

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกมเพื่อการเรียนรู้ส่วนประกอบไมโครคอมพิวเตอร์

CAI FOR COMPUTER HARDWARE



2/ค.
ส 394 ก
2547

เลขหมู่.....
เลขทะเบียน..... 58794
วัน,เดือน,ปี..... 10 ก.พ. 2549

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

b. 1148 9091

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAI FOR COMPUTER HARDWARE



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2004**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ เกมเพื่อการเรียนรู้ส่วนประกอบไมโครคอมพิวเตอร์
 CAI FOR COMPUTER HARDWARE

ชื่อนักศึกษา นายรัชชัย ขวัญเมือง 44050423
 นายเพชร เตมียะเสน 44050444

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ดร.กรกช ประชุมรัมย์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2547

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ผศ.ดร.จิรพร ศรีสวัสดิ์	
กรรมการ	รศ.ไพโรจน์ พันธ์รักษ์พงษ์	
กรรมการและอาจารย์ที่ปรึกษา	ดร.กรกช ประชุมรัมย์	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ เกมเพื่อการเรียนรู้ส่วนประกอบไมโครคอมพิวเตอร์
ชื่อนักศึกษา นายรัชชชัย ขวัญเมือง 44050423
 นายเพชร เตมียะเสน 44050444
ภาควิชา คณะศึกษาศาสตร์และวิทยาการคอมพิวเตอร์
สาขา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2547
อาจารย์ที่ปรึกษา ดร.กรกช ประชุมรักษ์

บทคัดย่อ

ในปัจจุบันนี้เป็นที่ยอมรับกันอย่างกว้างขวางว่าเครื่องคอมพิวเตอร์นั้นเข้ามามีบทบาทต่อชีวิตประจำวันมาก ไม่ว่าจะเป็นทางด้านการศึกษา ทางด้านคมนาคม หรือแม้กระทั่งทางด้านการบันเทิง ซึ่งคนส่วนใหญ่ก็น่าจะรู้แล้วที่ใช้คอมพิวเตอร์เพียงอย่างเดียวเท่านั้น เพราะเห็นว่าเครื่องคอมพิวเตอร์นั้นเป็นเครื่องใช้ไฟฟ้าทั่วไป ที่ประกอบด้วยอุปกรณ์หลายชนิดซึ่งยากที่จะเข้าใจ ดังนั้นโครงการพิเศษนี้ต้องการเสนอการสอนในรูปแบบของเกม ซึ่งเป็นเกมลักษณะที่ช่วยสอนการประกอบเครื่องคอมพิวเตอร์ ว่าเครื่องคอมพิวเตอร์ 1 เครื่องนั้นมีส่วนประกอบอะไรบ้าง และทำหน้าที่อะไรบ้าง โดยการที่จะประกอบคอมพิวเตอร์ได้นั้น เราต้องสะสมชิ้นส่วนของเครื่องคอมพิวเตอร์ให้ครบทุกชิ้นจากเกมต่างๆ ที่เราได้กำหนดไว้ โดยเราจะแบ่งเป็นออกเป็นเกมย่อยๆ ซึ่งในแต่ละเกมนั้นเราจะให้ความรู้แก่ผู้เล่นว่าเครื่องคอมพิวเตอร์นั้นมีส่วนประกอบอะไรบ้าง แต่ละชิ้นส่วนมีประโยชน์อย่างไรมีความสำคัญอย่างไรกับเครื่องคอมพิวเตอร์ และเสนอข้อคิดในการเลือกซื้ออุปกรณ์คอมพิวเตอร์ ซึ่งเกมแต่ละเกมนั้นจะไม่เหมือนกัน เช่น เกมตอบคำถาม เกมจับคู่ เป็นต้น เมื่อผู้เล่นผ่านด่านแต่ละด่านแล้วก็จะได้ชิ้นส่วนของเครื่องคอมพิวเตอร์ของด่านนั้นๆ ถ้าผู้เล่นสะสมชิ้นส่วนครบแล้วเราก็จะอธิบายขั้นตอนการประกอบคอมพิวเตอร์ให้แก่ผู้เล่น

Special Project Title	CAI FOR COMPUTER HARDWARE		
Students	Mr.Thawatchai	Kwanmoung	44050423
	Mr. Petch	Temceeyasen	44050444
Degree	Bachelor of Science		
Department	Mathematics and Computer Science, Faculty of Sciences		
Programme	Computer Science		
Academic Year	2004		
Special Project Advisor	Dr.Korakot Prachumrak		

ABSTRACT

Nowadays, computers are widely accepted and play an essential role in our life. Such as education, communication, entertainment etc. Many users only know how to use them because they think that computer is an electronic equipment and consists of many components which difficult to understand.

This project special presents Computer Aided Instruction (CAI) in term of game. The application will help users to learn about each component in a computer and it's functions to play this game. The users has to collect the whole parts of the computer from each game. The aim of this game is that the user has to collect all the components to build a computer. There are three sub-games in this game, the user has to win all sub-games, to get all the components. Each sub-game has an instruction about the computer, components and functions, and how to buy the computer components. Each sub-game is different from the others. When the users can collect all the component, there will be an instruction for the user of how to build a computer.

กิตติกรรมประกาศ

ในการทำโครงการปัญหาพิเศษเรื่อง เกมเพื่อการเรียนรู้ด้านฮาร์ดแวร์และการประกอบคอมพิวเตอร์ สามารถทำสำเร็จและผ่านลู่วงไปได้ด้วยดีนั้น ทางคณะผู้จัดทำต้องขอขอบพระคุณ ดร.กรกช ประชุมรักษ์ อาจารย์ผู้รับผิดชอบโครงการปัญหาพิเศษฉบับนี้ ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาในด้านต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ได้ประสานวิชาความรู้ทั้งทางด้านทฤษฎีและภาคปฏิบัติแก่ทางคณะผู้จัดทำจนกระทั่งปัญหาพิเศษนี้สัมฤทธิ์ผลด้วยดีทุกประการ

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ที่คอยให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์

นอกจากนี้ทางคณะผู้จัดทำต้องขอขอบพระคุณ เพื่อนๆ และรุ่นพี่ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ ที่เกี่ยวข้องกับปัญหาพิเศษไว้ ณ ที่นี้



คณะผู้จัดทำ
มีนาคม 2548

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	IX
สารบัญรูป.....	X
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของปัญหา.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนในการดำเนินการ.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 โปรแกรม Microsoft DirectX.....	4
2.1.1 ความเป็นมา.....	4
2.1.2 หลักการของ DirectX.....	5
2.1.3 องค์ประกอบของ DirectX.....	6
2.1.3.1 ชั้นติดต่อกับอุปกรณ์ฮาร์ดแวร์.....	6
2.1.3.2 ชั้นจำลองอุปกรณ์ฮาร์ดแวร์.....	6
2.1.3.3 การแสดงผลภาพบนจอภาพ (Direct Draw).....	6
2.1.3.4 การแสดงเสียงทั่วไป (Direct Sound).....	6
2.1.3.5 การแสดงเสียงแบบสามมิติ (Direct Sound 3D).....	6
2.1.3.6 การแสดงเสียงเพลง (Direct Music).....	6
2.1.3.7 การรับข้อมูลอินพุต (Direct Input).....	7
2.1.3.8 การเชื่อมต่อเครือข่าย (Direct Play).....	7
2.1.3.9 การแสดงภาพ 3 มิติในโหมดระดับสูง.....	8

สารบัญ (ต่อ)

	หน้า
2.1.3.10 การแสดงภาพ 3 มิติในโหมดระดับต่ำ.....	8
2.2 การเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์.....	8
2.2.1 ไฟล์ไอบรารีของ Win32 API.....	9
2.2.2 การเพิ่มไฟล์เข้าไปในแอปพลิเคชัน.....	9
2.2.3 ฟังก์ชันการทำงานหลักของวินโดวส์.....	9
2.2.4 คลาสของวินโดวส์.....	9
2.2.5 การขึ้นทะเบียนคลาสของวินโดวส์.....	10
2.2.6 การสร้างวินโดวส์ออปเจก.....	11
2.2.7 การแสดงวินโดวส์บนจอภาพ.....	12
2.2.8 ลูปของเหตุการณ์ในวินโดวส์ (Event Loop).....	12
2.2.8.1 ฟังก์ชันรับ Message หรือ GetMessage().....	13
2.2.8.2 ฟังก์ชันแปล Message ที่ได้รับ หรือ TranslateMessage().....	13
2.2.8.3 ฟังก์ชันจัดส่ง Event หรือ DispatchMessage().....	13
2.2.9 ตัวจัดการเหตุการณ์ต่าง ๆ หรือ Event handler.....	13
2.3 Event Driven Programming.....	14
2.4 วิธีการแสดงภาพบนจอภาพ.....	14
2.4.1 ความรู้เบื้องต้นในการแสดงภาพด้วยคำสั่งของ Direct Draw.....	14
2.4.2 การสร้างออปเจกของ DirectDraw.....	15
2.4.3 การกำหนดการแสดงผลของวินโดวส์ด้วยฟังก์ชัน SetCooperativeLevel ของ Direct Draw.....	15
2.4.4 การกำหนดโหมดการแสดงผล.....	17
2.4.5 การประยุกต์ใช้ DirectDraw.....	18
2.4.6 การวาดภาพด้วยคำสั่ง DirectDraw.....	18
2.4.7 หลักการสร้าง Surface.....	18
2.4.8 การนำภาพขึ้นแสดง.....	24
2.4.9 การวาดภาพลงบน Surface และการทำ Transparency.....	25
2.5 วิธีการรับข้อมูลจากเมาส์และคีย์บอร์ด.....	29
2.5.1 การใช้ DirectInput.....	30

สารบัญ (ต่อ)

	หน้า
2.5.2 การติดต่อกับคีย์บอร์ดผ่านทาง Direct Input.....	31
2.5.2.1 ออปเจกของ DirectInput.....	31
2.5.2.2 ทำการสร้างออปเจกของคีย์บอร์ด.....	32
2.5.2.3 กำหนดลักษณะการใช้งานของคีย์บอร์ด.....	33
2.5.2.4 กำหนดรูปแบบข้อมูลของคีย์บอร์ด.....	34
2.5.2.5 ทำการจองคีย์บอร์ดด้วยฟังก์ชัน Acquire().....	35
2.5.2.6 ทำการอ่านข้อมูลจากคีย์บอร์ด.....	35
2.5.3 การรับข้อมูลจากเมาส์.....	36
2.5.3.1 สร้างออปเจกเมาส์.....	37
2.5.3.2 ทำการกำหนดลักษณะการใช้งานของเมาส์.....	37
2.5.3.3 ทำการกำหนดรูปแบบของข้อมูลที่ได้รับ.....	37
2.5.3.4 ทำการจองเมาส์.....	38
2.5.3.5 การอ่านค่าจากเมาส์.....	38
2.6 วิธีการแสดงเสียง.....	40
2.6.1 การสร้าง Direct Sound Object.....	40
2.6.2 การตั้งค่าของ cooperation level.....	41
2.6.3 Primary และ secondary buffers.....	41
2.6.4 การสร้าง Secondary Buffer.....	41
2.6.5 การเขียนข้อมูลลงบน Secondary Buffer.....	43
2.6.6 การแสดงเสียง.....	44
2.6.7 การหยุดเสียง.....	45
2.6.8 การตั้งระดับความดังของเสียง.....	45
บทที่ 3 ขั้นตอนการดำเนินการ.....	46
3.1 ขั้นตอนการวิเคราะห์และการออกแบบ.....	47
3.1.1 ออกแบบระบบเกมและการกำหนดโครงสร้างของเกม.....	47
3.1.2 การออกแบบรูปภาพและกราฟิก.....	47
3.1.3 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น.....	48

สารบัญ (ต่อ)

	หน้า
3.1.4 การออกแบบการจัดเก็บข้อมูล.....	49
3.1.5 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้.....	49
3.2 ขั้นตอนการสร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม.....	49
3.3 ขั้นตอนการเขียนโปรแกรม.....	70
3.3.1 หน้าจอ Intro.....	71
3.3.2 หน้าจอการเลือกตัวละคร.....	73
3.3.3 หน้าจอฉากภายในห้อง.....	75
3.3.3.1 หน้าจอ Data.....	71
3.3.3.2 หน้าจอการ Create Computer.....	79
3.3.4 ฉากแผนที่ในเมือง.....	80
3.3.5 ฉาก Mini Game.....	81
3.3.6 เกมต่างๆ.....	83
3.3.7 โปรแกรมช่วยสร้างข้อมูลต่างๆ.....	91
3.4 ขั้นตอนการทดสอบเกม.....	92
บทที่ 4 ขั้นตอนการติดตั้งและการใช้งานโปรแกรม.....	93
4.1 เครื่องมือที่ใช้ในการทดลอง.....	93
4.1.1 ขั้นตอนการใช้งาน.....	93
4.1.1.1 การลงโปรแกรม.....	93
4.1.1.2 คู่มือการเล่นเกม.....	99
4.2 การออกแบบการทดลองเกม.....	128
4.2.1 ลักษณะการกดเมาส์.....	128
4.2.2 ลักษณะการกดคีย์บอร์ด.....	128
4.3 ผลการทดลอง.....	128
บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะ.....	129
5.1 ขั้นตอนการออกแบบ Story Board.....	129
5.2 ขั้นตอนการเขียนโปรแกรม.....	129

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
5.3 ขั้นตอนการสร้างตัวละคร.....	130
5.4 ข้อเสนอแนะ และแนวทางในการพัฒนา.....	130
บรรณานุกรม.....	131



สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแสดงแผนงานในการทำโครงการปัญหาพิเศษ.....	3
2.1 ค่าแฟลกของพารามิเตอร์ dwStyle.....	12
2.2 ค่าแฟลก Cooperative level ของ DirectDraw.....	16
2.3 ค่าแฟลกของ dwCaps.....	22
2.4 ค่าแฟลกควบคุมของฟังก์ชัน Blt().....	26
2.5 ค่าแฟลก dwFlag ลักษณะการใช้งาน ของ Keyboard.....	33
2.6 ค่าคีย์บนคีย์บอร์ด.....	36
2.7 ค่าแฟลกของ cooperation level DirectSound.....	41
2.8 ค่าแฟลกควบคุมของโครงสร้าง DirectSound Buffer.....	42
3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ที่ใช้ภายในเกม.....	48



สารบัญรูป

รูปที่	หน้า
2.1 การทำงานของ DirectX เปรียบเทียบกับ GDI/MCI.....	5
2.2 องค์ประกอบของ DirectX.....	7
2.3 กลไกการส่ง message ในวินโดวส์.....	8
2.4 การทำงานของฟังก์ชัน Winmain() กับ Event Handler.....	10
2.5 อุปเหตุการณ์หลัก.....	12
2.6 ทรัพยากรที่ใช้สร้างแอปพลิเคชันของ Win32 DirectX.....	14
2.7 การกำหนดขนาดของ Surface.....	19
2.8 การแสดงภาพบนหน้าจอ โดย DirectDraw.....	20
2.9 การ Flip สลับ Surface.....	24
2.10 การเปรียบเทียบระหว่างการกำหนดค่า Color Key.....	28
2.11 อุปกรณ์อินพุตต่าง ๆ.....	30
2.12 ระดับโครงสร้างภายในของ DirectInput.....	31
2.13 แสดงการเก็บค่าตำแหน่งเมื่อทำการเลื่อนเมาส์.....	38
2.14 รูปของข้อมูลอินพุต.....	39
2.15 การทำงานของ DirectSound.....	40
2.16 การเขียนข้อมูลเสียงลง Sound Buffer.....	44
2.17 การแสดงข้อมูลเสียง.....	44
3.1 รูปฉากหลังหน้าจอ Intro.....	49
3.2 รูปปุ่มต่างๆ ของหน้าจอ Intro.....	50
3.3 แสดงภาพเมาส์.....	50
3.4 รูปภาพ Mini Game ของเกมต่ออุปกรณ์.....	50
3.5 รูปภาพ Mini Game ของเกมทายสูงต่ำ.....	50
3.6 รูปภาพ Mini Game ของเกมตอบคำถาม.....	51
3.7 รูปภาพ Mini Game ของเกมจับคู่.....	51
3.8 รูปแสดงพื้นหลังของฉากการเลือก Mini Game.....	51
3.9 ปุ่มของฉาก Mini Game.....	52
3.10 รูปภาพแถบการเลือก Mini Game.....	52
3.11 ฉากหลังของการเลือกตัวละคร.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.12 รูปตัวละคร.....	53
3.13 รูปปุ่มต่างๆ ของฉากการเลือกตัวละคร.....	53
3.14 รูปแสดงเมนูของฉากภายในห้อง.....	53
3.15 รูปฉากภายในห้อง.....	54
3.16 รูปพื้นหลังของฉากห้อง.....	54
3.17 รูปปุ่มต่างๆ ภายในฉากห้อง.....	55
3.18 รูปแสดงฉากหลังของ Dialog ต่างๆ ภายในเกม.....	55
3.19 รูปปุ่มต่างๆ ของ Dialog ต่างๆ ภายในเกม.....	56
3.20 รูปแสดงฉากหลังของ Messages Box ต่างๆ ภายในเกม.....	56
3.21 รูปแสดงปุ่มต่างๆ ของ Messages Box.....	57
3.22 รูปแสดง Separate line ที่เอาไว้ช่วยในการจัดหน้าจอของเกม.....	57
3.23 รูปบอกรายละเอียดต่างๆ ของเมืองและตัวละคร.....	57
3.24 รูปบอกสถานะของวันและปี.....	57
3.25 รูปของสถานที่ต่างๆ ภายในเมือง.....	58
3.26 รูปแสดงสถานที่ต่างๆ ภายในเมือง.....	58
3.27 รูปแสดงฉากหลังของแผนที่เมือง.....	59
3.28 รูปแสดงการเลือก Event ต่างๆ.....	59
3.29 รูปแสดงแถบการเลือก Event.....	59
3.30 รูปแสดงหน้าต่างตัวละคร และวันเวลา.....	60
3.31 รูปแสดงรายละเอียดของตัวละครและบทสนทนา.....	60
3.32 รูปฉากพื้นหลังของการเลือกซื้อสินค้า.....	60
3.33 รูปตัวละครต่างๆ ภายในเกม.....	61
3.34 รูปฉากพื้นหลังของโรงเรียน.....	61
3.35 รูปฉากพื้นหลังของ IT mall ที่ 1 และ 3.....	61
3.36 รูปฉากพื้นหลังของสวนสาธารณะ.....	62
3.37 รูปฉากพื้นหลังของ IT mall ที่ 2.....	62
3.38 รูปฉากพื้นหลังภายในห้อง.....	62
3.39 ฉากหลังของเกมจับคู่ภายในเกมใหญ่.....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.40 ฉากหลังของเกมจับคู่ของ Mini Game.....	63
3.41 รูปของจำนวนไพ่ทั้งหมดที่ใช้ในเกม (เกมใหญ่และ Mini Game).....	63
3.42 รูปตัวอักษรต่างๆ ที่ใช้บอกสถานะของเกมจับคู่ใน Mini Game.....	64
3.43 รูปของแถบเวลาที่ใช้ในเกมจับคู่ใน Mini Game.....	64
3.44 รูปของตัวนับการจับคู่ว่าผิดไปแล้วกี่ครั้งของเกมจับคู่ในเกมใหญ่.....	64
3.45 รูปฉากพื้นหลังของเกมทายสูงต่ำ.....	65
3.46 รูปกล่องเก็บตัวเลข.....	65
3.47 รูปตัวเลขที่เอาไว้ทายสูงต่ำ.....	65
3.48 รูปลูกศรที่เอาไว้ทายสูงหรือต่ำ.....	66
3.49 รูปเครื่องหมายถูกผิดที่เอาไว้แสดงตอนทายตัวเลข.....	66
3.50 รูปพื้นหลังของเกมตอบคำถาม.....	66
3.51 รูปปุ่มต่างๆ ที่ใช้ในเกมตอบคำถาม.....	67
3.52 รูป Icon ที่เอาไว้เลือกข้อที่ต้องการจะทำข้อสอบ.....	67
3.53 รูปเครื่องหมายถูกที่เอาไว้แสดงการเลือกคำตอบในข้อนั้นๆ.....	67
3.54 รูปตัวเลขที่แสดงคำตอบที่เราเลือก.....	67
3.55 รูปสัญลักษณ์ที่เอาไว้แสดงอันดับของ Link.....	67
3.56 รูป CPU.....	67
3.57 รูปสาย Floppy Disk.....	67
3.58 รูป Main board (มองจากมุมบน).....	68
3.59 รูป Heat sink และพัดลม (มองจากมุมบน).....	68
3.60 รูปสายแพ IDE (มองจากมุมบน).....	68
3.61 รูปสายไฟจาก Power Supply แบบ 20 pin(มองจากมุมบน).....	69
3.62 รูปสายไฟจาก Power Supply แบบ 4 pin (มองจากมุมบน).....	69
3.63 รูป RAM (มองจากมุมบน).....	69
3.64 รูป VGA (มองจากมุมบน).....	69
3.65 รูปพื้นหลังที่ไว้แสดงข้อความ.....	70
3.66 รูปหน้าจอ Intro.....	71
3.67 แผนภาพการทำงานในหน้าจอ Intro.....	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.68 รูปแสดงหน้าจอการเลือกตัวละคร.....	73
3.69 แผนภาพการทำงานของหน้าจอการเลือกตัวละคร.....	74
3.70 รูปแสดงฉากภายในห้อง.....	75
3.71 แผนภาพการทำงานของฉากภายในห้อง.....	76
3.72 รูปแสดงรายละเอียดในส่วน Tips.....	77
3.73 รูปแสดงรายละเอียดในส่วน Parts.....	77
3.74 แผนภาพการทำงานของปุ่ม Data.....	78
3.75 รูปหน้าจอการเลือกอุปกรณ์คอมพิวเตอร์ที่จะนำไปประกอบ.....	79
3.76 แผนภาพการทำงานของปุ่ม Create Computer.....	79
3.77 รูปแผนที่ภายในเมือง.....	80
3.78 แผนภาพการทำงานของฉากแผนที่.....	81
3.79 รูปแสดงเกมย่อยต่างๆ.....	81
3.80 แผนภาพการทำงานของฉาก Mini Game.....	82
3.81 รูปของเกมจับคู่ในเกมใหญ่.....	83
3.82 รูปของเกมจับคู่ใน Mini Game.....	83
3.83 แผนภาพการทำงานของเกมจับคู่ในเกมใหญ่.....	84
3.84 แผนภาพการทำงานของเกมจับคู่ใน Mini Game.....	85
3.85 รูปของเกมทายสูงต่ำในเกมหลัก และ Mini Game.....	86
3.86 รูปของเกมตอบคำถามในเกมหลัก และ Mini Game.....	86
3.87 แผนภาพการทำงานของเกมทายสูงต่ำในเกมใหญ่.....	87
3.88 แผนภาพการทำงานของเกมทายสูงต่ำใน Mini Game.....	88
3.89 แผนภาพการทำงานของเกมตอบคำถามในเกมใหญ่.....	89
3.90 แผนภาพการทำงานของเกมตอบคำถาม Mini Game.....	90
3.91 รูปหน้าจอโปรแกรม Items Editor.....	91
3.92 รูปหน้าจอโปรแกรม Question Editor.....	91
3.93 รูปหน้าจอโปรแกรม Tips Editor.....	92
4.1 รูปหน้าจอของ Microsoft Visual C++.....	94
4.2 รูปหน้าจอของการปรับแต่ง Visual C++.....	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 รูปหน้าจอของการปรับแต่ง Visual C++ (Include Files).....	95
4.4 รูปหน้าจอของการปรับแต่ง Visual C++ (Choose Directory).....	95
4.5 รูปหน้าจอของการปรับแต่ง Visual C++ (Choose Directory) ในส่วนของ Sample.....	96
4.6 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Option ที่มีการจัดแล้ว.....	96
4.7 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library files.....	97
4.8 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library files (Choose Directory).....	97
4.9 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library ที่จัดเรียงเรียบร้อยแล้ว.....	98
4.10 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Settings.....	98
4.11 รูป DIY.EXE.....	99
4.12 แสดงหน้าจอเริ่มต้นของเกมของเกม.....	100
4.13 แสดงหน้าจอการเลือกตัวละคร.....	101
4.14 แสดงหน้าจอการตั้งชื่อ.....	101
4.15 แสดงหน้าจอในห้อง.....	102
4.16 แสดงหน้าจอในห้องสำหรับผู้เริ่มต้นเล่นเกมใหม่.....	103
4.17 หน้าจอ Objective.....	103
4.18 หน้าจอ Profile กรณีที่ยังไม่สามารถประกอบคอมพิวเตอร์ได้.....	104
4.19 หน้าจอ Profile กรณีที่สามารถประกอบคอมพิวเตอร์ได้.....	104
4.20 หน้าจอ New.....	105
4.21 หน้าจอ Data ในส่วนของ Tips สำหรับผู้ที่เล่นเกมใหม่.....	105
4.22 หน้าจอ Data ในส่วนของ Tests สำหรับผู้ที่เล่นเกมใหม่.....	106
4.23 หน้าจอ Data ในส่วนของ Parts สำหรับผู้ที่เล่นเกมใหม่.....	106
4.24 หน้าจอ Data ในส่วนของ Tips เมื่อผู้เล่นนั้นได้สะสมข้อมูลต่างๆ แล้ว.....	107
4.25 หน้าจอ Data ในส่วนของ Tests เมื่อผู้เล่นนั้นมีชิ้นส่วนคอมพิวเตอร์แล้ว.....	107
4.26 หน้าจอ Help.....	108
4.27 หน้าจอการประกอบคอมพิวเตอร์สำหรับผู้เล่นเกมใหม่.....	108
4.28 รูปหน้าจอการประกอบคอมพิวเตอร์ (กรณีที่มีอุปกรณ์คอมพิวเตอร์ครบแล้ว).....	109
4.29 หน้าจอการเลือกชิ้นส่วนที่จะประกอบคอมพิวเตอร์.....	109
4.30 หน้าจอบอกว่าเลือกอุปกรณ์ที่จะประกอบเสร็จเรียบร้อยแล้ว.....	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.31 หน้าจออธิบายการประกอบ CPU.....	110
4.32 หน้าจออธิบายการประกอบ Heat sink.....	111
4.33 หน้าจออธิบายการประกอบ RAM.....	111
4.34 หน้าจออธิบายการประกอบสายแพ.....	112
4.35 หน้าจออธิบายการประกอบสาย Floppy Disk.....	112
4.36 หน้าจออธิบายการประกอบสาย Power Supply.....	113
4.37 หน้าจออธิบายการประกอบสาย Power Supply แบบ 4 pin.....	113
4.38 หน้าจออธิบายการประกอบสาย VGA.....	114
4.39 หน้าจอบอกว่าประกอบคอมพิวเตอร์เสร็จสิ้นแล้ว.....	114
4.40 หน้าจอของ Calendar.....	115
4.41 รูปแสดง Massage Box ถามผู้เล่นก่อนจะออกจากห้อง.....	115
4.42 แสดงหน้าจอจากภายในเมือง.....	116
4.43 แสดงหน้าจอจากของโรงเรียน.....	117
4.44 รูปแสดงบทสนทนา.....	118
4.45 แสดงหน้าจอจากของห้างสรรพสินค้า IT ที่ 1.....	118
4.46 แสดงหน้าจอจากของห้างสรรพสินค้า IT ที่ 2.....	119
4.47 แสดงหน้าจอจากของห้างสรรพสินค้า IT ที่ 3.....	119
4.48 แสดงหน้าจอจากของสวนสาธารณะ.....	120
4.49 แสดงหน้าจอ New บอกว่าจะมีอะไรเกิดขึ้น.....	120
4.50 แสดงหน้าจอเมื่อสามารถเข้าร่วมกิจกรรมได้.....	121
4.51 แสดงหน้าจอเกมจับคู่.....	123
4.52 แสดงหน้าจอเกมทายสูงต่ำ.....	123
4.53 แสดง Massage Box ถามว่าจะเล่นต่อหรือไม่.....	124
4.54 แสดงหน้าจอจากเกมตอบคำถาม.....	125
4.55 รูปแสดงหน้าจอการเลือกเล่น Mini Game.....	126
4.56 แสดงหน้าจอเกมต่ออุปกรณ์คอมพิวเตอร์.....	127
4.57 แสดงหน้าจอเกมจับคู่.....	127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ (Introduction)

