่าง ๆ โดยจะต่างกับ DirectSound ตรงที่ DirectSound จะใช้กับเสียงทุกชนิดที่มีการเก็บในรูปแบบ wav คือเก็บเป็น sample ของระดับเสียง ส่วน DirectMusic จะทำงานกับเสียงดนตรีต่าง ๆ ที่สร้างจากเครื่องดนตรีเท่านั้น โดยจะเก็บข้อมูลเป็นรหัสของเสียงแล้วส่งผ่าน Channel คล้ายกับการใช้ Palette หรือตารางสีใน DirectDraw มี file format แบบ MIDI (Musical Instrument Digital Interface) และ Direct Music ยังสนับสนุน DLS (Downloadable Sound) อีกด้วย

Sound Card ส่วนใหญ่ในปัจจุบันจะมี wavetable synthesis ซึ่งเป็นที่เก็บข้อมูลเสียงและรหัสเพื่อใช้กับไฟล์ MIDI ระบบ DLS จะช่วยให้ wavetable สามารถปรับปรุงได้โดยการ Download ข้อมูลเสียงใส่ใน wavetable ได้ ทำให้เพิ่มชนิดของเสียงและเครื่องดนตรีใหม่ๆ ได้

สำหรับ PC ที่ไม่มี Sound Card ที่ใช้ wavetable และไม่สนับสนุน DLS DirectMusic จะใช้ software synthesizer เพื่อจำลองเสียงต่าง ๆ แทนการใช้ wavetable แต่การทำเช่นนี้จะต้องใช้ CPU ที่มีความเร็วพอสมควร ด้วยการจำลองเสียงและ DLS ทำให้สามารถสร้างเสียงจำลองได้มากมายเสียงดนตรีในปัจจุบันจึงสามารถมีได้มากกว่า 128 เสียงซึ่งเป็นมาตรฐานที่ตั้งขึ้นโดย Roland GM สำหรับ file MIDI (general MIDI) และด้วย DLS และ DirectMusic ปัญหาความแตกต่างของ wavetable ใน Sound Card ก็ได้ถูกแก้ไขไป ผลที่ได้ก็คือ ไฟล์ดนตรีแบบ MIDI จะสามารถแสดงผลได้เสียงเหมือนกันใน computer ทุก ๆ เครื่อง โดยไม่ขึ้นอยู่กับ Sound Card ในเครื่องนั้น ๆ อีกต่อไป

คุณสมบัติที่สำคัญอีกอย่างหนึ่งของ DirectMusic ก็คือการแสดงผลดนตรีในแบบ Interactive ผู้พัฒนาสามารถพัฒนาให้ดนตรีในแบบ Application ของตนเปลี่ยนแปลงไปตามต้องการได้ ความสามารถนี้เป็นประโยชน์อย่างมากในการพัฒนาเกมและมัลติมีเดียต่าง ๆ ดนตรีที่เปลี่ยนแปลงได้ตามอารมณ์การเล่นหรือตามภาพที่เห็นนั้นช่วยให้ผู้ใช้งานเกิดอารมณ์ร่วมได้อย่างมาก DirectMusic สามารถตัดสินใจเปลี่ยนเสียงดนตรีได้ (ตามที่ผู้พัฒนาเขียนขึ้น) ทำให้เกิดความยืดหยุ่นมากขึ้น

#### 2.4.2 Object ที่สำคัญของ DirectMusic

Object ที่สำคัญของ DirectMusic มีดังนี้

- 1) DirectMusic เป็น object หลักที่จำเป็นต้องสร้างขึ้นเป็นอันดับแรก ทำหน้าสร้างและควบคุม object อื่นๆ
- 2) DirectMusicPerformance ทำหน้าที่ควบคุมการเล่นเสียงดนตรีต่างๆ เช่น ปรับความเร็ว, สั่งเล่นวน loop เป็นต้น
- 3) DirectMusicLoader ทำหน้าที่ดึงข้อมูลเสียงจาก Disk เพื่อเตรียมสำหรับการแสดงผล โดยจะนำข้อมูลที่ได้เตรียมแบ่งลง segment
- 4) DirectMusicSegment เป็น object ที่ใช้เก็บข้อมูลเสียงดนตรีที่แบ่งออกเป็นส่วนเล็ก ๆ เตรียมพร้อมสำหรับแสดงผลต่อไป

#### 2.4.3 MIDI (Musical Instrument Digital Interface)

MIDI เป็น ไฟล์ที่ออกแบบมาสำหรับการส่งถ่ายเสียงดนตรีจากเครื่องดนตรีไปสู่ข้อมูลดิจิทัลเพื่อนำข้อมูลดิจิทัลนั้นมาแสดงผลผ่านเครื่องคอมพิวเตอร์ในแบบ real time โดยในไฟล์ MIDI จะเก็บข้อมูลว่าเสียงนี้เป็นโน้ตตัวใด มีระดับความดังขนาดไหนและจะต้องแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลในเวลาใด แต่ไม่ได้บอกว่าจะแสดงผลเป็นเสียงเครื่องดนตรีชนิดใดเพราะใช้ระบบ Channel ในการแสดงผล ระบบต้องทำการ map Channel นี้กับ wavetable ที่อยู่ใน Sound Card เองว่าจะให้ Channel นี้แทนด้วยเครื่องดนตรีชนิดใดซึ่ง wavetable ของผู้ผลิตแต่ละรายจะไม่เหมือนกัน ทำให้เกิดปัญหาที่เมื่อนำไฟล์ MIDI ไปเปิดที่ต่างเครื่องกันจะได้เสียงที่ไม่เหมือนกัน ต่อมา Roland จึงได้กำหนดค่าเสียงเครื่องดนตรีมาตรฐานขึ้นทั้งหมด 128 เสียงเพื่อแก้ปัญหาดังกล่าว แต่ในปัจจุบันมีความต้องการใช้เสียงของเครื่องดนตรีมากกว่า 128 เสียงทำให้เกิดระบบ DLS ขึ้นซึ่งจะช่วยให้สามารถใช้เสียงดนตรีได้อย่างไม่จำกัดแต่ก็ทำให้เกิดปัญหาเดิมขึ้น แต่เมื่อใช้ร่วมกับ DirectMusic ก็จะช่วยจัดปัญหาเดิมออกไปได้ดังที่กล่าวไปแล้ว อนึ่งแม้เสียงที่ได้จะเป็นเครื่องดนตรีชนิดเดียวกันแล้วแต่คุณภาพของเสียงยังคงขึ้นอยู่กับคุณภาพของ Sound Card ที่ใช้

ในไฟล์ MIDI จะเก็บข้อมูลเป็นจังหวะ , ระยะเวลา และค่าของตัวโน้ต Standard MIDI จะมี Channel ได้มากถึง 16 Channel แต่ละ Channel จะแทนเครื่องดนตรี 1 ชนิด ซึ่งจะคล้ายกับ Palette สีที่ 1 index แทนค่าสี 1 สี เพราะฉะนั้นการมีเพียง 16 Channel ไม่ได้หมายความว่า 16 ชนิดเครื่องดนตรี แต่หมายความว่าสามารถแสดงผลพร้อมกันได้ไม่เกิน 16 ชนิดเครื่องดนตรี ส่วนชนิดเครื่องดนตรีทั้งหมดที่สามารถแสดงผลได้จะขึ้นอยู่กับ Software และ Sound Hardware ที่ใช้ในเครื่องนั้นๆ เมื่อมีข้อมูลเสียงมาที่ Channel ใดอุปกรณ์ก็จะทำการเปิดและแสดงผลเสียงเครื่องดนตรีชนิดนั้น

### บทที่ 3

## ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนในการดำเนินการวิจัยทั้งหมดแบ่งออกได้เป็น 3 ส่วนใหญ่ ๆ คือ ส่วนของการออกแบบ ส่วนการสร้างภาพ และส่วนของการเขียนโปรแกรม ซึ่งแต่ละส่วนล้วนมีความสำคัญกับการพัฒนา เกมกระดานมหาสนุกอย่างมาก แต่ละขั้นตอนประกอบด้วยขั้นตอนย่อย ๆ ที่ต้องทำ ซึ่งแจกแจง รายละเอียดคร่าว ๆ ได้ดังนี้

#### 1. ขั้นตอนการออกแบบ จะต้องทำการออกแบบส่วนต่าง ๆ ดังนี้

- การออกแบบกติกาการเล่นและระบบของเกม ซึ่งเป็นหัวใจหลักของเกมทุกเกม
- ออกแบบภาพ graphic ที่จะใช้ในเกมซึ่งเป็นส่วนที่มีความสำคัญมากต่ออารมณ์การเล่นของเกม ภาพ graphic ที่สวยงามจะช่วยให้ผู้เล่นมีอารมณ์ร่วม และสนุกสนานกับเกมมากขึ้น
- ออกแบบ Interface ที่ใช้ติดต่อและตอบโต้กับผู้เล่น Interface ที่ดีสำหรับเกมจำเป็นจะต้องง่าย สามารถเรียนรู้และใช้งานได้อย่างรวดเร็ว
- ออกแบบการจัดเก็บข้อมูลที่เป็นต้องใช้ในเกม ให้เป็นระเบียบ, ประหยัด และสามารถนำมาใช้ได้ง่ายเพื่อเพิ่มประสิทธิภาพในการทำงานของตัวเกม
- ออกแบบวิธีการเขียน โปรแกรม และเลือกใช้ Hardware , Software รวมถึง Component และ tool ต่าง ๆ ที่จะนำมาใช้ในการเขียน โปรแกรม

2. ขั้นตอนการการสร้างภาพ เป็นขั้นตอนที่ต้องใช้ tool ต่าง ๆ สร้างภาพ graphic ที่ได้ออกแบบไว้ให้อยู่ในรูปแบบที่สามารถนำมาใช้งานในโปรแกรมได้ ซึ่งขั้นตอนนี้จะเริ่มทันที หลังจากออกแบบภาพ graphic ที่ต้องการใช้เสร็จเรียบร้อยแล้ว และจะทำงานควบคู่ไปกับการออกแบบและการเขียนโปรแกรมจนกระทั่งได้ภาพ graphic ที่ต้องการ

3. ขั้นตอนการเขียน โปรแกรม จะต้องทำการเขียน code ในส่วนต่าง ๆ เพื่อนำมาประกอบให้ เกมสามารถทำงานได้ตามที่ได้ออกแบบเอาไว้ โดยในเกมกระดานมหาสนุกอาจจะแบ่งส่วนของการเขียน code ออกเป็นส่วน ๆ ดังนี้

- ส่วนการแสดงผลภาพ ควบคุมการแสดงผลภาพในเกมให้ดำเนินไปอย่างถูกต้อง
- ส่วนการแสดงผลเสียง ควบคุมการแสดงผลเสียงในเกมให้ดำเนินไปอย่างถูกต้อง
- ส่วนการรับ input จากผู้เล่น ควบคุมการรับ input ให้ถูกต้องและสามารถนำ input นั้นไปประมวลผลต่อ เพื่อดำเนินเกมไปตามกติกาที่ตั้งไว้อย่างถูกต้อง
- ส่วนควบคุมการติดต่อกับไฟล์ข้อมูลต่าง ๆ ต้องสามารถบันทึกหรือดึงข้อมูลมาใช้ได้อย่างมีประสิทธิภาพ

- ส่วนควบคุมการเล่นหลัก ซึ่งจะควบคุมการเล่นให้เป็นไปตามกติกาและรูปแบบที่ออกแบบไว้ในขั้นตอนการออกแบบ ซึ่งเป็นส่วนของโปรแกรมที่สำคัญที่สุด

การดำเนินงานจะเรียงลำดับจากขั้นตอนที่ 1 ถึงขั้นตอนที่ 3 โดยในขั้นตอนของการสร้างภาพจะทำงานขนานไปพร้อม ๆ กับการทำงานในขั้นออกแบบและเขียนโปรแกรม เนื่องจากเป็นขั้นตอนที่ต้องใช้เวลามาก และในระหว่างการเขียน โปรแกรมยังไม่จำเป็นต้องใช้ภาพจริงก็ได้

### 3.1 ขั้นตอนการออกแบบ

#### 3.1.1 การออกแบบกติกาและระบบการเล่น

โจทย์ของการออกแบบเกมนี้ คือ ต้องการเกมที่สามารถเล่นได้ง่าย กติกาไม่ยุ่งยากหรือซับซ้อนมากจนเกินไปนัก และสามารถสอดแทรกการเสริมสร้างทักษะทางด้านต่าง ๆ ได้ง่าย คำตอบของโจทย์นี้ก็คือ รูปแบบเกมกระดานที่มีการเล่นกันมานานและเป็นที่ยอมรับกันดีของคนทั่วไป ตัวอย่างเกมกระดานที่มีชื่อเสียงในอดีตก็คือ เกมเศรษฐี (Monopoly) รูปแบบการเล่นหลักของเกมกระดานแบบนี้ก็คือการทอยลูกเต๋าแล้วเดินไปตามช่องต่าง ๆ บนกระดานเพื่อบรรลุจุดมุ่งหมายอย่างใดอย่างหนึ่งที่เกมกำหนด สำหรับเกมกระดานมหาสนุกที่มีกลุ่มเป้าหมายเป็นเด็ก และมีวัตถุประสงค์เพื่อช่วยเสริมสร้างทักษะต่าง ๆ ให้แก่ผู้เล่น เกมกระดานในรูปแบบที่กล่าวมามีความเหมาะสมมาก เพราะมีวิธีการเล่นที่ง่ายและสามารถสอดแทรกการเสริมสร้างทักษะต่าง ๆ ผ่านส่วนที่ผู้เล่นจะต้องทำเพื่อบรรลุจุดมุ่งหมายที่เกมนั้น ซึ่งก็คือการทำแต้มให้ได้ถึงจุดที่กำหนด โดยแต้มจะได้จากการเดินเกมย่อยที่กำหนดไว้ในแต่ละช่องของตาราง ส่วนระบบการเล่นของเกม ผู้เล่นแต่ละคนจะผลัดกันเล่น โดยทอยลูกเต๋าแล้วเดินไปตามแต้มที่ได้ เมื่อไปตกที่ช่องใดก็จะต้องทำตามคำสั่งที่ปรากฏในช่องนั้น ผลัดกันเดินสะสมแต้มไปเรื่อย ๆ เมื่อมีผู้ใดได้แต้มครบตามที่เกมกำหนดและสามารถกลับไปถึงจุด start ได้ก่อนจะเป็นผู้ชนะ จุดสำคัญที่จะทำให้เกมนี้มีความสนุกสนานมากขึ้น คือ การออกแบบคำสั่งหรือเกมย่อยในช่องบนกระดานให้มีความหลากหลาย เช่น หุุดเดิน 1 รอบ จุควาร์ปซึ่งผู้ที่ตกจะถูกเคลื่อนย้ายไปอีกตำแหน่งหนึ่ง ฯลฯ นอกจากนี้รูปแบบของเส้นทางการเดินบนตารางก็ยังสามารถปรับเปลี่ยนได้เพื่อให้มีความแปลกใหม่และน่าเล่นเพิ่มขึ้น ซึ่งในที่นี้ได้สร้างรูปแบบของกระดานที่แตกต่างกันขึ้นมา 3 แบบหรือก็คือ 3 จากในเกมนั่นเอง ส่วนเกมย่อยมีทั้งหมด 4 เกม เพื่อเสริมสร้างทักษะด้านต่าง ๆ ที่แตกต่างกันคือ

1. เกม Vocab เสริมสร้างทักษะทางภาษาอังกฤษของผู้เล่น
2. เกม Equation เสริมสร้างทักษะทางคณิตศาสตร์ของผู้เล่น
3. เกม Memory เสริมสร้างทักษะด้านความจำของผู้เล่น
4. เกม Image เสริมสร้างทักษะด้านจินตนาการการปะติดปะต่อภาพของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังทำการเพิ่มจุด checkpoint ซึ่งเป็นช่องที่เมื่อผู้เล่นเดินผ่านจะได้รับค่า checkpoint แต่ละแผนที่จะมี 4 ค่า ถ้าผู้เล่นมี checkpoint ครบ 4 ค่าแล้วเดินผ่านจุด start จะได้เติม bonus เพิ่มขึ้น ในบางครั้งการเล่นอาจใช้เวลานาน จึงจำเป็นต้องมีระบบการ Save ข้อมูลของเกมเอาไว้ เพื่อสามารถ Load Game กลับมาเล่นต่อได้ในครั้งต่อไป และสำหรับผู้ที่นิยมเล่นเฉพาะเกมย่อยแต่ไม่ชอบที่จะเล่นเกมหลักก็มีส่วนของการเลือกเล่นเฉพาะเกมย่อยได้ โดยเลือกหัวข้อ MiniGame จากหน้า title ซึ่งสามารถเลือกเล่นเฉพาะเกมย่อยที่ชอบได้ และยังสามารถเลือกฝึกทักษะที่ต้องการได้โดยตรงอีกด้วย สำหรับระดับความยากของโจทย์จะระบุให้ตรงกับกลุ่มเป้าหมายที่เป็นเด็กระดับประถมศึกษา โดยคละกันไปทั้งโจทย์ยากและง่าย เพื่อให้ผู้เล่นไม่เกิดความเบื่อหน่าย

### 3.1.2 การออกแบบภาพ

รูปแบบของภาพ Graphic ในเกมกระดานมหาสนุก มีลักษณะของรูปภาพเป็นการ์ตูนแบบ 3 มิติผสมกับรูปภาพที่เป็น 2 มิติ มีการใช้สีสันสดใส เพื่อดึงดูดความสนใจของผู้เล่น ซึ่งสาเหตุที่เกมกระดานมหาสนุกมีการเสนอรูปภาพ Graphic ในลักษณะนี้ เพราะว่ากลุ่มเป้าหมายเป็นเด็กในระดับชั้นประถมศึกษา

เกมกระดานมหาสนุกนี้มีการดำเนินการเล่นภายในเกม โดยผ่านตัวละคร 4 ตัว ซึ่งแบ่งเป็นตัวละครที่เป็นผู้ชาย 2 ตัว (Ton และ Mike) ตัวละครที่เป็นผู้หญิง 1 ตัว (Fon) และตัวละครที่เป็นตัวตลก 1 ตัว (Crow) โดยตัวละครแต่ละตัวนั้นจะมีการออกแบบมาเพื่อรองรับความชอบที่แตกต่างกันของผู้เล่นแต่ละคน ส่วนแผนที่หรือฉากที่มีให้เลือกเล่นทั้ง 3 ฉากถูกออกแบบมาให้มีบรรยากาศที่แตกต่างกัน โดยฉาก Island มีบรรยากาศแบบเกาะกลางทะเลที่สนุกสนาน มีชนพื้นเมืองออกมาเชียร์การแข่งขัน ฉาก Giant Room เป็นฉากห้องคนยักษ์สิ่งของต่าง ๆ ก็จะมีขนาดใหญ่โตมโหฬาร และฉาก Fantastic ประกอบขึ้นมาจากสิ่งก่อสร้างที่สำคัญต่าง ๆ ในโลก ที่สำคัญภาพในฉากนี้ส่วนใหญ่จะเป็นภาพที่สร้างจาก 3D Studio Max

### 3.1.3 การออกแบบ Interface ที่ใช้ติดต่อและตอบโต้กับผู้เล่น

Interface ที่ดีจะต้องเข้าใจได้ง่าย สามารถใช้ได้โดยไม่ต้องมีการฝึกฝนมากนัก โดยเฉพาะเกมนี้ที่มีกลุ่มเป้าหมายเป็นเด็กจึงต้องการ Interface ที่ใช้ง่ายและสวยงาม การโต้ตอบกับผู้เล่นจึงเลือกใช้ mouse เพื่อให้การตอบโต้ทำได้โดยง่าย และเพราะต้องการให้ใช้ mouse เป็นอุปกรณ์โต้ตอบเพียงพอย่างเดียว Interface ทั้งหมดจึงมีลักษณะเป็นปุ่มที่ตอบสนองกับ event ที่เกิดขึ้นจาก mouse ของปุ่ม ปุ่มแต่ละปุ่มจะให้ผลลัพธ์ที่แตกต่างกันไป ซึ่งในเกมกระดานมหาสนุกนี้ประกอบด้วย Interface หลัก ๆ ดังนี้

ตารางที่ 3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ของเกมกระดานมหาสนุก

Interface	ปุ่ม	หน้าที่
Title	NewGame	เข้าสู่ส่วนเลือกตัวละคร
	LoadGame	เข้าสู่ส่วน LoadGame
	MiniGame	เข้าสู่ส่วน MiniGame
	Help	เข้าสู่ส่วน Help
	Exit	ออกจากโปรแกรม
MiniGame	Vocab	เข้าสู่ส่วนการเล่นเกมน้อย Vocab
	Equation	เข้าสู่ส่วนการเล่นเกมน้อย Equation
	Memory	เข้าสู่ส่วนการเล่นเกมน้อย Memory
	Image	เข้าสู่ส่วนการเล่นเกมน้อย Image
	Back	กลับสู่หน้า Title
Help	Help หลัก	เข้าสู่คู่มือรวม
	Vocab	เข้าสู่คู่มือ Vocab
	Equation	เข้าสู่คู่มือ Equation
	Memory	เข้าสู่คู่มือ Memory
	Image	เข้าสู่คู่มือ Image
	Exit	ออกจากหน้า Help ไป Title
Vocab	a-z	รับตำแหน่งมาประมวลผล
Equation	+, -, *, ÷	รับตำแหน่งมาประมวลผล
Memory	a-z, 0-9	รับตำแหน่งมาประมวลผล
Image	Stop	หยุดเปิดภาพแล้วโชว์ตัวเด็ก
	a-j	รับตำแหน่งมาประมวลผล
เลือกตัวละคร	2,3,4 player	เลือกจำนวนผู้เล่น
	ตัวละครทั้ง 4	เลือกตัวละครสำหรับแต่ละผู้เล่น
เลือก Map	Map ทั้ง 3	เลือก Map ที่จะเล่น
การเล่นเกมนหลัก	Play	ทอยลูกเต๋า
	Save	ไปหน้า Save Game
	Exit	กลับ ไป Title
Load Page	File 1-3	ดึงข้อมูลจากไฟล์ที่เลือก
	Back	กลับ ไป Title
Save Page	File 1-3	บันทึกข้อมูลลงไฟล์ที่เลือก
	Back	กลับ ไป Title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4 การออกแบบการจัดเก็บข้อมูล

ข้อมูลที่จำเป็นจะต้องจัดเก็บเพื่อใช้งานในเกมนี้ จะประกอบด้วยข้อมูลเกี่ยวกับแผนที่และข้อมูลโจทย์ของแต่ละเกมย่อย ซึ่งจะไม่มีการเปลี่ยนแปลงค่าข้อมูล และมีจำนวนข้อมูลไม่มาก จึงเลือกใช้วิธีเก็บเป็นแบบ Random – Access File เพื่อให้มีการเข้าถึงข้อมูลได้ง่ายและรวดเร็ว อีกทั้งใช้เนื้อที่น้อยกว่าการเก็บเป็น DataBase ส่วน File ที่มีการเปลี่ยนแปลงบ่อยครั้งอย่างเช่น File Save Game ก็ใช้เป็น Random – Access File เช่นเดียวกัน ด้วยเหตุผลคล้าย ๆ กันคือ เข้าถึงได้ง่ายรวดเร็วและกินเนื้อที่น้อย

### 3.1.5 การออกแบบวิธีการเขียนโปรแกรมและกำหนด Hardware , Software ที่จะใช้

การเขียน โปรแกรมเกมกระดานมหาสนุกนี้ได้เลือกใช้คอมพิวเตอร์ภาษา Visual Basic 6 มาใช้ร่วมกับ DirectX7.0 ซึ่งเป็น component ที่ทำหน้าที่ติดต่อกับอุปกรณ์ต่าง ๆ โดยเฉพาะ อุปกรณ์แสดงผลทั้งภาพและเสียง จุดเด่นของ DirectX คือ ทำงานติดต่อกับอุปกรณ์โดยตรงไม่ต้องผ่าน OS ทำให้ประสิทธิภาพในการทำงานสูงขึ้น สามารถทำงานได้เร็วขึ้น ส่วนการสร้างภาพที่ใช้ภายในเกมใช้ Photoshop 6.0 กับ pen ในการสร้างภาพ 2 มิติ และใช้ 3D Studio Max ในการสร้างภาพ 3 มิติ ส่วนวิธีการเขียนโปรแกรมเป็นแบบ procedural โดยแบ่งออกเป็น module แต่ละ module ก็จะทำหน้าที่เฉพาะช่วงของมัน แล้วจะส่งต่อการทำงานไปเป็นช่วง ๆ

## 3.2 การสร้างภาพ Graphic

ภาพ Graphic ที่ใช้ในเกมกระดานมหาสนุกมีทั้งแบบ 2 มิติและ 3 มิติ ซึ่งวิธีในการสร้างภาพทั้ง 2 แบบนี้ก็มีความแตกต่างกันไปทั้งหลักการที่ทำและเครื่องมือที่ใช้ โดยจะอธิบายการสร้างรูปภาพทั้ง 2 แบบดังนี้

### 3.2.1 การสร้างรูปภาพ 2 มิติ

ภาพ 2 มิติจะเป็นภาพที่มองเห็นเป็นลักษณะแบนราบ ไม่ค่อยรู้สึกถึงความตื้นลึกหนาบางของรูปภาพมากสักเท่าไร โดยภาพที่เป็น 2 มิติจะเป็นภาพที่เห็นเพียงแค่มุมมองมุมมองเดียวคือ มุมมองในแนวแกน X,Y ซึ่งภาพ 2 มิติที่ใช้ในเกมกระดานมหาสนุกมีการใช้อุปกรณ์ในการสร้างภาพ คือ pen และ โปรแกรม Adobe Photoshop

#### 3.2.1.1 Pen

เป็นอุปกรณ์ที่มีลักษณะเหมือนปากกาซึ่งช่วยในการวาดรูปที่เป็น 2 มิติ อุปกรณ์ชนิดนี้ช่วยให้ผู้ใช้สามารถวาดภาพได้นับว่าการใช้เมาส์ เพราะรูปแบบของมันเหมือนปากกา จึงทำให้ผู้ใช้สามารถใช้งาน pen ในการวาดรูปหรือการเขียนได้นับว่าการใช้เมาส์



รูปที่ 3.1 รูปต้นมะพร้าวที่ใช้ pen ในการวาดเป็นภาพ 2 มิติ

### 3.2.1.2 Adobe Photoshop

เป็นโปรแกรมที่ใช้ในการจัดการด้านภาพ Photoshop หรือแมวไฟฟ้าที่ฝรั่งเค้าเรียกกันว่า Electric Cat ซึ่งเป็นสมญานามของโปรแกรมที่มีความสามารถรอบตัว โปรแกรม Photoshop แต่ก่อนอาจจะเป็นแค่เพียงความต้องการทางด้านความสามารถในการปรับปรุงด้านภาพ เพื่อนำเสนอลงแผ่นกระดาษหรือนำเสนอลงบนแผ่นฟิล์ม แต่ปัจจุบันนี้ Photoshop ได้พัฒนาความสามารถให้ทำได้มากกว่านั้นแล้ว โดยภาพที่ได้ออกแบบจะเก็บไว้ในนามสกุล bmp เพื่อให้โปรแกรม DirectX สามารถเรียกใช้ได้

### 3.2.2 การสร้างรูปภาพ 3 มิติ

ภาพ 3 มิติเป็นภาพที่มีทั้งความกว้าง ความยาว และความลึก จึงส่งผลให้ภาพมีความสมจริงมากขึ้นกว่าภาพที่เป็นลักษณะ 2 มิติ และในบางครั้งภาพ 3 มิติอาจจะสร้างความรู้สึกที่เหนือกว่าความเป็นจริงด้วยซ้ำ ในปัจจุบันการสร้างหรือการทำแอนิเมชันด้วยภาพ 3 มิตินั้นมีบทบาทมากขึ้นทั้งทางด้านภาพยนตร์ โฆษณา การจำลองแบบเหตุการณ์และเกม เป็นต้น ซึ่งในเกมกระดานมหาสนุกได้มีการนำภาพลักษณะ 3 มิติเข้ามาใช้สำหรับภาพที่เป็นตัวละคร สิ่งก่อสร้าง และของประกอบฉากที่ใช้ภายในเกม เครื่องมือที่ใช้ในการสร้างภาพ 3 มิติสำหรับเกมกระดานมหาสนุก คือ โปรแกรม 3D Studio Max 4



รูปที่ 3.2 รูปภาพที่สร้างจากโปรแกรม 3D Studio Max

สาเหตุที่ใช้โปรแกรม 3D Studio Max ในการสร้างภาพต่าง ๆ ที่ใช้ภายในเกม เนื่องจากต้องการความสะดวกในการทำแอนิเมชันของภาพเหล่านั้นซึ่งมีเป็นจำนวนมาก

### 3.2.2.1 3D Studio Max

เป็นโปรแกรมที่ใช้ในการสร้างงาน 3 มิติแบบแอนิเมชัน ซึ่งพัฒนาโดย Kinetix โปรแกรม 3D Studio Max สามารถรันได้บนระบบปฏิบัติการ Windows 95/98/ME และ Windows NT/2000 ซึ่งเวอร์ชันก่อน Release 2.5 จะรันได้เฉพาะบน Windows NT เท่านั้น ด้วยประสิทธิภาพที่สูงขึ้นอย่างมากในหลาย ๆ ด้าน ทั้งทางด้านการสร้าง Model รูปทรง 3 มิติ รูปแบบของพื้นผิววัตถุที่มากขึ้น การคำนวณสภาพการชนของวัตถุ (Simulation)

ขั้นตอนในการสร้างภาพจะเริ่มต้นด้วย การสร้างวัตถุ 3 มิติ นอกจากนี้ยังสามารถกำหนดเส้นทางการเคลื่อนที่พร้อมกำหนดการเปลี่ยนแปลงลักษณะต่าง ๆ ของวัตถุ 3 มิติ นั้น เมื่อทำการออกแบบวัตถุเสร็จแล้ว ก็ทำการ Render โดยเลือก Render Output เป็นนามสกุล bmp กำหนดขนาดของภาพตามความเหมาะสม จะได้ภาพเป็นนามสกุล bmp เพื่อให้โปรแกรม DirectX สามารถเรียกใช้ได้

### 3.2.2.2 การสร้างการเคลื่อนไหวให้กับตัวละคร

ตัวละครที่ใช้ในเกมกระดานมหาสนุกจะเป็นตัวดำเนินเนื้อเรื่องภายในตัวเกมทั้งหมด ดังนั้นตัวละครที่นำมาใช้นี้จะต้องมีการแสดงออกถึงกริยาท่าทางต่าง ๆ เพื่อที่จะให้ผู้เล่นเกมไม่รู้สึกรำคาญเมื่อกินไป เพราะตัวละครภายในเกมไม่มีการแสดงท่าทางหรือเคลื่อนไหวเลย การที่ตัวละครภายในเกมมีการแสดงท่าทางหรือเคลื่อนไหว มันจะทำให้เกมดูน่าสนุกขึ้นและดึงดูดผู้เล่นได้มากกว่าเกมที่ตัวละครไม่เคลื่อนไหวหรือเคลื่อนไหวแต่ดูไม่ค่อยสมคูลกัน การสร้างให้ตัวละครภายในเกมมีการเคลื่อนไหวนั้นในโปรแกรม 3D Studio Max มีอยู่หลายวิธีด้วยกัน โดยมากจะเป็นพวกโปรแกรม Plug in ในโปรแกรม 3D

Studio Max สามารถทำได้โดยการใช้ bone ซึ่งในเกมกระดานมหาสนุกนี้ใช้ Plug in Character Studio ช่วยในการสร้างการเคลื่อนไหวให้กับตัวละครภายในเกม

### 3.2.3 ภาพที่ใช้ภายในเกมกระดานมหาสนุก

ภาพเคลื่อนไหวที่ใช้ภายในเกมกระดานมหาสนุกจะมีการสร้างภาพในจังหวะต่าง ๆ ออกมา 4 รูป เพื่อรองรับกับการทำงานของโปรแกรมที่มีการเขียนไว้แบบสลับ 4 เฟรม ซึ่งประเภทของภาพที่ใช้ภายในเกมกระดานมหาสนุกแบ่งออกได้ดังนี้

#### 3.2.3.1 ภาพตัวละคร

ภาพของตัวละครที่ใช้ภายในตัวเกมกระดานมหาสนุกแบ่งออกได้ 2 ลักษณะ คือ

- 1) ภาพแสดงการเดินของตัวละคร ประกอบด้วย การเดิน 4 ทิศทาง ดังนี้



รูปที่ 3.3 แสดงจังหวะการเดินของตัวละครในแนวเฉียงขึ้นทางด้านซ้าย

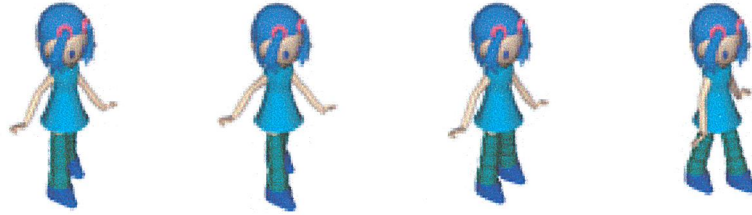


รูปที่ 3.4 แสดงจังหวะการเดินของตัวละครในแนวเฉียงขึ้นทางด้านขวา



รูปที่ 3.5 แสดงจังหวะการเดินของตัวละครในแนวเฉียงลงทางด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงจังหวะการเดินของตัวละครในแนวเฉียงลงทางด้านขวา

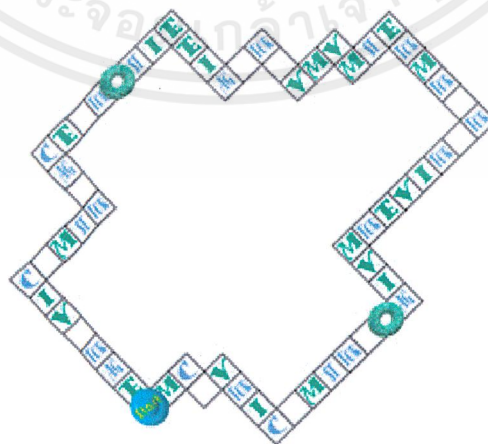
- 2) ภาพแสดงการขึ้นของตัวละคร ใช้ในขณะที่ตัวละครรอให้ผู้เล่นทอยลูกเต๋าและรอให้ถึงตาตัวเองเดินมีดังนี้



รูปที่ 3.7 แสดงจังหวะต่าง ๆ ในขณะขึ้นของตัวละคร

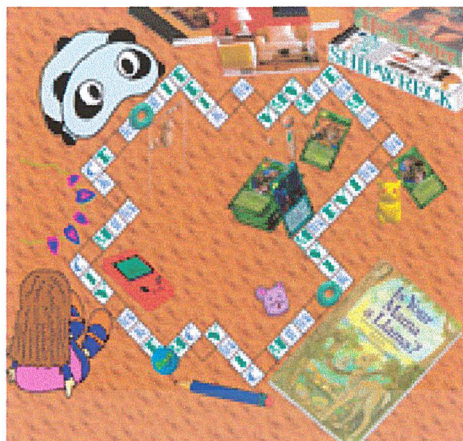
### 3.2.3.2 ภาพแผนที่และทางเดิน

ภาพเส้นทางการเดินกับภาพของแผนที่นั้นเราทำแยกส่วนกันก่อน แล้วค่อยนำมารวมกันทีหลัง โดยเราจะเริ่มจากกำหนดเส้นทางในการเดินเสียก่อนแล้วจึงค่อยทำแผนที่ เพราะว่าเส้นทางในการเดินนั้นจะเป็นตัวกำหนดลักษณะของแผนที่ว่าจะออกมาในรูปแบบลักษณะใด



รูปที่ 3.8 ภาพของเส้นทางที่มีการกำหนดมาก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ภาพของแผนที่หลังจากมีการรวมกับภาพของเส้นทาง

### 3.2.3.3 ภาพสิ่งของประกอบจากแผนที่

เป็นภาพที่ใช้ช่วยประดับแผนที่เพื่อให้แผนที่ไม่ดูว่างจนเกินไป และเพื่อเพิ่มการเคลื่อนไหวให้กับแผนที่ ซึ่งจะทำให้แผนที่ดูมีชีวิตชีวาขึ้นและทำให้ผู้เล่นรู้สึกสนุกกับการเล่นมากขึ้นด้วย โดยภาพที่สร้างขึ้นนั้นจะมีการสร้างมา 4 ภาพ ( เฉพาะภาพที่มีการขยับ ) เนื่องจากในเกมนี้มีการสับเฟรมแบบ 4 เฟรมคั้งนั้นภาพที่จะมีการขยับจึงต้องมี 4 ภาพด้วย ส่วนภาพที่ไม่มีการขยับก็จะสร้างเพียงภาพเดียวก็พอ



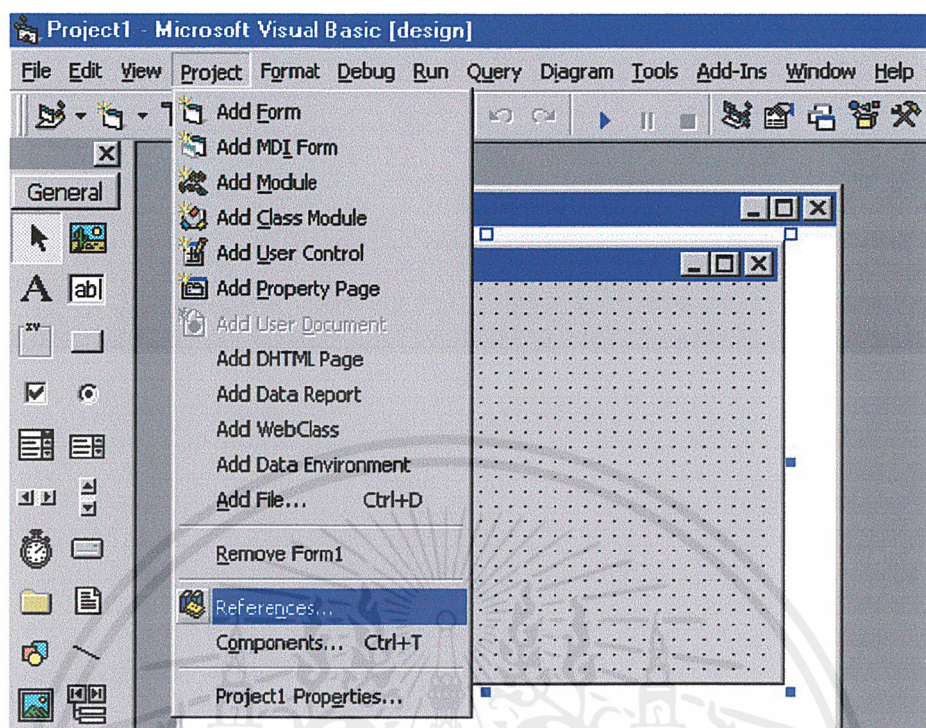
รูปที่ 3.10 ภาพสุนัขทั้ง 4 จังหวะที่นำมาประกอบฉาก

## 3.3 ขั้นการเขียนโปรแกรม

เราได้เลือกภาษา Visual Basic 6 มาเป็นคอมไพเลอร์และนำ DirectX 7.0 มาใช้ในการแสดงผลภาพและเสียง การนำ DirectX มาใช้กับ Visual Basic จะอยู่ในรูปแบบเดียวกับการใช้ Library โดยจะต้องทำการติดตั้ง DirectX SDK ก่อนแล้วจึงเข้าไปตั้งค่าใน Visual Basic ดังนี้

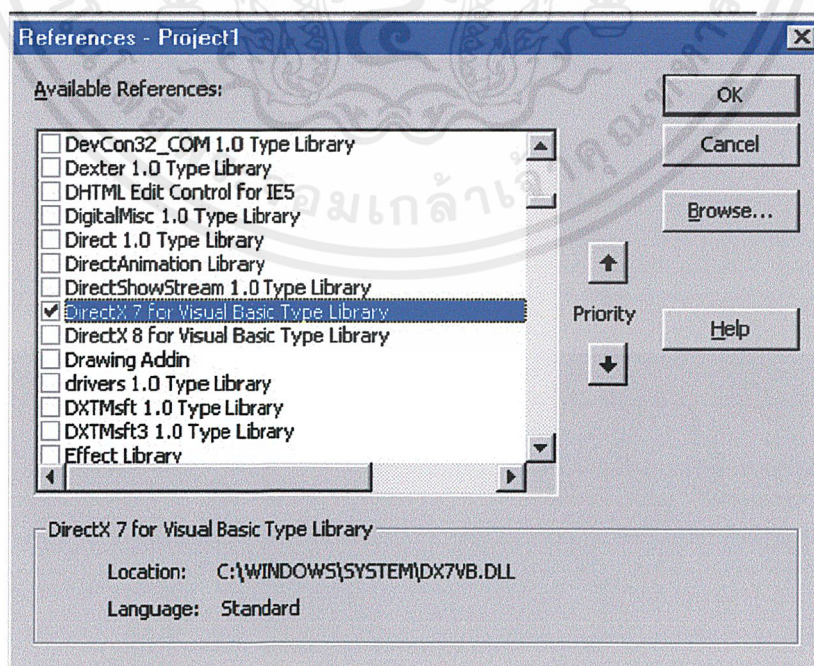
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เลือก Project จากแถบ menu แล้วเลือก References เพื่อเปิดหน้าต่าง References ขึ้นมา



รูปที่ 3.11 แสดงการเลือก References จากเมนู Project

2. คลิกเลือก DirectX 7 for Visual Basic Type Library แล้วกด OK ก็จะสามารถใช้ DirectX บน Visual Basic ได้ (ในที่นี้ใช้ DirectX 7)



รูปที่ 3.12 แสดงการเลือก DirectX 7 จากหน้าจอ References

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Component ของ DirectX 7 ที่ใช้ในโปรแกรมนี้นี้มี 3 ตัวคือ

1. DirectDraw ทำหน้าที่ควบคุมการแสดงผลภาพ
2. DirectMusic ทำหน้าที่ควบคุมการแสดงผลเสียงดนตรีในรูปแบบ Midi
3. DirectSound ทำหน้าที่ควบคุมการแสดงผลเสียงดนตรีในรูปแบบ Wave File Format

ซึ่งรายละเอียดหลักการการทำงานของ Component ต่าง ๆ เหล่านี้ ได้แสดงไว้แล้วในบทที่ 2 ในบทนี้จะกล่าวถึงหลักการเขียนโปรแกรมที่ใช้ในเกมนี้ นอกจากนี้ยังต้องทำการเขียนโปรแกรมเพื่อควบคุมและประมวลการเล่นให้เป็นไปตามกติกาที่ตั้งไว้ ลักษณะของการเขียนโปรแกรมเป็นการเขียนแบบ Procedural โดยแบ่งออกเป็น module ตามส่วนต่าง ๆ ของเกม เช่น Title , MiniGame , SaveLoad และในแต่ละ module ก็จะแบ่งย่อยออกเป็น function ซึ่งทำหน้าที่ต่าง ๆ กัน แต่ในที่นี้เราจะแบ่งการเขียนโปรแกรมตามลักษณะการทำงานดังนี้