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันเครื่องคอมพิวเตอร์ถือว่าเป็นอุปกรณ์ที่คนทั่วไปรู้จัก และส่วนมากก็เคยสัมผัสและทดลองใช้หรือเป็นเจ้าของ เครื่องคอมพิวเตอร์นั้นนอกจากจะช่วยในเรื่องของการทำงานแล้วยังจะช่วยในด้านความบันเทิง เช่น อินเทอร์เน็ต ดูหนัง ฟังเพลง และอื่นๆ ซึ่งการใช้งานคอมพิวเตอร์แต่ละด้านนั้นทำได้ง่ายดาย แต่เมื่อเกิดอาการผิดปกติขึ้นมาหลายคนก็เลือกที่จะโทรเรียกช่างหรือยกเครื่องเอาไปให้ที่ร้านซ่อม เนื่องจากความไม่รู้ของผู้ใช้ว่าปัญหาเกิดขึ้นนั้นเกิดจากอะไร ซึ่งบางครั้งอาการที่เกิดขึ้นนั้นอาจเกิดจากการที่สายไฟหลุดเท่านั้น ถ้าร้านที่เราไปซ่อมนั้นเป็นร้านที่ดี ก็จะบอกว่าปัญหาเกิดจากอะไรและไม่คิดค่าใช้จ่าย ทำให้เราไม่ต้องเสียเงิน แต่อย่างน้อยๆก็ต้องเสียเวลาและค่าเดินทาง แต่ถ้าเป็นร้านที่หวังแต่ผลประโยชน์ส่วนตัวก็จะโกงเราได้โดยบอกว่าปัญหานั้นเกี่ยวกับตัวเครื่องต้องเปลี่ยนอะไหล่ทำให้เราต้องเสียเงินเพิ่มขึ้นกับค่าอะไหล่ที่เปลี่ยน

ดังนั้นการศึกษาค้นคว้าความรู้ในเรื่องเหล่านี้จะมีประโยชน์ต่อบุคคลทั่วไปที่ทำงาน, ใช้งานหรือคิดจะซื้อเครื่องคอมพิวเตอร์ ซึ่งจะนำเสนอให้รู้จักชนิดของอุปกรณ์, การเลือกซื้อ, การประกอบ, การวิเคราะห์และแก้ปัญหา, การใช้งาน และ เกร็ดความรู้ต่างๆ โดยจะทำการสื่อโดยผ่านทางโปรแกรมประเภทเกมที่สามารถเข้าถึงผู้ใช้ได้มากที่สุด

1.2 วัตถุประสงค์

- 1) เพื่อเพิ่มความรู้เกี่ยวกับอุปกรณ์คอมพิวเตอร์
- 2) เพื่อให้รู้หน้าที่และคุณสมบัติของอุปกรณ์คอมพิวเตอร์
- 3) สามารถนำความรู้ที่ได้จากการเล่นเกมไปใช้ในชีวิจริงได้
- 4) เพื่อสอนวิธีการประกอบคอมพิวเตอร์
- 5) แนะนำการเลือกซื้ออุปกรณ์คอมพิวเตอร์

1.3 ขอบเขตของปัญหา

- 1) ใช้ Microsoft DirectX SDK เป็น Library สำคัญในการศึกษาวิธีการพัฒนาเกม
- 2) การสร้างโปรแกรมเกมนั้นจะพัฒนาบนเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป (PC- based) ที่มีระบบปฏิบัติการ Windows และเป็นแบบ Stand - alone

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รู้จักการพัฒนาเกมอย่างมีระบบ
- 2) รู้จักออกแบบเกมเพื่อให้สามารถนำไปใช้กับชีวิตจริงได้
- 3) มีความรู้เรื่องเกี่ยวกับคอมพิวเตอร์มากขึ้น
- 4) ให้ความรู้เพิ่มเติมกับบุคคลที่มาเล่นเกม

1.5 ขั้นตอนในการดำเนินการ

- 1.) ออกแบบเกมและคิด Story Board
- 2.) ศึกษา Microsoft Visual C++ และ Direct X
- 3.) สร้างเกมตามที่ได้ออกแบบไว้
- 4.) ทดสอบเกมที่สร้างขึ้น
- 5.) ปรับปรุงข้อผิดพลาดและแก้ไข



ตารางที่ 1.1 ตารางแสดงแผนงานในการทำโครงการปัญหาพิเศษ

ขั้นตอน \ เวลา	มี.ช.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
ออกแบบเกมและคิด Story Board	■	■								
ศึกษา Microsoft Visual C++ และ Direct X			■	■						
สร้างเกมตามที่ได้ออกแบบไว้					■	■	■			
ทดสอบเกมที่สร้างขึ้น								■		
ปรับปรุงข้อผิดพลาดและแก้ไข									■	■



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 โปรแกรม Microsoft DirectX

2.1.1 ความเป็นมา

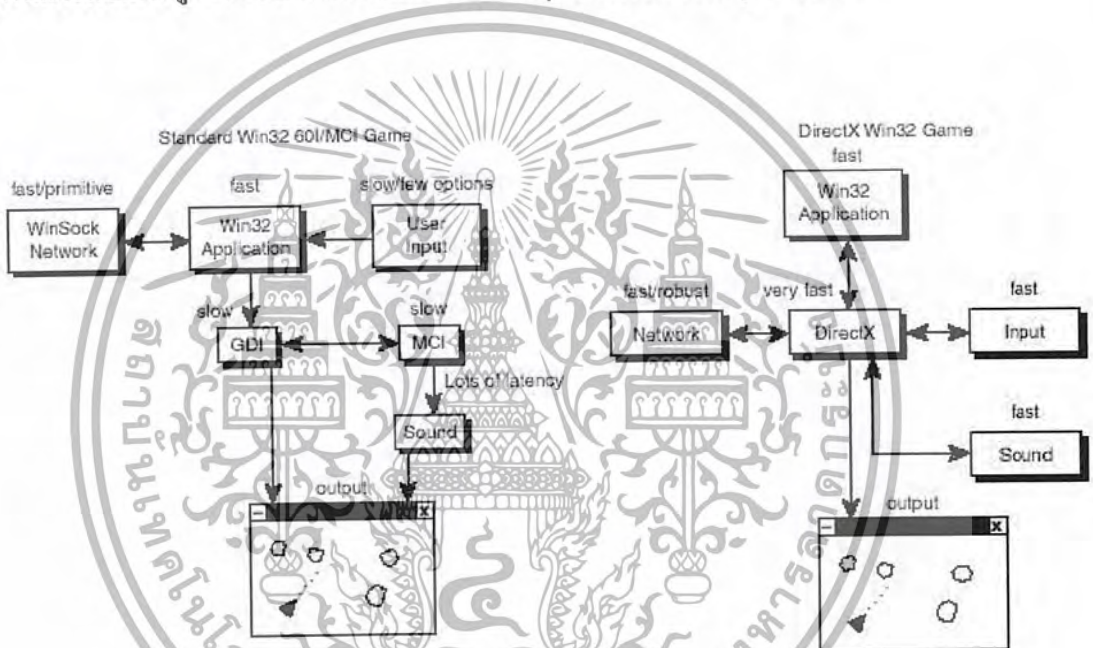
ในการเขียนโปรแกรมติดต่อกับอุปกรณ์ฮาร์ดแวร์ต่าง ๆ โดยตรงแต่เดิมนั้นเป็นสิ่งที่ยุ่งยากและซับซ้อนมาก เนื่องจากผู้พัฒนาเกมต้องศึกษาคำสั่งต่าง ๆ ของอุปกรณ์ฮาร์ดแวร์แต่ละรุ่น แต่ละยี่ห้อ โดยคำสั่งต่าง ๆ ที่ใช้ควบคุมอุปกรณ์ฮาร์ดแวร์นั้นเขียนจากภาษาแอสเซมบลี ซึ่งเป็นภาษาระดับล่างที่มีโครงสร้างของภาษาใกล้เคียงกับภาษาเครื่อง ซึ่งเป็นภาษาที่มีความสามารถในการควบคุมอุปกรณ์ฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ แต่เป็นภาษาที่ยากต่อการทำความเข้าใจของผู้พัฒนาเกมยิ่งไปกว่านั้นเมื่ออุปกรณ์ฮาร์ดแวร์นั้น ๆ ถูกพัฒนาขึ้นโดยการเพิ่มคำสั่งใหม่ ๆ เข้าไป หรือเปลี่ยนคำสั่งเดิมให้มีประสิทธิภาพมากขึ้น เป็นผลให้ผู้พัฒนาเกมจำเป็นต้องทำการศึกษาคำสั่งต่าง ๆ นั้นใหม่อีกครั้ง ทำให้ผู้พัฒนาเกมเสียเวลาและสร้างความลำบากในการพัฒนาโปรแกรมอีกด้วย

ในช่วงที่ไมโครซอฟท์สร้างระบบปฏิบัติการวินโดวส์ขึ้นมาใหม่นั้น สิ่งที่เป็นเอกลักษณ์ของวินโดวส์ คือ GUI (Graphic User Interface) โดยมี GDI (Graphic Device Interface) เป็นเครื่องมือสำหรับจัดการทางด้านภาพ ในขณะที่เดียวกันการพัฒนาเกมยังคงพัฒนาอยู่บนระบบปฏิบัติการคอสตั้นั้นบริษัทไมโครซอฟท์จึงสังเกตเห็นถึงการพัฒนาเกมบนวินโดวส์ขึ้น ในระบบปฏิบัติการวินโดวส์ตั้งแต่เวอร์ชัน 3.1 ขึ้นไป ได้มีการใช้ไดรฟ์เวอร์ของอุปกรณ์ฮาร์ดแวร์ขึ้น โดยไดรฟ์เวอร์ประกอบด้วยชุดคำสั่งเพื่อที่จะควบคุมการเข้าถึงอุปกรณ์ฮาร์ดแวร์ต่าง ๆ ทำให้การเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ฮาร์ดแวร์ของผู้พัฒนาเกมง่ายและสะดวกขึ้น แต่กระนั้นผู้พัฒนาเกมยังคงคาดหวังว่าระบบปฏิบัติการวินโดวส์รุ่นใหม่ ๆ จะพัฒนาชุดคำสั่งที่ทำหน้าที่เป็นตัวประสานงานระหว่างอุปกรณ์ฮาร์ดแวร์ต่าง ๆ โดยทำส่วนอินเทอร์เฟสเพียงตัวเดียวได้ ซึ่งจะทำให้การพัฒนาโปรแกรมสะดวกขึ้น เพราะผู้พัฒนาเกมเพียงแต่ทำการศึกษาคำสั่งของอินเทอร์เฟสตัวนั้นเพียงอย่างเดียวก็เพียงพอกับการพัฒนาโปรแกรมแล้ว

ด้วยเหตุนี้บริษัทไมโครซอฟท์ได้พัฒนาชุดคำสั่งพัฒนาซอฟต์แวร์ (Software Development Kits: SDK) ที่ทำหน้าที่เป็นตัวประสานงานระหว่างผู้พัฒนาเกมกับอุปกรณ์ฮาร์ดแวร์ในการพัฒนาซอฟต์แวร์บนวินโดวส์ เพื่อให้การเข้าถึงอุปกรณ์ฮาร์ดแวร์ทำได้ง่าย โดยตอนแรกได้เน้นเพื่อใช้พัฒนาเกมคอมพิวเตอร์โดยเฉพาะ โดยมีชื่อเรียกว่า “Game Kits” ต่อมาได้เพิ่มความสามารถทางด้านกราฟิกส์, งานด้านมัลติมีเดีย, งานด้านเน็ตเวิร์ก, งานด้านการรับข้อมูลอินพุต และงานกราฟิกส์ 3 มิติ ดังนั้นชุดคำสั่งนี้จึงถูกเรียกใหม่ว่า “Microsoft DirectX SDK” และปัจจุบันบริษัทไมโครซอฟท์ได้พัฒนา DirectX จนถึงรุ่น 9.0 แล้ว

2.1.2 หลักการของ DirectX

DirectX คือ ชุดคำสั่งของ Dynamic Link Library (DLL) ซึ่งทำหน้าที่เป็นตัวประสานระหว่าง ชั้นแอปพลิเคชันกับอุปกรณ์ฮาร์ดแวร์ให้มีความสัมพันธ์กันเสมือนเป็นชั้นเดียวกัน เนื่องจาก ต้องการให้แอปพลิเคชันสามารถทำงานกับอุปกรณ์ฮาร์ดแวร์ได้อย่างรวดเร็ว แต่เดิมบริษัท ไมโครซอฟท์มีชุดคำสั่งที่ใช้ติดต่อกับการ์ดแสดงผลซึ่งเรียกว่า GDI (Graphic Device Interface) ซึ่ง มีการทำงานคล้ายกับ DirectX โดยติดต่อกับอุปกรณ์ฮาร์ดแวร์ต่าง ๆ โดยตรงโดยไม่ต้องใช้ ไดรฟ์เวอร์ของการ์ดจอเอง แต่มีความเร็วช้ากว่า และมีคุณสมบัติบางอย่างที่ไม่สามารถทดแทน DirectX ได้ เช่น ในกรณีที่แอปพลิเคชันเรียกใช้อุปกรณ์ฮาร์ดแวร์แต่ DirectX ตรวจสอบไม่พบอุปกรณ์ ฮาร์ดแวร์ตัวนั้น DirectX จะสามารถทำการจำลองอุปกรณ์ฮาร์ดแวร์ตัวนั้น เสมือนว่ามีอุปกรณ์ ฮาร์ดแวร์ตัวนั้นอยู่ในระบบ แต่ในทางตรงกันข้ามคุณสมบัตินี้ไม่มีอยู่ใน GDI



รูปที่ 2.1 การทำงานของ DirectX เปรียบเทียบกับ GDI/MCI

ยิ่งไปกว่านั้น DirectX ยังมีคุณสมบัติพิเศษโดยเกมที่เขียนด้วย DirectX เวอร์ชันต่ำ สามารถ นำมาประมวลผลใน DirectX เวอร์ชันสูงก็ได้ อีกทั้งบริษัท ไมโครซอฟท์ยังพัฒนาเทคโนโลยีของ DirectX ให้สนับสนุนการทำงานเชิงวัตถุ เพื่อให้สามารถทำงานกับภาษาอื่น ๆ ได้ และยังนำ เทคโนโลยี COM (Component Object Model) มาใช้งาน ซึ่งทำให้ผู้พัฒนาเกมได้รับความ สะดวกสบายในการใช้งานเป็นอย่างยิ่ง และยังสามารถควบคุมและสร้างการทำงานได้อย่างอิสระ

2.1.3 องค์ประกอบของ DirectX

2.1.3.1 ชั้นติดต่อกับอุปกรณ์ฮาร์ดแวร์ (Hardware Abstraction Layer: HAL)

HAL เป็นชั้นที่ใช้ติดต่อกับอุปกรณ์ฮาร์ดแวร์ระดับล่างสุด ซึ่งมีไคร์ฟเวอร์อุปกรณ์ฮาร์ดแวร์ของผู้ผลิตนั้น ๆ เพื่อใช้ควบคุมอุปกรณ์ฮาร์ดแวร์โดยตรง และให้ประสิทธิภาพสูงสุด เพราะว่าเป็นชั้นที่ติดต่อกับอุปกรณ์ฮาร์ดแวร์โดยตรง

2.1.3.2 ชั้นจำลองอุปกรณ์ฮาร์ดแวร์ (Hardware Emulation Layer: HEL)

HEL เป็นชั้นที่อยู่ถัดจากชั้น HAL เข้ามา ซึ่งเป็นชั้นที่ทำการจำลองอัลกอริทึมของอุปกรณ์ฮาร์ดแวร์นั้น ๆ ในกรณีที่ DirectX ตรวจไม่พบ แต่การทำงานจะช้าลงเพราะเป็นการจำลองการทำงานจริงของอุปกรณ์ฮาร์ดแวร์นั้น

2.1.3.3 การแสดงภาพบนจอภาพ (Direct Draw)

เป็นส่วนที่แสดงภาพกราฟฟิกสู่ทุกชนิดออกทางหน้าจอเบื้องต้น จึงถือเป็นส่วนที่สำคัญที่สุด เพราะว่ามันทำหน้าที่ควบคุมการ์ดแสดงผล โดยที่ Direct Draw รู้จักการเซต โหมดของจอภาพทุก ๆ โหมด อีกทั้งยังสนับสนุนการจัดการสี, การกำหนดขอบเขต และการทำแอนิเมชัน

2.1.3.4 การแสดงเสียงทั่วไป (Direct Sound)

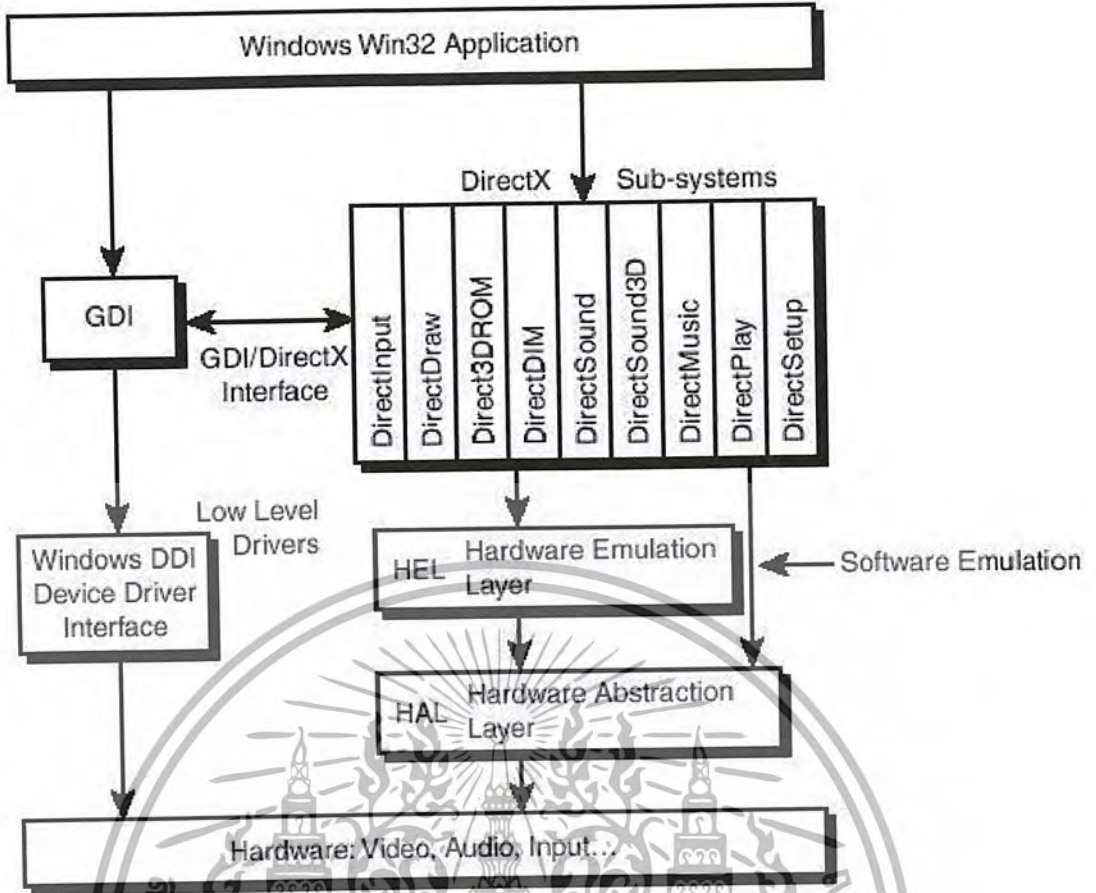
เป็นส่วนที่สนับสนุนเสียงแบบดิจิทัล แต่ไม่สนับสนุนเสียงแบบ MIDI ใช้จัดการเสียงแบบโมโนและแบบสเตอริโออย่างมีประสิทธิภาพ ซึ่งประกอบด้วยหน่วยความจำ และการผสมเสียงฮาร์ดแวร์

2.1.3.5 การแสดงเสียงแบบสามมิติ (Direct Sound 3D)

เสียงแบบ 3 มิติเป็นส่วนที่มีความสามารถเหนือกว่า Direct Sound ทำให้เราสามารถกำหนดตำแหน่งของเสียงแบบ 3 มิติในอวกาศได้ ซึ่งต้องมีการคำนวณทางคณิตศาสตร์ โดยคำนวณพิกัดตำแหน่งของเสียงในระบบ 3 มิติ ทำให้เสียงมีความสมจริงมากขึ้น เสมือนวัตถุอยู่รอบตัวเรา

2.1.3.6 การแสดงเสียงเพลง (Direct Music)

เป็นส่วนที่สนับสนุนเสียงแบบ MIDI อีกทั้งยังมีระบบ DLS (Downloadable Sound) ซึ่งเป็นระบบที่ทำให้คุณสามารถสร้างเครื่องมือที่สร้างเสียงแบบดิจิทัลที่สามารถควบคุมด้วยเทคโนโลยี MIDI ได้



รูปที่ 2.2 องค์ประกอบของ DirectX

2.1.3.7 การรับข้อมูลอินพุต (Direct Input)

ระบบนี้จะควบคุมอุปกรณ์อินพุตทุกชนิดไม่ว่าจะเป็นเมาส์, คีย์บอร์ด, จอยสติ๊กส์, แพดเดิล, สเปซบอล ฯลฯ ซึ่งไปกว่านั้นมันยังสนับสนุนอุปกรณ์ Force Feedback ซึ่งมีเครื่องกระตุ้นไฟฟ้าและเซ็นเซอร์ของแรงสั่นสะเทือนที่ทำให้ผู้เล่นรู้สึกถึงแรงนั้น

2.1.3.8 การเชื่อมต่อเครือข่าย (Direct Play)

เป็นส่วนที่จัดการเกี่ยวกับเครือข่าย โดยทำให้ผู้พัฒนาเกมสามารถเชื่อมต่อแบบย่อโดยใช้อินเทอร์เน็ต, โมเด็ม, ไคเร็กคอนเนกชัน ฯลฯ ซึ่งทำให้ไม่ต้องเขียนไดรฟ์เวอร์ หรือไม่ต้องใช้ซ็อกเก็ตเลย นอกจากนั้นมันยังสนับสนุนหลักการของเซสชัน ซึ่งเป็นเกมที่กำลังดำเนินอยู่ และล็อบบี้ ซึ่งเป็นสถานที่ที่ผู้เล่นเกมเล่นอยู่ด้วยกัน

Direct Play สนับสนุนสถาปัตยกรรมเครือข่ายแบบหลายผู้เล่น (Multiplayer Network Architecture) โดยมันรับและส่ง packets เท่านั้น

2.1.3.9 การแสดงภาพ 3 มิติในโหมดระดับสูง(Direct 3D Retained Mode:Direct 3D RM)

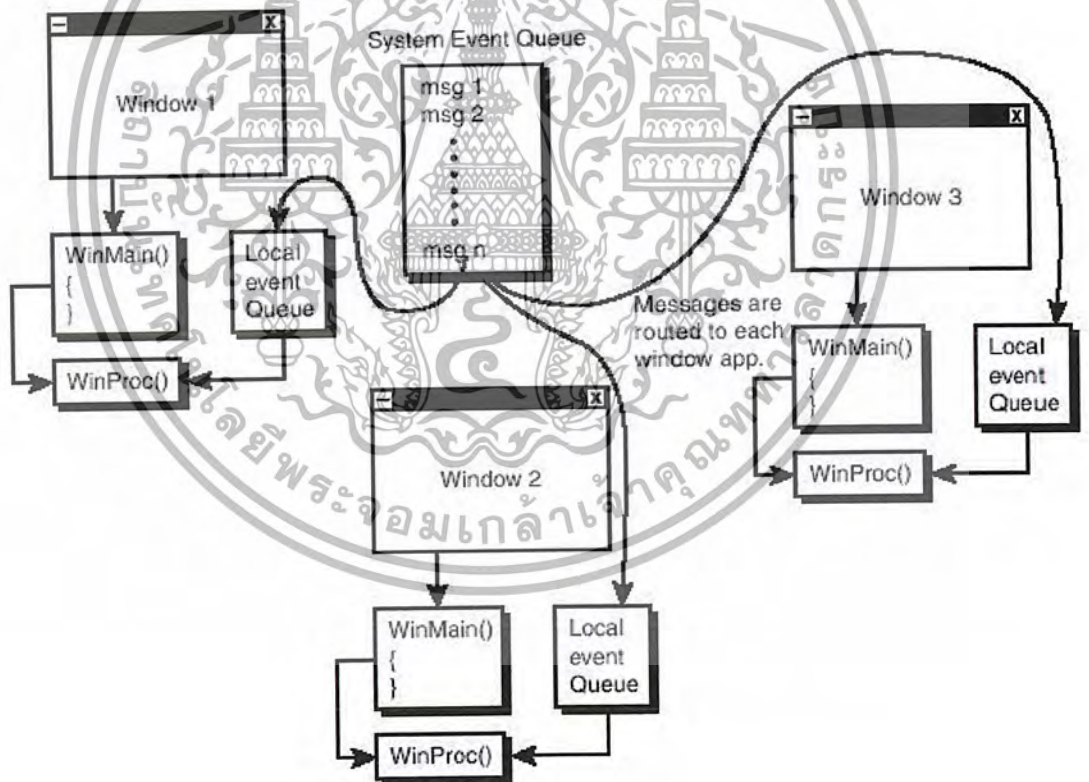
เป็นระบบ 3 มิติเชิงวัตถุและเชิงโครงร่างระดับสูง ที่ช่วยให้ผู้พัฒนาเกมสามารถเขียนโปรแกรม 3 มิติ แบบ walkthrough ได้

2.1.3.10 การแสดงภาพ 3 มิติในโหมดระดับต่ำ(Direct 3D Immediate Mode:Direct 3D IM)

เป็นระบบ 3 มิติระดับต่ำที่สนับสนุนการทำงานกับ framewar ด้วย OpenGL ผ่านฟังก์ชัน DrawPrimitive() ซึ่งมีอยู่ใน DirectX 5.0 ขึ้นไป โดยทำหน้าที่ส่ง strip ภาพไปประมวลผลยัง rendering engine และเปลี่ยนสถานะด้วยฟังก์ชัน call แทนการประมวลผลบัฟเฟอร์

2.2 การเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์

วินโดวส์เป็นระบบปฏิบัติการแบบ event-based โดยมี message ที่ส่งไปยังวินโดวส์เพื่อบอกให้รู้ว่าต้องทำอะไรและทำอย่างไร เป็นต้น ดังนั้นเราจึงควรศึกษาโครงสร้างของ WIN32 API ก่อนที่จะสร้างเกมขึ้นมาตามหัวข้อต่อไปนี้



รูปที่ 2.3 กลไกการส่ง message ในวินโดวส์

2.2.1 ไฟล์ไลบรารีของ Win32 API

เมื่อผู้พัฒนาเกมสร้างแอปพลิเคชันใด ๆ ขึ้นมา คอมไพเลอร์จะเชื่อมโยงกับส่วนของไฟล์ไลบรารีให้เองทั้งหมด ยกเว้นไฟล์ไลบรารีที่ไม่ได้ถูกกำหนดไว้ก่อนในตอนแรก เช่น เมื่อต้องการสร้างแอปพลิเคชันที่เกี่ยวกับการแสดงเสียงซึ่งต้องใช้ไฟล์ไลบรารี dsound.lib แต่คอมไพเลอร์ยังไม่รู้จักกับไฟล์ไลบรารีนี้ ดังนั้นผู้พัฒนาเกมต้องเพิ่มไฟล์นี้เข้าไปในรายการเอง

2.2.2 การเพิ่มไฟล์เข้าไปในแอปพลิเคชัน

ไฟล์ที่จะเพิ่มเข้าไปในแอปพลิเคชันเป็นไฟล์ไลบรารีและไฟล์เฮดเดอร์ที่จำเป็นสำหรับแอปพลิเคชันนั้นๆ ซึ่งโดยทั่วไปผู้พัฒนาเกมจะกำหนดไฟล์ไลบรารีและไฟล์เฮดเดอร์พื้นฐานไว้ดังนี้

```
#define WIN32_LEAN_AND_MEAN
#include<windows.h>
#include<windowsx.h>
```

ผู้พัฒนาเกมอาจจะต้องเพิ่มไฟล์ที่จำเป็นสำหรับแอปพลิเคชันนั้นๆ เข้าไปอีก เช่น stdio.h, conio.h หรือ math.h เป็นต้น

2.2.3 ฟังก์ชันการทำงานของวินโดวส์

ฟังก์ชันหลักของวินโดวส์ทำหน้าที่เหมือนกับฟังก์ชัน main() ในภาษา C/C++ ซึ่งทำหน้าที่กำหนดค่าเริ่มต้นให้กับวินโดวส์ โดยการประกาศต้นแบบของฟังก์ชัน WinMain() เป็นดังนี้

```
int WINAPI WinMain(HINSTANCE hinstance, //handle of current of instance
                  HINSTANCE hpreinstance, //handle of previous instance
                  LPSTR lp cmdline, //address of command line
                  int ncmdshow ); //show state of window
```

2.2.4 คลาสของวินโดวส์

คลาสของวินโดวส์เป็นคลาสที่ประกอบด้วยองค์ประกอบทั่วไปของวินโดวส์ ทำให้ผู้พัฒนาเกมสามารถสร้างวินโดวส์ได้ตามคลาสของวินโดวส์ที่กำหนดขึ้น โดยการประกาศโครงสร้างคลาสของวินโดวส์เป็นดังนี้

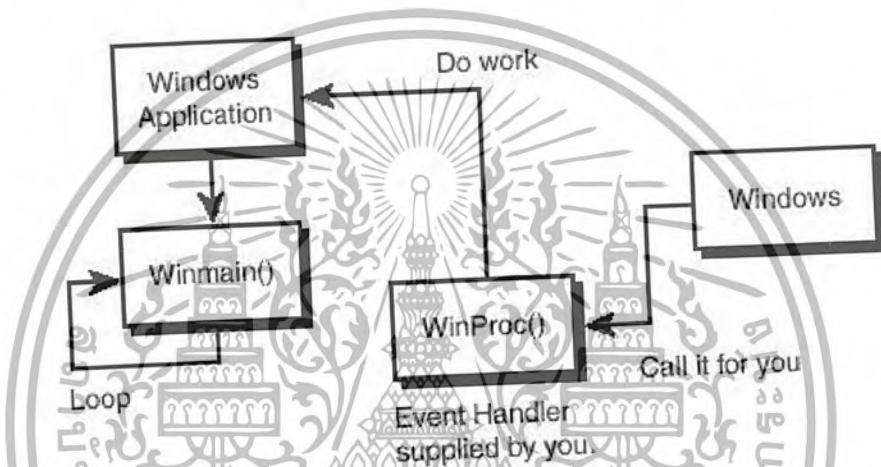
```
typedef struct _WNDCLASS
{
    UINT style; // style flags
    WNDPROC lpfnWndProc; // pointer to event handler
    Int cbClsExtra; // extra byte
```

```

Int          cbWndExtra; //extra byte
HANDLE      hInstance; //handle of instance
HICON       hIcon; //handle to Icon
HCURSOR     hCursor; //handle to cursor
HBRUSH      hbrBackground; //handle to background brush
LPCTSTR     lpszMenuName; //name of Menu
LPCTSTR     lpszClassName; //name of class
} WNDCLASS;

WNDCLASS    wndclass // a declaration of a window class

```



รูปที่ 2.4 การทำงานของฟังก์ชัน Winmain() กับ Event Handler

2.2.5 การขึ้นทะเบียนคลาสของวินโดวส์

เป็นการขึ้นทะเบียนคลาสของวินโดวส์เพื่อเอาไว้เรียกใช้ในภายหลังเมื่อผู้พัฒนาเกมเรียกใช้ฟังก์ชัน `CreateWindow` หรือ `CreateWindowEx` ถ้าขึ้นทะเบียนสำเร็จฟังก์ชันจะคืนค่า `atom` มาให้ แต่ถ้าไม่สำเร็จฟังก์ชันจะคืนค่าศูนย์มาให้แทน การรีจิสเตอร์คลาสทำได้ดังนี้

```

if (RegisterClass(&wndclass) == 0)
{
    //error
} // end if

```

การประกาศโครงสร้างของ `RegisterClass` เป็นดังนี้

```
typedef struct tagWNDCLASS
```

```
{
    UINT        style;
    WNDPROC     lpfnWndProc;
    Int         cbClsExtra;
    Int         cbWndExtra;
    HANDLE      hInstance;
    HICON       hIcon;
    HCURSOR     hCursor;
    HBRUSH      hbrBackground;
    LPCTSTR     lpzMenuName;
    LPCTSTR     lpzClassName;
} WNDCLASS;
```

2.2.6 การสร้างวินโดวส์ออปเจกต์

การสร้างวินโดวส์ออปเจกต์ขึ้นมาต้องทำการเรียกฟังก์ชัน `CreateWindow()` โดยการประกาศโครงสร้างของฟังก์ชัน `CreateWindow()` เป็นดังนี้

```
HWND CreateWindow (
    LPCTSTR     lpClassName, //pointer to class name (string)
    LPCTSTR     lpWindowName, //pointer to window title (string)
    DWORD       dwStyle, // windows style flags
    int         x, // ตำแหน่งแนวแกน x ของวินโดวส์
    int         y, // ตำแหน่งแนวแกน y ของ วินโดวส์
    int         nWidth, // ความกว้างของวินโดวส์
    int         nHeight, // ความสูงของวินโดวส์
    HWND        hWndParent, //handle to parent (usually NULL)
    HMENU       hMenu, //handle to menu (usually NULL)
    HANDLE      hInstance, //handle to application instance
    LPVOID      lpParam //pointer to startup creation data (NULL)
);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ค่าแฟลกของพารามิเตอร์ dwStyle

ค่าแฟลกของ dwStyle	ความหมาย
WS_BORDER	สร้างวินโดวส์ที่มีเส้นขอบบาง
WS_CAPTION	สร้างวินโดวส์ที่มีไทเทิลบาร์ (รวมทั้ง WS_BORDER)
WS_DISABLE	สร้างวินโดวส์ที่ไม่สามารถรับอินพุตของผู้ใช้ได้
WS_OVERLAPPED	สร้างวินโดวส์ที่ซ้อนเหลื่อมกันได้
WS_POPUP	สร้างวินโดวส์แบบ Pop-up
WS_VISIBLE	สร้างวินโดวส์ที่สามารถมองเห็นได้

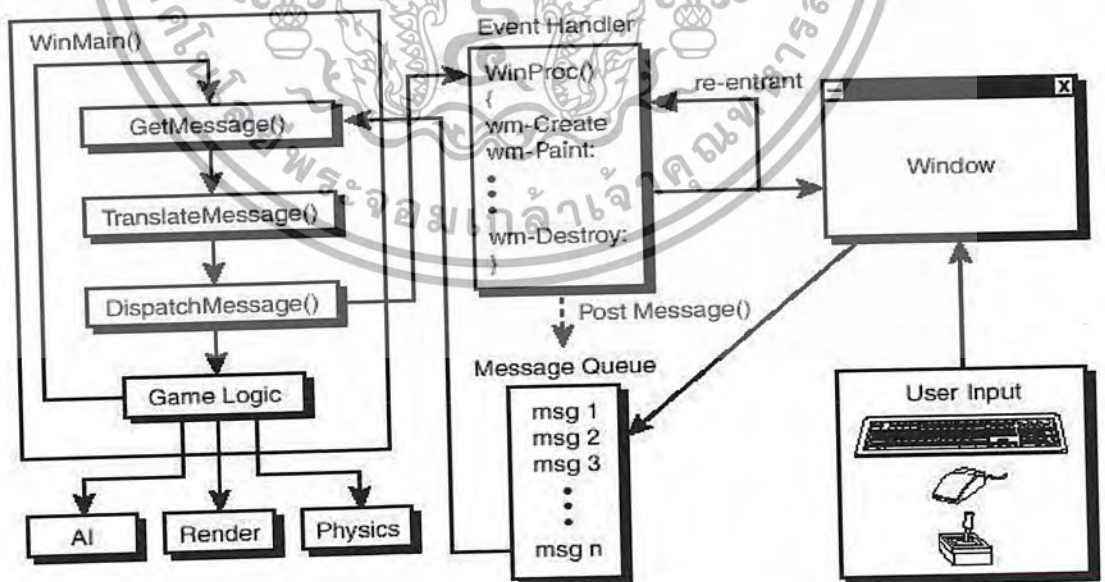
2.2.7 การแสดงวินโดวส์บนจอภาพ

การแสดงวินโดวส์บนจอภาพ ในกรณีที่ตั้งค่าแฟลกของพารามิเตอร์ dwStyle ไม่เป็น WS_VISIBLE ผู้พัฒนาเกมต้องทำการเรียกฟังก์ชัน ShowWindow() ขึ้นมาด้วยเสมอ

```
BOOL ShowWindow(HWND hWnd, // handle of window
                INT nCmdShow); // show state of window
```

2.2.8 ลูปของเหตุการณ์ในวินโดวส์ (Event Loop)

ลูปของเหตุการณ์หลักเป็นลูปที่คอยรับ message จากวินโดวส์แล้วนำไปเก็บไว้ในคิวของ message จากนั้นส่งต่อไปยัง Event Handler เพื่อทำหน้าที่ตาม message ที่ได้รับ ดังรูปที่ 2.5



รูปที่ 2.5 ลูปเหตุการณ์หลัก

จากรูป 2.5 ฟังก์ชันใน WinMain() ประกอบด้วย 3 ฟังก์ชันย่อย คือ

2.2.8.1 ฟังก์ชันรับ Message หรือ GetMessage()

เป็นฟังก์ชันที่ทำหน้าที่รับ message จากคิวของ message ที่เกี่ยวข้องกับวินโดวส์ออปเจคนั้นมา และทำให้อยู่ในรูปโครงสร้างเฉพาะ โดยฟังก์ชันจะคืนค่าจริงกลับมาเมื่อฟังก์ชันทำงานสำเร็จ ซึ่งถ้า message วางฟังก์ชันนี้จะรอจนกว่าจะมี message เข้ามาใหม่ การประกาศรูปแบบของฟังก์ชันเป็นดังนี้

```

BOOL GetMessage (LPMSG lpMsg, //pointer to message structure
                 HWND hWnd, //handle of window
                 UINT wMsgFilterMin, //first message
                 UINT wMsgFilterMax); //last message

```

2.2.8.2 ฟังก์ชันแปล Message ที่ได้รับ หรือ TranslateMessage()

เป็นฟังก์ชันที่แปล message ที่อยู่ในคิว จากประเภทคีย์เสมือนให้เป็นประเภทอักขระ โดย message ประเภทอักขระจะถูกทำไปไว้ในคิวของ message เพื่อให้ฟังก์ชัน GetMessage() อ่าน message ในครั้งต่อไป โดยฟังก์ชันจะคืนค่าจริงกลับมาเมื่อฟังก์ชันทำงานสำเร็จ การประกาศรูปแบบของฟังก์ชันเป็นดังนี้

```

BOOL TranslateMessage(CONST MSG *lpmsg); //pointer to message structure

```

2.2.8.3 ฟังก์ชันจัดส่ง Event หรือ DispatchMessage()

เป็นฟังก์ชันที่ทำการส่ง message ไปยังฟังก์ชัน WindowProc() เพื่อทำงานตาม message ที่ได้รับ โดยฟังก์ชันจะคืนค่าเฉพาะที่มีความหมายขึ้นกับ message ที่ส่งไปประมวลผล และเราจะไม่สนใจค่าที่ส่งกลับ การประกาศรูปแบบของฟังก์ชันเป็นดังนี้

```

LONG DispatchMessage ( CONST MSG *lpmsg); //pointer to message structure

```

2.2.9 ตัวจัดการเหตุการณ์ต่าง ๆ หรือ Event handler

เป็นฟังก์ชัน WindowProc() ซึ่งจัดการกับทุก ๆ message ของวินโดวส์ที่สนใจในขณะนั้น โดยฟังก์ชันจะคืนผลลัพธ์ของการประมวลผล message มาให้ และค่าที่คืนนั้นจะขึ้นอยู่กับ message ที่ได้รับ การประกาศรูปแบบของฟังก์ชันเป็นดังนี้

```

LRESULT CALLBACK WindowProc ( HWND hwnd, //the window
                              UINT msg, //the message itself
                              WPARAM wParam, //more info on message
                              LPARAM lParam ); //more info on message

```

2.3 Event Driven Programming

สิ่งที่ต้องคำนึงถึงเป็นอย่างแรกในการเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์ คือ Event Driven Programming ซึ่งเป็นการโปรแกรมที่ทำงานตามเหตุการณ์ที่เกิดขึ้น โดยระบบปฏิบัติการจะเป็นตัวบอกโปรแกรมว่ามีเหตุการณ์อะไรเกิดขึ้นบ้าง หากโปรแกรมได้รับเหตุการณ์ที่สนใจ มันก็จะทำงานตามที่ผู้พัฒนาเกมได้เขียนโปรแกรมไว้

หากเปรียบเทียบกับกรเขียนโปรแกรมโดยทั่วไปซึ่งจะทำงานแบบเรียงลำดับตามผู้พัฒนาเกมได้เขียนโปรแกรมไว้ แต่สำหรับการเขียนโปรแกรมตามเหตุการณ์จะให้ผู้เขียนโปรแกรมควบคุมโปรแกรมได้มากกว่าผู้ใช้

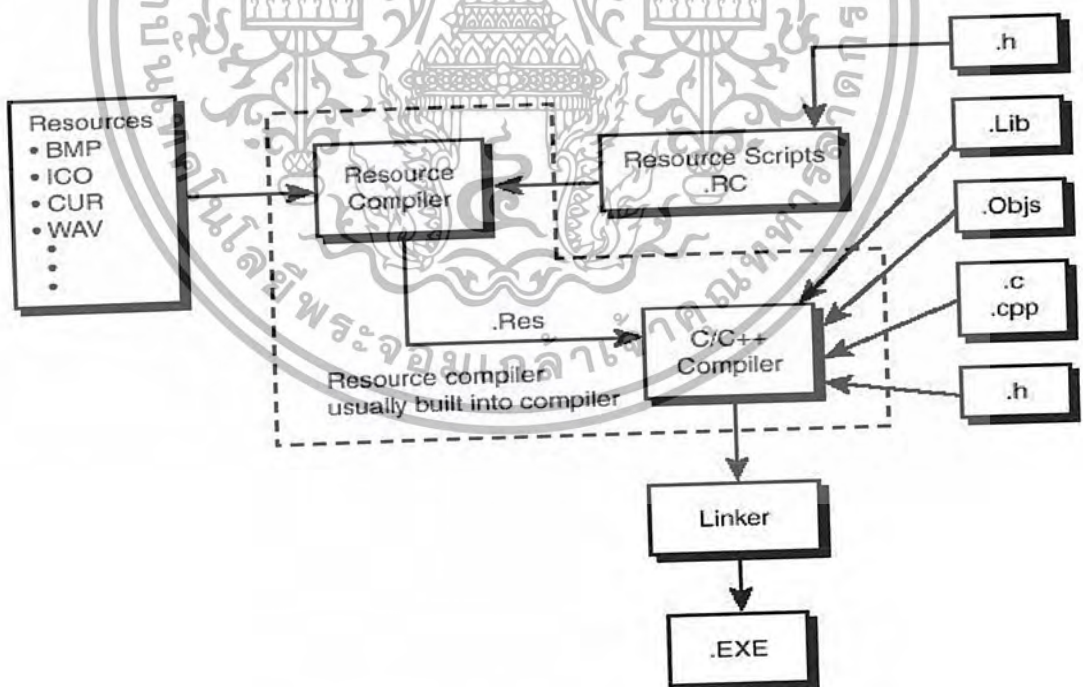
2.4 วิธีการแสดงภาพบนจอภาพ

2.4.1 ความรู้เบื้องต้นในการแสดงภาพด้วยคำสั่งของ Direct Draw

Direct Draw เป็นไลบรารีของชุดคำสั่งพัฒนาโปรแกรม DirectX ของภาษาซี++ ซึ่งประกอบด้วยไฟล์เฮดเดอร์ ddraw.h และไฟล์ไลบรารี ddraw.lib เวลาเรียกใช้ให้ประกาศดังนี้

```
#include <ddraw.h>
```

และกำหนดไฟล์ไลบรารี ddraw.lib ให้กำหนดในไคเรกทอรีลิงค์ไลบรารี



รูปที่ 2.6 ทรัพยากรที่ใช้สร้างแอปพลิเคชันของ Win32 DirectX

2.4.2 การสร้างออปเจกของ DirectDraw

ผู้พัฒนาเกมจะสร้างออปเจกของ Direct Draw หลังจากที่สร้างออปเจกของวินโดวส์แล้ว โดยรูปแบบของฟังก์ชันเป็นดังนี้

```
HRESULT DirectDrawCreate (GUID FAR *lpGUID, //guid of object
LPDIRECTDRAW FAR *lpDD, //receive interface
IUnknown FAR *pUnkOuter); //com stuff
```

ฟังก์ชันนี้จะทำการสร้างออปเจกของ DirectDraw ไว้ที่ตำแหน่งแอดเดรส lpDD และแสดงผลแบบ lpGUID ซึ่งถ้ากำหนดค่าเป็น NULL การแสดงผลจะเป็นแบบ Default ฟังก์ชันนี้จะแจ้งผลผ่านค่า HRESULT เป็นชนิดข้อมูลชนิดหนึ่งของ DirectX ซึ่งใช้สำหรับตรวจสอบการทำงานของฟังก์ชัน ไม่ว่าจะทำงานสำเร็จหรือไม่ก็ตาม ถ้าสำเร็จจะส่งค่า DD_OK หรือ DirectDraw Okay หากฟังก์ชันคืนค่าอื่น ๆ ที่ไม่ใช่ค่านี้ นั้นหมายถึงการทำงานของฟังก์ชันมีความผิดพลาดเกิดขึ้น

ตัวอย่าง

```
LPDIRECTDRAW lpdd; //สร้างตัวแปรที่ใช้กับค่าแอดเดรสออปเจกของ DirectDraw
//สร้างออปเจกและตรวจสอบข้อผิดพลาด
if (DirectDrawCreate(NULL, &lpdd, NULL) != DD_OK)
{
    //จัดการกับข้อผิดพลาด
}
```

การเรียกฟังก์ชันของออปเจกเป็นดังนี้

```
lpdd->ชื่อฟังก์ชัน();
หลังจากนั้นต้องลบออปเจกเพื่อคืนหน่วยความจำให้กับระบบ โดยใช้ฟังก์ชัน Release()
lpdd->Release();
```

2.4.3 การกำหนดการแสดงผลของวินโดวส์ด้วยฟังก์ชัน SetCooperativeLevel ของ Direct Draw

การเขียนโปรแกรมเพื่อทำงานบนระบบปฏิบัติการวินโดวส์นั้น ผู้พัฒนาเกมต้องจัดสรรทรัพยากรต่าง ๆ โดยใช้ฟังก์ชันดังนี้

```
HRESULT SetCooperativeLevel (HWND hWnd, DWORD dwFlags);
```

HWND hWnd คือ โปรแกรมที่กำลังทำงานอยู่ในขณะนั้น (Handle) โดยปกติแล้วผู้พัฒนาเกมจะกำหนดค่าตัวแปรนี้ให้คอมไพเลอร์สร้างให้โดยอัตโนมัติ

ตารางที่ 2.2 ค่าแฟลก Cooperative level ของ DirectDraw

ค่าแฟลกของ dwFlag	ความหมาย
DDSCL_ALLOWMODEX	อนุญาตให้แสดงภาพในโหมด X ได้ (320x200, 320x240) โดยค่านี้จะต้องใช้คู่กับ DDSCL_EXCLUSIVE และ DDSCL_FULLSCREEN
DDSCL_ALLOWREBOOT	อนุญาตให้ใช้ปุ่ม Ctrl+Alt+Delete ในขณะที่อยู่ในโหมด Exclusive (fullscreen) เพื่อ Reboot
DDSCL_EXCLUSIVE	กำหนดให้โปรแกรมมีลำดับความสำคัญสูงสุด โดยจะต้องทำงานคู่กับ DDSCL_FULLSCREEN
DDSCL_FULLSCREEN	อนุญาตให้เขียนข้อมูลลงบนหน่วยความจำแสดงผลทั้งหมด
DDSCL_NORMAL	บอกให้ระบบปฏิบัติการวินโดวส์ทราบว่าโปรแกรมนี้เป็นเหมือนแอปพลิเคชันทั่วไป โดยผู้พัฒนาเกมจะไม่สามารถกำหนดค่า DDSCL_ALLOWMODEX DDSCL_EXCLUSIVE หรือ DDSCL_FULLSCREEN ได้
DDSCL_NOWINDOWCHANGES	บอกว่าโปรแกรมนี้ไม่สามารถย่อหรือขยายขนาดวินโดวส์ของโปรแกรมอื่น ๆ ในขณะที่โปรแกรมนี้ทำงานอยู่ได้

ตัวอย่าง

```
lpdd->SetCooperativeLevel(hwnd, DDSCL_ALLOWMODEX|
DDSCL_FULLSCREEN|
DDSCL_EXCLUSIVE|
DDSCL_ALLOWREBOOT);
```

โปรแกรมนี้เป็นแบบโหมด X แบบเต็มหน้าจอ อีกทั้งยังมีความลำดับสำคัญสูงสุด และยังสามารถออกจากโปรแกรมได้โดยใช้ปุ่ม Ctrl+Alt+Delete อีกด้วย

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.4.4 การกำหนดโหมดการแสดงผล

เป็นฟังก์ชันที่ใช้เปลี่ยนโหมดการแสดงผล ฟังก์ชันนี้จะแจ้งผลผ่านค่า HRESULT ถ้าสำเร็จจะส่งค่า DD_OK หรือ DirectDraw Okay รูปแบบของฟังก์ชันเป็นดังนี้