- 1 ส่วนแสดงผลภาพโดยใช้ DirectDraw
- 2 ส่วนแสดงผลเสียงดนตรีในรูปแบบ Midi โดยใช้ DirectMusic
- 3 ส่วนแสดงผลเสียงในรูปแบบ Wave File Format โดยใช้ DirectSound
- 4 ส่วนการรับ input จากผู้เล่น
- 5 ส่วนควบคุมการเล่นเกมหลัก

ซึ่งในการเขียนโปรแกรมจริง ๆ แล้วจะต้องทำควบคู่กันไปทั้ง 5 ส่วน เพราะส่วนการแสดงผลทั้ง 3 ส่วนจะต้องถูกใช้ในส่วนควบคุมการเล่น ซึ่งเป็นส่วนการทำงานหลัก แต่ละส่วนมีรายละเอียดดังนี้

### 3.3.1 ส่วนแสดงผลภาพโดยใช้ DirectDraw

ส่วนนี้จะเราจะเขียน function ที่เกี่ยวกับ DirectDraw ไว้ใน module DDraw เพื่อให้ module อื่น ๆ เรียกใช้ การเรียกใช้ DirectDraw มีชนิดตัวแปรที่เกี่ยวข้องดังนี้

- 1) DirectX7 เป็น Class หลักใน DirectX จำเป็นต้องประกาศทุกครั้งประกอบด้วย function ที่ทำหน้าที่สร้าง object อื่นๆ ใน DirectX
- 2) DirectXDraw7 เป็น Class หลักของการใช้งาน DirectDraw ประกอบด้วย function ที่ทำหน้าที่สร้าง object อื่นๆ ใน DirectDraw
- 3) DirectDrawSurface7 เป็น Class ของ Surface ที่ใช้ในการเก็บรูปภาพและแสดงผล
- 4) DDSURFACEDESC2 เป็นชนิดที่เก็บ description ต่าง ๆ ของ Surface ใช้กับคำสั่ง DirectDraw7.CreateSurface
- 5) DDSCAPS2 เป็นชนิดตัวแปรที่จะนิยามการทำงานของ Surface
- 6) RECT เป็นชนิดตัวแปรของ Visual Basic ซึ่งจะเก็บค่าจุดมุมซ้ายบน และมุมขวาล่างของรูปสี่เหลี่ยมใช้ในการกำหนด source กับ destination ในการทำ BLT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน function ของ DirectDraw ที่ใช้ในเกมนี้นี้มีดังนี้

- 1) DX.DirectDrawCreate : สร้าง object ของ DirectDraw
- 2) DD.SetCooperativeLevel : กำหนดระดับในการติดต่อทำงานกับอุปกรณ์ผ่าน DirectX
- 3) DD.SetDisplayMode : กำหนด resolution และ ค่าสีที่จะใช้ในการแสดงผล
- 4) DD.CreateSurface : สร้าง Surface ภาพ
- 5) PrimarySurface.GetAttachedSurface : นำ Surface ที่สร้างไว้เป็น BackBuffer มาเชื่อมโยงกับ PrimarySurface
- 6) BackBuffer.SetFontTransparency : set ให้คำสั่ง DrawText แล้วพื้นหลังตัวหนังสือเป็น Transparency
- 7) DD.CreateSurfaceFromFile : สร้าง Surface จากไฟล์ภาพ โดยให้ Surface นั้นเก็บภาพจากไฟล์
- 8) Surface.SetColorKey : set สีที่จะทำ ColorKey ใน Surface นั้น
- 9) Surface.BlitFast : ทำการ Blit ภาพลงใน Surface โดยไม่มีการใช้ effect พิเศษใดๆ
- 10) Surface.Blit : ทำการ Blit ภาพลงใน Surface สามารถใช้ effect พิเศษได้
- 11) PrimarySurface.Flip : ทำการ Flip สลับ BackBuffer กับ PrimarySurface

หมายเหตุ กำหนดให้ DX คือ object ของ DirectX , DD คือ object ของ DirectDraw และ Surface คือ Surface ใดๆ

การเริ่มการติดต่อกับ DirectDraw ทำได้โดยสร้าง object ของ DirectX ก่อนแล้วจึงสร้าง object ของ DirectDraw โดยใช้คำสั่ง DirectDrawCreate จาก object ของ DirectX ต่อจากนั้นต้องทำการกำหนดระดับการติดต่อด้วยคำสั่ง SetCooperativeLevel และกำหนดโหมดการแสดงผลด้วยคำสั่ง SetDisplayMode หลังจากนั้นสร้าง PrimarySurface ซึ่งจะเป็น Surfaces ในการแสดงผลด้วยคำสั่ง CreateSurface และเตรียมสร้าง BackBuffer เพื่อใช้ในการ Flip ซึ่งเมื่อทำทุกอย่างเสร็จเรียบร้อยแล้วก็พร้อมที่จะใช้งาน DirectX

ในการใช้งานมีหลักการทำงานตามที่กล่าวไปแล้วในบทที่ 2 คือ เมื่อต้องการแสดงผลภาพจะต้องสร้าง Surface ขึ้นมารองรับภาพจากไฟล์ชนิด Bitmap แล้วนำไป blit ลงบน BackBuffer ก่อน จึงค่อยทำการ Flip สลับไปแสดงผลที่ PrimarySurface เพื่อการแสดงผลที่ต่อเนื่อง ถ้าทำการ blit ลงไปบน PrimarySurface โดยตรงจะทำให้การแสดงผลภาพกระพริบไม่ต่อเนื่อง การ flip ภาพไปเรื่อย ๆ ยังสามารถทำให้เกิด effect การกระพริบของภาพที่ต้องการได้ นอกจากนี้ในการ blit ยังสามารถใส่ effect การเล่นเกมต่าง ๆ เพิ่มเติมได้

ส่วนการแสดงผลนี้จะทำงานเพื่อตอบสนองกับ Event ที่ได้รับมาจะผู้เล่น โดยเป็นการแสดงผลภาพเพื่อทำงานตามผลที่ได้จะการประมวลผล Event เสียเป็นส่วนใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ส่วนแสดงผลเสียงดนตรีในรูปแบบ Midi โดยใช้ DirectMusic

ส่วนนี้จะเราจะเขียน function ที่เกี่ยวกับ DirectMusic ไว้ใน module DMusic เพื่อให้ module อื่น ๆ เรียกใช้ การเรียกใช้ DirectMusic มีชนิดตัวแปรที่เกี่ยวข้องดังนี้

- 1) DirectMusic เป็น object หลักของ DirectMusic
- 2) DirectMusicLoader ใช้ในการค้นหาและอ่านข้อมูลจากไฟล์ที่เป็นชนิด MIDI
- 3) DirectMusicPerformance ใช้ทำการติดต่อกับอุปกรณ์เพื่อทำการแสดงผลเสียง MIDI
- 4) DirectMusicSegment ใช้เก็บส่วนเล็ก ๆ ของเสียงดนตรีที่กำลังจะถูกส่งไปแสดงผล
- 5) DirectMusicSegmentState ใช้ตรวจสอบสถานะของ Segment ที่กำลังแสดงผลอยู่

ส่วน function ของ DirectMusic ที่ใช้ในเกมนี้มีดังนี้

- 1) DX.DirectMusicLoaderCreate : สร้าง object ของ DirectMusicLoader
- 2) DX.DirectMusicPerformanceCreate : สร้าง object ของ DirectMusicPerformance
- 3) DMP.Init : เป็น method ที่ทำการตั้งค่าการทำงานและทำงานสัมพันธ์ DirectSound
- 4) DMP.SetPort : ตั้งค่า port ให้กับการทำงาน โดยต้องทำก่อนการใช้งาน DirectMusicPerformance
- 5) DMP.SetMasterAutoDownload : ใช้เปิด - ปิด automatic downloading of Instrument
- 6) DMS = DM.LoadSegment : ใช้ loadSegment จากไฟล์มาส่ง DirectMusicSegment
- 7) DMS.SetStandardMidiFile : บอก DirectMusic ว่า Segment นั้นเป็นไฟล์ MIDI แบบมาตรฐาน
- 8) DMS.SetRepeats : ตั้งค่าการเล่นวนซ้ำ
- 9) DMState = DMP.PlaySegment : แสดงผลเสียงจาก Segment ที่กำหนด
- 10) DMP.Stop : หยุดการแสดงผลเสียงที่กำลังแสดงอยู่

หมายเหตุ กำหนดให้ DMP คือ DirectMusicPerformance , DM คือ DirectMusic , DMS คือ DirectMusicSegment และ DMState คือ DirectMusicSegmentState

การทำงานของ DirectMusic จะเริ่มจากสร้าง object ของ DirectMusicLoader กับ DirectMusicPerformance แล้วตั้งค่าต่าง ๆ ให้กับ DirectMusicPerformance เพื่อให้พร้อมที่จะทำการแสดงผลดังนี้

- Call DMP.Init(Nothing, hdl)
- Call DMP.SetPort(-1, 1)
- Call DMP.SetMasterAutoDownload(True)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วเมื่อจะทำการแสดงผลให้ load ไฟล์เข้าสู่ Segment โดยใช้คำสั่ง DMS = DM.LoadSegment แล้วบอกประเภทของไฟล์ MIDI ในที่นี้ คือ MIDI Standard Format จึงใช้ DMS.SetStandardMidiFile แล้วใช้คำสั่ง Call DMS.SetRepeats(999) เพื่อบอกให้ทำการเล่นวนซ้ำ 999 ครั้ง แล้วสั่งเล่น MIDI โดยใช้คำสั่ง DMState = DMP.PlaySegment โดยให้ DirectMusicSegmentState สถานะเป็นกำลังเล่นอยู่ เมื่อต้องการหยุดการเล่น MIDI ใช้คำสั่ง DMP.Stop

### 3.3.3 ส่วนแสดงผลเสียงในรูปแบบ Wave File Format โดยใช้ DirectSound

ส่วนนี้จะเราจะเขียน function ที่เกี่ยวกับ DirectSound ไว้ใน module DSound เพื่อให้ module อื่น ๆ เรียกใช้ การเรียกใช้ DirectSound มีชนิดตัวแปรที่เกี่ยวข้องดังนี้

- 1) DirectSound เป็น object หลักของ DirectSound
- 2) DirectSoundBuffer เป็น Buffer ที่ใช้เก็บและแสดงเสียงไฟล์ WAV
- 3) DSBUFFERDESC เป็นตัวแปรที่ใช้ระบุค่าต่าง ๆ ที่ buffer ปลายทาง
- 4) WAVEFORMATEX เป็นตัวแปรที่ใช้บอก format ของ WAV file

ส่วน function ของ DirectSound ที่ใช้ในเกมนี้นี้มีดังนี้

- 1) DX.DirectSoundCreate : สร้าง object ของ DirectSound
- 2) DS.SetCooperativeLevel : กำหนดระดับการทำงานติดต่อกับอุปกรณ์
- 3) DS.CreateSoundBufferFromFile : สร้าง object ของ DirectSoundBuffer เพื่อใช้รับกับไฟล์เสียง
- 4) SoundBuffer.Play : เรียก SoundBuffer นั้นๆ ให้ทำการเล่นเสียง
- 5) SoundBuffer.Stop : ใช้หยุดเล่นเสียงใน SoundBuffer ที่สั่ง
- 6) SoundBuffer.SetCurrentPosition : ใช้ตั้งค่าตำแหน่งของเสียงที่ต้องการแสดง

หมายเหตุ กำหนดให้ DS คือ object ของ DirectSound และ SoundBuffer คือ object ของ DirectSoundBuffer

การทำงานกับ DirectSound เริ่มด้วยการสร้าง object ของ DirectSound กำหนดระดับการทำงานติดต่อกับ function DS.SetCooperativeLevel แล้วทำการกำหนดเสียงให้กับ SoundBuffer ด้วยคำสั่ง DS.CreateSoundBufferFromFile โดยจะต้องระบุ WAV format ด้วย เมื่อต้องการจะเรียกไฟล์เสียงนั้นแสดงผล ใช้ function SoundBuffer.Play แล้วเมื่อต้องการจะหยุดเล่นให้เรียก function SoundBuffer.Stop

### 3.3.4 ส่วนการรับ input จากผู้เล่น

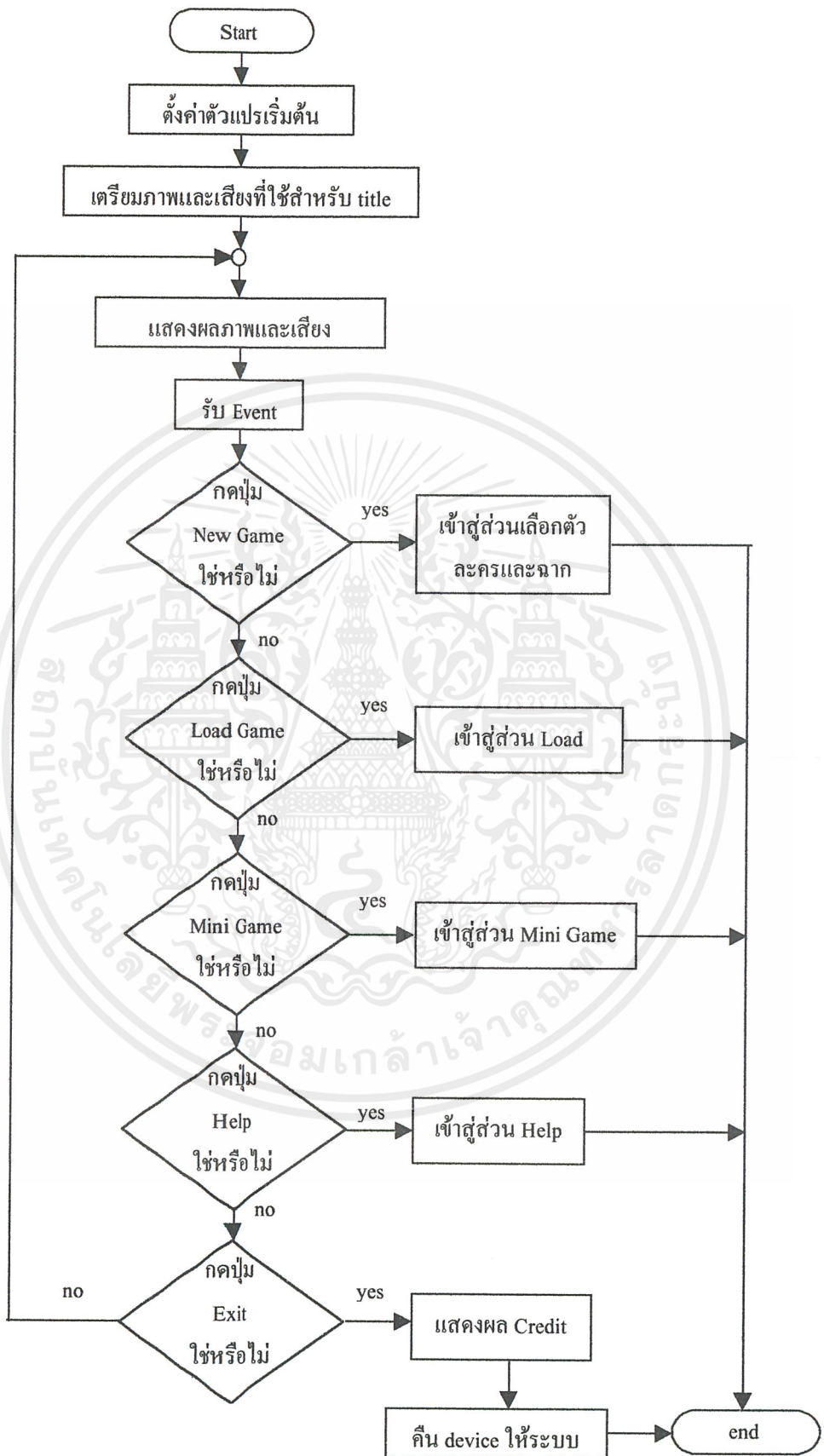
การรับ input ในเกมนี้จะใช้ MouseUp Event ของ Visual Basic เพื่อให้ใช้กับจอร์บบ สัมผัสได้ ( เนื่องจาก DirectX ไม่สามารถทำงานกับจอร์บบสัมผัสได้ ทำให้ไม่สามารถนำ Event ไปใช้ได้โดยตรง จึงเลือกใช้ MouseUp Event ของ Visual Basic แทน ) โดยรับ event จาก form แล้วให้ DirectX ตรวจสอบว่ามี Event หรือไม่เมื่อพบคำสั่ง DoEvents ถ้ามี Event จะเปลี่ยนค่าตัวแปร boolean สำหรับ Check ให้เป็นจริง คือ มีการคลิก mouse แล้วคืนค่า X,Y ใน ตำแหน่งที่ถูกคลิกกลับไปให้โปรแกรมหลักทำการประมวลผล

### 3.3.5 ส่วนควบคุมการเล่นหลัก

ส่วนควบคุมการเล่นนี้เป็นส่วนที่ใหญ่ที่สุดของโปรแกรม และมีความสำคัญต่อการเล่น มากที่สุด โดยส่วนนี้จะเขียนเป็นแบบ Procedural ซึ่งในที่นี้จะขอแสดงเป็นแผนผังการทำงาน ของโปรแกรมหาดังต่อไปนี้



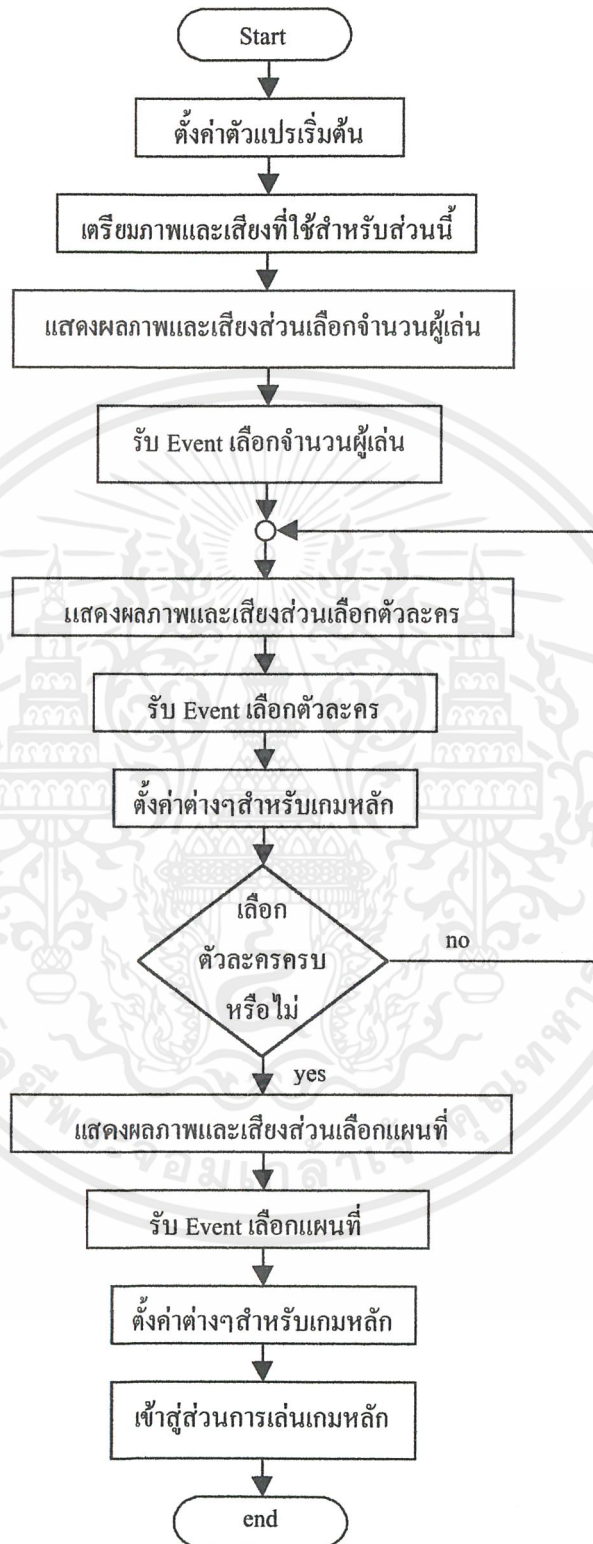
## 1) แผนผังการทำงานในส่วนของ title



รูปที่ 3.13 แผนผังการทำงานในส่วนของ title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