```
HRESULT SetDisplayMode(DWORD dwWidth,
                      DWORD dwHeight,
                      DWORD dwBPP,
                      DWORD dwRefreshRate,
                      DWORD dwFlags);
```

การกำหนดพารามิเตอร์เป็นดังนี้

พารามิเตอร์ dwWidth กำหนดความกว้างของการแสดงผล โดยมีหน่วยเป็นพิกเซล

พารามิเตอร์ dwHeight กำหนดความยาวของการแสดงผล โดยมีหน่วยเป็นพิกเซล

พารามิเตอร์ dwBPP กำหนดจำนวนข้อมูลสีต่อหนึ่งพิกเซล (8,16,32 bit)

ส่วนพารามิเตอร์ dwRefreshRate และ dwFlags ให้กำหนดเป็น 0 ทั้งหมดเนื่องจากคอมพิวเตอร์จะทำการตรวจสอบค่าที่เหมาะสมให้เอง

การใช้งานฟังก์ชันของ Direct Draw มีตามขั้นตอนดังนี้

- 1.) สร้างออบเจกต์ DirectDraw
- 2.) กำหนดระดับความสำคัญและจัดสรรทรัพยากร โดยใช้ฟังก์ชัน SetCooperativeLevel
- 3.) ทำการกำหนดขนาดของการแสดงผลโดยใช้ฟังก์ชัน SetDisplayMode

ตัวอย่าง

```
LPDIRECTDRAW lpdd;
if (DirectDrawCreate(NULL, &lpdd, NULL) != DD_OK)
{ //ERROR/}
lpdd->SetCooperativeLevel(hwnd, DDSCL_ALLOWMODEX|
DDSCL_FULLSCREEN|
DDSCL_EXCLUSIVE|
DDSCL_ALLOWREBOOT);
if ((lpdd->SetDisplayMode(640, 480, 8)) != DD_OK)
{ //ERROR }
```

2.4.5 การประยุกต์ใช้ DirectDraw

DirectDraw สามารถนำมาใช้ได้ ในหลายๆ ด้าน แต่ด้านที่ถือว่ามีนิยมนิยมที่สุดและนำ DirectDraw มาใช้มากที่สุดก็คือด้านการสร้างเกมคอมพิวเตอร์ในปัจจุบันเกมส่วนมากนำ DirectDraw มาใช้ ในการสร้าง Animation ซึ่ง DirectDraw สามารถเข้าถึง Hardware ด้านการแสดงผลได้โดยตรง ดังนั้น การทำงานต่างๆ จึงรวดเร็วมาก และผู้ใช้ไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับคำสั่งในระดับ

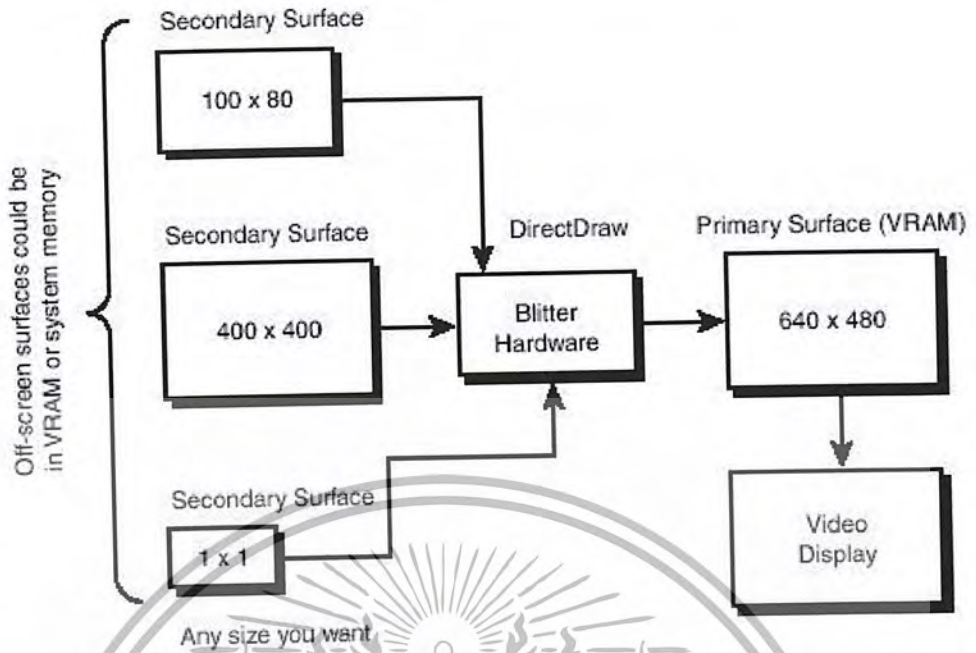
Hardware เลย

2.4.6 การวาดภาพด้วยคำสั่ง DirectDraw

DirectDraw มี Surface เป็นหน่วยความจำที่เก็บรูปภาพที่จะแสดงผลไว้ใน Primary Surface ซึ่ง เป็น Surface ที่เรามองเห็นขณะนั้นบนจอภาพ และเราสามารถกำหนด Surface ได้มาก 1 Surface แต่ต้องขึ้นอยู่กับหน่วยความจำที่มีอยู่ในระบบ

2.4.7 หลักการสร้าง Surface

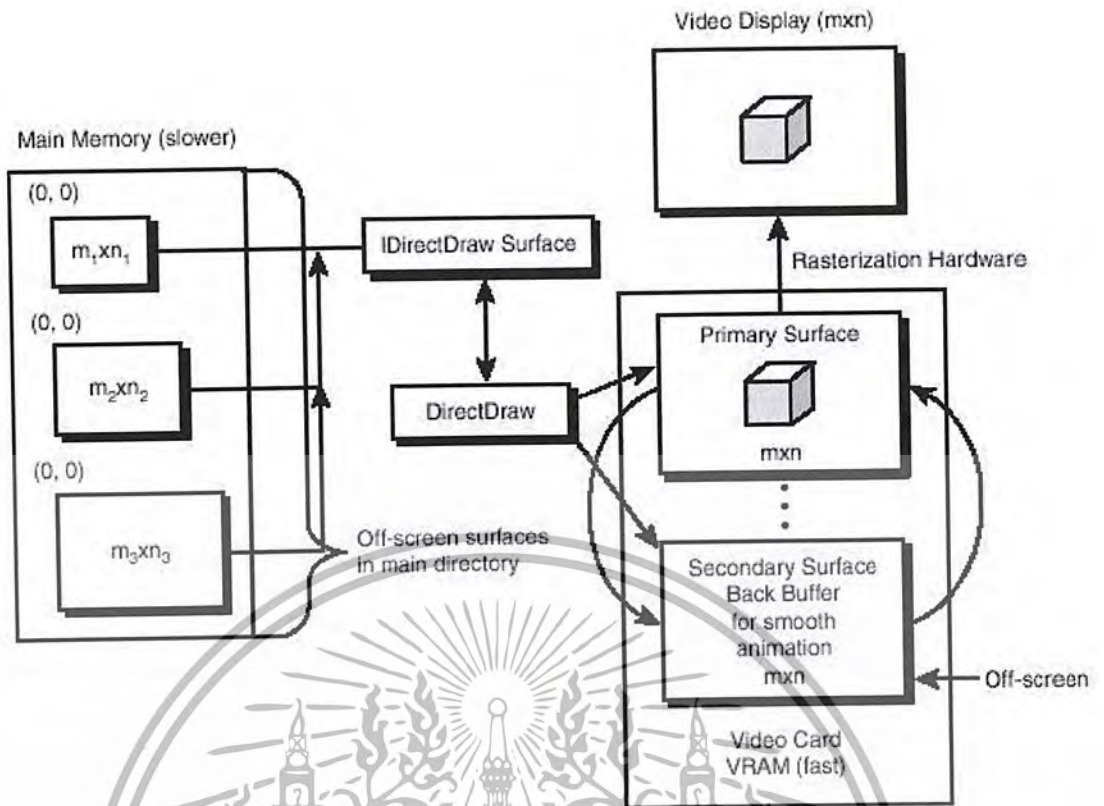
- 1.) Surface ที่สร้างนั้นมีขนาดเท่าใดก็ได้ขึ้นอยู่กับภาพที่เราจะนำมาแสดง แต่ Surface ที่เป็น Primary Surface ต้องมีขนาดเท่ากับขนาดของความละเอียดของจอภาพขณะนั้น
- 2.) การสร้าง Surface ผู้พัฒนาเกมควรจะใช้หน่วยความจำของการ์ดแสดงผล (VRAM) หรือหน่วยความจำหลัก (System Memory) ซึ่งถ้าใช้ VRAM ได้จะทำให้การแสดงผลมีความเร็วสูงขึ้น แต่ VRAM มีขนาดเล็ก ดังนั้นผู้พัฒนาเกมสามารถใช้ System Memory แทนได้
- 3.) เมื่อสร้าง Surface ขึ้นมา ระดับสีใน Surface ต้องเข้ากันได้กับระดับสีของ Primary Surface ตามปกติ Surface ที่ขึ้นสร้างมาจากออปเจกของ Direct Draw ตัวเดียวกันจะมีระดับสีเป็นระดับเดียวกันหมด แต่ถ้ามีออปเจกของ Direct Draw สองตัว อาจเป็นไปได้ที่ระดับสีของ Surface ไม่เท่ากัน ทำให้ไม่สามารถทำการถ่ายโอนข้อมูลระหว่าง Surface ได้



รูปที่ 2.7 การกำหนดขนาดของ Surface

โดยทั่วไปเกมที่สร้างขึ้นจะใช้ Surface ดังนี้

- 1.) Primary Surface หรือ FrontBuffer ทำหน้าที่แสดงภาพที่เป็นหน้าจอปัจจุบัน โดยมีโครงสร้างและขนาดเท่ากับความละเอียดของหน้าจอ
- 2.) Secondary Surface หรือ BackBuffer ทำหน้าที่วาดภาพหน้าจอต่อไปขณะที่ Primary Surface โดย Surface นี้มีโครงสร้างทุกอย่างเหมือน Primary Surface เพื่อที่จะทำงานสลับกับ Primary Surface
- 3.) OffScreen Surface เป็น Surface สำหรับเก็บภาพต่างๆ ที่ใช้ในโปรแกรม เพื่อที่จะเพิ่มประสิทธิภาพในการแสดงผลโดยไม่ต้องไปอ่านภาพจากไฟล์รูปที่อยู่ในฮาร์ดดิสก์ ทำให้เข้าถึงได้เร็วกว่า ส่วนมากแล้วเกมจะมีจำนวน Surface นี้มากที่สุด เพื่อใช้เก็บรูปภาพต่างๆ ลักษณะของการแสดงผลจะเป็นวัฏจักรดังนี้



รูปที่ 2.8 การแสดงภาพบนหน้าจอโดย DirectDraw

ขั้นแรกนำภาพที่เราต้องการใช้งานเก็บไว้ใน OffScreen Surface โดยอาจมีได้หลาย Offscreen Surface จากนั้นทำการ Blit (Bit Block Transfer) โดยการถ่ายโอนข้อมูลจาก Offscreen ไปยัง Secondary Surface ณ ตำแหน่งที่ต้องการแสดงผลโดยอ้างอิงจากจอภาพ จากนั้นก็แสดงผลต่อผู้ใช้ ด้วยวิธี FLIP หรือการสลับหน้าจอกับ Primary Surface โดยจะทำการสลับเฉพาะตำแหน่งหน่วยความจำที่อ้างอิง จากนั้นทำการวาดภาพที่จะใช้แสดงผลต่อไปใน Secondary Surface ในตำแหน่งที่เพิ่งสลับมาและจะเป็นเช่นนี้ไปเรื่อย ๆ จนจบโปรแกรม

การสร้าง Surface จะต้องทำการใส่ข้อมูลลงในโครงสร้างข้อมูลแบบ DDSURFACEDESC (Direct Draw Surface Description) แล้วจึงทำการเรียกฟังก์ชัน CreateSurface() เพื่อทำการสร้าง Surface นั้น ๆ ถ้าฟังก์ชันนี้ทำงานสำเร็จจะคืนค่า DD_OK กลับมา โดยรูปแบบฟังก์ชันเป็นดังนี้

```
HRESULT CreateSurface ( LPDDSURFACEDESC lpDDSurfaceDesc,
                        LPDIRECTDRAWSURFACE FAR *lpDDSurface,
                        IUnknown FAR *pUnkOuter);
```

พารามิเตอร์ lpDDSurfaceDesc เป็นข้อมูลประเภทพอยเตอร์ชี้ไปยังโครงสร้างข้อมูลแบบ DDSURFACEDESC ซึ่งมีการใส่ข้อมูลลงไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ lpDDSurface

ถ้าฟังก์ชันทำการสร้าง Surface ได้สำเร็จ จะคืนค่านี้กลับมา
เป็นพอยเตอร์ที่ชี้ไปยัง Surface ที่สร้างขึ้น

พารามิเตอร์ pUnkOuter

เป็นพารามิเตอร์ชั้นสูง โดยปกติจะกำหนดให้เป็น NULL

DDSURFACEDESC มีรูปแบบของโครงสร้างดังนี้

```
typedef struct _DDSURFACEDESC
```

```
{
```

```
    DWORD          dwSize;
```

```
    DWORD          dwFlags;
```

```
    DWORD          dwHeight;
```

```
    DWORD          dwWidth;
```

```
    Union
```

```
{
```

```
        LONG          lpPitch;
```

```
        DWORD        dwLinearSize;
```

```
};
```

```
    DWORD dwBackBufferCount;
```

```
    union
```

```
{
```

```
        DWORD        dwMipMapCount;
```

```
        DWORD        dwZBufferBitDepth;
```

```
        DWORD        dwRefreshRate;
```

```
};
```

```
    DWORD        dwAlphaBitDepth;
```

```
    DWORD        dwReserved;
```

```
    LPVOID lpSurface;
```

```
    DDCOLORKEY   ddckCKDestOverlay;
```

```
    DDCOLORKEY   ddckCKDestBlt;
```

```
    DDCOLORKEY   ddckCKSrcOverlay;
```

```
    DDCOLORKEY   ddckCKSrcBlt;
```

```
    DDPIXELFORMAT ddpfPixelFormat;
```

```
    DDSCAPS      ddsCaps;
```

```
} DDSURFACEDESC;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพารามิเตอร์ที่ต้องกำหนด คือ

- พารามิเตอร์ dwSize คือ ค่าขนาดของโครงสร้าง DDSURFACEDESC
- พารามิเตอร์ dwFlags คือ ส่วนที่ใช้กำหนดค่าให้กับ CreateSurface ทราบว่าจะสร้าง Surface แบบใด ซึ่งมีหลายค่าดังนี้

ค่าแฟลก DDSD_ALL เป็นตัวกำหนดให้ผู้พัฒนาสามารถเรียกใช้การทำงานของได้ทุก ๆ ส่วนของ Surface คือ DDSD_ALPHABITDEPTH, DDSD_BACKBUFFERCOUNT, DDSD_CAPS, DDSD_CKDESTBLT, DDSD_CKDESTOVERLAY, DDSD_CKSRCLBLT, DDSD_CKSRCOVERLAY, DDSD_HEIGHT, DDSD_LINEARSIZE, DDSD_LPSURFACE, DDSD_MIPMAPCOUNT, DDSD_PITCH, DDSD_PIXELFORMAT, DDSD_REFRESHRATE, DDSD_WIDTH, DDSD_ZBUFFERBITDEPTH

- dwBackBufferCount คือ ส่วนที่ใช้กำหนดจำนวน BackBuffer โดยทั่วไปมีเพียง 1 Surface
- ddsCaps คือ ส่วนที่ใช้กำหนดคุณสมบัติต่าง ๆ ของ Surface โดยมีโครงสร้างดังนี้

```
typedef struct _DDSCAPS
{
    DWORD dwCaps;
} DDSCAPS, FAR* LPDDSCAPS;
```

- dwcaps เป็นส่วนที่ใช้เก็บค่าคุณสมบัติของ Surface โดยส่วนใหญ่จะใช้ในระดับสูง

ตารางที่ 2.3 ค่าแฟลกของ dwCaps

ค่าแฟลกของ dwCaps	ความหมาย
DDSCAPS_BACKBUFFER	Surface ถูกกำหนดให้เป็น BackBuffer
DDSCAPS_COMPLEX	บอก DirectDrawSurface ว่าเราจะใช้หลาย Surface
DDSCAPS_FLIP	บอกว่า Surface นี้สามารถ Flip ได้
DDSCAPS_FRONTBUFFER	Surface ถูกกำหนดให้เป็น FrontBuffer
DDSCAPS_MODEX	กำหนดให้ Surface เป็น ModeX หรือ 320x200 หรือ 320x240 พิกเซล
DDSCAPS_OFFSCREENPLAIN	กำหนดให้ Surface เป็น Offscreen ซึ่งไม่ใช่ Frontbuffer หรือ Backbuffer โดยปกติจะใช้สำหรับเก็บภาพเคลื่อนไหวต่าง ๆ
DDSCAPS_SYSTEMMEMORY	กำหนดให้ Surface นี้จองหน่วยความจำในหน่วยความจำหลัก

ตัวอย่างการสร้าง PrimarySurface และ SecondarySurface

```

LPDIRECTDRAW          lpdd; // ประกาศตัวแปรพอยเตอร์ของ DirectDraw
DDSURFACEDESC         ddsd; //ประกาศตัวแปรที่ใช้เก็บค่าของ
                        //โครงสร้างข้อมูล DDSURFACEDESC
LPDIRECTDRAW_SURFACE lpddsprimary; //เก็บ address ของ Surface
//ทำการสร้างออปเจคของ DirectDraw
DirectDrawCreate(NULL, &lpdd, NULL);
//กำหนดระดับความสำคัญและจัดสรรทรัพยากร
lpdd->SetCooperatiiveLevel(hwnd,
                        DDSCL_ALLOWREBOOT|
                        DDSCL_ALLOWMODEX|
                        DDSCL_FULLSCREEN|
                        DDSCL_EXCLUSIVE);
//กำหนดขนาดของการแสดงผล
lpdd->SetDisplayMode(SCREEN_WIDTH, SCREEN_HEIGHT, SCREEN_BPP);
//กำหนดขนาดของ Primary Surface
ddsd.dwSize = sizeof(ddsd);
//หลังจากนั้นจึงกำหนดแอฟโลกของ Surface
ddsd.dwFlags = DDSD_CAPS| DDSD_BACKBUFFERCOUNT);
//เมื่อ DDSD_CAPS สามารถเรียกใช้งานได้นั้นหมายถึงสามารถกำหนด Primary Surface ได้แล้ว
ddsd.dwCaps.dwCaps = DDSCAPS_PRIMARYSURFACE|
                        DDSCAPS_FLIP|
                        DDSCAPS_COMPLEX;
ddsd.dwBackBufferCount = 1; //มี BackBuffer 1 อัน
lpdd->CreateSurface(&ddsd, &lpddsprimary, NULL); //สร้าง Primary Surface
ddscaps.dwCaps = DDSCAPS_BACKBUFFER; //ทำการกำหนด BackBuffer
//กำหนดให้ lpddsback เป็น Backbuffer ซึ่งใช้ทำงานร่วมกับ lpddsprimary
lpddsprimary->GetAttachedSurface(&ddscaps, &lpddsback);

```

2.4.8 การนำภาพขึ้นแสดง

เมื่อเราทำการวาดภาพลงบน Backbuffer เรียบร้อยแล้ว หลังจากนั้นทำการ Flip สลับกับ Frontbuffer กับ PrimarySurface โดยใช้ฟังก์ชัน Flip()

HRESULT Flip(LPDIRECTDRAW_SURFACE lpDDSurfaceOverride, DWORD dwFlags);

ค่าพารามิเตอร์ที่กำหนด คือ

พารามิเตอร์ lpDDSurfaceOverride

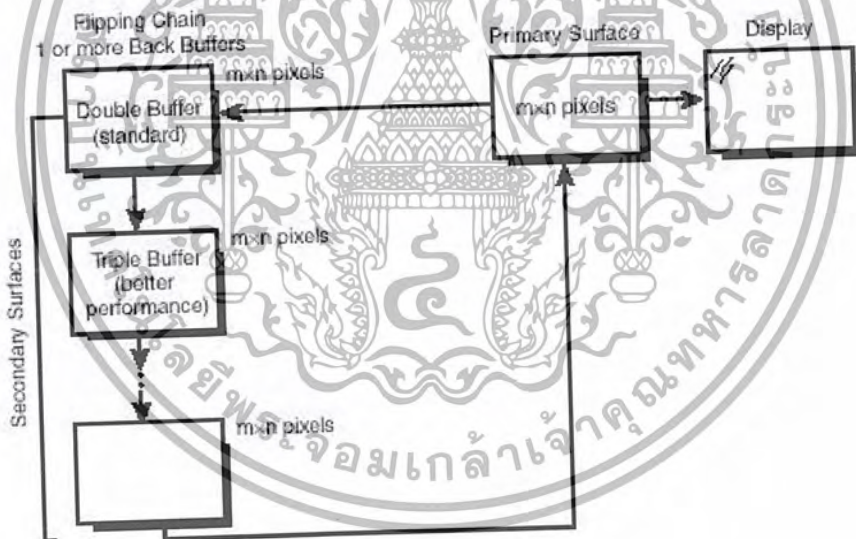
โดยปกติกำหนดให้เป็น NULL

พารามิเตอร์ dwFlags

โดยปกติกำหนดให้เป็น DDFLIP_WAIT เนื่องจากบางครั้ง SecondarySurface ยังอยู่ในสถานะที่กำลังวาดอยู่ และยังไม่สามารถ Flip ได้ โดยค่านี้จะทำให้ฟังก์ชันทำการลอง Flip ไปจนกว่าจะ Flip ได้

วิธีใช้เป็นดังนี้

lpddsprimary->Flip(NULL, DDFLIP_WAIT);



รูปที่ 2.9 การ Flip สลับ Surface

2.4.9 การวาดภาพลงบน Surface และการทำ Transparency

ก่อนที่จะทำการ Flip ผู้พัฒนาเกมจะต้องทำการวาดลงบน Surface ให้ได้ก่อน โดยใช้ฟังก์ชัน Blt ในการคัดลอกข้อมูล โดยเราสามารถคัดลอกข้อมูลจาก Surface หนึ่ง ๆ ไป Surface ใดก็ได้ โดยฟังก์ชันมีโครงสร้างดังนี้

```
HRESULT Blt(LPRECT          lpDestRect,
             LPDIRECTDRAW_SURFACE lpDDSrcSurf,
             DWORD            dwFlags,
             LPDDBLTFX        lpDDBltFx);
```

lpDestRect คือ พอยเตอร์ที่ชี้ไปยังตัวแปรประเภท RECT ของ Surface ปลายทาง ซึ่งตัวแปรประเภท RECT นี้มีโครงสร้างดังนี้

```
typedef struct _RECT
```

```
{
    LONG top; //ตำแหน่งพิกเซลที่อยู่บนสุด
    LONG left; //ตำแหน่งพิกเซลซ้ายสุด
    LONG right; //ตำแหน่งพิกเซลขวาสุด
    LONG bottom; //ตำแหน่งพิกเซลล่างสุด
} RECT
} RECT
```

พารามิเตอร์ lpDestRect คือ ตำแหน่งที่จะวางรูปภาพนั่นเอง แต่หากค่านี้กำหนดเป็น NULL จะหมายถึงใช้ Surface ปลายทางทั้งหมด

lpDDSrcSurf คือ Surface ที่เป็นรูปภาพต้นทาง ซึ่งอาจจะเป็น OffScreen ก็ได้

lpSrcRect คือ ตำแหน่งของรูปภาพที่ต้นทาง ถ้ากำหนดให้เป็น NULL หมายถึงใช้รูปทั้งรูป

dwFlags คือ ส่วนที่ใช้ควบคุมการวาดภาพ

lpDDBltFx คือ พอยเตอร์ที่ชี้ไปที่โครงสร้างข้อมูล DDBltFx ซึ่งใช้กำหนดความสามารถพิเศษต่าง ๆ ของฟังก์ชัน เช่น การหมุนภาพ หรือ ระบายสี

ตารางที่ 2.4 ค่าแฟลคควบคุมของฟังก์ชัน Blit()

ค่าแฟลคของ dwFlags	ความหมาย
DDBLT_COLORFILL	ใช้ร่วมกับ dwFillColor ในโครงสร้างข้อมูล DDBLTFX โดยใส่เป็นสีรูปแบบ RGB (Red, Green, Blue) ซึ่งจะทำการระบายลงบน RECT ที่ Surface ปลายทาง
DDBLT_DDFX	ใช้ร่วมกับ dwDDFX ซึ่งกำหนดในโครงสร้างข้อมูล DDBLTFX ซึ่งใช้กำหนด ลักษณะพิเศษบางอย่างในการแสดงผล
DDBLT_DDROPS	ใช้ร่วมกับ dwDDROPS ซึ่งกำหนดในโครงสร้างข้อมูล DDBLTFX โดยใช้กับ การทำงานชั้นสูง
DDBLT_KEYDEST	เมื่อกำหนดแล้วจะสามารถกำหนด Color Key ที่ Surface ปลายทางได้
DDBLT_KEYSRC	เมื่อกำหนดแล้วจะสามารถกำหนด Color Key ที่ Surface ต้นทางได้
DDBLT_ROP	ใช้ร่วมกับ dwROP โดยกำหนดในโครงสร้างข้อมูล DDBLTFX โดยทำงานตรง กันข้ามกับ DDBLT_DDROPS
DDBLT_ROTATIONANGLE	ใช้ร่วมกับ dwRotationAngle ใน โครงสร้างข้อมูล DDBLTFX โดยใช้หมุนภาพที่จะ Blit ไปใส่ที่ Surface โดยมีหน่วยเป็น 1/100 องศา
DDBLT_WAIT	ตั้งทำการ Blit จนกว่าจะ Blit สำเร็จ เมื่อใช้คำสั่งนี้ฟังก์ชัน Blit จะไม่ส่งข้อผิดพลาดกลับมาถึงแม้จะยังทำการ Blit ไม่ได้ก็ตาม

ในโครงสร้างข้อมูล DDBLTFX มีข้อมูลจำนวนมากที่ต้องกำหนด แต่ข้อมูลส่วนใหญ่จะใช้สำหรับการทำงานระดับสูง เช่น Z-buffer ที่ใช้กับภาพสามมิติ ดังนั้นเราจะกล่าวถึงเพียงเท่าที่จำเป็นเท่านั้น

```
typedef struct _DDBLTFX
```

```
{
```

```
    DWORD    dwSize;
    DWORD    dwDDFX;
    DWORD    dwROP;
    DWORD    dwDDROP
    DWORD    dwRotationAngle;
    DWORD    dwZBufferOpCode;
    DWORD    dwZbufferLow;
    DWORD    dwZbufferHigh;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DWORD          dwZBufferBaseDest;
        DWORD          dwZDestConstBitDepth;

union
{
        DWORD          dwZDestConst;
        LPDIRECTDRAW SURFACE  lpDDSZBufferDest;
};
        DWORD dwZSrcConstBitDepth;
union
{
        DWORD          dwzSrcConst;
        LPDIRECTDRAW SURFACE  lpDDSZBufferSrc;
};
        DWORD          dwAlphaEdgeBlendBitDepth;
        DWORD          dwAlphaEdgeBlend;
        DWORD          dwReserved;
        DWORD          dwAlphaDestConstBitDepth;
union
{
        DWORD          dwAlphaDestConst;
        LPDIRECTDRAW SURFACE  lpDDSAAlphaDest;
};
        DWORD dwAlphaSrcConstBitDepth;
union
{
        DWORD          dwAlphaSrcConst;
        LPDIRECTDRAW SURFACE  lpDDSAAlphaSrc;
};
union
{
        DWORD          dwFillColor;
        DWORD          dwFillDepth;
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DWORD          dwFillPixel;
LPDIRECTDRAW_SURFACE lpDDSPattern;
};
DDCOLORKEY     ddckDestColorkey;
DDCOLORKEY     ddckSrcColorkey;
} DDBLTFX.FAR* LPDDBLTFX;

```

พารามิเตอร์ที่กำหนดค่า คือ

พารามิเตอร์ dwSize เป็นขนาดของโครงสร้างข้อมูล DDBLTFX มีค่าเป็นไบต์

พารามิเตอร์ dwFillColor สีที่ต้องการระบายลง Surface

พารามิเตอร์ ddckDestColorkey เป็น Color Key ของทาง Surface ปลายทาง

พารามิเตอร์ ddckSrcColorkey เป็น Color Key ของทาง Surface ต้นทาง

พารามิเตอร์ dwRotationAngle กำหนดเป็นองศาเพื่อจะใช้หมุนภาพ

Color Key คือ สีที่เรากำหนดให้เป็นสีโปร่งใส (Transparency) ซึ่งจะใช้นักในเกมส์คอมพิวเตอร์ทั่ว ๆ ไป โดยปกติรูปภาพที่สร้างขึ้นจะมีขอบสีเหลี่ยมที่เป็นส่วนเกิน เมื่อนำภาพไปสร้างเกมเราต้องตัดส่วนนี้ออกโดยการกำหนดสีที่เป็น Color Key แล้วเราก็ใช้สีเดียวกันนี้ระบายลงไปที่ยอบหรือส่วนเกินของภาพ หลังจากทำการ Blit แล้ว คอมพิวเตอร์จะทำการจับคู่สีที่เป็น Color Key แล้วจะไม่แสดงสีนั้นลงบนจอภาพ ดังรูป

รูปที่ 2.10 การเปรียบเทียบระหว่างการกำหนดค่า Color Key

จากรูปที่ 2.10 รูปเดิมจะเป็นรูปด้านซ้ายมีขอบเกินมาด้วย แต่เมื่อกำหนดให้ Color Key เป็นสีดำแล้ว เมื่อเราทำการ Blit บน Surface เราจะได้ภาพด้านขวาที่ไม่มีขอบเกินออกมา ทำให้ภาพดูสมจริงมากขึ้น

ตัวอย่างการ Blit โดยใช้ Color Key เป็นสีดำ

```

DDBLTFX     ddbltfx; //ข้อมูลประกาศโครงสร้าง
RECT        blit_area, src_area; //ประกาศ RECT ที่จะวางภาพปลายทางและภาพต้นทาง
memset(&ddbltfx, 0, sizeof(DDBLTFX));
ddbltfx.dwSize = sizeof(DDBLTFX);
DDCOLORKEY  col_key; //กำหนดตัวแปรแบบ DDCOLORKEY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
col_key.dwColorSpaceLowValue = 0; //ทำการเซ็ตค่าสีหนึ่งซึ่งต้องมีค่าสีต่ำกว่า
col_key.dwColorSpaceHighValue = 0; //ไปยังอีกสีหนึ่ง ทำให้เราสามารถมี transparent ได้หลายสี
lpddsback->SetColorKey(DDCKEY_SRCBLT, &col_key); //ทำการเซ็ต Color Key โดยเป็น
//Source Colorkey ซึ่งก็คือสีที่ไม่ต้องการ ณ ภาพต้นทาง
//โดยต้องทำการกำหนดที่ BackBuffer
//เพื่อบอกให้ BackBuffer ทราบว่าจะไม่สนใจสีนี้
//และไม่ทำการ Blit ลงไป แต่หากเป็น DestinationColorkey
//จะหมายถึงว่าเป็นสีที่จะไม่ให้วาดทับแทน
```

```
blit_area.top = top;
blit_area.left = left;
blit_area.bottom = bottom;
blit_area.right = right;
src_area.top=top_src;
src_area.left=left_src;
src_area.bottom=bottom_src;
src_area.right=right_src;
lpddsprimary->Blit(&blit_area, lpddsback, &src_area, DDBLT_KEYSRC | DDBLT_WAIT,
NULL);
```

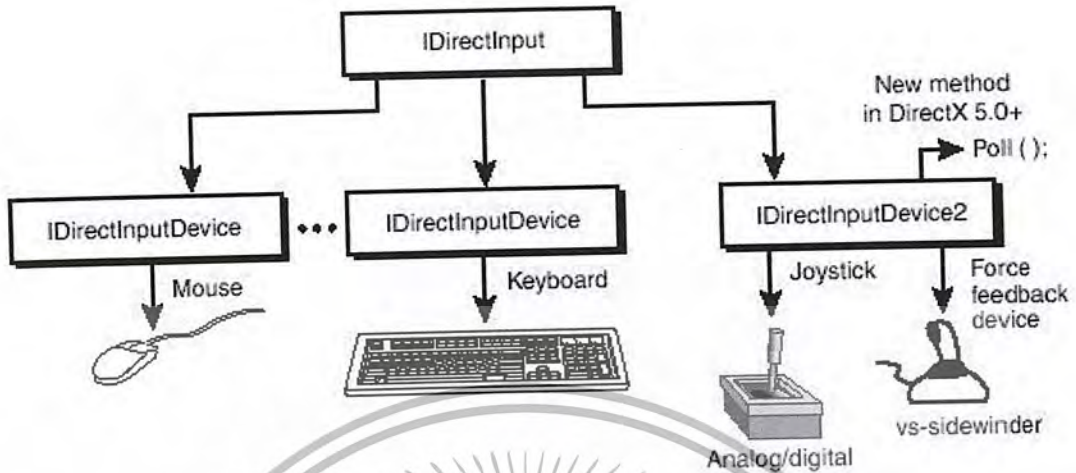
2.5 วิธีการรับข้อมูลจากเมาส์และคีย์บอร์ด

DirectInput เป็นไลบรารีที่ช่วยให้ผู้พัฒนาเกมทำการติดต่อและเรียกใช้อุปกรณ์ในการรับอินพุตชนิดต่าง ๆ ได้อย่างมีประสิทธิภาพ Direct Input ครอบคลุมอุปกรณ์ในการรับอินพุตมาตรฐานทุกชนิด อันประกอบไปด้วย

- คีย์บอร์ดมาตรฐาน
- เมาส์ทั้ง 2 และ 3 ปุ่ม
- จอยสติ๊ก, จอยโยก (Paddle) ทั้งที่เป็นแบบอนาล็อกและดิจิตอล
- อุปกรณ์ควบคุมการขับเคลื่อนใช้สำหรับจำลองการขับรถทั้งที่เป็นแบบอนาล็อกและ

ดิจิตอล

- อุปกรณ์ที่ตอบสนองแรงสั่นสะเทือน, จอยสติ๊กหรืออุปกรณ์อื่น ๆ ที่สามารถคำนวณเพื่อทำให้เกิดแรงสั่นได้ตอบกับผู้ใช้



รูปที่ 2.11 อุปกรณ์อินพุตต่าง ๆ

2.5.1 การใช้ DirectInput

Direct Input ประกอบด้วยไฟล์เฮดเดอร์ `dinput.h` และไฟล์ไลบรารี `dinput.lib` สำหรับโปรแกรมที่ต้องการติดต่อกับอุปกรณ์อินพุต

ขั้นตอนในการใช้ Direct Input

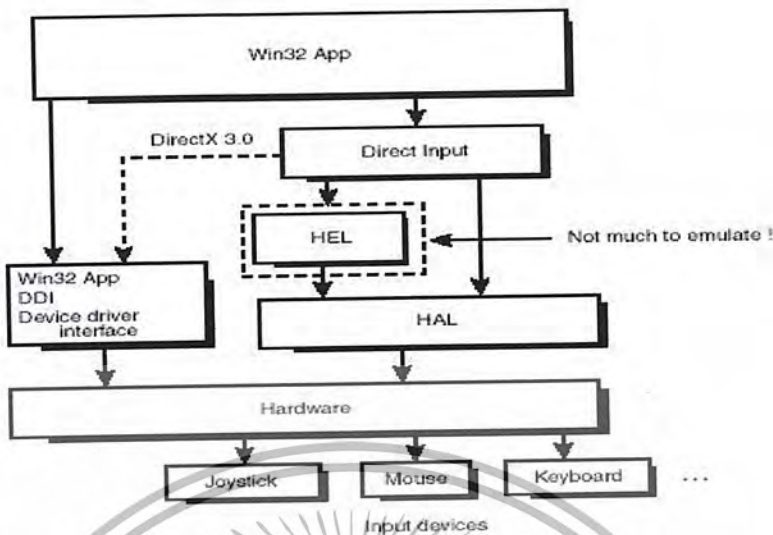
- สร้างออบเจกต์ของ Direct Input หลักด้วยฟังก์ชัน `DirectInputCreate()`
- สร้างออบเจกต์ของอุปกรณ์อินพุตด้วยฟังก์ชัน `CreateDevice()`
- กำหนดลักษณะการใช้งานของแต่ละอุปกรณ์ด้วยฟังก์ชัน `SetCooperativeLevel()`
- กำหนดรูปแบบการรับข้อมูลของแต่ละอุปกรณ์ด้วยฟังก์ชัน `SetDataFormat()` และทำการกำหนดคุณสมบัติพิเศษของอุปกรณ์ต่าง ๆ ด้วยฟังก์ชัน `SetProperty()`
- ทำการจองอุปกรณ์ต่าง ๆ ด้วยฟังก์ชัน `Acquire()`
- รับข้อมูลจากแต่ละอุปกรณ์ที่ได้ทำการจองไว้แล้วด้วยฟังก์ชัน `GetDeviceState()`

การอ่านข้อมูลของแต่ละอุปกรณ์จะถูกอ่านเป็นเรคคอร์ด โดยอุปกรณ์แต่ละตัวมีรูปแบบของเรคคอร์ดที่แตกต่างกันแต่มีลักษณะใกล้เคียงกันมาก ทำให้เกิดการสร้างข้อมูลอินพุตที่เป็นรูปแบบหลัก ๆ ขึ้น โดยผู้พัฒนาเกมสามารถร้องขอข้อมูลอินพุตจาก `DirectInput` ได้ 2 วิธี ดังนี้

- 1.) แบบทันทีทันใด (Immediate Input) ข้อมูลที่ได้จะเป็นสถานะปัจจุบันของอุปกรณ์อินพุตขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) แบบมีการพักข้อมูล (Buffered Input) ข้อมูลถูกเก็บเป็นฐานข้อมูลของเหตุการณ์ที่เกิดขึ้นตั้งแต่การร้องขออินพุตครั้งสุดท้าย



รูปที่ 2.12 ระดับโครงสร้างภายในของ DirectInput

2.5.2 การติดต่อกับคีย์บอร์ดผ่านทาง Direct Input

2.5.2.1 ออปเจกต์ของ DirectInput

การกำหนดออปเจกต์ของ Direct Input มีจุดประสงค์เพื่อใช้ในการจัดการติดต่อกับอุปกรณ์อินพุตชนิดอื่น

LPDIRECTINPUT lpdi;

การสร้างออปเจกต์หลักของอุปกรณ์อินพุตทำได้โดยใช้ฟังก์ชัน DirectInputCreate() โดยมีรูปแบบการของฟังก์ชันดังนี้

```
HRESULT DirectInputCreate(HINSTANCE hinst,
    DWORD dwversion,
    LPDIRECTINPUT * lplpDirectInput,
    LPUNKNOWN punkOuter);
```

ฟังก์ชันจะคืนค่า DI_OK(DirectInput Okey) ถ้าการเรียกใช้ฟังก์ชันนี้ผ่าน มันจะคืนค่าเป็นค่าผิดพลาดอื่น ๆ ถ้าฟังก์ชันนี้ไม่ผ่าน สำหรับค่าพารามิเตอร์ต่าง ๆ ของ DirectInputCreate มีดังนี้

hinst ใช้อ้างอิงแอปพลิเคชันที่ต้องการใช้งาน Direct Draw ให้แน่ใจว่าค่า hinst นี้ ถูกส่งค่ามาจากค่า hinstance ใน WinMain()

dwversion	กำหนดเวอร์ชันของ Direct Inputว่าจะใช้เวอร์ชันใดมีไว้เพื่อให้ผู้พัฒนาเกมที่ต้องการใช้ Direct Input เวอร์ชันเก่าแต่โดยทั่วไปจะถูกกำหนดค่าเป็น DIRECTINPUT_VERSION ซึ่ง DirectInput ที่กำหนดจะใช้เป็นเวอร์ชันล่าสุด
lpDlDirectInput	เป็นพอยเตอร์ที่อ้างอิงไปยังออปเจกต์หลักของ DirectInput
punkOuter	ถูกกำหนดเป็น NULL เสมอ ตัวอย่างการสร้างออปเจกต์หลักของ DirectInput

LPDIRECTINPUT lpdI;

DirectInputCreate(hinstance, DIRECTINPUT_VERSION, &lpdi, NULL);

เมื่อไม่ต้องการใช้งานออปเจกต์หลักของ DirectInput ก็ทำการลบด้วยฟังก์ชัน Release() เช่น

lpdi->Release();

2.5.2.2 ทำการสร้างออปเจกต์ของคีย์บอร์ด

ด้วยฟังก์ชัน CreateDevice() ฟังก์ชันนี้จะใช้เมื่อต้องการสร้างออปเจกต์อินพุทใด ๆ รูปแบบการใช้งานฟังก์ชัน CreateDevice() มีดังนี้

```
HRESULT CreateDevice( REFGUID rguid,
LPDIRECTINPUTDEVICE * lplpDiDev,
LPUNKNOWN punkOuter);
```

ค่าพารามิเตอร์ต่าง ๆ ของ DirectInput มีดังนี้

rguid คือ ค่า GUID (Globally Unique Identifier) เป็นหมายเลขของอุปกรณ์อินพุทที่ต้องการสร้าง โดยปกติจะใช้ฟังก์ชัน EnumDevice () เพื่อหาค่า GUID ของอุปกรณ์อินพุทนั้น ๆ แต่ใน DirectInput มีค่าที่กำหนดค่า

GUID ของคีย์บอร์ดและเมาส์ไว้ให้เรียบร้อยแล้ว โดยจะมีค่าเป็น

GUID_SysKeyboard สำหรับคีย์บอร์ดมาตรฐาน

GUID_SysMouse สำหรับเมาส์มาตรฐาน

อย่างไรก็ตามการใช้ค่า GUID ที่ถูกประกาศไว้แล้วนี้จำเป็นต้องเพิ่ม #define INITGUID ไว้ส่วนบนของทุกไฟล์ C/C++ ที่มีการประกาศ OBJBASE.H เพื่อคอมไพเลอร์จะรวมข้อมูลเกี่ยวกับ COM และค่า GUID ที่ได้ถูกประกาศไว้แล้ว

lpDlDiDev เป็นตำแหน่งที่พอยเตอร์ชี้เพื่ออ้างอิงถึงอุปกรณ์อินพุทชนิดนั้น ๆ

punkOuter ถูกกำหนดให้เท่ากับ NULL เสมอ

ตัวอย่างการสร้างออปเจตของคีย์บอร์ด

```
#define INITGUID
#include <OBJBASE.H>
#include "DINPUT.H"
//สร้างออปเจตหลักตามที่ได้กล่าวมาในขั้นก่อนหน้านี้นี้หลังจากสร้างออปเจตหลักแล้วจึงทำ
การสร้าง
//ออปเจตของคีย์บอร์ด
LPDIRECTINPUTDEVICE    lpdikey;
lpdi->CreateDevice(GUID_SysKeyboard, &lpdikey, NULL);
```

2.5.2.3 กำหนดลักษณะการใช้งานของคีย์บอร์ด

แต่ละอุปกรณ์ก็จะมีลักษณะการใช้งานเป็นของตัวเอง SetCooperativeLevel() มีรูปแบบการเรียกใช้ดังนี้

```
HRESULT SetCooperativeLevel(HWND hwnd,
                             DWORD dwFlag);
```

ค่าพารามิเตอร์ต่าง ๆ ของ SetCooperativeLevel() มีดังนี้

hwnd ใช้อ้างถึงวินโดวส์ของแอปพลิเคชันนั้น ๆ

dwFlag เป็นค่าที่กำหนดลักษณะของอุปกรณ์อื่น ๆ มีทั้งหมด 4 ค่า

ตารางที่ 2.5 ค่าแฟลก dwFlag ลักษณะการใช้งาน ของ Keyboard

ค่าแฟลก dwFlag	ความหมาย
DISCL_BACKGROUND	กำหนดให้แอปพลิเคชันสามารถใช้อุปกรณ์อื่น ๆ ได้ แม้ว่า แอปพลิเคชันจะอยู่ในโหมด Background
DISCL_FOREGROUND	แอปพลิเคชันไม่สามารถเรียกใช้อุปกรณ์อื่น ๆ ได้ใน โหมด Background เพราะอุปกรณ์อื่นจะถูกปล่อยทันทีเมื่อ แอปพลิเคชันเปลี่ยนเป็นโหมด Background
DISCL_EXCLUSIVE	หลังจากทำการจองอุปกรณ์อื่น ๆ แล้ว จะมีได้เพียง แอปพลิเคชันเดียวที่ ร้องขอ DISCL_EXCLUSIVE ส่วน แอปพลิเคชันอื่นจะต้องร้องขอการใช้งานอุปกรณ์อื่นแบบ DISCL_NONEXCLUSIVE ได้
DISCL_NONEXCLUSIVE	แอปพลิเคชันสามารถใช้งานอุปกรณ์อื่น ๆ ร่วมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเกมที่ใช้โหมดแบบ Full Screen จะมีการกำหนดลักษณะของอุปกรณ์อินพุตเป็นแบบ DISCL_BACKGROUND และ DISCL_NONEXCLUSIVE

ตัวอย่างการใช้ฟังก์ชัน SetCooperativeLevel()

```
lpdikey->SetCooperativeLevel(hwnd, DISCL_BACKGROUND|DISCL_NONEXCLUSIVE);
```

2.5.2.4 กำหนดรูปแบบข้อมูลของคีย์บอร์ด

เพื่อกำหนดว่าข้อมูลที่จะรับเข้ามาจากคีย์บอร์ดมีรูปแบบอย่างไร โดยใช้ฟังก์ชัน SetDataFormat() โดยมีรูปแบบการเรียกใช้ดังนี้

```
HRESULT SetDataFormat(LPCDIDATAFORMAT lpdf);
```

ฟังก์ชัน SetDataFormat มีพารามิเตอร์เพียงตัวเดียว คือ lpdf พารามิเตอร์นี้จะใช้โครงสร้างของ DIDATAFORMAT ตามที่ได้แสดงด้านล่างนี้

Typedef struct

```
{
    DWORD dwSize; //ขนาดของโครงสร้างรูปแบบนี้หน่วยเป็นไบต์
    DWORD dwObjSize; //ขนาดของ DIOBJECTDATAFORMAT เป็นไบต์
    DWORD dwFlags; //กำหนดการรายงานผลว่าเป็นรูปแบบ Relative หรือว่า Absolute
    DWORD dwDataSize; //ขนาดของ packet ข้อมูล
    DWORD dsNumObjs; //จำนวนออบเจกต์กำหนดให้เป็นขนาดของอาร์เรย์
    LPDIOBJECTDATAFORMAT rgodf; //พอยเตอร์ชี้ไปยังอาร์เรย์ออบเจกต์
} DIDATAFORMAT, *LPDIDATAFORMAT;
```

แต่ DirectInput ได้กำหนดรูปแบบของข้อมูลสำหรับอุปกรณ์อินพุตมาตรฐานไว้เรียบร้อยแล้ว นั่นคือ

c_dfDIKeyboard กำหนดรูปแบบข้อมูลสำหรับคีย์บอร์ดมาตรฐาน

c_dfDIMouse กำหนดรูปแบบข้อมูลสำหรับเมาส์มาตรฐาน

c_dfDIJoystick กำหนดรูปแบบข้อมูลสำหรับจอยสติ๊กมาตรฐาน

ดังนั้นผู้พัฒนาเกมสามารถใช้รูปแบบข้อมูลที่ถูกกำหนดขึ้นไว้แล้ว ถ้าแอปพลิเคชันนั้นใช้ อุปกรณ์อินพุตมาตรฐาน

ตัวอย่างการใช้ฟังก์ชัน SetDataFormat

```
lpdikey->SetDataFormat(&c_dfDIKeyboard);
```

2.5.2.5 ทำการจอบอร์ดด้วยฟังก์ชัน Acquire()

มีรูปแบบการเรียกใช้ดังนี้

```
HRESULT Acquire();
```

ตัวอย่างการใช้ฟังก์ชัน Acquire();

```
lpdikey->Acquire();
```

2.5.2.6 ทำการอ่านข้อมูลจากคีย์บอร์ด

สำหรับลักษณะข้อมูลของคีย์บอร์ดที่ถูกกำหนดโดย DirectInput มีลักษณะเป็น State Data นั้นหมายความว่าเมื่ออ่านข้อมูลขึ้นมาจากคีย์บอร์ดข้อมูลที่ได้จะเป็นสถานะปัจจุบันของคีย์บอร์ด สำหรับการอ่านข้อมูลจากคีย์บอร์ดจะใช้ฟังก์ชัน GetDeviceState() ฟังก์ชันนี้จะอ่านค่าสถานะปัจจุบันของคีย์บอร์ดขึ้นมา หรือสิ่งที่คีย์บอร์ดได้ทำครั้งสุดท้ายในอินพุทโดยมีรูปแบบการเรียกใช้ฟังก์ชันดังนี้

```
HRESULT GetDeviceState (DWORD cbData, //ขนาดของเรคคอร์ดข้อมูล
LPVOID lpvData); // พ้อยเตอร์ชี้ไปยังตำแหน่งที่เก็บข้อมูล
```

ตัวอย่างการใช้ฟังก์ชัน GetDeviceState

```
UCHAR keystate[256];
```

```
lpdikey ->GetDeviceState(256, keystate)
```

DirectInput จะมีค่าคงที่ของแต่ละคีย์บนคีย์บอร์ดเพื่อให้ผู้พัฒนาเกมสามารถตรวจสอบสถานะของคีย์บนคีย์บอร์ดได้ว่าเป็นอย่างไร โดยค่าคงที่นี้จะขึ้นต้นด้วย DIK_ แล้วตามด้วยคำศัพท์ ตามตารางด้านล่างนี้

ตารางที่ 2.6 ค่าคีย์บนคีย์บอร์ด

ค่าแฟลก	ความหมาย
DIK_ESCAPE	ปุ่ม ESC
DIK_0 – DIK_9	หมายเลข 0 ถึง 9
DIK_A – DIK_Z	A ถึง Z
DIK_RETURN	ปุ่ม Enter
DIK_LCONTROL	ปุ่ม Ctrl ทางซ้ายมือ
DIK_RCONTROL	ปุ่ม Ctrl ทางขวามือ
DIK_SPACE	ปุ่ม Spacebar
DIK_F1 – DIK_F12	ปุ่ม F1 ถึง F12
DIK_UP	ปุ่ม ลูกศรขึ้น
DIK_DOWN	ปุ่ม ลูกศรลง
DIK_LEFT	ปุ่ม ลูกศรไปทางซ้าย
DIK_RIGHT	ปุ่ม ลูกศรไปทางขวา
DIK_PRIOR	ปุ่ม PageUp
DIK_NEXT	ปุ่ม PageDown

หลังจากทำการอ่านค่าสถานะของคีย์บอร์ดแล้ว จะสามารถตรวจสอบได้ว่าคีย์ที่สนใจมีสถานะเป็นอย่างไรด้วยการตรวจสอบค่าบิตของคีย์นั้น ๆ ถ้าคีย์ถูกกด ค่าบิต 0x80 จะมีค่าเป็น 1 แต่ถ้าคีย์นั้น ๆ ไม่ได้ถูกกด หรือคีย์นั้น ถูกปล่อยไปแล้วค่าบิต 0x80 จะมีค่าเป็น 0

ตัวอย่างการตรวจสอบค่าของคีย์

```
int ship_x = 100, ship_y = 100;
UCHAR keystate[256];
lpdikey->GetDeviceState(256, keystate)
if (keystate[DIK_RIGHT] & 0x80) ship_x++;
if (keystate[DIK_DOWN] & 0x80) ship_y++;
```

จากตัวอย่างเป็นการตรวจสอบว่าคีย์ลูกศรขึ้น หรือลูกศรไปด้านขวาถูกกดหรือไม่

2.5.3 การรับข้อมูลจากเมาส์

ขั้นตอนต่าง ๆ ในการใช้ DirectInput เพื่อติดต่อกับเม้าส์นั้นจะคล้ายกับขั้นตอนในการใช้ DirectInput เพื่อติดต่อกับคีย์บอร์ดจะต่างกันเพียง ขั้นตอนในการสร้างออปเจกเพื่อใช้ติดต่อกับและอ่านค่าจากเม้าส์มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) สร้างออปเจคเมาส์
- 2) ทำการกำหนดลักษณะการใช้งานของเมาส์
- 3) กำหนดรูปแบบข้อมูลที่ได้จากการอ่านค่าสถานะของเมาส์
- 4) ทำการจองเมาส์

2.5.3.1 สร้างออปเจคเมาส์

สำหรับการสร้างออปเจคเมาส์นี้ จะใช้ฟังก์ชัน Create Device() พารามิเตอร์ GUID ที่จะใช้เป็นหมายเลขของเมาส์จะใช้ค่า GUID ที่ถูกประกาศไว้แล้วโดย DirectInput นั่นคือใช้ค่า GUID_SysMouse