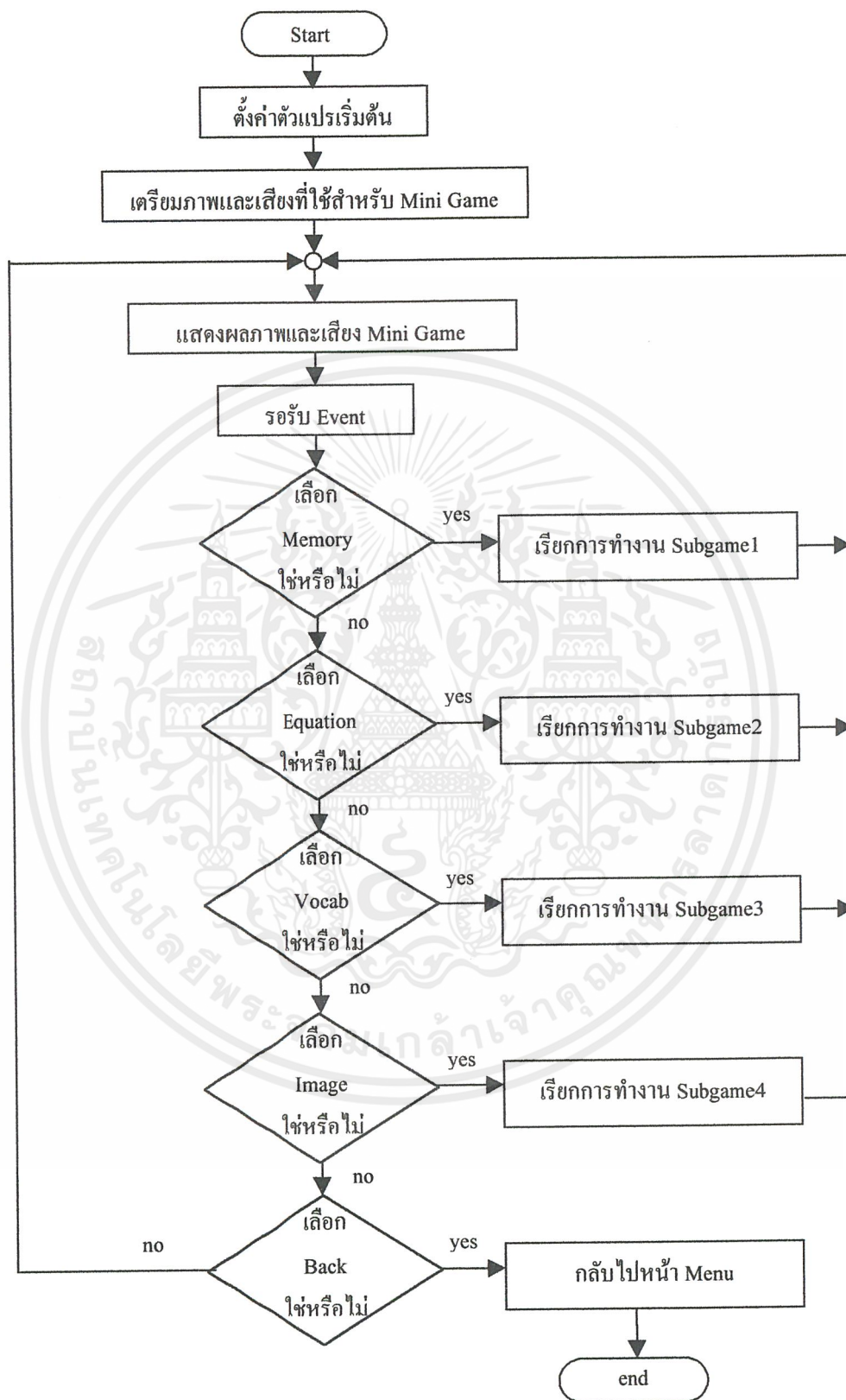
## 2) แผนผังการทำงานในส่วนของ Character Select



รูปที่ 3.14 แผนผังการทำงานในส่วนของ Character Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

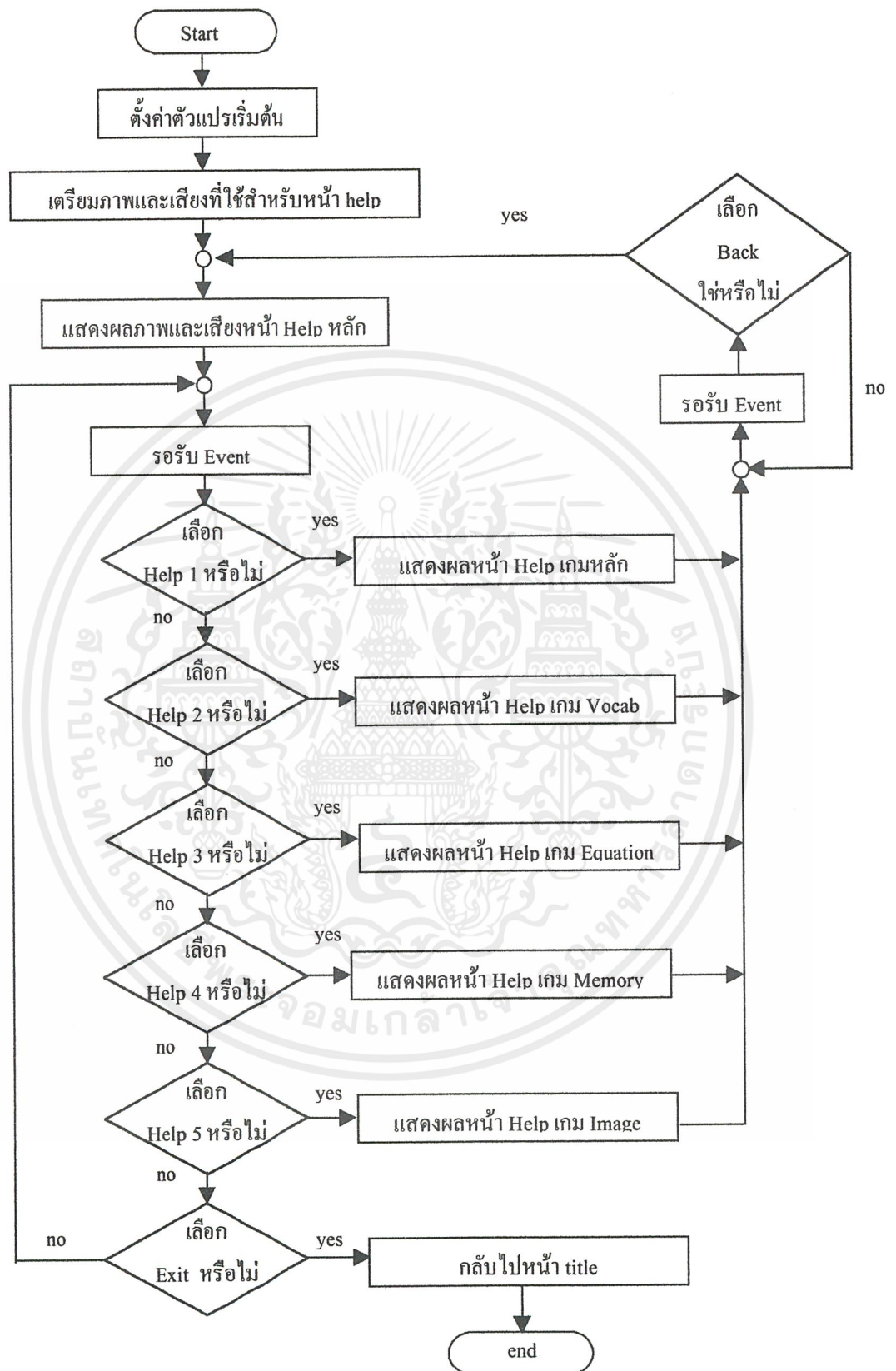
### 3) แผนผังการทำงานในส่วนของ Mini Game



รูปที่ 3.15 แผนผังการทำงานในส่วนของ Mini Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

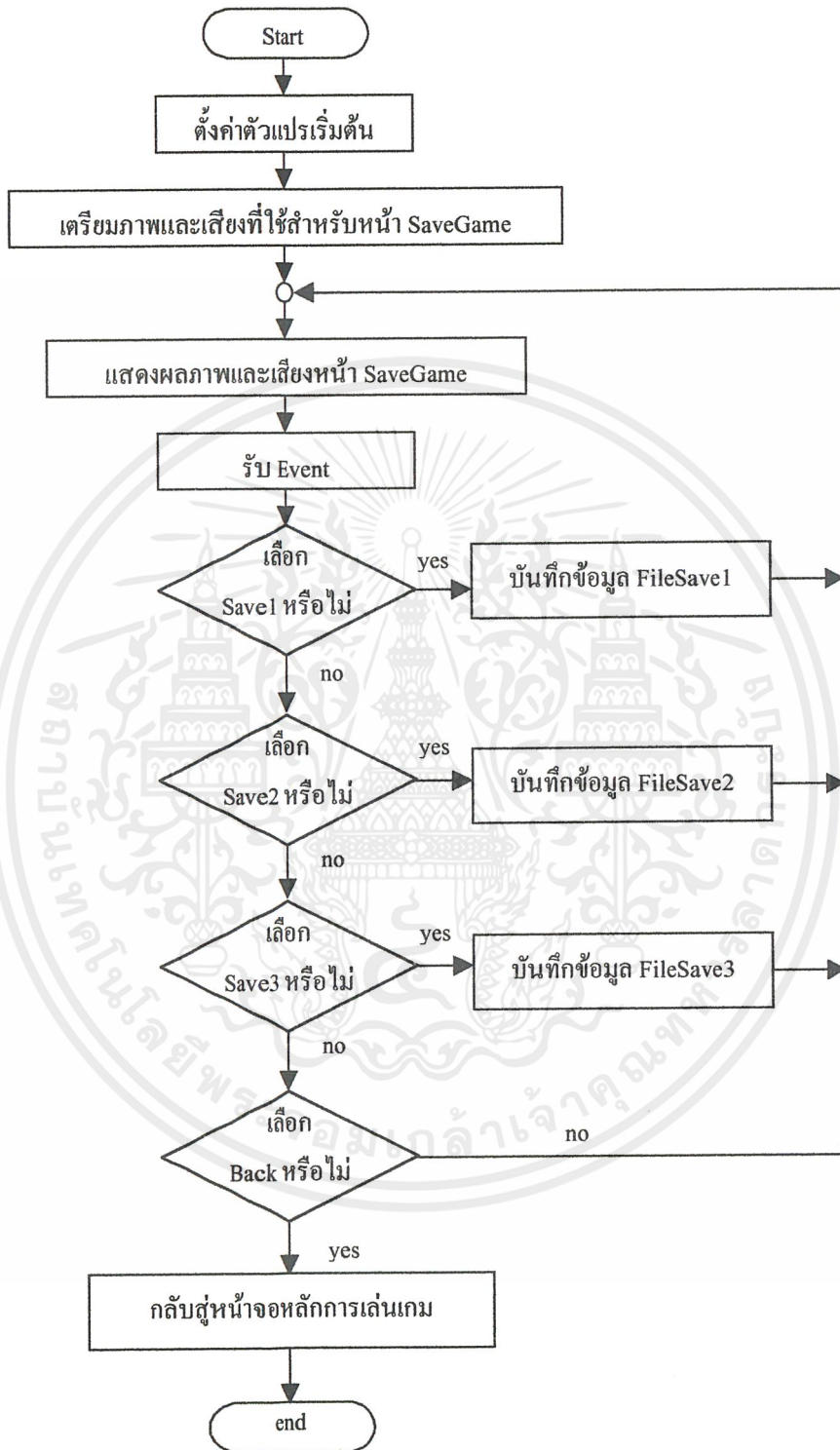
## 4) แผนผังการทำงานในส่วนของ Help



รูปที่ 3.16 แผนผังการทำงานในส่วนของ Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

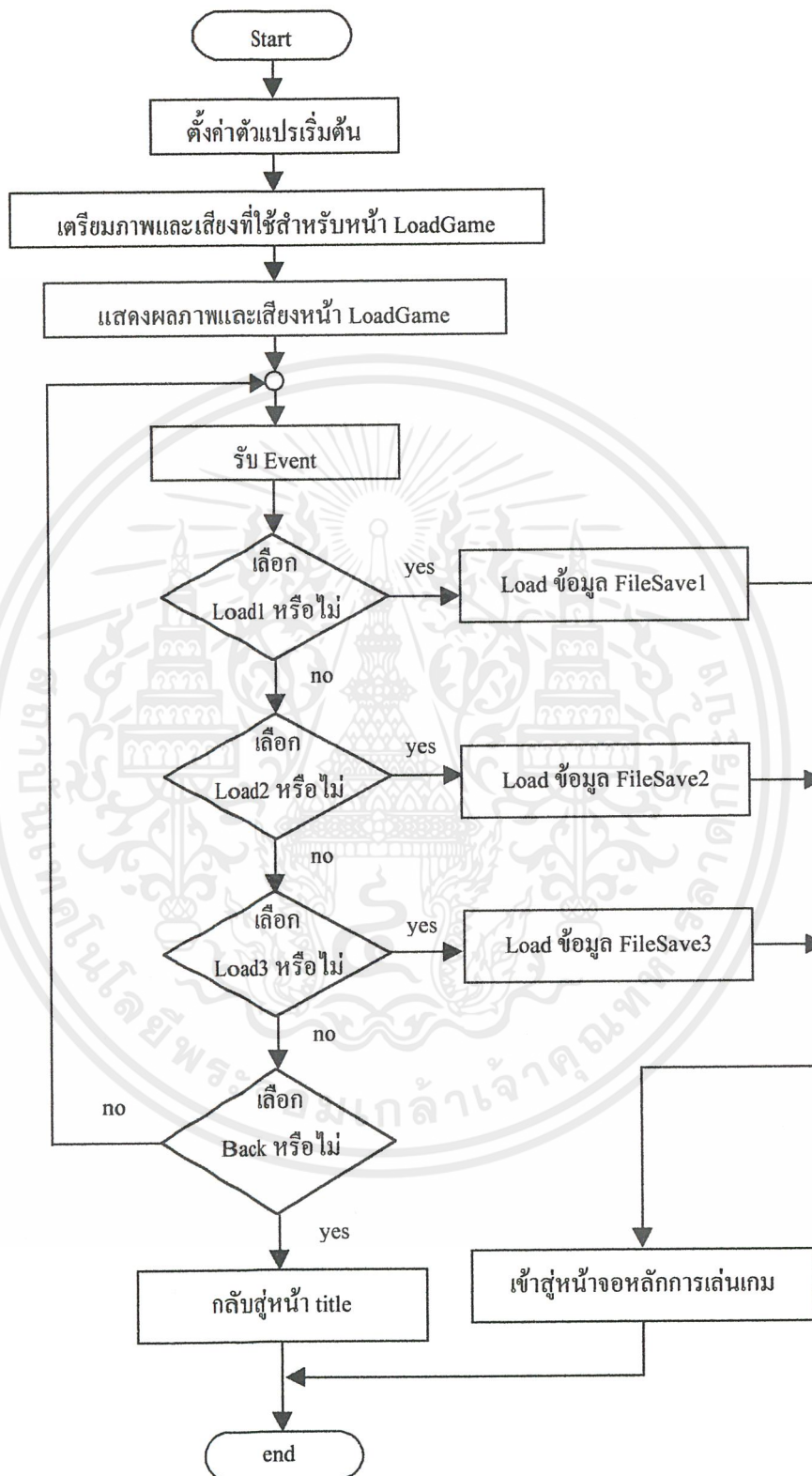
## 5) แผนผังการทำงานในส่วนของ Save Game



รูปที่ 3.17 แผนผังการทำงานในส่วนของ Save Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

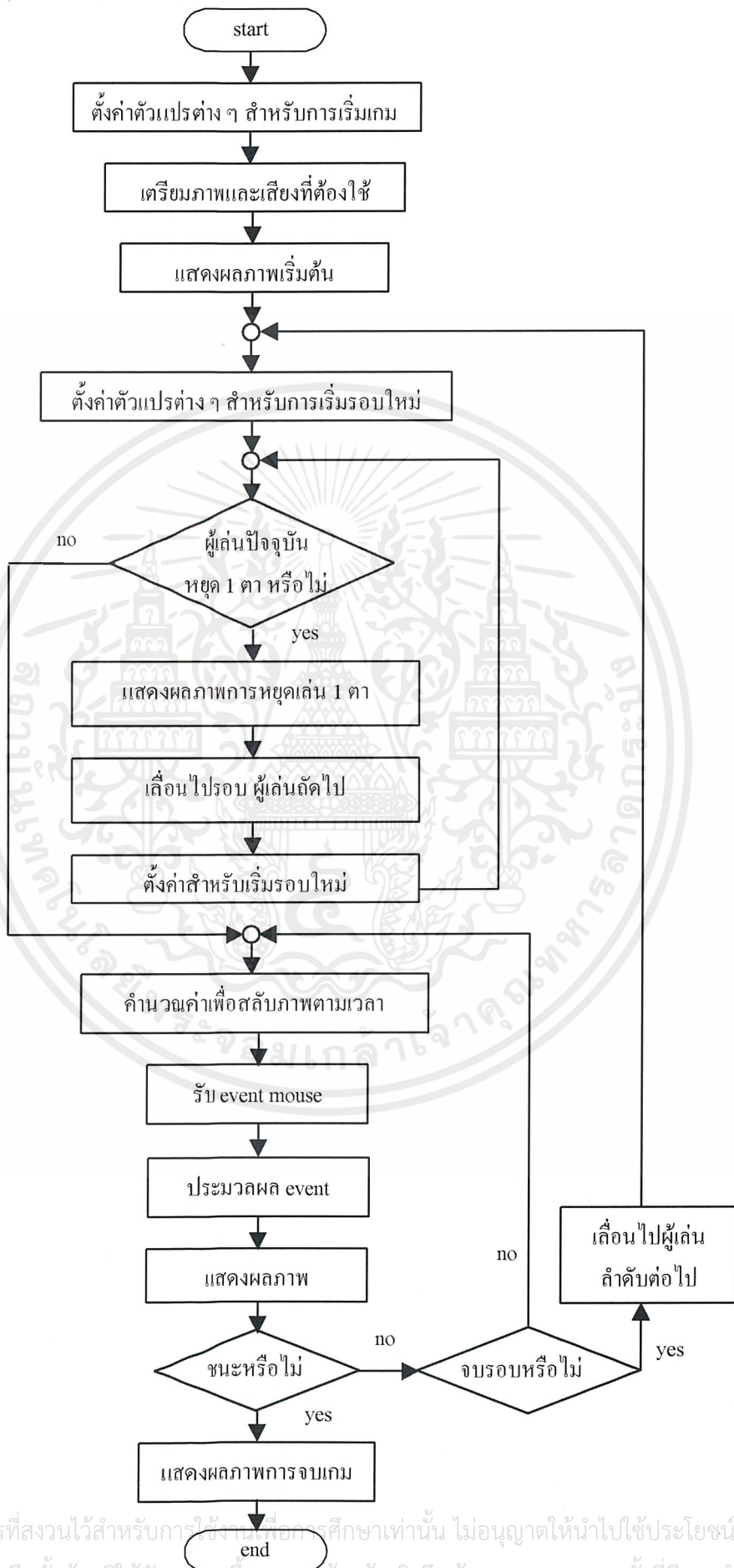
### 6) แผนผังการทำงานในส่วนของ Load Game



รูปที่ 3.18 แผนผังการทำงานในส่วนของ Load Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

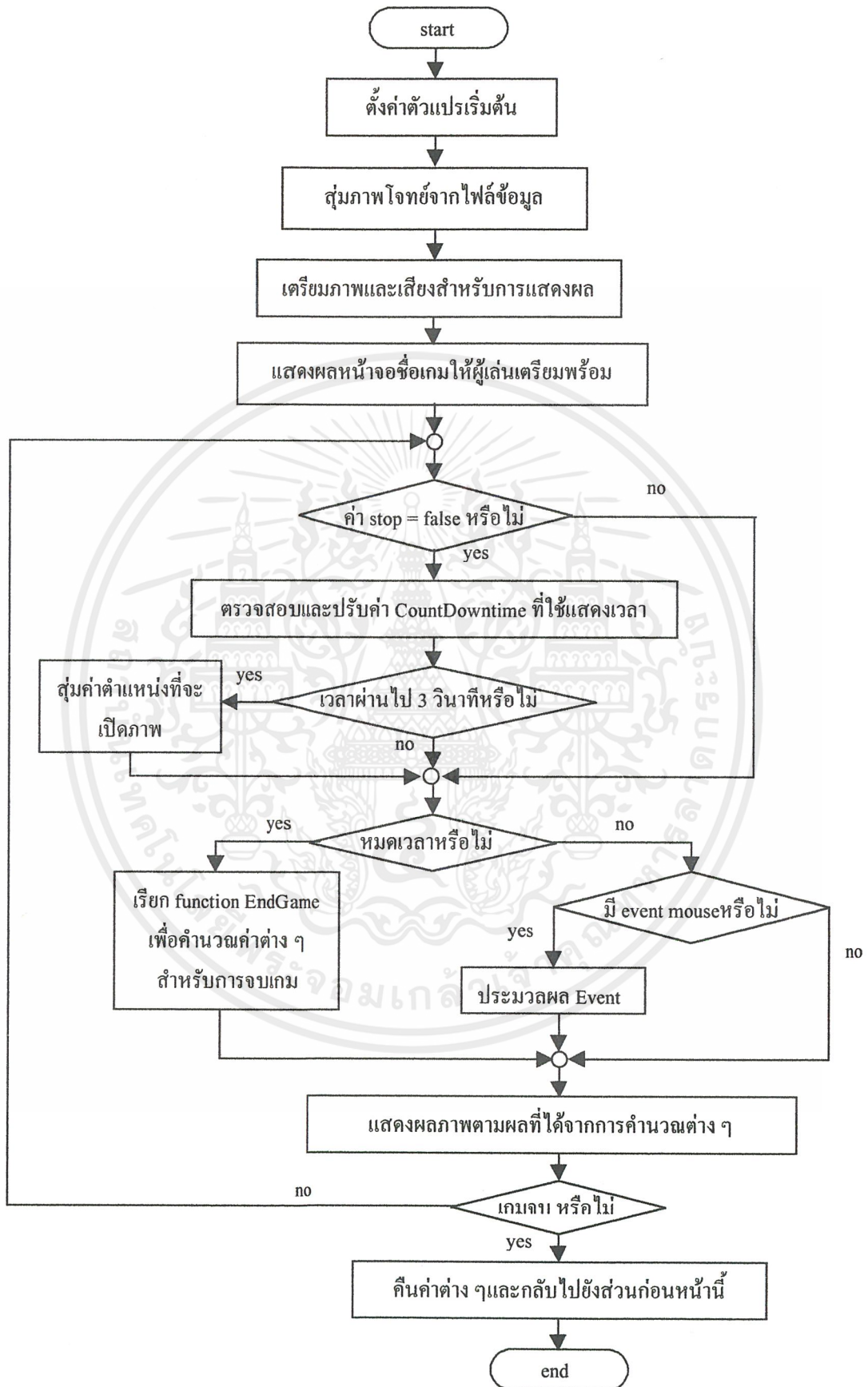
7) แผนผังการทำงานในส่วนของเกมหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 แผนผังการทำงานในส่วนของเกมหลัก