ตัวอย่าง การเรียกใช้ฟังก์ชัน CreateDevice()

```
LPDIRECTINPUTDEVICE lpdimouse; //ทำการสร้างเมาส์ออปเจค
lpdi->CreateDevice(GUID_SysMouse, &lpdimouse, NULL);
```

2.5.3.2 ทำการกำหนดลักษณะการใช้งานของเมาส์

หลังจากการ CreateDevice () ได้สำเร็จ ขั้นตอนต่อไปก็จะทำการกำหนดลักษณะการใช้งานของอุปกรณ์อินพุตกับแอปพลิเคชันหนึ่ง ๆ อุปกรณ์อินพุตในที่นี้ คือ เมาส์ จะทำการกำหนดลักษณะการใช้งานได้ด้วยฟังก์ชัน SetCooperativeLevel() สำหรับพารามิเตอร์ที่ใช้ในฟังก์ชันนี้จะเหมือนกับพารามิเตอร์ที่ใช้ในการกำหนดลักษณะการใช้งานให้กับคีย์บอร์ด โดยมีค่าแฟลก 4 ค่า คือ

- DISCL_BACKGROUND
- DISCL_FOREGROUND
- DISCL_EXCLUSIVE
- DISCL_NONEXCLUSIVE

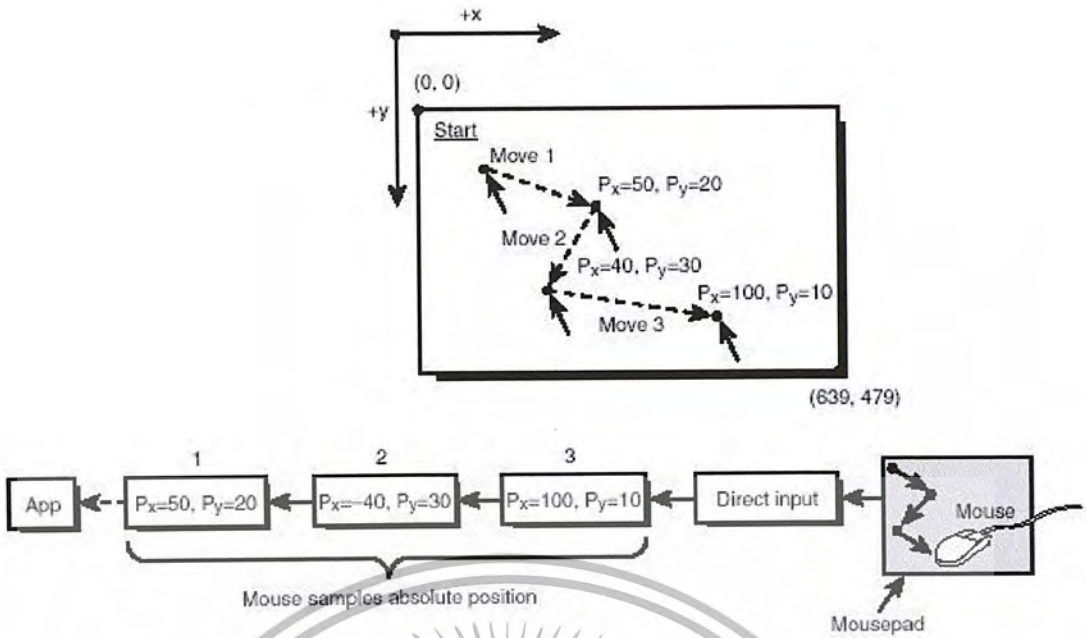
ค่าแฟลกทั้ง 4 ค่านี้มีความหมายเหมือนกับค่าแฟลกที่ใช้กำหนดลักษณะการใช้งานของคีย์บอร์ด สำหรับเกมที่ใช้ในโหมด Full Screen จะกำหนดค่าแฟลกเป็น DISCL_BACKGROUND และ ISCL_NONEXCLUSIVE เพื่อหลีกเลี่ยงข้อผิดพลาดจากการจองอุปกรณ์อินพุตชนิดเดียวกัน

2.5.3.3 ทำการกำหนดรูปแบบของข้อมูลที่ได้รับ

ขั้นตอนนี้เหมือนกับขั้นตอนในการกำหนดรูปแบบของข้อมูลให้กับคีย์บอร์ด จะมีการใช้รูปแบบของข้อมูลที่ได้ถูกประกาศไว้แล้ว คือ c_dfDIMouse แต่รูปแบบของข้อมูลที่ได้จากการใช้ c_dfDIMouse จะเป็นแบบสัมพัทธ์ (relative) นั้นหมายความว่าทุก ๆ ครั้งที่มีการร้องขอข้อมูลจากเมาส์ ค่าตำแหน่งของเมาส์จะอยู่ในรูปผลต่างของตำแหน่งที่เมาส์ชี้เปลี่ยนไป จะใช้ฟังก์ชัน SetDataFormat() เพื่อกำหนดรูปแบบของข้อมูลของฟิลด์ที่ได้จากการอ่านค่าสถานะของเมาส์

```
lpdimouse->SetDataFormat(&c_dfDIMouse);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงการเก็บค่าตำแหน่งเมื่อทำการเลื่อนเมาส์

2.5.3.4 ทำการจองเมาส์

DirectInput จะทำการจองเมาส์ไว้ที่แอปพลิเคชันนั้น ๆ ด้วยฟังก์ชัน `Acquire()`
`lpdimouse->Acquire();`

2.5.3.5 การอ่านค่าจากเมาส์

จะสามารถอ่านค่าจากเมาส์ได้ด้วยฟังก์ชัน `GetDeviceState()` แต่ข้อมูลที่ส่งคืนกลับมาจะต่างจากข้อมูลที่ถูกส่งคืนจากคีย์บอร์ด ข้อมูลที่ได้จากการอ่านค่าเมาส์จะถูกส่งคืนมาในรูปแบบของ `DIMOUSESTATE`

```
typedef struct {
    LONG lx; //ค่าแกน x ของเมาส์
    LONG ly; //ค่าแกน y ของเมาส์
    LONG lz; //ค่าแกน z ของเมาส์ หมายถึงปุ่มที่วงล้อ
    BYTE rgbButtons[4] //ปุ่มของเมาส์ บิตสูงจะหมายถึงปุ่มนั้นถูกกด
} DIMOUSESTATE, *LPDIMOUSESTATE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างข้อมูลแบบนี้ ค่าตำแหน่งของแกน x y และค่าของวงล้อจะถูกส่งมายังพารามิเตอร์ lx ly และ lz และค่าสถานะของปุ่มจะถูกส่งมายัง rgbButtons[4] เป็นอาร์เรย์ขนาด 4 ไบต์ สำหรับวิธีการอ่านค่าสถานะจากเมาส์จะใช้คำสั่ง GetDeviceState()

```
DIMOUSESTATE mouse_state;
```

```
lpdimouse->GetDeviceState(sizeof(DIMOUSESTATE),(LPVOID)&mouse_state);
```

เมื่อได้ข้อมูลมาแล้วการตรวจสอบว่าข้อมูลที่ได้มานั้นมีค่าเป็นอะไรสามารถตรวจสอบได้ดังนี้

```
// เป็นการตรวจสอบว่ามีกรกดเมาส์ปุ่มซ้าย ถ้าพบว่าปุ่มถูกกดก็ให้ทำงานตามหลังวงเล็บปีกกา
if (mouse_state.rgbButtons[0])
```

```
{
```

```
// เป็นการตรวจสอบว่ามีกรกดเมาส์ปุ่มขวาถ้าพบว่าปุ่มถูกกดก็ให้ทำงานตามหลังวงเล็บปีกกา
```

```
if (mouse_state.rgbButtons[1])
```

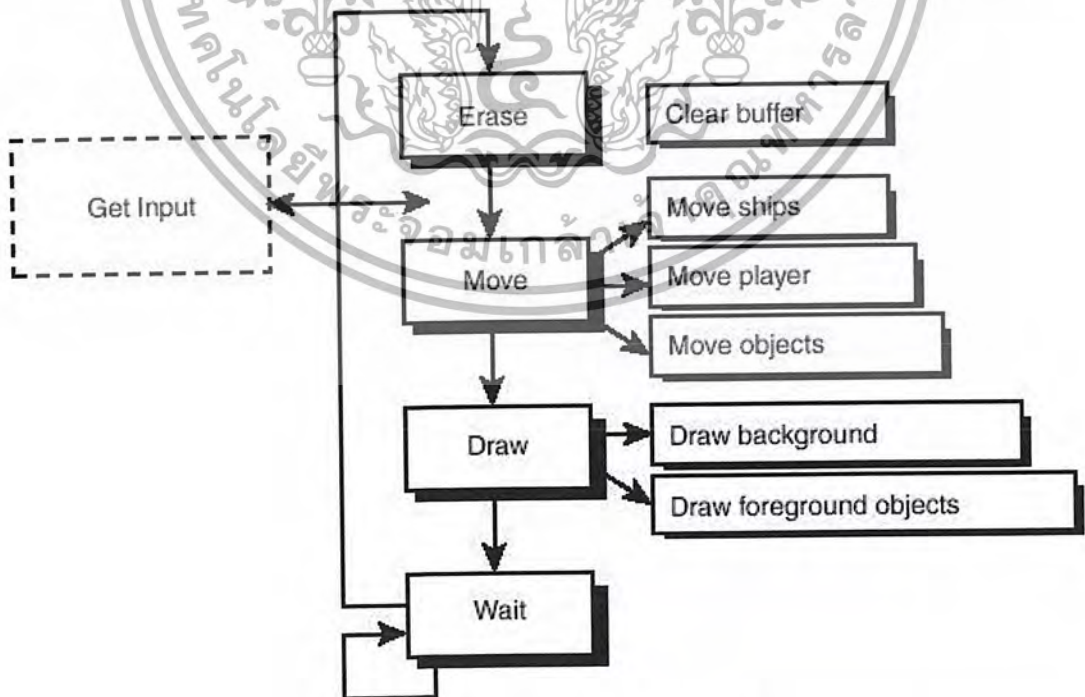
```
{
```

เมื่อต้องการยกเลิกการใช้เมาส์จะต้องทำการคืนเมาส์ให้สู่ระบบด้วยฟังก์ชัน Unacquire() และทำการลบ

ออปเจกของเมาส์ทิ้งไปด้วยฟังก์ชัน Release()

```
lpdimouse->Unacquire();
```

```
lpdimouse->Release();
```

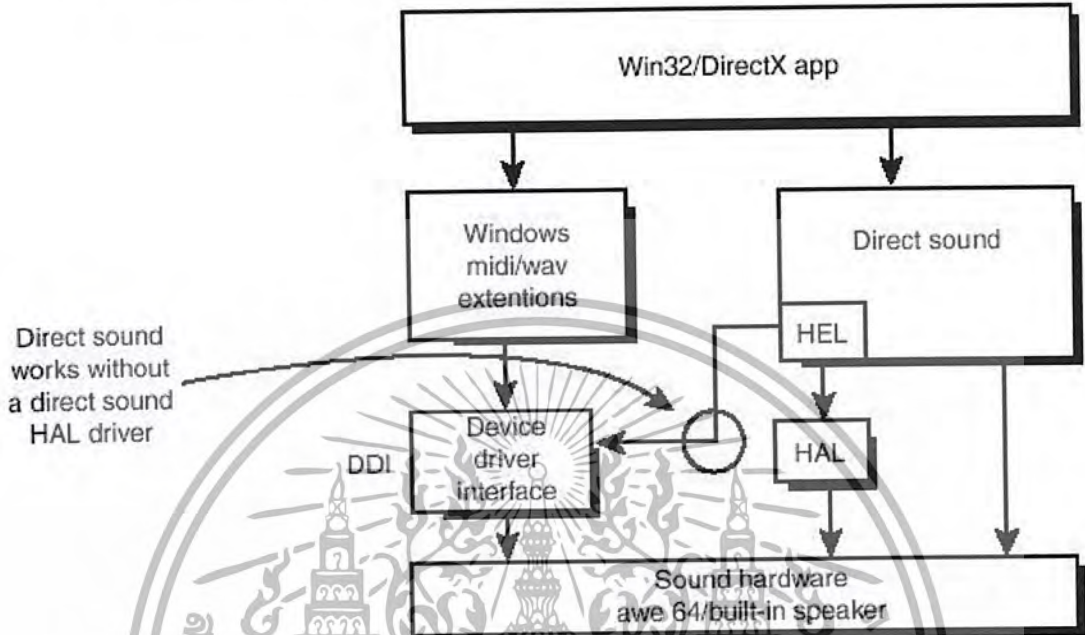


รูปที่ 2.14 รูปของข้อมูลอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 วิธีการแสดงเสียง

ในส่วนนี้จะกล่าวถึง Direct Sound ที่ใช้ในการสร้างเสียงและมีประโยชน์อย่างมากในกรณีที่ไม่
มีไดรฟ์เวอร์ของการ์ดเสียงตัวนั้น โดย Direct Sound จะติดต่อกับการ์ดเสียงได้โดยตรงไม่ต้องอาศัย
ไดรฟ์เวอร์ของการ์ดเสียง ดังรูป



รูปที่ 2.15 การทำงานของ DirectSound

ส่วนประกอบพื้นฐานของ Direct Sound มีอยู่ 3 ส่วน ได้แก่

- ขณะ run-time .DLL จะถูก โหลดขณะเรียกใช้ Direct Sound
- ขณะ compile-time Dsound.Lib จะทำการสร้าง Direct sound application
- ขณะ compile-time Dsound.h จะทำการสร้าง Direct sound application

ดังนั้นจะต้องทำการเพิ่ม file เหล่านี้เข้าไปใน application ที่ต้องการสร้างด้วย
ขั้นตอนการสร้าง Direct sound Object

2.6.1 การสร้าง Direct Sound Object

การที่จะสร้าง Direct sound object จะต้องทำการเรียก Function DirectSoundCreate() ซึ่ง
กำหนด prototype ดังนี้

```
HRESULT DirectSoundCreate(LPGUID lpGuid, // GUID ของ การ์ดเสียง
LPDIRECTSOUND *lpDS, //interface pointer to obj lunknow FAR *pUnkOuter)//ปกติNULL
```

โดยจะต้องประกาศ pointer เพื่อใช้ชี้ Direct Sound Object ดังนี้

```
LPDIRECTSOUND lpds;
```

เมื่อทำการสร้าง Object ของ Direct Sound ขึ้นมาแล้ว และเมื่อเรียกใช้เสร็จก็ต้องทำการคืนค่า Object นั้นด้วยคำสั่ง

lpds -> Release();

2.6.2 การตั้งค่าของ cooperation level

จะคล้ายกับการ set cooperation ของ Direct Draw โดยการเรียก Function SetCooperativeLevel() โดยมี prototype ดังนี้

HRESULT SetCooperativeLevel (HWND hwnd, // window handle

DWORD dwLevel); // ตั้งค่า Cooperation

ตารางที่ 2.7 ค่าแฟล็กของ cooperation level DirectSound

ค่าแฟล็ก dwLevel	ความหมาย
DSSCL_NORMAL	ตั้งเป็นแบบปกติโดยมีขนาด Primary Buffer สำหรับความถี่เสียง 22 kHz, Stereo แบบ 8 bit เป็นที่นิยมที่สุด
DSSCL_PRIORITY	อนุญาตให้เปลี่ยนขนาดของ Primary Buffer ได้ตามต้องการและสามารถเปลี่ยนรูปแบบเสียงเป็นแบบ 16 bit ได้
DSSCL_EXCLUSIV	เหมือนกับ Priority แต่จะได้ยินเสียงเพียงเสียงเดียว
DSSCL_WRITEPRIMAR Y	เป็นระดับสูงสุดจะควบคุมทุกอย่างของ Primary Buffer เหมาะสำหรับการที่จะสร้างเสียงขึ้นมาเอง

2.6.3 Primary และ secondary buffers

การสร้างเสียงขึ้นมาบนการ์ดเสียง โดยที่ตัวโปรแกรมไม่ต้องไปสนใจ Direct Sound จะเป็นตัวจัดการแทนให้ และไม่จำเป็นต้องสร้าง primary buffer โดยเพียงแค่ว่า set ค่า cooperative level ต่ำสุด Direct sound ก็จะจัดการให้ ส่วนที่จะต้องสนใจจริงๆ ก็คือ Secondary buffer ซึ่งส่วนนี้จะทำการเล่นเสียงที่โปรแกรมต้องการจะมีขนาดเท่าไรก็ได้ ขึ้นอยู่กับหน่วยความจำที่มีอยู่ อย่างไรก็ตาม SRAM ที่อยู่บนการ์ดเสียงก็เพียงพอที่จะเก็บเสียงได้มากตามที่ต้องการแล้ว

2.6.4 การสร้าง Secondary Buffer

การจะสร้าง secondary sound buffer ต้องทำการเรียก Function CreateSoundBuffer() โดยกำหนด prototype ไว้ดังนี้

HRESULT CreateSoundBuffer(LPCDSBUFFERDESC lpCDsBufferDesc,

// ตัวชี้ไปยัง DSBUFFERDESC

```
LPLPDIRECTSOUNDBUFFER lpLpDSBuff,
```

```
//ตัวชี้ไปยัง sound buffer
```

```
IUnknown FAR *pUnkOuter); // ปกติใช้ NULL
```

ก่อนที่จะสร้าง secondary sound buffer จะต้องกำหนดโครงสร้างของ DirectSoundbuffer เสียก่อนโดยมีรูปแบบดังนี้

```
typedef struct {
```

```
    DWORD          dwSize; // ขนาดของ โครงสร้างนี้
```

```
    DWORD          dwFlags; // ค่าแฟล็กควบคุม
```

```
    DWORD          dwBufferBytes; // ขนาดของบัพเฟอร์ของเสียงในรูป ไบต์
```

```
    DWORD          dwReserved; // ไม่ใช้
```

```
    LPWAVEFORMATEX lpwfxFormat; // รูปแบบรูปแบบของเสียง
```

```
} DSBUFFERDESC; *LPDSBUFFERDESC;
```

ตารางที่ 2.8 ค่าแฟล็กควบคุมของโครงสร้าง DirectSound Buffer

ค่าแฟล็กของ dwFlags	ความหมาย
DSBCAPS_CTRLALLBUFFER	ต้องมีการควบคุมขนาด
DSBCAPS_CTRLDEFAULT	เป็นการตั้งค่า default
DSBCAPS_CTRLFREQUENCY	ควบคุมความถี่ของเสียง
DSBCAPS_CTRLPAN	ควบคุมความ balance ของเสียง
DSBCAPS_CTRLVOLUME	ควบคุมความดังของเสียง
DSBCAPS_STATIC	บอกว่า buffer จะใช้สำหรับข้อมูลเสียงแบบ static
DSBCAPS_LOCHARDWARE	ใช้ Hardware ทำการสร้างเสียงขึ้นมาและใช้ หน่วยความจำ แทน buffer
DSBCAPS_LOCSOFTWARE	บังคับให้ buffer ใช้หน่วยความจำของ software เป็นตัวเก็บ
DSBCAPS_PRIMARYBUFFER	บอกให้รู้ว่าเป็นแบบ primarybuffer เพื่อสร้าง primary buffer ด้วยตัวเอง

โดยส่วนใหญ่ จะกำหนดค่า Control flag เป็น DSBCAPS_CTRLDEFAULT|
DSBCAPS_STATIC|DSBCAPS_LOCSOFTWARE

ต่อไปเป็นการ กำหนดรายละเอียดของเสียงโดยสร้างรูปแบบของ Waveformatex ขึ้นมาดังนี้

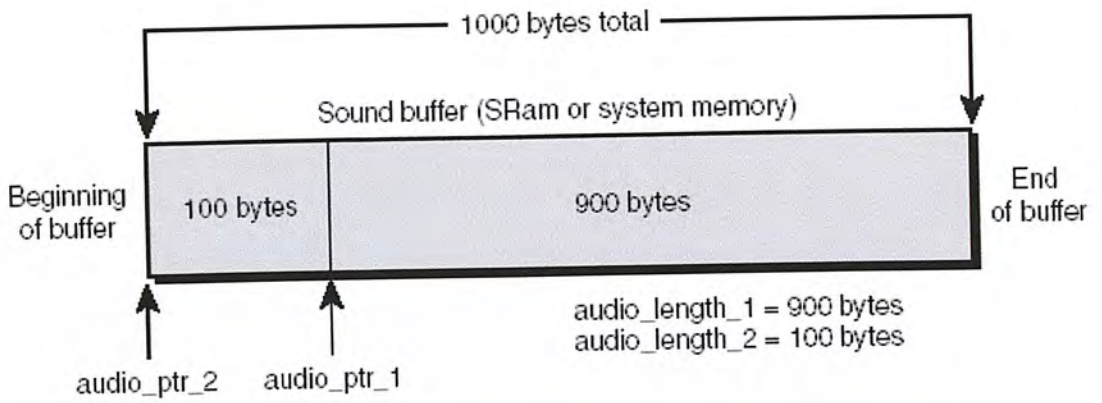
```
typedef struct {
WORD wFormatTag ; //ปกติใช้ WAVE_FORMAT_PCM
WORD nChannels ; //จำนวนช่องสัญญาณจะใช้ 1 สำหรับเสียงแบบโมโนและ 2 สำหรับสเตอริโอ
DWORD nSamplesPerSec ; //samples per second
DWORD nAvgBytesPerSec ; //average data rate
WORD nBlockAlign ; //nchannels * bytespersample
WORD wBitsPerSample ; //bits per sample
WORD cbSize ; // ตั้งเท่ากับ 0
} WAVEFORMATEX;
```

2.6.5 การเขียนข้อมูลลงบน Secondary Buffer

ใน Direct Draw Surface จะต้องทำการ lock surface memory ก่อน แล้วจึงเขียน ใน Direct Sound ก็เช่นเดียวกัน แตกต่างกันว่า Direct sound มี 2 พอยเตอร์ ที่ใช้ในการเขียน โดยพอยเตอร์แรก จะใช้เขียนข้อมูลในส่วนแรกและข้อมูลที่เหลือสำหรับพอยเตอร์ที่สอง โดยมีการกำหนด prototype Lock() ไว้ดังนี้

```
HRESULT Lock(DWORD dwWriteCursor,
             DWORD dwWriteBytes, // ขนาดของ Buffer ที่ต้องการ lock
             LPVOID lplpvAudioPtr1,
             LPDWORD lpdwAudioBytes1,
             LPVOID lpvpvAudioPtr2,
             LPDWORD lpdwAudioBytes2,
             DWORD dwFlags );
```

ตัวอย่าง เช่น ต้องการสร้าง Sound buffer ขนาด 1000 bytes เมื่อทำการ lock buffer เพื่อจะทำการเขียน จะต้องใช้ สองพอยเตอร์ในการอ้างอิง โดยขนาดของพอยเตอร์ตัวแรก อาจจะมีขนาด 900 bytes ส่วน ที่เหลือ 100 bytes จะเป็นของพอยเตอร์ตัวที่สอง ดังรูปภาพที่ 2.17



Note: ptrs are descending order

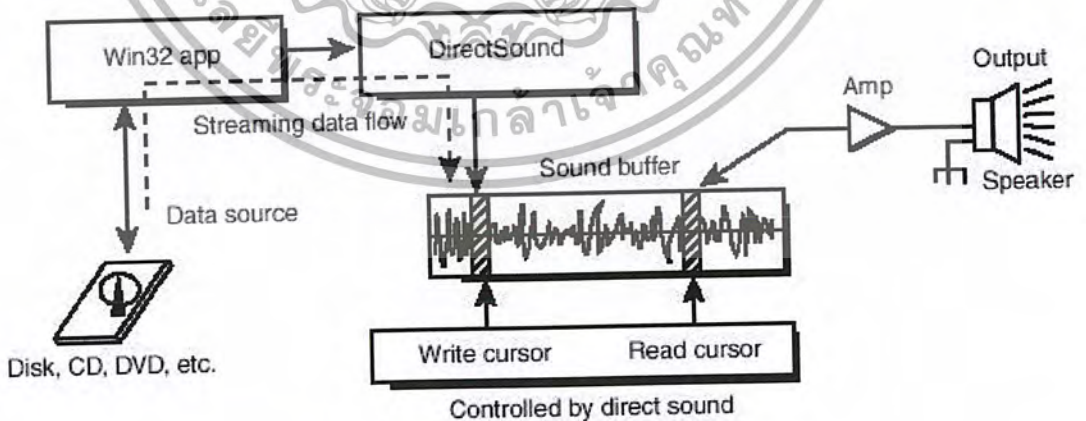
รูปที่ 2.16 การเขียนข้อมูลเสียงลง Sound Buffer

หลังจากทำการ สร้าง Sound Buffer และ โหลด sound ทั้งหมด ขึ้นคို့ไปก็จะเป็นการ นำ sound ที่พร้อมนำมาแสดง โดย Direct X มี Function มากมายในการแสดงเสียงที่จำเป็นได้แก่

2.6.6 การแสดงเสียง

ใช้ฟังก์ชัน Play() ในการแสดงเสียงจาก sound buffer โดยมีตัวต้นแบบดังนี้

```
HRESULT Play(DWORD dwReserved1,
              DWORD dwReserved2, // กำหนดเป็น 0 ทั้งสองค่า
              DWORD dwFlags); // ควบคุม flags ในการเล่น ถ้าต้องการเล่นวนรอบ
```



รูปที่ 2.17 การแสดงข้อมูลเสียง

2.6.7 การหยุดเสียง

ถ้าต้องการหยุดเสียงที่กำลัง เล่นอยู่ ใช้ฟังก์ชัน Stop () โดยกำหนดตัวคืนแบบดังนี้

```
HRESULT Stop();
```

2.6.8 การตั้งระดับความดังของเสียง

ถ้าต้องการปรับความดังของเสียงจะเรียกใช้ ฟังก์ชัน SetVolume() มีการกำหนดคืนแบบดังนี้

```
HRESULT SetVolume( LONG VOLUME );
```



บทที่ 3

ขั้นตอนการดำเนินการ

การสร้างเกมที่ดีนั้นควรที่จะมีการวางแผนให้รอบคอบก่อนที่จะทำการสร้างเกม รวมถึงการคำนึงถึงความเป็นไปได้ว่าจะสามารถพัฒนาได้หรือไม่ ดังนั้นการเขียนโครงสร้างของเกมก่อนทำการพัฒนาจึงเป็นเรื่องที่ควรกระทำโดยเฉพาะกับตัวเกมที่มีขนาดใหญ่ มีแผนงานทั้งหมดดังต่อไปนี้

1) ขั้นตอนการวิเคราะห์และออกแบบเกม

- การวิเคราะห์ความต้องการว่าต้องการจะสร้างเกมในรูปแบบใด เนื้อหาทั้งหมดมีลักษณะอย่างไร ออกแบบกติกาการเล่น ระบบของเกม และลักษณะแนวทางการเล่น รูปแบบตัวละคร ฉาก เสียง และเนื้อหาของเกม ว่าควรมีรูปแบบใด
- ออกแบบหน้าจอส่วนที่ใช้ติดต่อกับผู้เล่น ซึ่งต้องเป็นรูปแบบที่ใช้งานง่าย และดึงดูดความสนใจของผู้เล่น
- ออกแบบวิธีการจัดเก็บข้อมูลที่เป็นต้องใ้ภายในเกม โดยที่ควรใช้พื้นที่ในการจัดเก็บน้อย เป็นระเบียบ และสามารถนำมาใช้ได้ง่ายเพื่อเพิ่มประสิทธิภาพการทำงานให้กับเกม
- ออกแบบวิธีการเขียนโปรแกรม และเลือกใช้ฮาร์ดแวร์, ซอฟต์แวร์ รวมถึง component และเครื่องมือ ต่าง ๆ ที่นำมาใช้ในการสร้างเกม

2) ขั้นตอนการลงมือปฏิบัติการจริง

- ด้านภาพและกราฟิก การทำงานในส่วนนี้จะใช้โปรแกรมทางด้านกราฟิกเข้ามาช่วยในการสร้าง ฉากที่ได้ทำการออกแบบไว้แล้ว
- ด้านโปรแกรมมิ่ง การทำงานในส่วนนี้จะเกี่ยวข้องกับการเขียนคำสั่งต่างๆ โดยมีการนำฉากตัวละคร และเสียงประกอบ ที่ได้สร้างไว้มาใช้ในส่วนนี้ ซึ่งมีการจัดลำดับขั้นตอนการทำงานดังนี้

- ใช้โปรแกรม Visual C++ ในส่วนของการเขียนและแก้ไข Source code ของเกม
- ใช้ DirectX ช่วยในการพัฒนาโปรแกรมเกม
- นำฉาก ตัวละคร และเสียง มาประกอบเข้าด้วยกัน
- เขียนคำสั่งที่ทำให้เกิดการกระทำตามที่ต้องการ

3) ขั้นตอนการทดสอบเกม การทำงานในส่วนนี้จะขั้นตอนในการทดสอบเกมที่เราผลิตขึ้นว่ามี ความสมบูรณ์มากน้อยเพียงใด เพื่อหาข้อผิดพลาดและประเมินประสิทธิภาพของเกม เมื่อพบข้อผิดพลาดก็จะทำการแก้ไขโดยทันที

3.1 ขั้นตอนการวิเคราะห์และการออกแบบ

การออกแบบระบบเกมเป็นส่วนที่สำคัญที่สุดในขั้นตอนการออกแบบทั้งหมด เพราะระบบเกมคือ ทุกสิ่งทุกอย่างตั้งแต่แนวคิดของเกม การดำเนินเรื่องของเกม จนถึงการทำงานของระบบภายในเกม โดยการออกแบบมีรายละเอียดดังต่อไปนี้

3.1.1 ออกแบบระบบเกมและการกำหนดโครงสร้างของเกม

เป็นการกำหนดลักษณะของเกมตามที่เราต้องการ เพื่อให้ภายในเกมนั้นเป็นไปตามที่เราร้องต้องการ โดยระยะเวลาต่างๆ ภายในเกมนั้นเราจะกำหนดให้มีลักษณะดังนี้

กำหนดให้ 1 ปี มีทั้งหมด 48 สัปดาห์ และในแต่ละ 1 สัปดาห์นั้นจะมี 2 ช่วงเวลา คือ ช่วงเช้า และ ช่วงบ่าย ซึ่งในการทำกิจกรรมแต่ละครั้งนั้นเมื่อผู้เล่นได้ทำกิจกรรมไปแล้วผู้เล่นจะเสียเวลาช่วงเวลานั้นไปเลย เช่น เมื่อผู้เล่นเข้าไปทำกิจกรรมคือคุยกับคุณชายในช่วงเวลาเช้าแล้วเมื่อผู้เล่นคุยกับคุณชายเสร็จแล้วผู้เล่นจะเสียเวลาในช่วงเช้าไปเลย

ซึ่งกติกาในการเล่นเกมนั้นผู้เล่นจะต้องทำการรวบรวมข้อมูลต่างๆ ให้ได้ครบตามที่ระบุไว้ซึ่งก็มีทั้งหมด 9 อย่างด้วยกัน และผู้เล่นนั้นต้องทำการสอบวัดระดับด้วยซึ่งการสอบวัดระดับนั้นจะมีการจัดขึ้นที่โรงเรียนเท่านั้น เมื่อผู้เล่นรวบรวมข้อมูลได้ครบตามที่กำหนดแล้วและมีอุปกรณ์ในการประกอบคอมพิวเตอร์ครบตามที่กำหนดผู้เล่นก็สามารถประกอบคอมพิวเตอร์ได้เมื่อผู้เล่นประกอบคอมพิวเตอร์ได้แล้วก็จะมีข้อมูลขั้นตอนการอธิบายการประกอบคอมพิวเตอร์ตั้งแต่การติดตั้ง CPU มาให้จนกระทั่งสามารถประกอบคอมพิวเตอร์ 1 เครื่อง ได้สำเร็จ การที่ผู้เล่นจะสามารถประกอบคอมพิวเตอร์ได้นั้น ผู้เล่นต้องทำการสะสมชิ้นส่วนของอุปกรณ์คอมพิวเตอร์ให้ครบ โดยผู้เล่นนั้นสามารถเข้าไปเล่นเกมเพื่อชิงอุปกรณ์คอมพิวเตอร์ก็ได้ ถ้าผู้เล่นเล่นเกมนั้นขณะผู้เล่นก็จะได้อุปกรณ์คอมพิวเตอร์กับเงินเป็นของรางวัล แต่ถ้าผู้เล่นเล่นเกมนั้นแพ้ก็จะได้เงินรางวัลตามที่ตัวผู้เล่นนั้นสะสมได้เป็นของรางวัล โดยเกมที่สามารถชิงรางวัลได้นั้นคือ เกมจับคู่ เกมทายสูงต่ำ และ เกมตอบคำถาม หรือผู้เล่นสามารถซื้อชิ้นส่วนอุปกรณ์คอมพิวเตอร์จากร้านค้าก็ได้

3.1.2 การออกแบบรูปภาพและกราฟิก

รูปแบบของภาพกราฟิกในเกม มีลักษณะของรูปภาพเป็นแบบ 2 มิติ ใช้สีที่สดใสเพื่อดึงดูดความสนใจผู้เล่น ในเกมประกอบด้วยหน้าจอต่าง ๆ ดังนี้ หน้าจอหลักแสดงเมนูต่าง ๆ เพื่อเข้าสู่เกม หน้าจอขณะเล่นเกม หน้าจอ Load Game เพื่อเข้าสู่การเล่นเกมต่อจากครั้งที่แล้ว หน้าจอ Exit เพื่อต้องการออกจากเกม ซึ่งโปรแกรมที่นำมาใช้สำหรับสร้างและตกแต่งภาพที่เลือกใช้สำหรับเกมนี้ได้แก่ โปรแกรม Adobe Photoshop CS

3.1.3 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น

ในเกมเราใช้อุปกรณ์ที่ใช้ติดต่อกับผู้เล่น 2 อย่าง คือ เมาส์ และ คีย์บอร์ด โดยใช้เมาส์สำหรับเลือกกดปุ่มเมนูต่างๆ และใช้เมาส์เพื่อกดเวลาเล่นเกมย่อยๆ ที่เกิดขึ้นภายในเกม ส่วนคีย์บอร์ดนั้นใช้สำหรับการตั้งชื่อให้กับตัวละครที่เราเลือก ภายในเกมนั้นประกอบด้วย Interface หลักๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ที่ใช้ภายในเกม

Interface	ปุ่ม	หน้าที่
หน้าจอแรกของเกม		
Intro	New Game	เป็นการเริ่มเล่นเกมใหม่
	Load Game	เป็นการเล่นเกมต่อจากครั้งที่แล้ว
	Mini Game	เป็นการเข้าสู่การเล่นเกมน้อยต่างๆ
	Exit	ออกจากเกม
New Game		
หน้าจอการเลือกตัวละคร	Create Game	เป็นการเข้าสู่เกม
	Back to main Menu	กลับไปสู่หน้าจอเมนูหลักของเกม
Create Game		
หน้าจอการในห้อง	Profile	แสดงข้อมูลของตัวละคร
	New	แสดงข่าวสารต่างที่เกิดขึ้นภายในเกม
	Data	แสดงข้อมูลที่เราได้มาจากการเหตุการณ์ต่างๆ
	Create Computer	สร้างคอมพิวเตอร์เมื่อสะสมอุปกรณ์คอมพิวเตอร์ครบทุกชิ้นแล้ว
	Go to Town	ออกจากบ้านเพื่อเข้าสู่แผนที่เมือง
	Calendar	บอกวันที่
	Objective	บอกเป้าหมายที่ต้องทำ
	Exit Game	กลับสู่เมนูหลัก
Load Game		
Load Game	Load Game	เมื่อเลือกแล้วจะเป็นการเล่นต่อจากครั้งที่แล้ว
Mini Game		
หน้าจอ Mini Game	เกมต่อสู้	เข้าสู่ส่วนสู่การเล่นเกมน้อยต่อสู้
	เกมทายแค้นสูงต่ำ	เข้าสู่ส่วนสู่การเล่นเกมน้อยทายแค้นสูงต่ำ
	เกมตอบคำถาม	เข้าสู่ส่วนสู่การเล่นเกมน้อยตอบคำถาม
	เกมจับคู่	เข้าสู่ส่วนสู่การเล่นเกมน้อยจับคู่

3.1.4 การออกแบบการจัดเก็บข้อมูล

ข้อมูลที่เป็นที่ใช้นในเกมนี้ที่จะต้องจัดเก็บจะประกอบไปด้วยพารามิเตอร์ที่จำเป็นต่างๆ ของเกม สำหรับไฟล์ทั้งหมดนั้นต้องการให้เข้าถึงได้อย่างรวดเร็ว และมีความซับซ้อนน้อย จึงเลือกใช้วิธีเก็บเป็นแบบ Text File เพื่อให้มีการเข้าถึงข้อมูลได้ง่าย และยังใช้เนื้อที่น้อยกว่าการเก็บแบบ Database

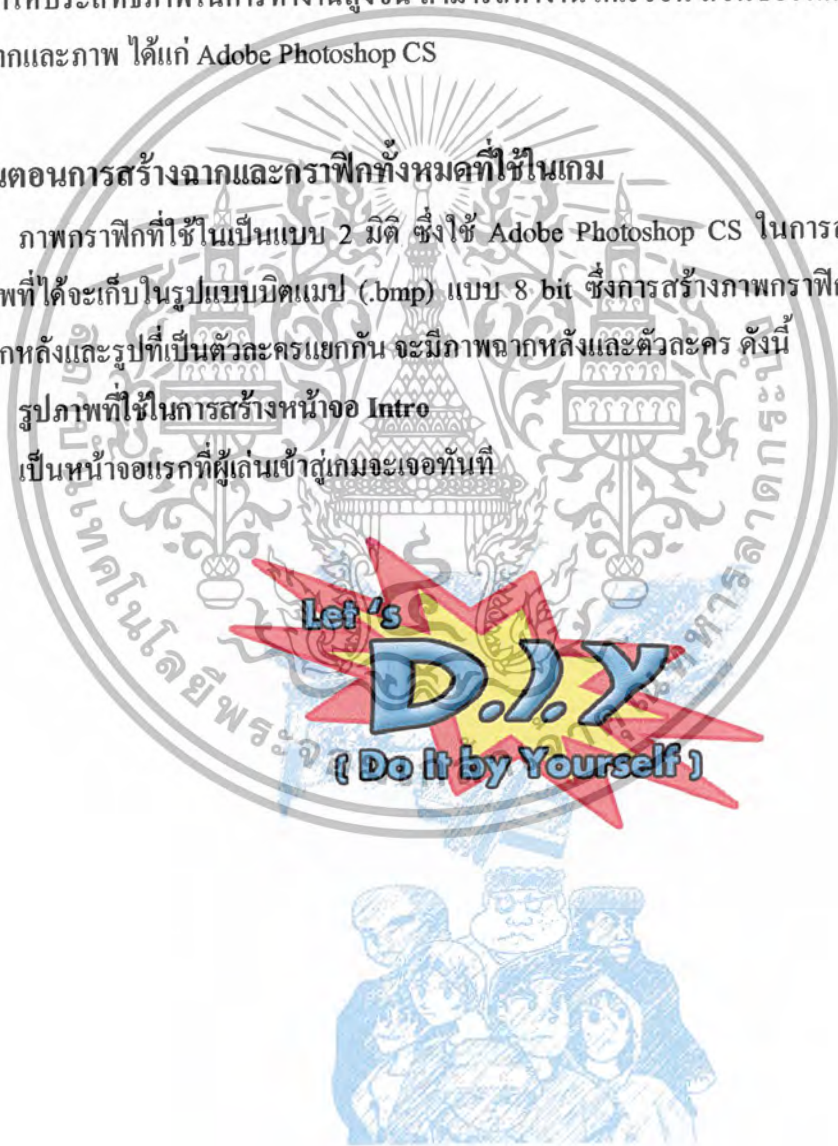
3.1.5 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้

การเขียนโปรแกรมเกมนี้ ได้เลือกใช้คอมพิวเตอร์ภาษา Visual C++ 6.0 มาใช้ร่วมกับ Direct X ซึ่ง DirectX จะเป็นผู้ติดต่อกับฮาร์ดแวร์ต่าง ๆ โดยตรงโดยเฉพาะอุปกรณ์แสดงผลทั้งภาพ และเสียง จุดเด่นของ DirectX นั่นคือ สามารถทำงานติดต่อกับฮาร์ดแวร์โดยไม่ต้องผ่านระบบปฏิบัติการ (OS) ทำให้ประสิทธิภาพในการทำงานสูงขึ้น สามารถทำงานได้เร็วขึ้น ส่วนซอร์ฟแวร์ที่ใช้ในการสร้างฉากและภาพ ได้แก่ Adobe Photoshop CS

3.2 ขั้นตอนการสร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม

ภาพกราฟิกที่ใช้เป็นแบบ 2 มิติ ซึ่งใช้ Adobe Photoshop CS ในการสร้างภาพกราฟิก โดยภาพที่ได้จะเก็บในรูปแบบบิตแมป (.bmp) แบบ 8 bit ซึ่งการสร้างภาพกราฟิก ต้องสร้างรูปที่เป็นฉากหลังและรูปที่เป็นตัวละครแยกกัน จะมีภาพฉากหลังและตัวละคร ดังนี้

รูปภาพที่ใช้ในการสร้างหน้าจอ Intro
เป็นหน้าจอแรกๆที่ผู้เล่นเข้าสู่เกมจะเจอทันที



รูปที่ 3.1 รูปฉากหลังหน้าจอ Intro

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NEW GAME NEW GAME
LOAD GAME LOAD GAME
MINI GAME MINI GAME
EXIT GAME EXIT GAME

รูปที่ 3.2 รูปปุ่มต่างๆ ของหน้าจอ Intro

รูปภาพที่ใช้ในการสร้างเม้าส์
 สำหรับเป็น Icon ที่ใช้ในการบอกตำแหน่งของเม้าส์ภายในเกม



รูปที่ 3.3 แสดงภาพเม้าส์

รูปภาพที่ใช้ในการสร้างปุ่มต่างๆ ของ Mini Game
 เป็นปุ่มที่เอาไว้กดเพื่อเลือกเล่นเกม

รูปที่ 3.4 รูปภาพ Mini Game ของเกมต่ออุปกรณ์

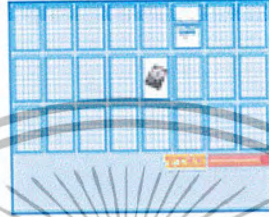


รูปที่ 3.5 รูปภาพ Mini Game ของเกมทายสูงต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 รูปภาพ Mini Game ของเกมตอบคำถาม



รูปที่ 3.7 รูปภาพ Mini Game ของเกมจับคู่



รูปที่ 3.8 รูปแสดงพื้นหลังของฉากรเลือก Mini Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

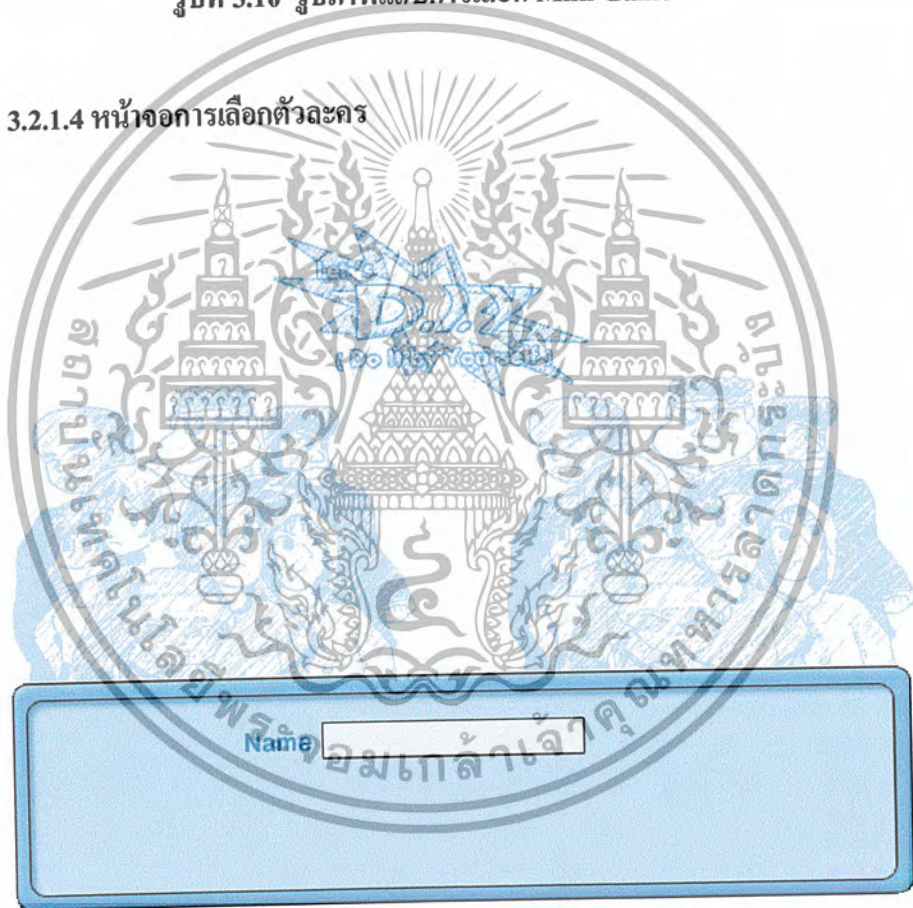
CLOSE CLOSE

รูปที่ 3.9 ปุ่มของฉาก Mini Game



รูปที่ 3.10 รูปภาพแถบการเลือก Mini Game

3.2.1.4 หน้าจอการเลือกตัวละคร

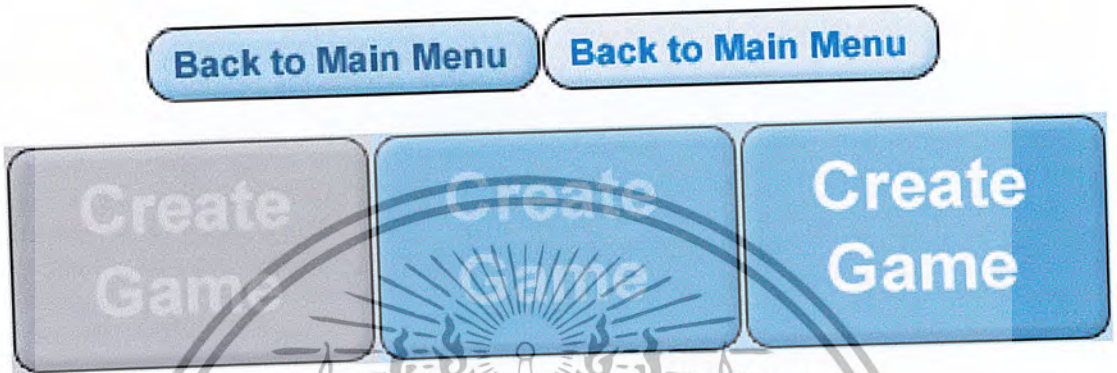


รูปที่ 3.11 ฉากหลังของการเลือกตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 รูปตัวละคร



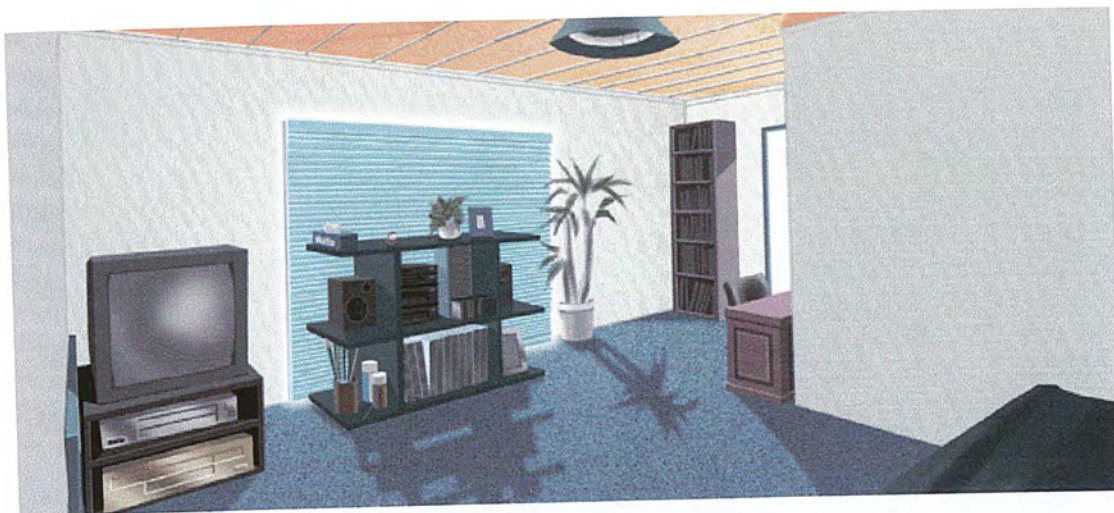
รูปที่ 3.13 รูปปุ่มต่างๆ ของฉากการเลือกตัวละคร

รูปภาพที่ใช้ในการสร้างหน้าจอภายในห้อง
จะแสดงรายละเอียดที่ใช้ภายในห้อง



รูปที่ 3.14 รูปแสดงเมนูของฉากภายในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

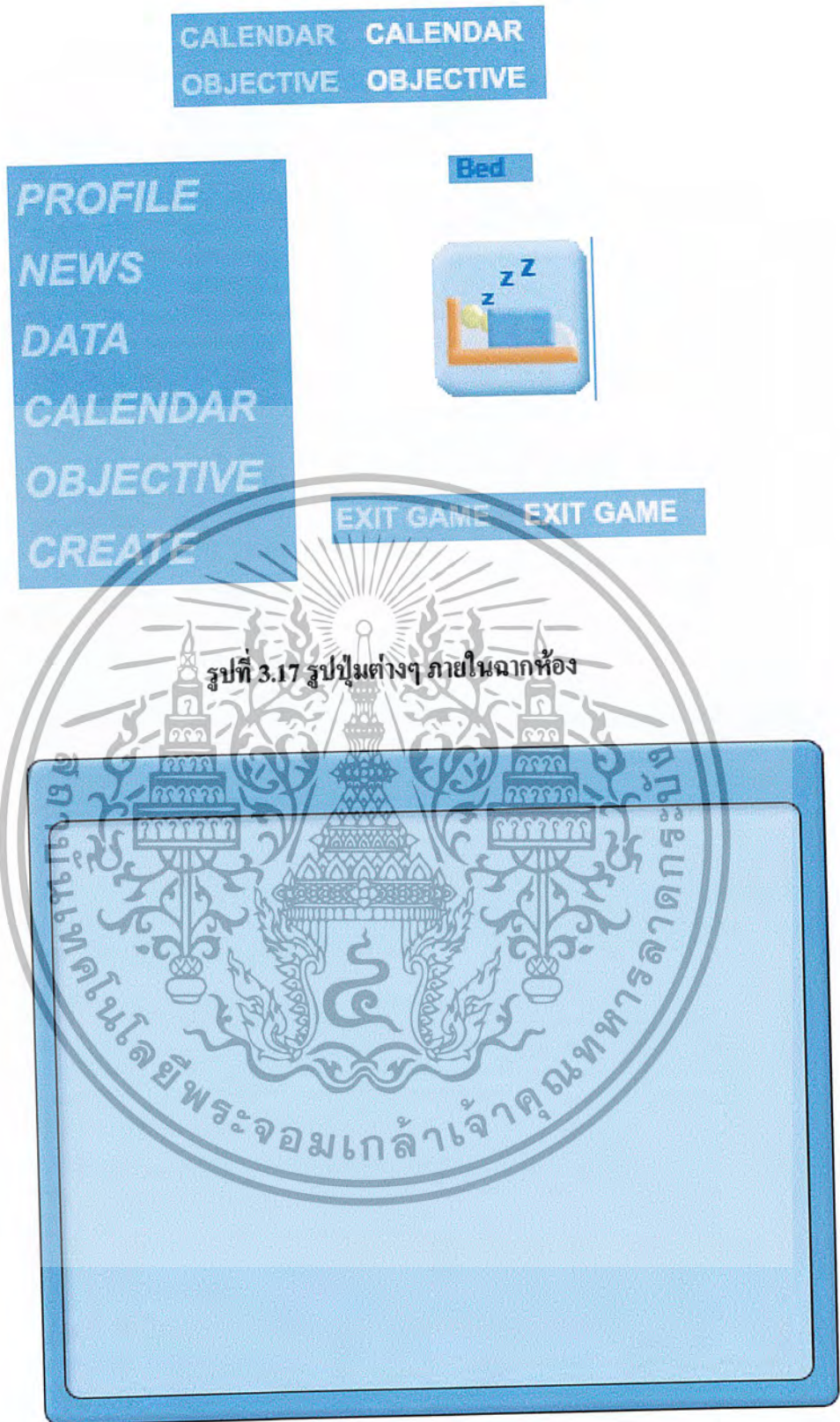


รูปที่ 3.15 รูปฉากภายในห้อง



รูปที่ 3.16 รูปพื้นหลังของฉากห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 รูปแสดงฉากหลังของ Dialog ต่างๆ ภายในเกม