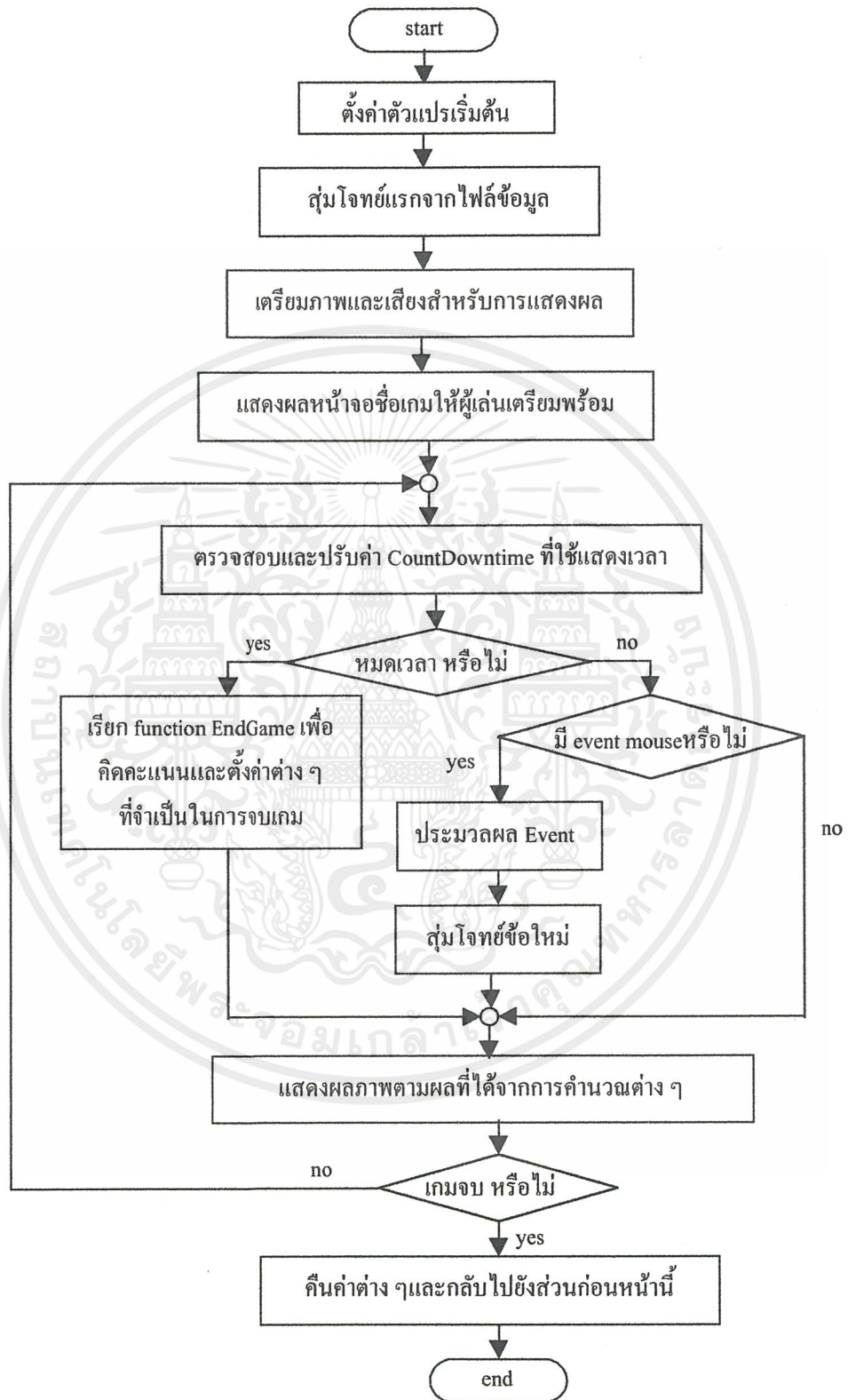
## 8) แผนผังการทำงานของเกมย่อย Magic Image



รูปที่ 3.20 แผนผังการทำงานของเกมย่อย Magic Image

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

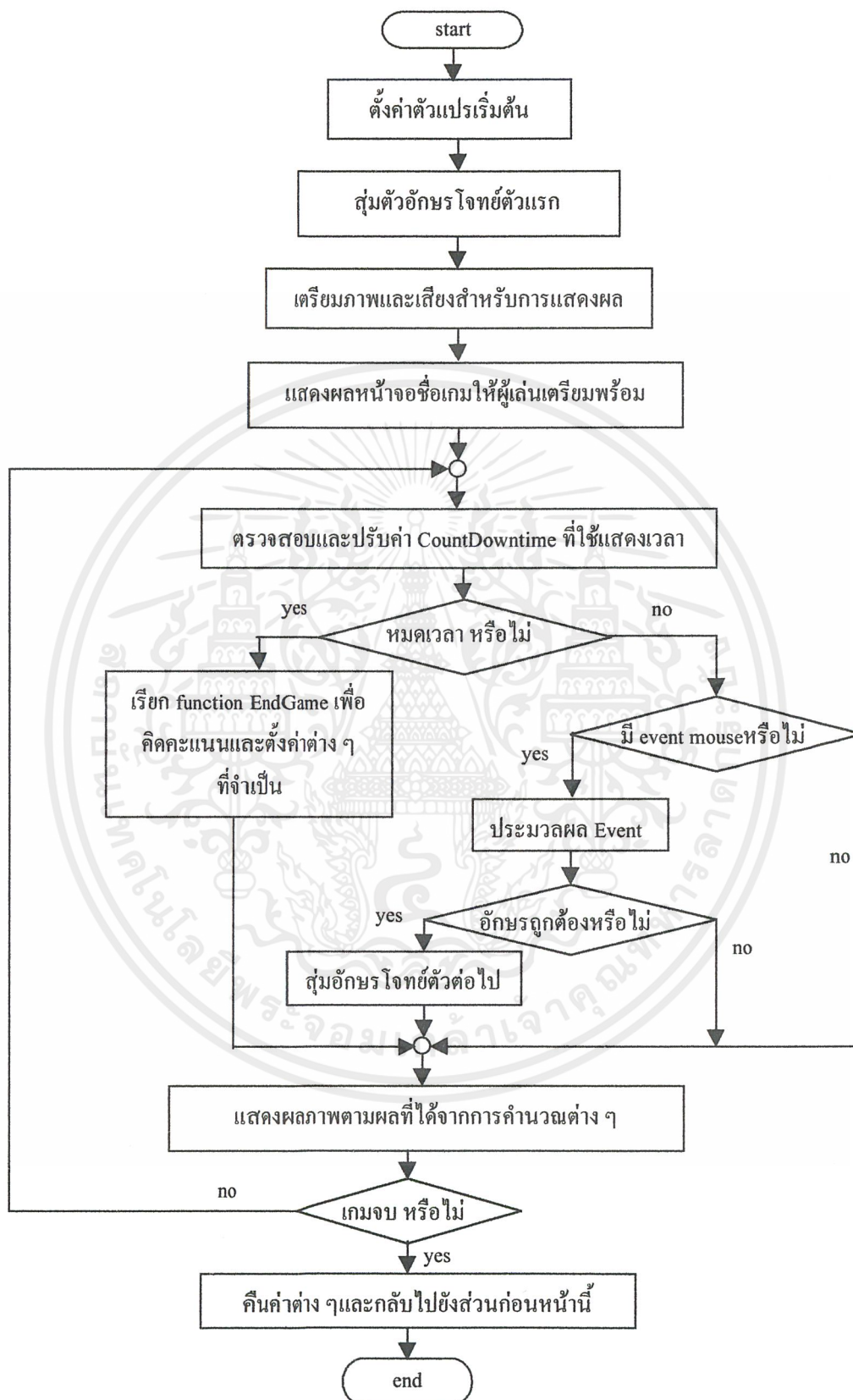
### 9) แผนผังการทำงานของเกมย่อย Magic Equation



รูปที่ 3.21 แผนผังการทำงานของเกมย่อย Magic Equation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

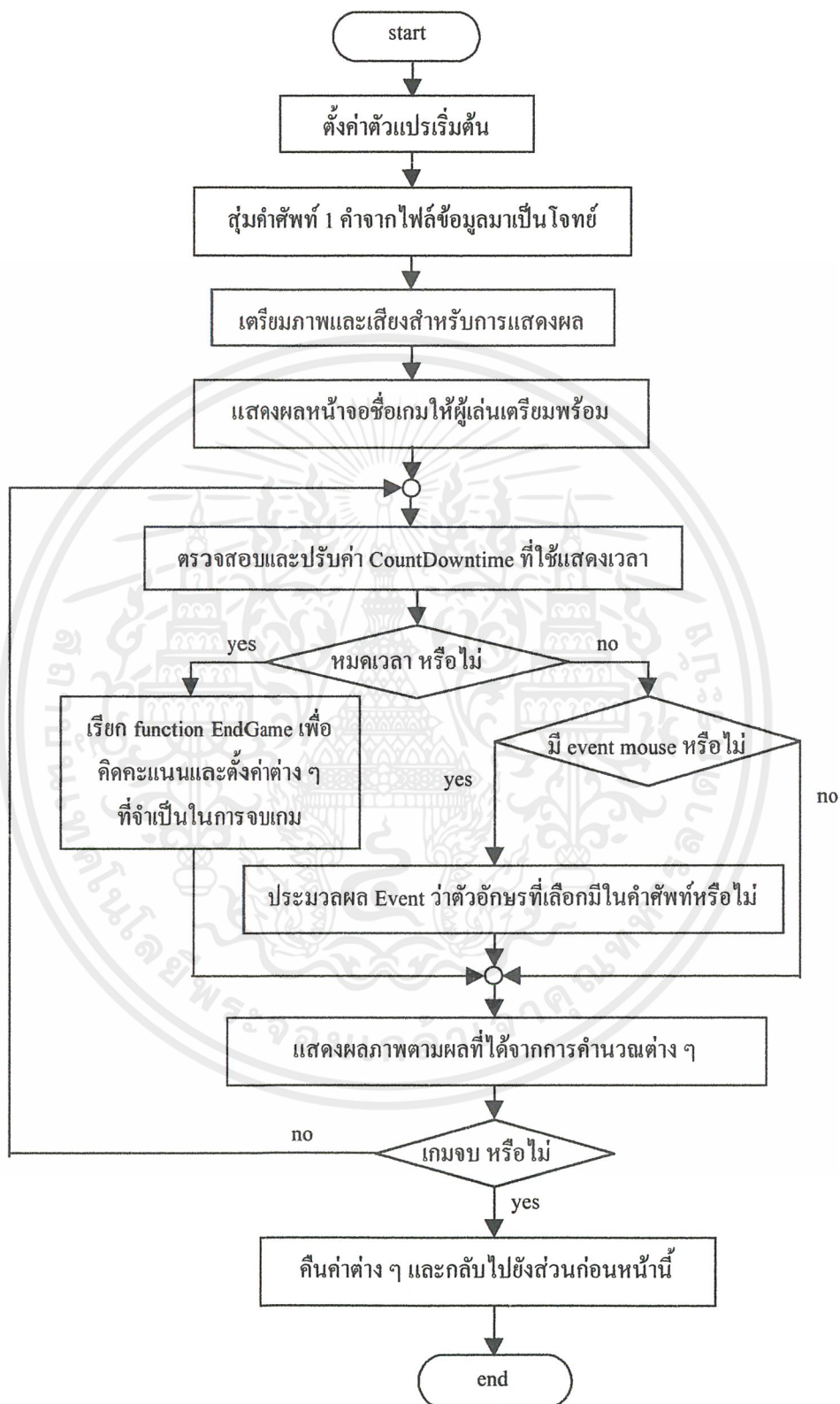
## 10) แผนผังการทำงานของเกมย่อย Magic Memory



รูปที่ 3.22 แผนผังการทำงานของเกมย่อย Magic Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 11) แผนผังการทำงานของเกมย่อย Magic Vocab



รูปที่ 3.23 แผนผังการทำงานของเกมย่อย Magic Vocab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลองและการวิเคราะห์ปัญหา

การทดลองและการวิเคราะห์ปัญหาที่กล่าวถึงในบทนี้ จะเป็นขั้นตอนการทดสอบและผลการทดสอบที่ได้ในแต่ละขั้น โดยผลการทดสอบนี้ จะถูกนำไปวิเคราะห์ถึงปัญหาและแนวทางในการพัฒนาต่อไปในอนาคต โดยทางผู้จัดทำหวังว่าจะเป็นประโยชน์แก่ผู้ที่นำไปศึกษาเพื่อพัฒนาต่อ และเพื่อให้เห็นถึงปัญหา และข้อดีข้อเสียของปัญหานี้โดยที่ผู้ศึกษาไม่จำเป็นต้องทำการทดสอบเพื่อหาปัญหาและนำไปพัฒนาต่ออีกครั้งหนึ่ง

#### คุณสมบัติของระบบที่นำมาทดสอบ

- 1) ระบบปฏิบัติการ Microsoft Windows 98
- 2) เครื่องคอมพิวเตอร์ที่มีซีพียู Pentium III MMX 500 MHz
- 3) หน่วยความจำหลักขนาด 128 MB
- 4) VDO Card ของ Matrox MGA G200 AGP
- 5) Sound Card ของ Crystal SoundFusion(tm)

#### ขั้นตอนการดำเนินการทดสอบ

- 1) ทำการติดตั้งตัวโปรแกรมและซอฟต์แวร์ที่จำเป็นต้องใช้
- 2) ทำการรันและประมวลผลภายใต้ระบบที่กำหนด
- 3) ตรวจสอบการใช้คำสั่งและการทำงาน
- 4) ตรวจสอบการหา Error

#### การประเมินผล

- 1) หลังจากติดตั้งตัวโปรแกรมแล้วสามารถใช้โปรแกรมได้
- 2) การใช้คำสั่งและการทำงานต้องเป็นไปตามที่กำหนดไว้
- 3) จำนวน Error ที่เกิดขึ้นจะต้องไม่มี หรือน้อยที่สุด

ต่อไปจะกล่าวถึงรายละเอียดของขั้นตอนการทดสอบและผลการทดสอบที่ได้

#### 4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็น

- 1) ทำการติดตั้งโปรแกรมและซอฟต์แวร์
- 2) ผู้ใช้ทำการรัน โปรแกรมได้โดยคลิกที่ไฟล์ “ setup.exe ” ใน folder “ Install ”

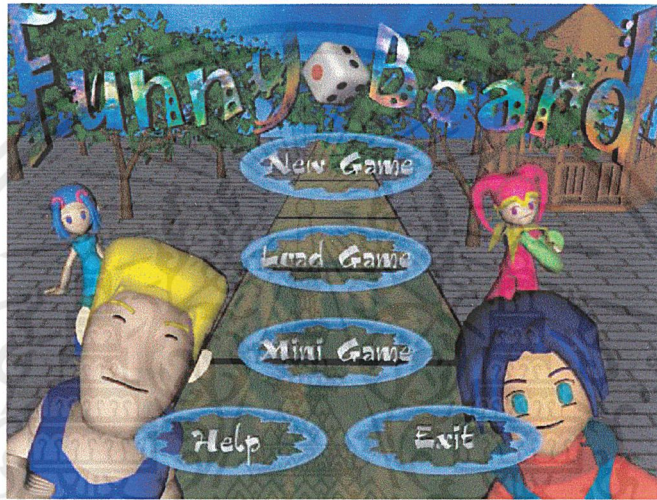
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ผลการทดสอบ

- 1) ติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็นลงบนเครื่องคอมพิวเตอร์ได้
- 2) ผู้ใช้สามารถรันโปรแกรมได้โดยการคลิกที่ไอคอน ซึ่งจะแสดงหน้าจอของเกมขึ้นมา

## 4.2 ขั้นตอนการทดสอบการรันและประมวลผลภายใต้ระบบที่กำหนด

รันโปรแกรมภายใต้เครื่องคอมพิวเตอร์ที่กำหนด โดยคลิกที่ไฟล์ “Funny Board.exe ” จะแสดงหน้าจอ Title ดังรูปที่ 4.1



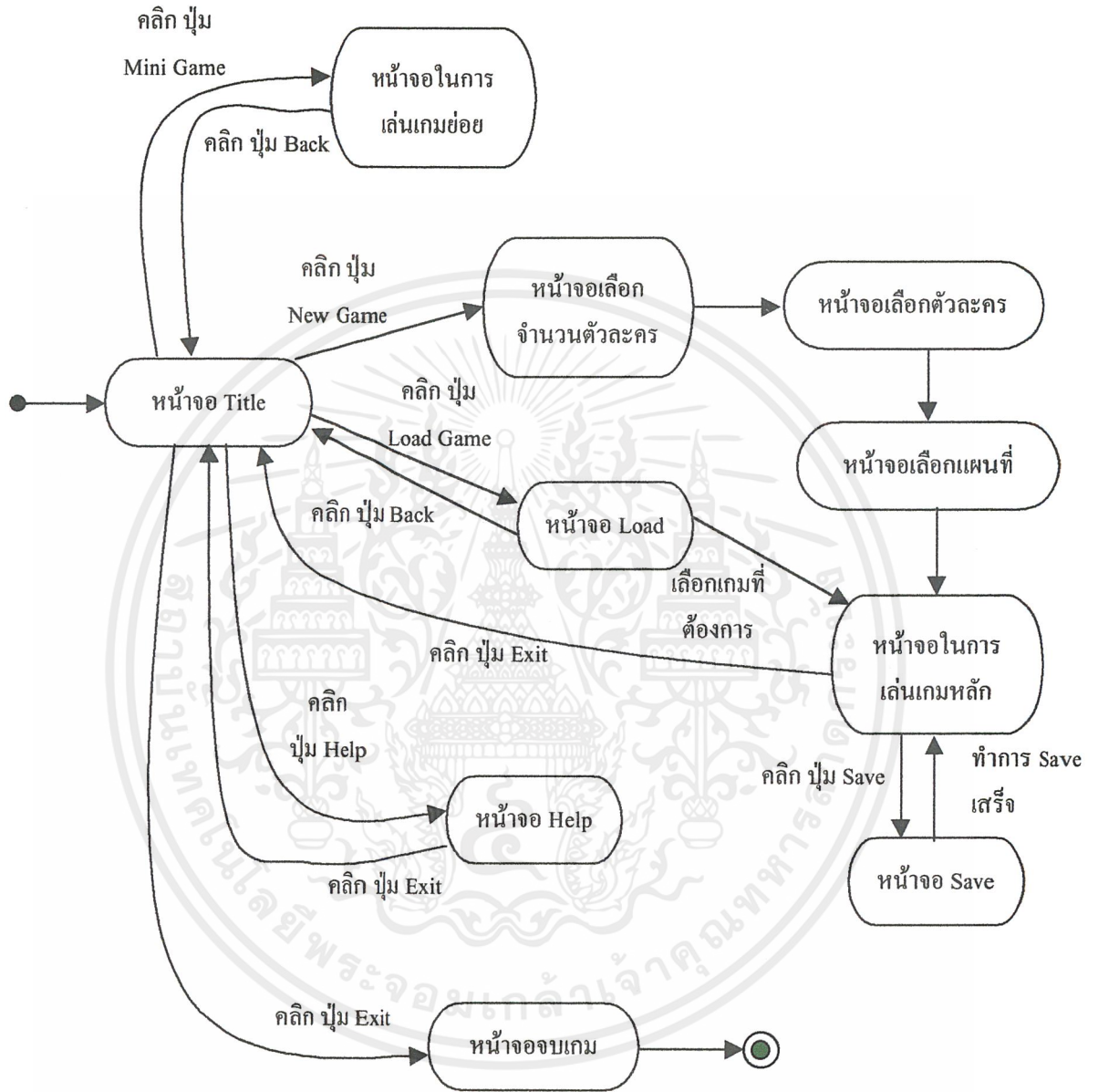
รูปที่ 4.1 หน้าจอTitleของโปรแกรม

โดยมีปุ่มต่างๆ ดังนี้

- ปุ่ม New Game จะเป็นการเข้าสู่การเล่นเกมนหลัก
- ปุ่ม Load Game จะเป็นการโหลดเกมที่ save ไว้กลับมาเล่นต่อ
- ปุ่ม Mini Game จะเป็นการเข้าไปสู่การเล่นเกมนย่อยต่าง ๆ
- ปุ่ม Help จะเป็นการเข้าไปสู่หน้าคู่มือ วิธีการเล่นเกม Funny Board
- ปุ่ม Exit จะเป็นการออกจากโปรแกรม

หลังจากนั้นทำการทดสอบการทำงานของโปรแกรมโดยการคลิก mouse ที่ปุ่มต่าง ๆ

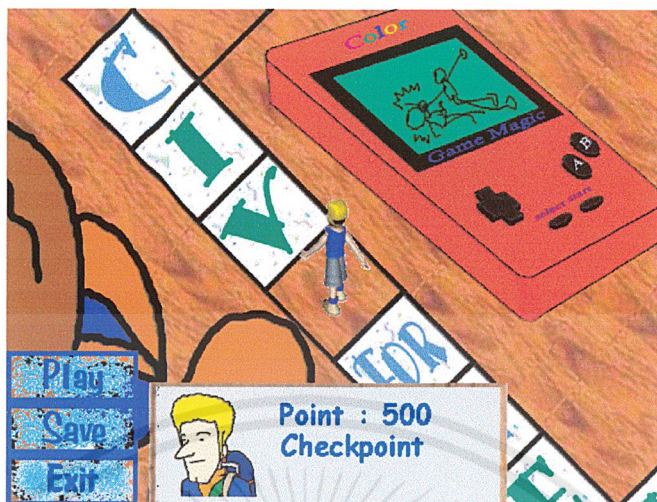
ผลการทดลอง คือ โปรแกรมสามารถทำงานได้ตามที่ต้องการ และมีลำดับในการแสดงของหน้าจอเป็นดังนี้



รูปที่ 4.2 แผนผังแสดงลำดับการแสดงผลหน้าจอของโปรแกรมเกมกระดานมหาสนุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ขั้นตอนการทดสอบการทำงานในส่วนของเกมหลัก



รูปที่ 4.3 หน้าจอแผนที่ในการเล่นเกมนหลัก

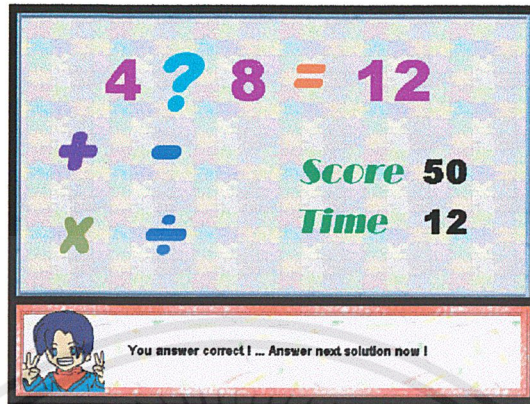
ตารางที่ 4.1 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกมหลัก

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การแสดงผลการเดินของตัวละคร	คลิก mouse ที่ปุ่ม Play	ตัวละครเดินตามทางในแผนที่และเป็นจำนวนช่องตามที่ทอยลูกเต๋าได้อย่างถูกต้อง
การคำนวณคะแนนที่ได้ของผู้เดิน	ทำการรันโปรแกรมหลาย ๆ ครั้ง แล้วพิจารณากรณีต่าง ๆ ที่ทำให้เกิดการเปลี่ยนแปลงของคะแนน	โปรแกรมสามารถคำนวณคะแนนได้อย่างถูกต้อง
การทำตามคำสั่งในแต่ละช่องของแผนที่	ทำการรันโปรแกรมหลาย ๆ ครั้ง แล้วพิจารณาว่าโปรแกรมทำงานตรงตามที่แต่ละช่องในแผนที่กำหนดไว้หรือไม่	โปรแกรมสามารถทำงานได้ตรงตามที่กำหนดไว้ในแต่ละช่องของแผนที่
การทำงานของปุ่ม Play	คลิก mouse ที่ปุ่ม Play	แสดงผลภาพเต็มลูกเต๋าตามที่ทอยได้
การทำงานของปุ่ม Save	คลิก mouse ที่ปุ่ม Save	เปลี่ยนหน้าจอเป็นหน้าจอ Save
การทำงานของปุ่ม Exit	คลิก mouse ที่ปุ่ม Exit	กลับสู่หน้าจอ Title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 ขั้นตอนการทดสอบการทำงานของเกมย่อย

### 4.4.1 เกม Magic Equation

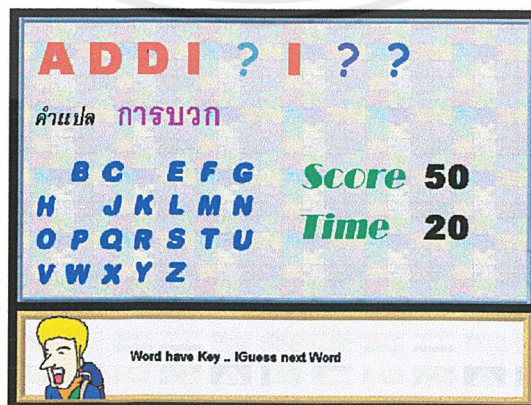


รูปที่ 4.4 หน้าจอการเล่นเกม Magic Equation

### ตารางที่ 4.2 วิธีการทดสอบและผลการทดสอบการทำงานของส่วนของเกม Magic Equation

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การตอบคำถาม ได้ถูกต้อง	คลิก mouse ที่คำตอบที่ถูกต้อง	Score ถูกบวกเพิ่มทีละ 10
การตอบคำถามผิด	คลิก mouse ที่คำตอบที่ผิด	Time จะถูกลบ 3
การแสดงข้อความ	ทำการเล่นเกมนี้ให้เกิดกรณีต่าง ๆ	แสดงข้อความได้สอดคล้องกับกรณีต่าง ๆ ที่เกิดขึ้น
การจบเกมจากการหมดเวลา	ทำการเล่นเกมจนกระทั่งเวลาเป็น 0	คะแนนที่ได้จะเท่ากับ Score
การจบเกมจากการเล่นครบ 10 คำถาม	ทำการเล่นให้ครบ 10 คำถามก่อนที่เวลาจะหมด	คะแนนที่ได้จะเท่ากับ Score + ( 10 x Time )

### 4.4.2 เกม Magic Vocab



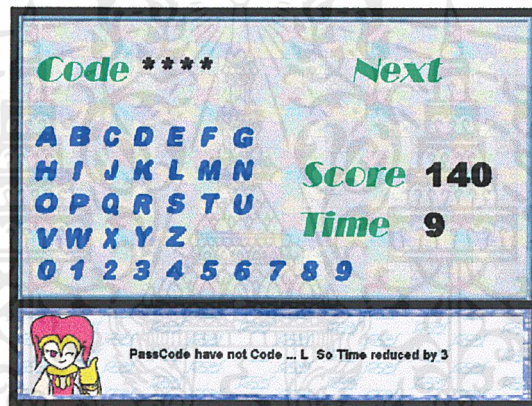
รูปที่ 4.5 หน้าจอการเล่นเกม Magic Vocab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Vocab

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การทายตัวอักษรที่มีในคำศัพท์	คลิก mouse ที่ตัวอักษรที่ถูก	Score = Score + ( 10 x จำนวนตัวอักษรที่มีอยู่ในคำศัพท์ )
การทายตัวอักษรที่ไม่มีในคำศัพท์	คลิก mouse ที่ตัวอักษรที่ผิด	Time จะถูกลบ 3
การแสดงข้อความ	ทำการเล่นเกมนี้ให้เกิดกรณีต่าง ๆ	แสดงข้อความได้สอดคล้องกับกรณีต่าง ๆ ที่เกิดขึ้น
การจบเกมจากการหมดเวลา	ทำการเล่นจนกระทั่งเวลาเป็น 0	คะแนนที่ได้จะเท่ากับ Score
การจบเกมจากการทายคำศัพท์ได้ถูกต้องก่อนเวลาหมด	ทำการทายคำศัพท์ให้ถูกก่อนที่เวลาจะหมด	คะแนนที่ได้จะเท่ากับ Score + ( 10 x Time )

#### 4.4.3 เกม Magic Memory



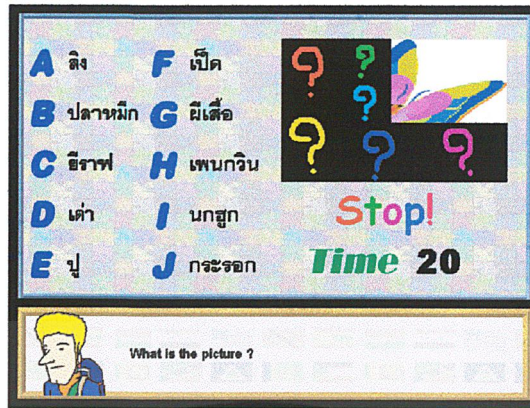
รูปที่ 4.6 หน้าจอการเล่นเกม Magic Memory

ตารางที่ 4.4 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Memory

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การคลิก code ที่ถูกต้อง	คลิก mouse ที่ตัวอักษรตามที่ตัวอักษรนั้นปรากฏ	Score เพิ่มขึ้นทีละ 10 ตามจำนวน code แต่ละตัวที่ตอบถูก
การคลิก code ที่ผิด	คลิก mouse ที่ตัวอักษรอื่นที่ตัวอักษรนั้นไม่ได้ปรากฏ	Time จะถูกลบ 3
การแสดงข้อความ	ทำการเล่นเกมนี้ให้เกิดกรณีต่าง ๆ	แสดงข้อความได้สอดคล้องกับกรณีต่าง ๆ ที่เกิดขึ้น
การจบเกมจากการหมดเวลา	ทำการเล่นจนกระทั่งเวลาเป็น 0	คะแนนที่ได้จะเท่ากับ Score + ( 10 x ชุดของรหัสที่ทำได้สมบูรณ์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.4 เกม Magic Image



รูปที่ 4.7 หน้าจอการเล่นเกม Magic Image

#### ตารางที่ 4.5 วิธีการทดสอบและผลการทดสอบการทำงานในส่วนของเกม Magic Image

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การถ่ายภาพได้ถูกต้อง	คลิก mouse ที่คำตอบที่ถูกต้อง	คะแนนที่ได้เท่ากับ Time x 10
การถ่ายภาพผิด	คลิก mouse ที่คำตอบที่ถูกต้อง	คะแนนที่ได้เท่ากับ 0
การแสดงข้อความ	ทำการเล่นเกมนี้ให้เกิดกรณีต่าง ๆ	แสดงข้อความได้สอดคล้องกับกรณีต่าง ๆ ที่เกิดขึ้น

#### 4.5 ขั้นตอนการทดสอบการ Save และ Load เกม

วิธีการทดสอบการ Save เกม ทำได้โดยการเลือก Save เกม ณ เวลาใด ๆ ที่ต้องการ  
ผลการทดสอบ คือ โปรแกรมสามารถทำการ Save เกม ได้ทั้งหมด 3 เกม

วิธีการทดสอบการ Load เกม ทำได้โดยเลือก Load เกมใด ๆ ที่ต้องการ  
ผลการทดสอบ คือ สามารถเล่นเกมต่อจากครั้งที่ได้ทำการ Save ไว้

#### 4.6 ขั้นตอนการทดสอบการหาข้อผิดพลาดของโปรแกรม

ในขั้นตอนการทดสอบเพื่อหาข้อผิดพลาดจะไม่มีรูปแบบที่ชัดเจน แต่จะทำการทดสอบโดยการเล่นเกมหลาย ๆ รอบเพื่อหาข้อผิดพลาด และจากการทดสอบในบางครั้งพบข้อผิดพลาดเกี่ยวกับการแสดงผลเสียงไฟล์ midi โดยเสียงที่แสดงอยู่ในขณะนั้นจะเกิดการแสดงผลเสียงค้างไว้ แต่ความผิดพลาดที่เกิดขึ้นนี้ไม่ได้ส่งผลกระทบต่อการทำงานส่วนอื่น ๆ ของโปรแกรมแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.7 ปัญหาและการวิเคราะห์

จากการทดสอบโปรแกรมทำให้ทราบถึงปัญหา ดังนี้

1) การรับ event จากการคลิก mouse

เนื่องจากการทำงานที่ไม่สอดคล้องกันของ Microsoft VisualBasic กับ DirectX จึงทำให้ต้องมีเคลีย event ที่ตกค้างทุกครั้งก่อนที่จะทำการรับ event ถัดไป

2) การเคลื่อนไหวของภาพ animation

เนื่องจากสร้างภาพเคลื่อนไหวต่าง ๆ เหล่านั้นเพียงไม่กี่เฟรม จึงทำให้ได้ภาพเคลื่อนไหวที่ไม่สมจริงมากนัก แต่ก็เป็นการประหยัดทรัพยากรที่ต้องใช้ในการรันโปรแกรม



## บทที่ 5

# สรุปผลการศึกษาและข้อเสนอแนะ

### 5.1 สรุปผลการศึกษา

#### 5.1.1 การศึกษารวบรวมข้อมูล

การศึกษารวบรวมข้อมูลทำโดยการศึกษาความเป็นไปได้ของเกมว่าสามารถที่จะพัฒนาด้วยการเขียนโปรแกรมให้เป็นเกมบนเครื่องคอมพิวเตอร์ได้หรือไม่ เมื่อตัดสินใจเลือกแล้ว ก็ทำการศึกษารวบรวมข้อมูลต่าง ๆ ของเกมนั้น ๆ เช่น กฎและกติกา เป็นต้น อีกทั้งยังต้องศึกษาค้นคว้าในเรื่องวิธีการต่าง ๆ ที่ใช้ในการเขียนโปรแกรมเกม ทั้งหนังสือ หรือศึกษาจากตัวเกมจริงที่มีอยู่ทั่วไป เพื่อที่จะนำเอาข้อมูลต่าง ๆ ที่ได้มานั้นมาเป็นเครื่องมือช่วยในการตัดสินใจและคิดค้นต่าง ๆ เช่น การออกแบบหน้าตาเกม , การออกแบบโครงสร้างและวิธีการของเกม และการเลือกเครื่องมือต่าง ๆ ที่จะใช้ในการพัฒนาโปรแกรมเกม เป็นต้น

#### 5.1.2 การวิเคราะห์และการออกแบบเกม

การออกแบบระบบเกมนั้นจะต้องใช้เวลาพอสมควร เนื่องจากจะต้องทำการออกแบบระบบเกมทั้งหมด ซึ่งระบบเกมที่ต้องออกแบบนั้นจะต้องสามารถใช้งานได้ง่าย โดยการนำข้อมูลต่าง ๆ ที่รวบรวมมาประกอบกับความรู้ความสามารถที่มีอยู่ หรือที่ได้มาจากการศึกษาและประสบการณ์ มาเป็นเครื่องมือช่วยในการวิเคราะห์หารูปแบบของเกมที่สามารถจะเป็นไปได้แล้วทำการวางโครงสร้างและออกแบบหน้าตา รวมทั้งวิธีการต่าง ๆ ของเกมและกำหนดขอบเขตเพื่อให้สามารถทำงานเสร็จตามระยะเวลาที่กำหนด แล้วจึงทำการแบ่งงานและมอบหมายหน้าที่ให้แต่ละคนในกลุ่มจัดการ

#### 5.1.3 การสร้างตัวละครและภาพต่าง ๆ

ขั้นตอนนี้สามารถทำควบคู่ไปได้กับขั้นตอนการเขียนโปรแกรม ซึ่งเกมกระดานมหาสนุกนี้ได้ใช้เครื่องมือหลายอย่างในการพัฒนาและสร้างภาพกราฟฟิกแบบต่าง ๆ รวมทั้งการสร้างภาพเคลื่อนไหวที่เกิดขึ้นภายในเกม เครื่องมือที่ใช้ในการสร้างภาพกราฟฟิก ได้แก่ โปรแกรมตกแต่งรูปภาพ Adobe Photoshop และโปรแกรม 3D Studio MAX โดยโปรแกรม 3D Studio Max นั้น สามารถสร้างภาพกราฟฟิกได้เสมือนจริงในแบบ 3 มิติ อีกทั้งยังสามารถกำหนดการเคลื่อนไหวของภาพให้สมจริง ด้วยเหตุนี้ การทำงานของเกมในส่วนกราฟิก จึงใช้ 3D Studio Max ในการสร้างภาพลักษณะ 3 มิติ ส่วนการออกแบบหรือตกแต่งภาพจะใช้โปรแกรม Adobe Photoshop ในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 5.1.4 การพัฒนาโปรแกรม

ขั้นตอนนี้จะใช้เวลานานที่สุดในงานวิจัย เนื่องจากเป็นเรื่องใหม่ซึ่งผู้วิจัยต้องทำศึกษานานพอสมควร จึงลงมือเขียนโปรแกรม ซึ่งระหว่างการเขียนโปรแกรมนั้น ผู้วิจัยจะต้องทำการทดสอบโปรแกรมที่เขียนขึ้นมาด้วย จึงทำให้ใช้เวลานานพอสมควร ซึ่งขั้นตอนการทำงานในส่วนนี้มีซอฟต์แวร์ที่ใช้พัฒนา คือ Microsoft Visual Basic 6.0 และ DirectX7

#### 5.2 ข้อเสนอแนะ

เนื่องจากเกมกระดานมหาสนุกนี้มีพื้นฐานมาจากเกมเดินกระดานโดยวิธีทอยลูกเต๋า ซึ่งมีกติกาและข้อปลีกย่อยมากมาย ไม่ว่าจะเป็นการทำตามคำสั่งในช่องต่าง ๆ หรือทิศทางในการเดินเมื่อถึงทางแยกต่าง ๆ อันจะทำให้โครงสร้างของเกมมีความซับซ้อนมากขึ้น เพื่อที่จะทำให้เกมนั้นครอบคลุมเงื่อนไขทั้งหมดที่มีอยู่ ด้วยเหตุนี้การออกแบบ algorithm ของเกมจึงต้องกระทำด้วยความรอบคอบ และต้องคอยตรวจสอบผลการทำงานของเกมไปทุกระยะ เพื่อตรวจหาข้อผิดพลาดที่อาจจะเกิดขึ้น นอกจากนี้ยังสามารถนำโปรแกรมนี้ไปทำการเพิ่มเติม แก้ไขดัดแปลงเงื่อนไข หรือกฎกติกาต่าง ๆ ของเกม ให้มีความเหมาะสมมากยิ่งขึ้นได้ตามต้องการ



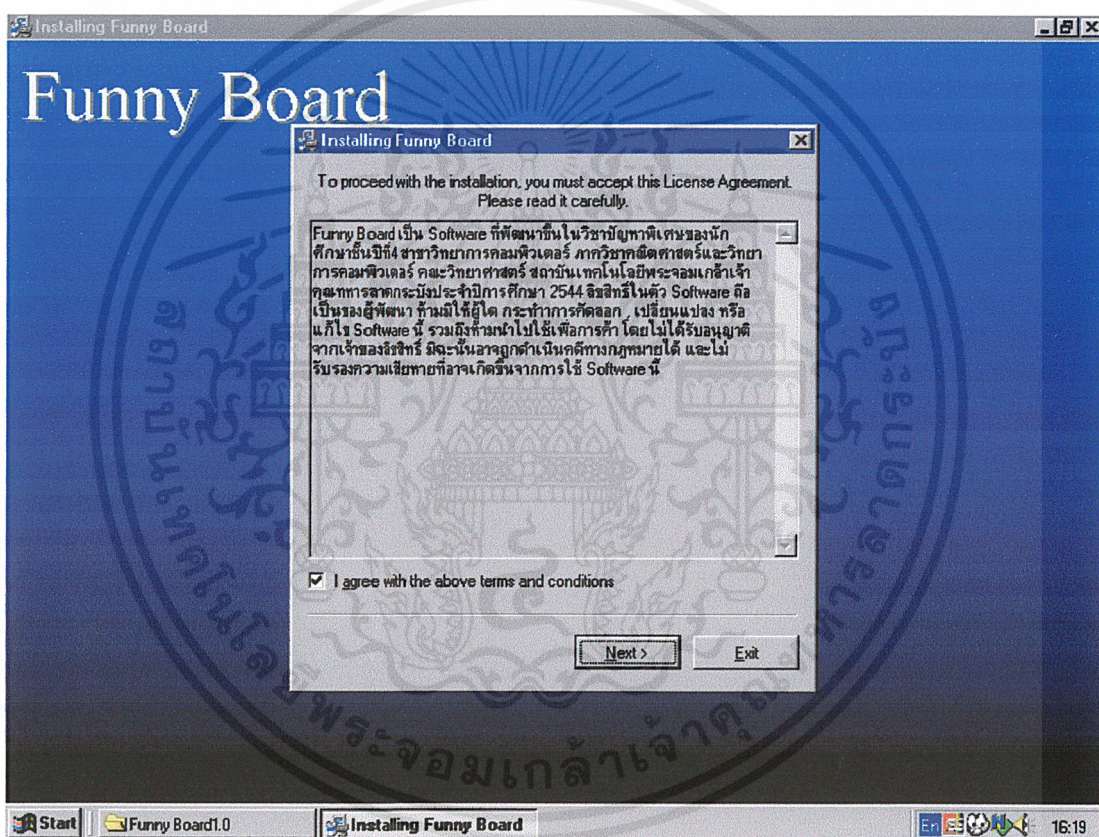
## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก. การติดตั้งและใช้งานโปรแกรม

### 1. ขั้นตอนการติดตั้งเกมกระดานมหาสนุก (Funny Board)

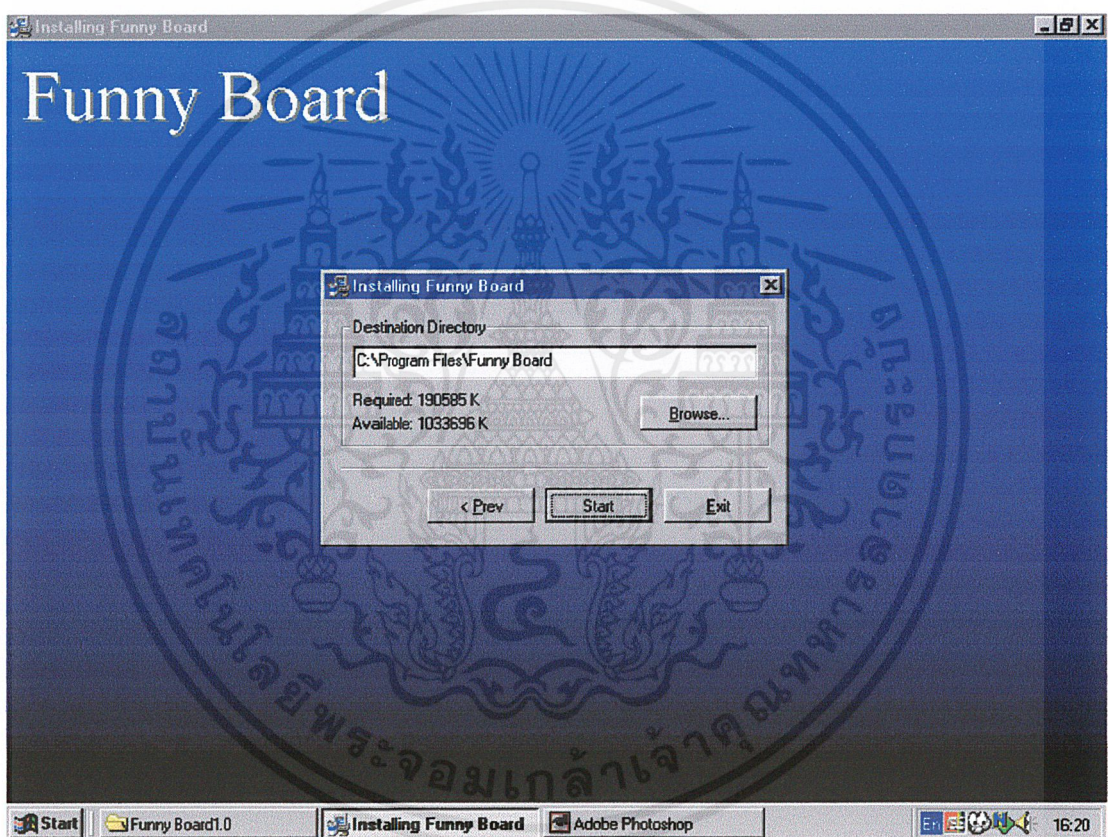
- 1.1 doubleclick ไฟล์ Setup.exe ในแผ่น CD จะเข้าสู่หน้าจอการเริ่มติดตั้งเกม
- 1.2 อ่านคำเตือนเกี่ยวกับลิขสิทธิ์แล้วคลิกที่ checkbox ด้านล่างเพื่อแสดงการยอมรับตามคำเตือนเมื่อคลิก checkbox แล้วให้กดปุ่ม next ด้านล่าง



รูปที่ ก-1 หน้าจอแสดงคำเตือนเกี่ยวกับลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

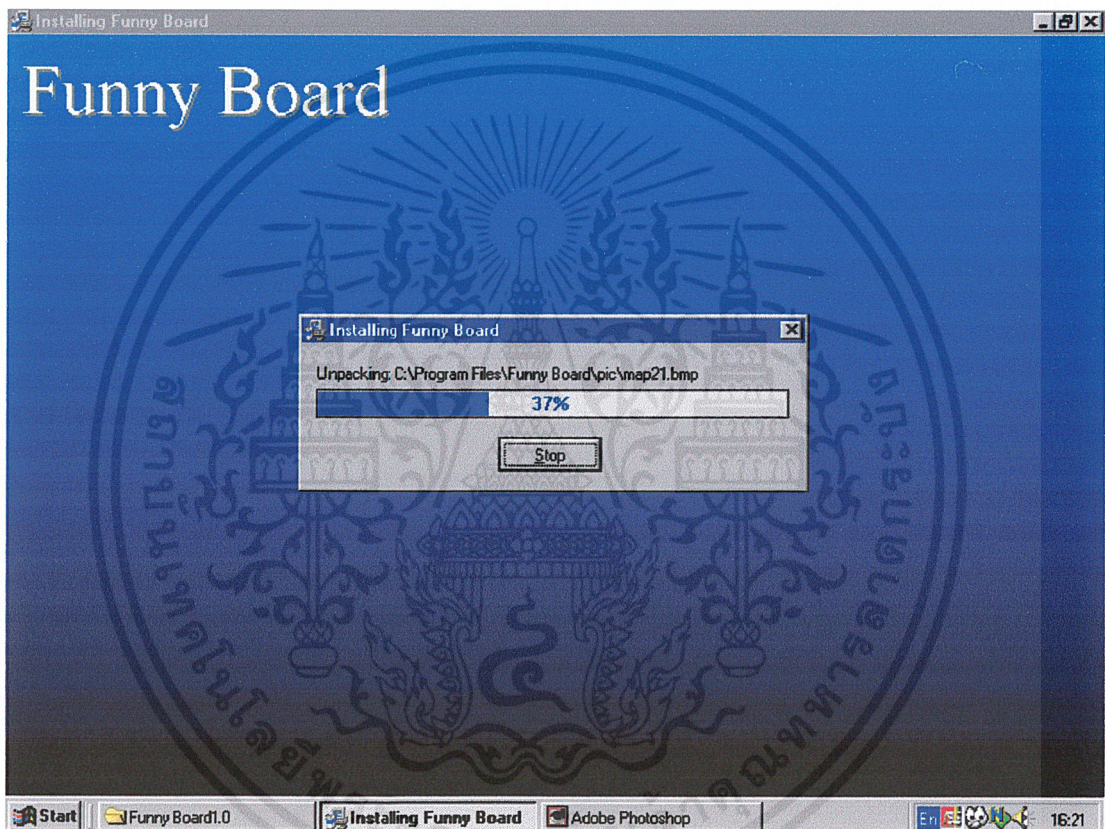
1.3 เลือก directory ที่ต้องการติดตั้งเกมกระดานมหาสนุกแล้วกดปุ่ม start ด้านล่างเพื่อเริ่มการติดตั้ง



รูปที่ ก-2 หน้าจอเลือก directory ที่ต้องการติดตั้งเกมกระดานมหาสนุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

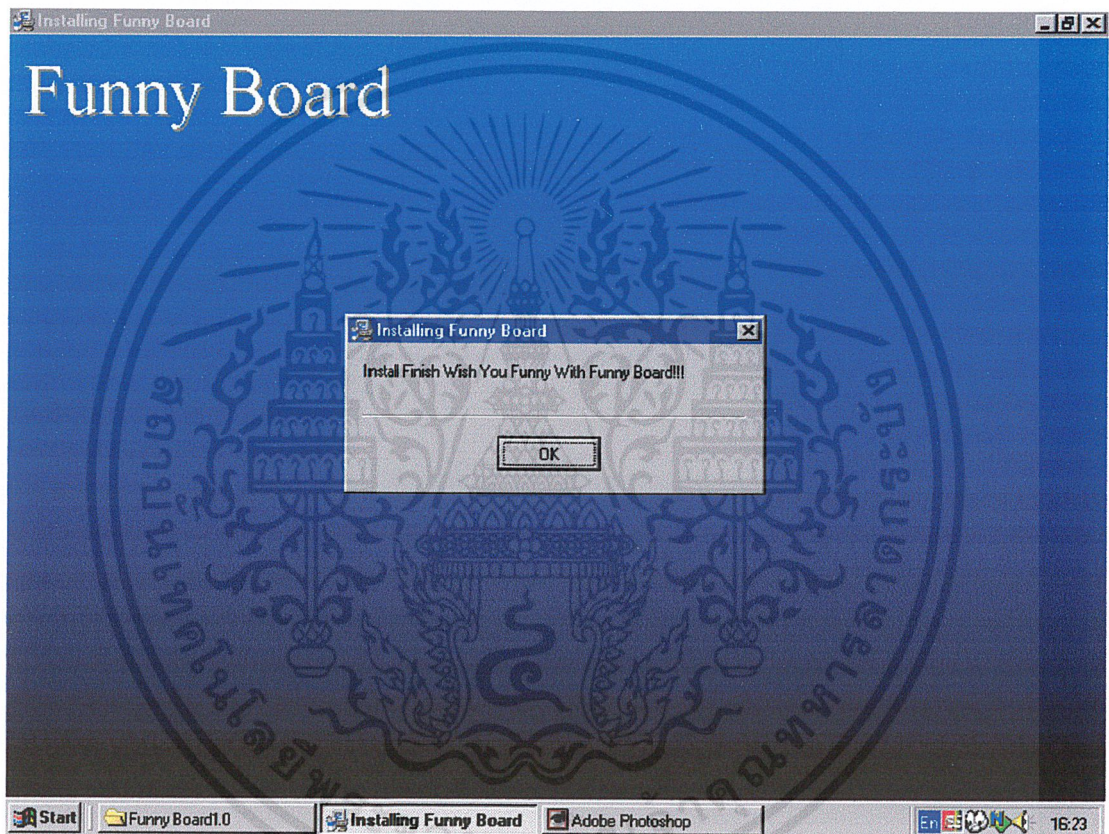
#### 1.4 โปรแกรมจะทำการติดตั้งเกมกระดานมหาสนุกลงใน directory ที่เลือกไว้



รูปที่ ก-3 หน้าจอขณะกำลังติดตั้งเกมกระดานมหาสนุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

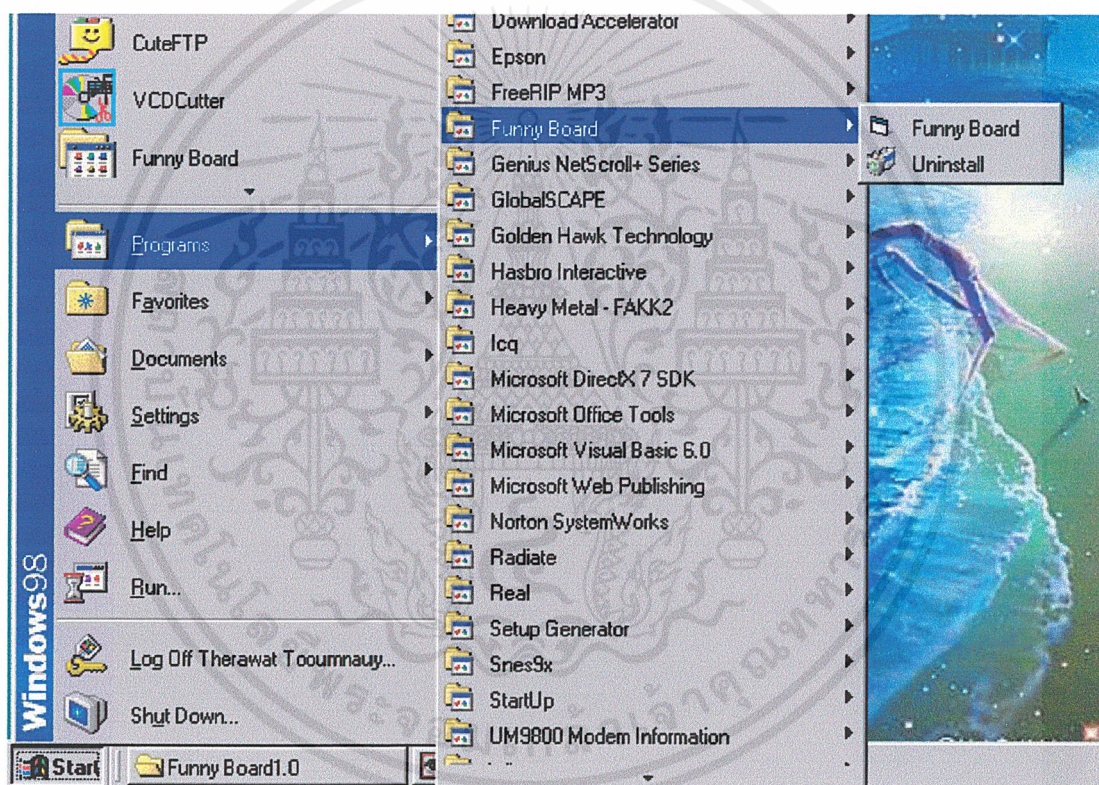
1.5 เมื่อโปรแกรมติดตั้งเสร็จเรียบร้อยแล้วคลิก ok เพื่อสิ้นสุดการติดตั้ง



รูปที่ ก-4 หน้าจอเมื่อติดตั้งเกมกระดานมหาสนุกเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

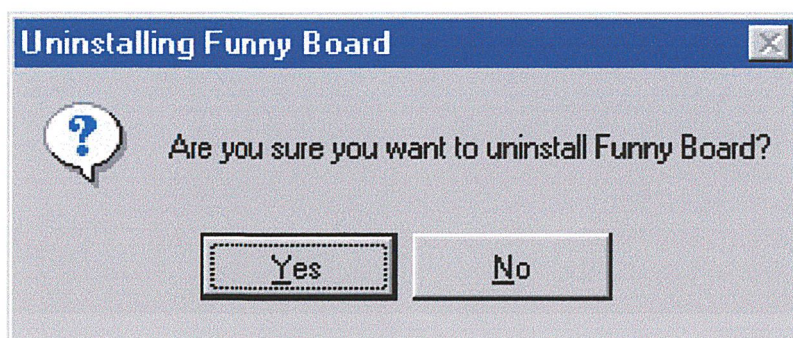
1.6 หลังจากติดตั้งแล้วสามารถเข้าไปเล่นเกมได้ที่ start menu → Program → Funny Board → Funny Board และถ้าต้องการถอดเกมกระดานมหาสนุกออกให้เลือก uninstall จากที่เดียวกัน



รูปที่ ก-5 การเลือกเล่นเกมกระดานมหาสนุก

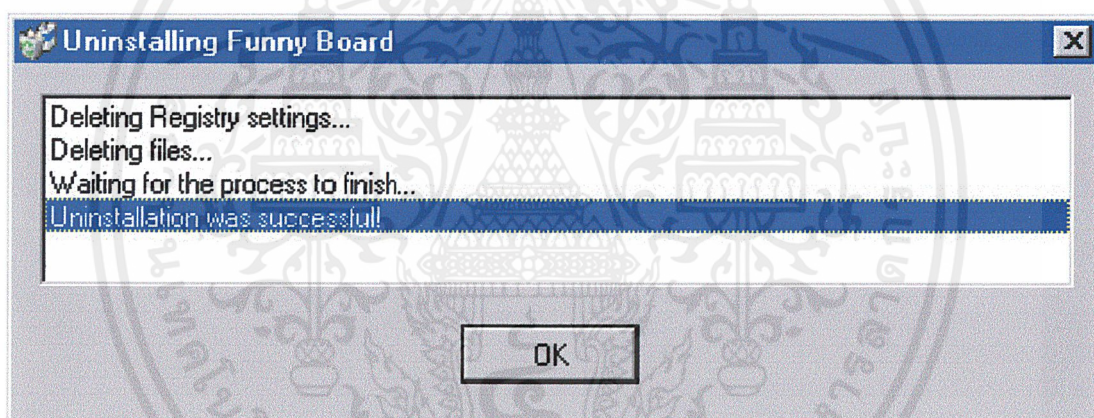
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 โปรแกรมจะถามยืนยันขั้นถ้าต้องการถอดเกมกระดานมหาสนุกออกจากระบบ กด Yes



รูปที่ ก-6 หน้าจอยืนยันขั้นการถอดเกมกระดานมหาสนุกออกจากระบบ

1.8 โปรแกรมจะทำการลบเกมกระดานมหาสนุกออกจากระบบเมื่อเสร็จเรียบร้อยแล้วกด ok เป็นอันเสร็จ



รูปที่ ก-7 หน้าจอเมื่อทำการถอดเกมกระดานมหาสนุกเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การใช้งานเกมกระดานมหาสนุก

ในที่นี้จะอธิบายการใช้งานในแต่ละหน้าจอของเกมกระดานมหาสนุกดังนี้

### 2.1 หน้าจอ Title

เมื่อเปิดเกม Funny Board จะเข้าสู่หน้าจอเริ่มเล่นเกม ดังรูปที่ ก-8 ซึ่งจะมีปุ่มต่างๆดังนี้



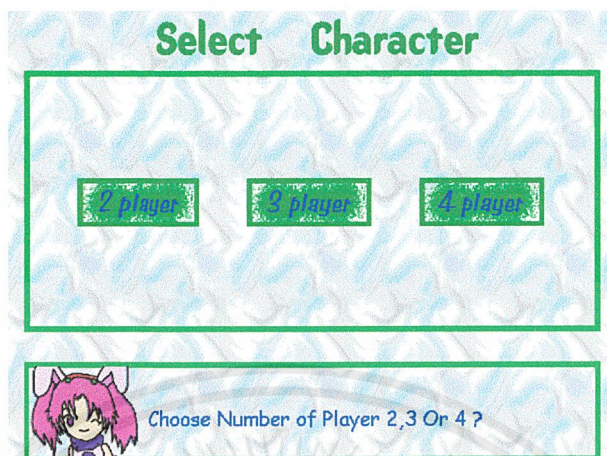
รูปที่ ก-8 หน้าจอเริ่มเล่นเกม

- 2.1.1 New Game – เป็นปุ่มเริ่มเล่นเกมรอบใหม่  
เมื่อเลือกแล้วจะเข้าไปสู่หน้าเลือกจำนวนผู้เล่น ดังรูปที่ ก-9
- 2.1.2 Load Game – เป็นปุ่มเปิดเกมที่เรานับที่การเล่นครั้งก่อนไว้  
เมื่อเลือกแล้วจะเข้าไปสู่หน้า Load Game ดังรูปที่ ก-15
- 2.1.3 Mini Game – เป็นปุ่มเปิดเล่นเกมย่อย ๆ ที่มีอยู่ในบอร์ดเกม  
เมื่อเลือกแล้วจะเข้าไปสู่หน้า Mini Game ดังรูปที่ ก-16
- 2.1.4 Help – เป็นปุ่มเปิดหน้าที่ใช้ในการอธิบายการเล่นเกม  
เมื่อเลือกแล้วจะเข้าไปสู่หน้า Help ดังรูปที่ ก-21
- 2.1.5 Exit – เป็นปุ่มที่ใช้ในการออกจากเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 หน้าจอเลือกจำนวนผู้เล่น

เป็นส่วนที่ให้เลือกจำนวนผู้เล่นเมื่อเริ่มเล่นเกมใหม่มีปุ่มต่างๆดังนี้



รูปที่ ก-9 หน้าจอเลือกจำนวนผู้เล่น

2.2.1 2 Player – เป็นปุ่มสำหรับเลือกเล่นเกมโดยมีผู้เล่น 2 คน

2.2.2 3 Player – เป็นปุ่มสำหรับเลือกเล่นเกมโดยมีผู้เล่น 3 คน

2.2.3 4 Player – เป็นปุ่มสำหรับเลือกเล่นเกมโดยมีผู้เล่น 4 คน

หลังจากเลือกจำนวนผู้เล่นแล้วจะเข้าสู่หน้าจอเลือกตัวละคร รูปที่ ก-10 ต่อไป

## 2.3 หน้าจอเลือกตัวละคร

หน้าจอจะเลือกตัวละครที่จะใช้แทนตัวผู้เล่น เริ่มเลือกตั้งแต่ผู้เล่นที่ 1 ไปตามลำดับจนครบตามจำนวนผู้เล่นที่เลือกไว้ โดยคลิกเลือกที่ตัวละครเลย และไม่สามารถเลือกตัวละครซ้ำกันได้ เมื่อผู้เล่นเลือกตัวละครจนครบแล้วจะเข้าสู่หน้าจอเลือกแผนที่ รูปที่ ก-11 ต่อไป



รูปที่ ก-10 หน้าจอเลือกตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 หน้าจอเลือกแผนที่

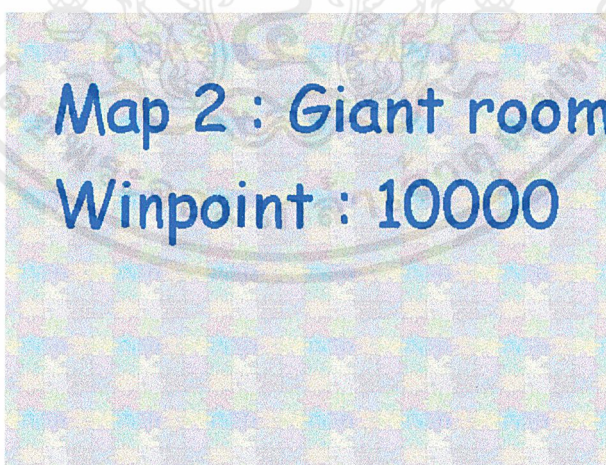
เลือกแผนที่ที่ต้องการเล่นจาก 3 แผนที่ โดยคลิกเลือกแผนที่ที่ต้องการเล่นเมื่อเลือกเสร็จเรียบร้อยแล้วจะเข้าสู่หน้าจอเริ่มต้นเกมหลัก รูปที่ ก-12



รูปที่ ก-11 หน้าจอเลือกแผนที่

## 2.5 หน้าจอเริ่มต้นเกมหลัก

หน้าจอนี้จะเป็นการเตรียมตัวเล่นเกมหลัก โดยโปรแกรมจะทำการ load ข้อมูลต่างๆที่จำเป็นในการเล่น ส่วนที่หน้าจอจะแสดงชื่อแผนที่ที่เลือกเล่นกับแต้ม winpoint ของแผนที่นั้น โดย winpoint นี้เป็นแต้มที่ต้องทำให้ถึงเพื่อชนะเกม เมื่อผ่านหน้าจอนี้ไปก็จะเข้าสู่หน้าจอการเล่นเกมหลัก รูปที่ ก-13



รูปที่ ก-12 หน้าจอเริ่มต้นเกมหลัก

## 2.6 หน้าจอการเล่นเกมหหลัก

หน้าจอนี้จะเป็นส่วนที่ใช้เล่นเกมหลักซึ่งจะมีปุ่มต่างๆดังนี้



รูปที่ ก-13 หน้าจอการเล่นเกมหหลัก

**2.6.1 Play** – ปุ่มที่ใช้กดเพื่อทอยลูกเต๋าลงเล่นเกม เมื่อทอยลูกเต๋าลงแล้วก็จะทำการเดินไปตามแต้มที่ทอยได้เมื่อตกในช่องใดก็จะเกิดเหตุการณ์ตามที่ช่องนั้นกำหนดไว้ โดยสามารถดูได้จากสัญลักษณ์บนช่องนั้นดังนี้

C – เป็นจุด Checkpoint ซึ่งแผนที่หนึ่งมี 4จุด ถ้าผ่านครบจะได้แต้มพิเศษเมื่อผ่านจุด Start

FOR – เมื่อตกลงช่องนี้จะเข้าสู่หน้า fortune เพื่อเปิดป้ายเสี่ยงโชค

ST – เมื่อตกช่องนี้จะหยุดเดินหนึ่งรอบ

AG – เมื่อตกช่องนี้แล้วจะได้เล่นเพิ่มหนึ่งรอบ

E – เมื่อตกช่องนี้จะเข้าไปเล่น mini game Equation

M – เมื่อตกช่องนี้จะเข้าไปเล่น mini game Memory

I – เมื่อตกช่องนี้จะเข้าไปเล่น mini game Image

V – เมื่อตกช่องนี้จะเข้าไปเล่น mini game Vocab

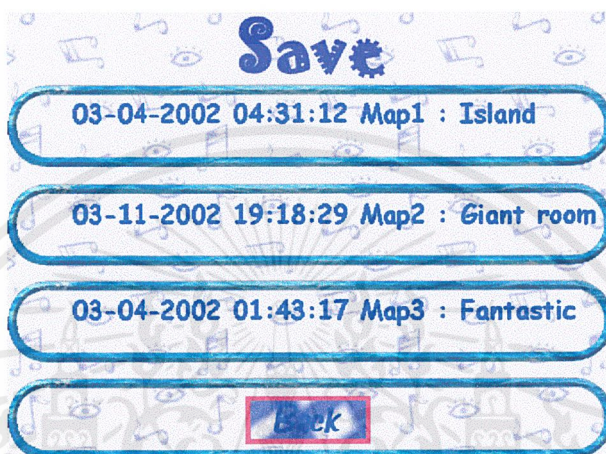
**2.6.2 Save** – กดเพื่อเข้าสู่หน้า Save Game ดังรูปที่ ก-14

**2.6.3 Exit** – กดเพื่อเลิกเล่นเกมและกลับไปสู่หน้า Title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 หน้าจอ Save Game

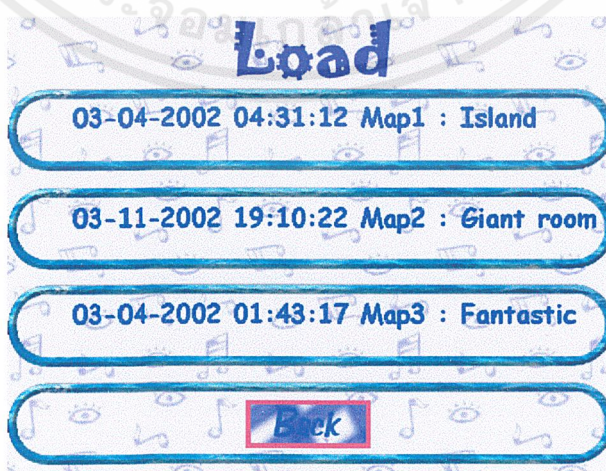
หน้าจอนี้ทำหน้าที่ Save Game ไว้เล่นต่อในครั้งต่อไป โดยผู้เล่นสามารถเลือกไฟล์ Save ได้โดยคลิกที่กรอบที่แสดงข้อมูลแต่ละไฟล์ กรอบนี้จะแสดงข้อมูลของไฟล์ Save แต่ละไฟล์ มีทั้งหมด 3 ไฟล์ ข้อมูลที่แสดงประกอบด้วย วันและเวลาที่ Save กับชื่อแผนที่ที่เลือกเล่น ถ้าช่องไหนเป็น Empty แสดงว่าไฟล์นั้นว่าง และถ้าคลิกเลือกไฟล์ที่มีข้อมูลอยู่แล้ว โปรแกรมจะทำการ Save ข้อมูลทับข้อมูลเดิม ส่วนปุ่ม Back ด้านล่างใช้เพื่อกลับไปยังหน้าการเล่นเกมนหลัก



รูปที่ ก-14 หน้าจอ Save Game

## 2.8 หน้าจอ Load Game

หน้าจอนี้จะมีองค์ประกอบเหมือนกับหน้า Save Game ทุกประการ คือมีกรอบสำหรับแต่ละไฟล์ 3 ไฟล์ และมีปุ่ม Back เหมือนกัน ต่างกันตรงที่หน้าจอนี้เข้าจากปุ่ม Load Game ที่หน้า Title โดยทำหน้าที่ Load ไฟล์กลับมาเพื่อเล่นต่อจากที่ Save ไว้ และปุ่ม Back ในหน้านี้จะกลับไปหน้า Title ส่วนข้อมูลในกรอบที่แสดงก็จะเหมือนหน้าจอ Save Game



รูปที่ ก-15 หน้าจอ Load Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 หน้าจอ Mini Game

หน้าจอนี้จะเป็นหน้าจอที่สามารถเลือกเล่นเกมย่อยใดๆก็ได้โดยไม่ต้องเล่นเกมหลัก ผู้เล่นสามารถเข้าสู่หน้าจอนี้ได้โดยคลิกปุ่ม Mini Game ที่หน้าจอ Title ผู้เล่นสามารถเลือกเล่นเกมย่อยได้โดยคลิกที่ Panel ตามชื่อเกมที่ต้องการเล่นแล้วเข้าสู่การเล่นเกมย่อยนั้น เมื่อเล่นเกมย่อยนั้นเสร็จเรียบร้อยแล้วก็จะกลับมาสู่หน้าจอนี้ ถ้าต้องการออกจากหน้าจอนี้กลับไปสู่หน้าจอ Title ให้คลิกปุ่ม Back



รูปที่ ก-16 หน้าจอ Mini Game

## 2.10 หน้าจอเกมย่อย Magic Vocab

หน้าจอนี้สามารถเข้ามาได้จาก 2 ทาง คือ เล่นเกมหลักแล้วตกที่ช่องสัญลักษณ์ V หรือเข้าจากหน้าจอ Mini Game ในเกมนี้ผู้เล่นจะสามารถทายตัวอักษรได้ โดยคลิกที่ตัวอักษรที่ต้องการทายเมื่อคลิกแล้วตัวอักษรนั้นจะหายไป เพื่อให้ไม่สามารถทายตัวเดิมซ้ำได้ เมื่อทายได้ครบหรือเวลาหมดก็จะจบการเล่น

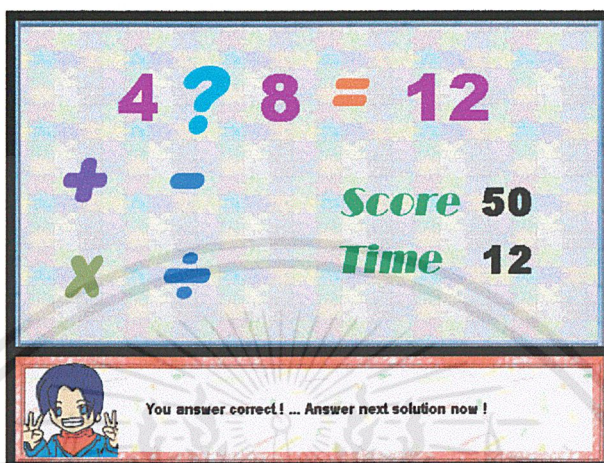


รูปที่ ก-17 หน้าจอเกมย่อย Magic Vocab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.11 หน้าจอเกมย่อย Magic Equation

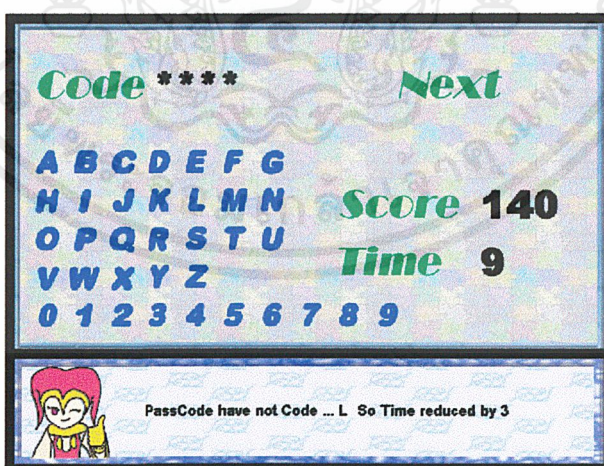
หน้าจอนี้สามารถเข้ามาได้จาก 2 ทาง คือ เล่นเกมหลักแล้วตกที่ช่องสัญลักษณ์ E หรือ เข้าจากหน้าจอ Mini Game ในหน้านี้ผู้เล่นจะต้องคลิกที่เครื่องหมายต่างๆ เพื่อทายเครื่องหมาย ที่ถูกต้องสำหรับโจทย์ข้อนั้นๆ เมื่อทำโจทย์ถูกต้องครบ 10 ข้อ หรือเวลาหมดก็จะจบการเล่น



รูปที่ ก-18 หน้าจอเกมย่อย Magic Equation

### 2.12 หน้าจอเกมย่อย Magic Memory

หน้าจอนี้สามารถเข้ามาได้จาก 2 ทาง คือ เล่นเกมหลักแล้วตกที่ช่องสัญลักษณ์ M หรือ เข้าจากหน้าจอ Mini Game เกมนี้จะต้องคลิกที่ตัวอักษรเหมือนเกม Magic Vocab เพื่อทวน Code ที่เกมให้ไป แต่ตัวอักษรที่คลิกแล้วจะไม่หายไป เกมนี้จะจบได้กรณีเดียวคือหมดเวลา

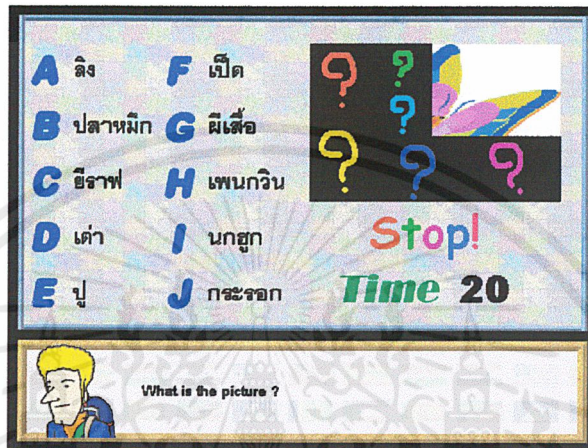


รูปที่ ก-19 หน้าจอเกมย่อย Magic Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.13 หน้าจอเกมย่อย Magic Image

หน้าจอนี้สามารถเข้ามาได้จาก 2 ทาง คือ เล่นเกมหลักแล้วตกที่ช่องสัญลักษณ์ I หรือ เข้าจากหน้าจอ Mini Game เกมนี้จะแบ่งออกเป็น 2 ขั้นตอน คือเริ่มต้นโดยเปิดภาพไปเรื่อยๆ เมื่อผู้เล่นต้องการทาสให้คลิกที่ stop เพื่อหยุดการเปิดภาพ แล้วเกมจะแสดงตัวเลือกให้เลือกทาส แล้วผู้เล่นจะเลือกทาสโดยคลิกที่ตัวอักษรหน้าตัวเลือกที่เลือก เกมนี้จะจบลงเมื่อผู้เล่นทาสภาพ ไม่ว่าจะทาสถูกหรือไม่หรือเวลาหมด



รูปที่ ก-20 หน้าจอเกมย่อย Magic Image

### 2.14 หน้าจอ Help หลัก

เป็นหน้าจอหลักซึ่งจะแสดง Menu หน้า Help ต่างๆ ให้เลือกเข้าไปดูได้ โดย Help นี้จะเป็น กติกาการเล่นของเกมหลักและเกมย่อยต่างๆ ต้องการเข้าไปดู Help ของเกมใดก็คลิกที่ปุ่มนั้น เพื่อเข้าไปสู่หน้าจอ Help ย่อยของเกมนั้นๆ ดังรูปที่ ก-20 ส่วนปุ่ม Exit เป็นการออกจาก หน้านี้กลับไปยังหน้าจอ Title

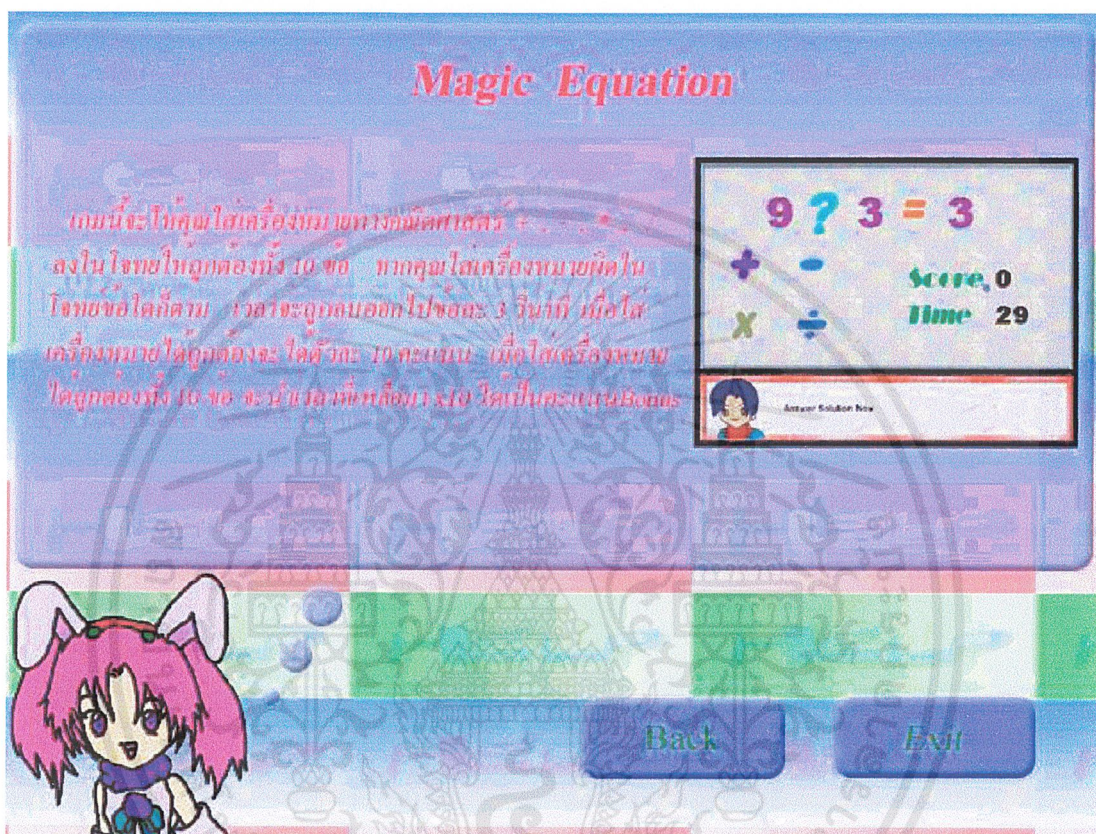


รูปที่ ก-21 หน้าจอ Help หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.15 หน้าจอ Help ย่อย

เป็นหน้าจอที่แสดงกติกาการเล่นเกมที่ทั้งหมด 5 หน้าคือ หน้าเล่นเกม , Magic Memory , Magic Equation , Magic Image และ Magic Vocab ตามลำดับ โดยจะสามารถเลือกดูได้จากหน้าจอ Help หลัก หน้าจอ Help ย่อยแต่ละหน้าจะมีปุ่ม 2 ปุ่ม คือ ปุ่ม Back ใช้สำหรับกลับไปสู่หน้าจอ Help หลัก และปุ่ม Exit สำหรับกลับไปสู่หน้าจอ Title



รูปที่ ก-22 ตัวอย่างหน้าจอ Help ย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข. กติกาการเล่น

### 1. กติกาในการเล่นเกมหลัก

เกมนี้เป็นเกมกระดานซึ่งมีรูปแบบคล้ายคลึงกับเกมกระดานทั่วไป คือ ใช้การทอยลูกเต๋าแล้วเดินไปตามแต้มที่ได้ เมื่อตกในช่องใดก็ให้ทำตามคำสั่งในช่องนั้น โดยผู้เล่นจะเริ่มเล่นตั้งแต่ผู้เล่นคนแรกจนถึงคนสุดท้ายตามลำดับ ถ้าผู้เล่นคนใดเดินไปตกในช่องที่มีผู้เล่นอื่นอยู่ก่อนแล้วจะได้เดินเลยไปยังหน้าอีก 1 ช่อง เงื่อนไขในการชนะของเกมนี้คือต้องทำคะแนนให้ได้ตามเป้าหมายที่กำหนดไว้ในแต่ละฉากแล้วกลับไปจุด start โดยสามารถสะสมคะแนนได้จากการเล่นเกมต่างๆ และการผ่าน check point ครบทุกจุด เมื่อทำคะแนนได้ตามเป้าหมายที่กำหนดไว้ และสามารถกลับไปจุด start ได้ก่อนเป็นคนแรกก็จะเป็นผู้ชนะ

### 2. กติกาสำหรับเล่นเกมย่อย

#### 2.1 เกมย่อย Magic Memory

เกมนี้จะให้จำตัวอักษรภาษาอังกฤษกับตัวเลขซึ่งจะเรียกว่า code โดย code นี้จะถูกสุ่มมาแล้วแสดงให้ทราบที่ ด้านหลังคำว่า NEXT แต่ละตัวจะแสดงอยู่ประมาณ 3 วินาที ผู้เล่นจะต้องกดปุ่มตัวอักษรหรือตัวเลขตาม code ที่ได้มา เมื่อผู้เล่นกดได้ถูกต้อง จะปรากฏเครื่องหมาย \* ที่ตำแหน่ง code ในแต่ละครั้งผู้เล่นจะต้องกด code ทวนตั้งแต่ตัวแรกตามลำดับจนถึงตัวล่าสุด เมื่อทวน code จนครบแล้วเครื่องหมาย \* จะหายไปพร้อมกับขึ้น code ตัวถัดไป หากกด code ผิด เวลาจะถูกลบออกไปตัวละ 3 วินาที ผู้เล่นจะเล่นได้เรื่อยๆจนกว่าจะหมดเวลา 30 วินาที การคิดคะแนนจะเป็นดังนี้

- เมื่อกด code ถูกต้อง 1 ตัว (\* ปรากฏขึ้น 1 ครั้ง) จะได้ตัวละ 10 คะแนน
- ทุกครั้งที่ทวน code ครบ (\* หายไป) จะได้โบนัส ครั้งละ 10 คะแนน

#### 2.2 เกมย่อย Magic Image

เกมนี้จะให้คุณทายรูปภาพปริศนาว่า รูปที่เห็นนั้นเป็นสัตว์ชนิดใด โดยเมื่อเริ่มต้นภาพปริศนาจะถูกปิดไว้ทั้งหมด แล้วเกมจะค่อยเปิดภาพให้เห็นทีละส่วนทุกๆ 3 วินาที เมื่อผู้เล่นต้องการจะตอบให้กดที่ปุ่ม STOP! เพื่อหยุดการเปิดภาพ และเวลา แล้วจะมีตัวเลือกขึ้นให้ให้เลือกตอบ 10 ตัวเลือก ถ้าตอบถูกก็จะนำเวลาที่เหลือ x15 เป็นคะแนนที่ได้รับ แต่หากตอบผิดก็จะได้ไม่ได้คะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 เกมย่อย Magic Vocab

เกมนี้จะให้ผู้เล่นทายศัพท์ภาษาอังกฤษจากคำแปลภาษาไทยที่ให้มา เลือกทายตัวอักษรโดยคลิกที่ตัวอักษรภาษาอังกฤษ ถ้าตัวอักษรนั้นถูกต้อง ก็จะปรากฏขึ้นในคำศัพท์ แต่ถ้าตัวอักษรนั้นผิดคือไม่มีในคำศัพท์ เวลาจะถูกลบออกไปตัวละ 3 วินาที ตัวอักษรที่ทายไปแล้วจะหายไปเพื่อไม่ให้มีการทายอักษรซ้ำตัวเดิม การคิดคะแนนทำได้ดังนี้

- เมื่อทายตัวอักษรได้ถูกต้องจะได้ตัวละ 10 คะแนน
- เมื่อทายคำศัพท์ได้ถูกต้อง จะนำเวลาที่เหลือมา  $\times 10$  เป็นคะแนนโบนัส

### 2.4 Magic Equation

เกมนี้จะให้ผู้เล่นเลือกใส่เครื่องหมายทางคณิตศาสตร์  $+$ ,  $-$ ,  $\times$ ,  $\div$  ลงในโจทย์ที่กำหนดให้ถูกต้อง หากผู้เล่นเลือกเครื่องหมายผิด เวลาจะถูกลบออกไปครั้งละ 3 วินาที และเกมจะสุ่มโจทย์ข้อใหม่ขึ้นมาให้แทน เกมจะจบลงเมื่อผู้เล่นสามารถตอบโจทย์ได้ถูกต้องครบ 10 ข้อหรือเวลาหมด การคิดคะแนนทำได้ดังนี้

- เมื่อทำโจทย์ถูกต้อง 1 ข้อจะได้ข้อละ 10 คะแนน
- เมื่อทำโจทย์ถูกต้องครบ 10 ข้อ จะนำเวลาที่เหลือมา  $\times 10$  ได้เป็นคะแนนโบนัส

### 2.5 เกมย่อย Fortune

เกมนี้จะให้คุณเลือกเปิดป้าย 1 ป้ายจากทั้งหมด 12 ป้าย โดยจะได้คะแนนตามที่ป้ายระบุ

## บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และจำลอง ครูอุตสาหะ. 2543. **Visual Basic 6 ฉบับโปรแกรมเมอร์.**

หจก.ไทยเจริญการพิมพ์

ชัยวัฒน์ คำรัตน์. 2537. **2D GAME With DirectX7.** ส.วิชาญการพิมพ์

พุดิพงษ์ นาคะปัท. 2542. **การเขียน GAME บน วินโดวส์ ด้วย Visual Basic.**

บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

Michael D. Root and James R. Boer. 1999. **DirectX Complete.** McGraw-Hill



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้