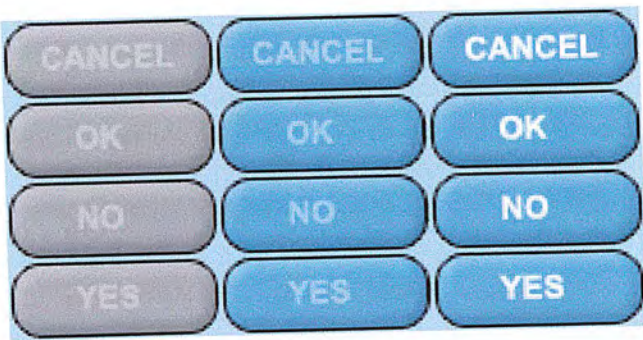
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 รูปปุ่มต่างๆ ของ Dialog ต่างๆ ภายในเกม

รูปที่ 3.20 รูปแสดงฉากหลังของ Messages Box ต่างๆ ภายในเกม

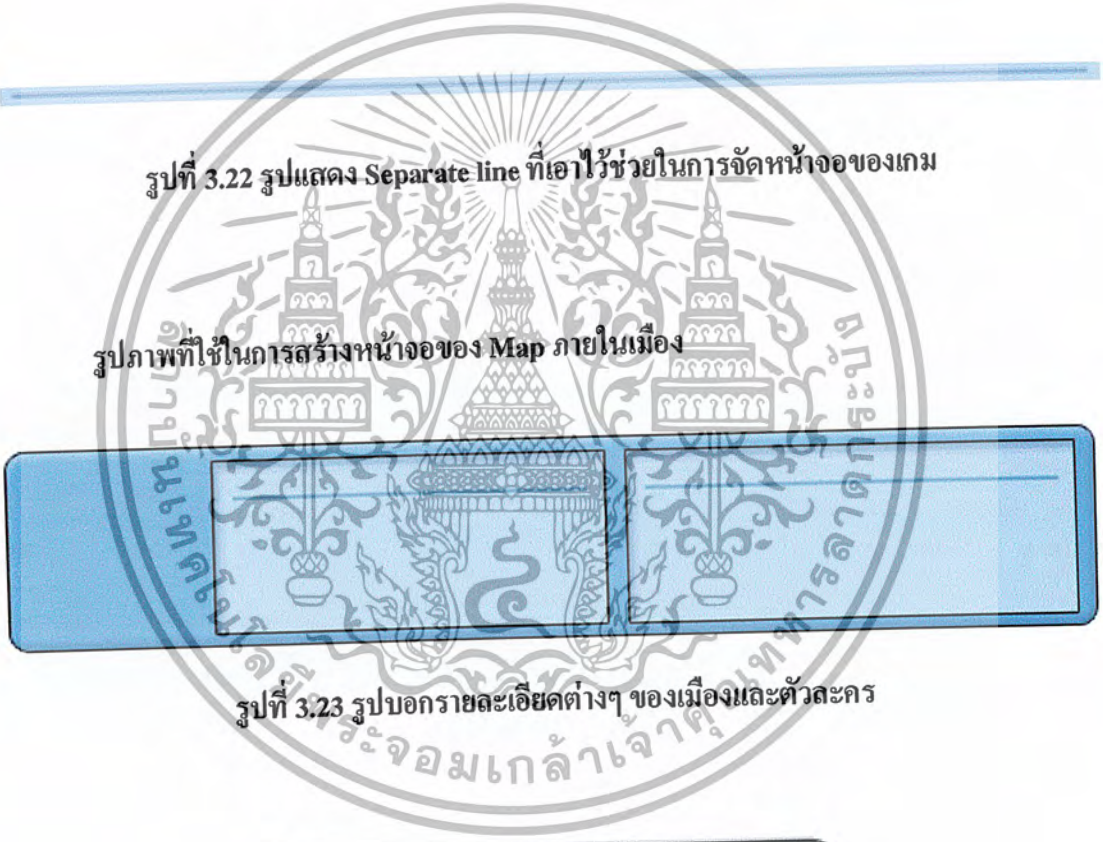
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 รูปแสดงปุ่มต่างๆ ของ Messages Box

รูปที่ 3.22 รูปแสดง Separate line ที่เอาไว้ช่วยในการจัดหน้าของเกม

รูปภาพที่ใช้ในการสร้างหน้าจอของ Map ภายในเมือง



รูปที่ 3.23 รูปบอกรายละเอียดต่างๆ ของเมืองและตัวละคร



รูปที่ 3.24 รูปบอกสถานะของวันและปี

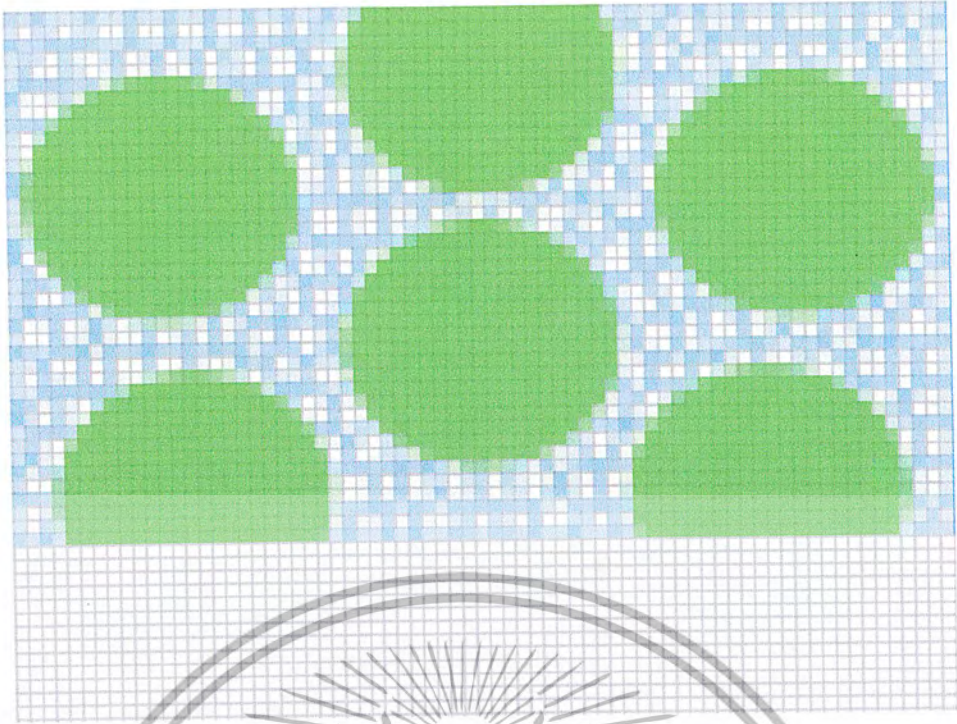


รูปที่ 3.25 รูปของสถานที่ต่างๆ ภายในเมือง



รูปที่ 3.26 รูปแสดงสถานที่ต่างๆ ภายในเมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 รูปแสดงฉากหลังของแผนที่เมือง
รูปภาพที่ใช้ในการสร้างรูปภาพ Event ต่างๆ ภายในเกม

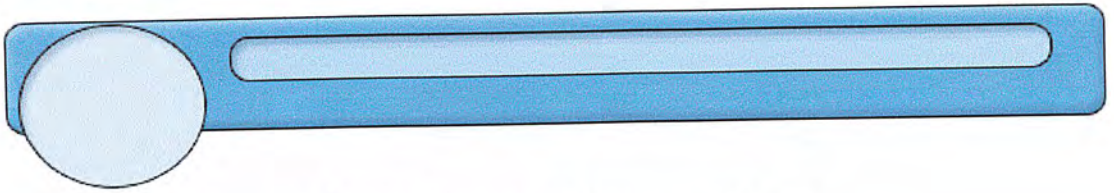


รูปที่ 3.28 รูปแสดงการเลือก Event ต่างๆ

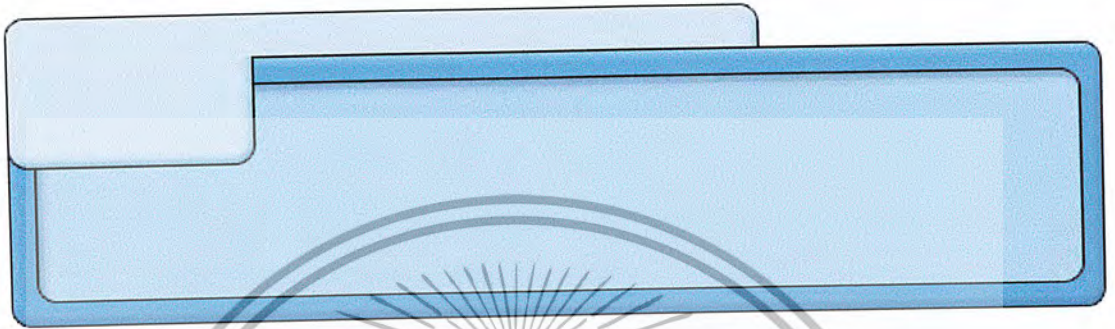


รูปที่ 3.29 รูปแสดงแถบการเลือก Event

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 รูปแสดงหน้าตาตัวละคร และวันเวลา



รูปที่ 3.31 รูปแสดงรายละเอียดของตัวละครและบทสนทนา

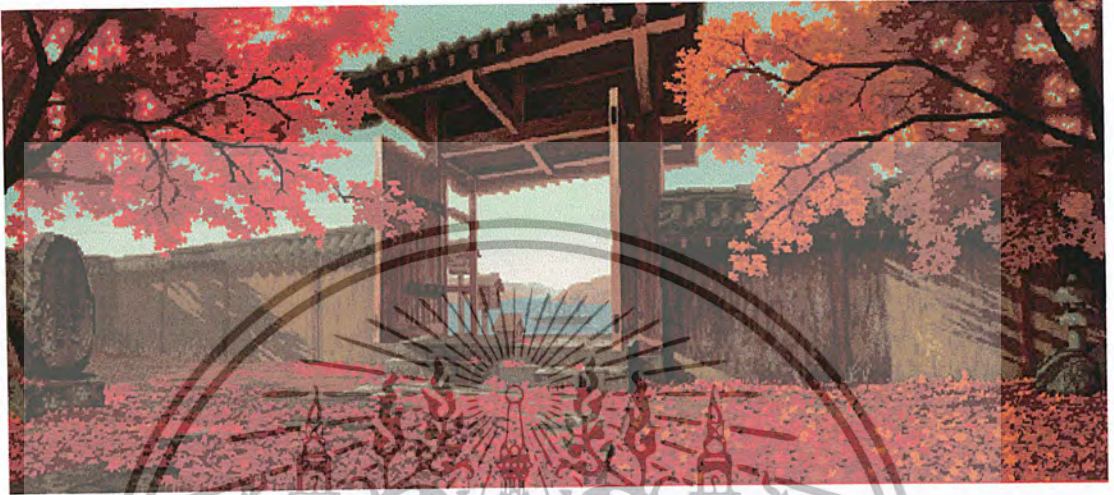


รูปที่ 3.32 รูปฉากพื้นหลังของการเลือกซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.33 รูปตัวละครต่างๆ ภายในเกม



รูปที่ 3.34 รูปฉากพื้นหลังของโรงเรียน

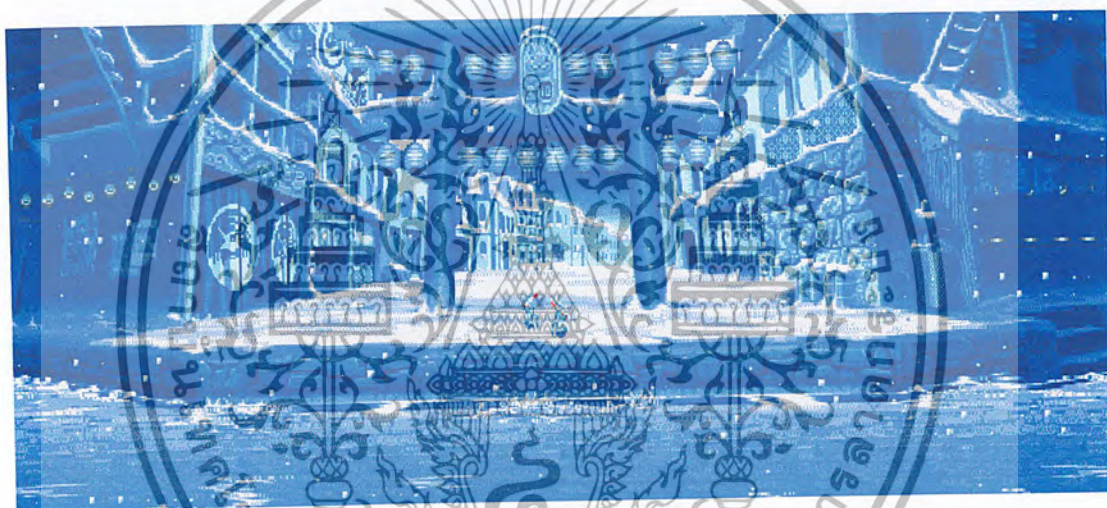


รูปที่ 3.35 รูปฉากพื้นหลังของ IT mall ที่ 1 และ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 รูปฉากพื้นหลังของสวนสาธารณะ



รูปที่ 3.37 รูปฉากพื้นหลังของ IT mall ที่ 2

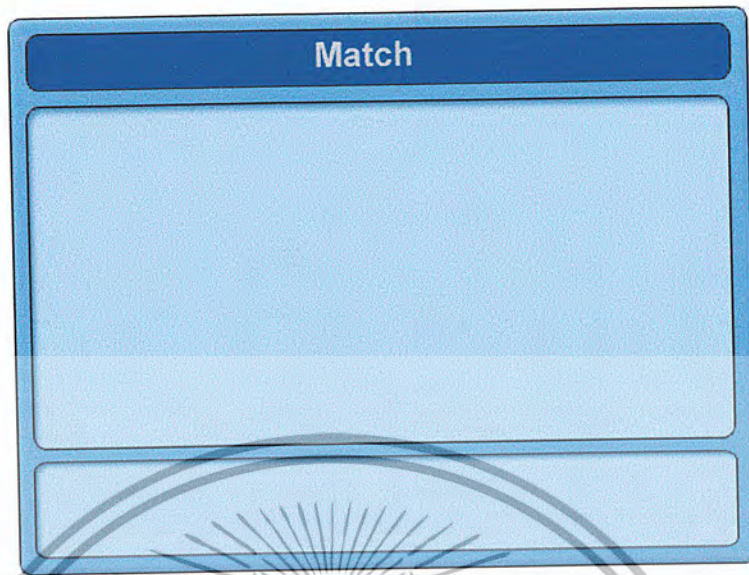


รูปที่ 3.38 รูปฉากพื้นหลังภายในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพที่ใช้ในการสร้างรูปภาพเกมย่อยต่างๆ

1 รูปภาพของเกมจับคู่



รูปที่ 3.39 ฉากหลังของเกมจับคู่ภายในเกมใหญ่



รูปที่ 3.40 ฉากหลังของเกมจับคู่ของ Mini Game



รูปที่ 3.41 รูปของจำนวนไฟทั้งหมดที่ใช้ในเกม (เกมใหญ่และ Mini Game)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MATCH READY YOU WIN GAME OVER

TIME

รูปที่ 3.42 รูปตัวอักษรต่างๆ ที่ใช้บอกสถานะของเกมจับคู่ใน Mini Game



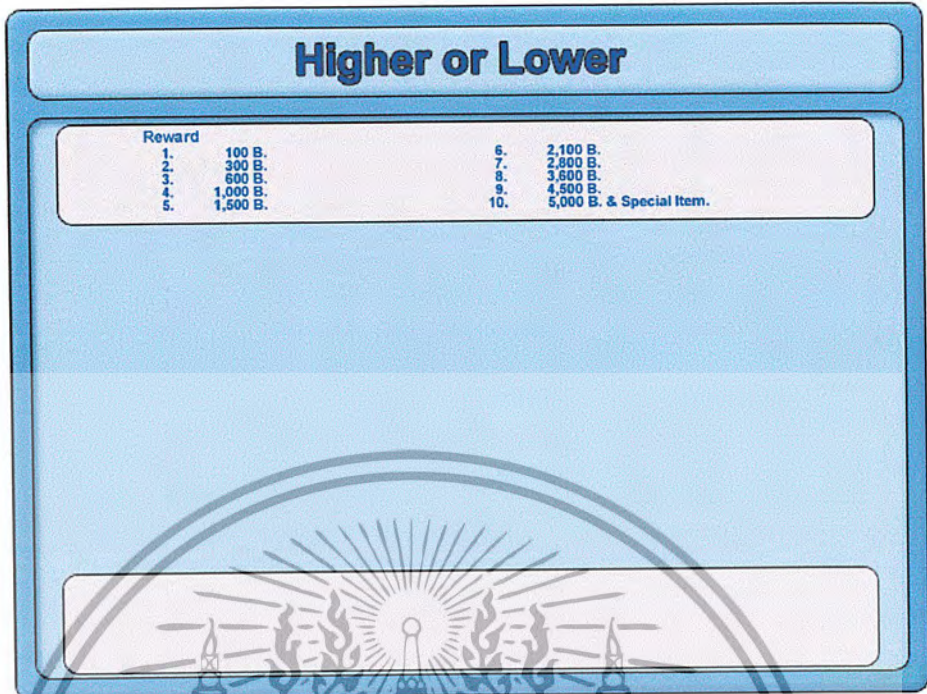
รูปที่ 3.43 รูปของแถบเวลาที่ใช้ในเกมจับคู่ใน Mini Game



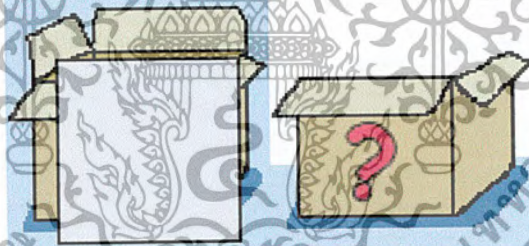
รูปที่ 3.44 รูปของตัวนับการจับคู่ที่ผิดไปแล้วก็ครั้งของเกมจับคู่ในเกมใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 รูปภาพของเกมทายสูงต่ำ



รูปที่ 3.45 รูปฉากพื้นหลังของเกมทายสูงต่ำ

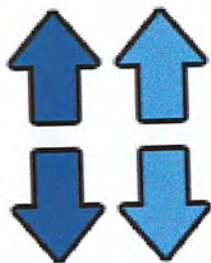


รูปที่ 3.46 รูปกล่องเก็บตัวเลข

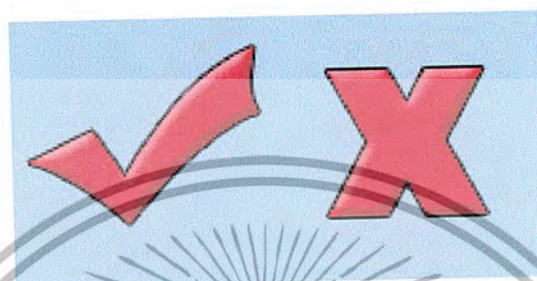


รูปที่ 3.47 รูปตัวเลขที่เอาไว้ทายสูงต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

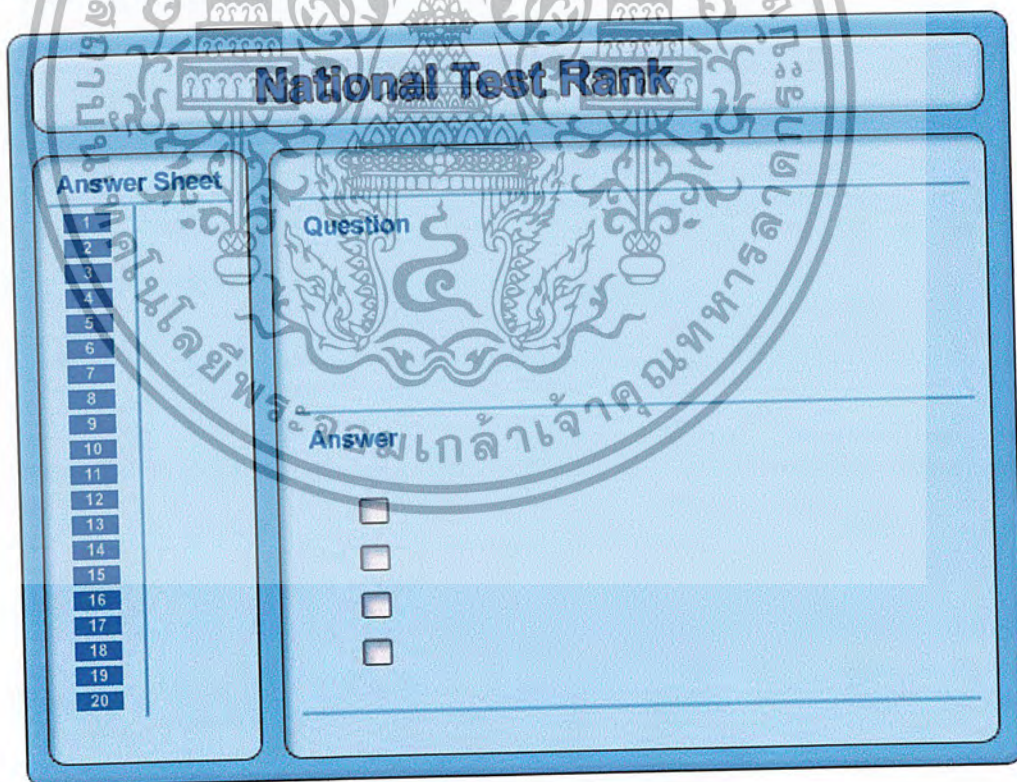


รูปที่ 3.48 รูปลูกศรที่เอาไว้ท่ายสูงหรือต่ำ



รูปที่ 3.47 รูปเครื่องหมายถูกผิดที่เอาไว้แสดงตอนท้ายตัวเลข

3 รูปภาพของเกมตอบคำถาม



รูปที่ 3.50 รูปพื้นหลังของเกมตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Next > Next >
< Back < Back

Finish Finish

รูปที่ 3.51 รูปปุ่มต่างๆ ที่ใช้ในเกมตอบคำถาม

รูปที่ 3.52 รูป Icon ที่เอาไว้เลือกข้อที่ต้องการจะทำข้อสอบ

รูปที่ 3.53 รูปเครื่องหมายถูกที่เอาไว้แสดงการเลือกคำตอบในข้อนั้นๆ

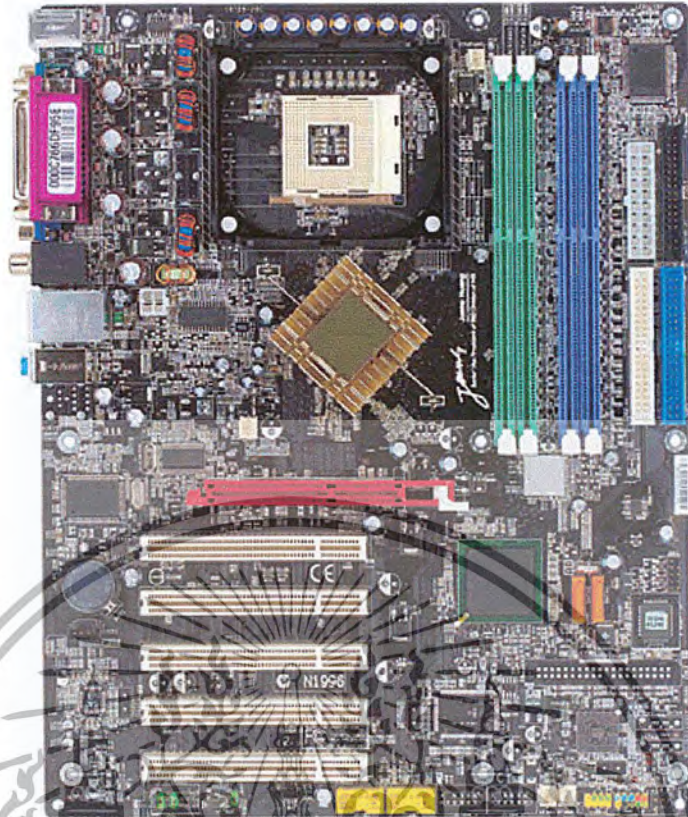
รูปที่ 3.54 รูปตัวเลขที่แสดงคำตอบที่เราเลือก

รูปที่ 3.55 รูปสัญลักษณ์ที่เอาไว้แสดงอันดับของ Lank

รูปที่ 3.56 รูป CPU

รูปที่ 3.57 รูปสาย Floppy Disk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.58 รูป Main board (มองจากมุมบน)

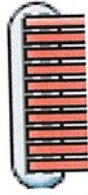


รูปที่ 3.59 รูป Heat sink และพัดลม (มองจากมุมบน)



รูปที่ 3.60 รูปสายแพ IDE (มองจากมุมบน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.61 รูปสายไฟจาก Power Supply แบบ 20 pin(มองจากมุมบน)



รูปที่ 3.62 รูปสายไฟจาก Power Supply แบบ 4 pin (มองจากมุมบน)



รูปที่ 3.63 รูป RAM (มองจากมุมบน)

รูปที่ 3.64 รูป VGA (มองจากมุมบน)



รูปที่ 3.65 รูปพื้นหลังที่ไว้แสดงข้อความ

3.3 ขั้นตอนการเขียนโปรแกรม

ศึกษารูปแบบวิธีการเขียนโปรแกรมที่เป็นลักษณะ Real time และ การเขียนโปรแกรมแบบเชิงวัตถุ เพื่อให้ได้โปรแกรมที่กะทัดรัดและทำงานได้ถูกต้อง เข้าใจง่าย และพยายามเขียนโปรแกรมให้มีลักษณะที่เป็นไปในทิศทางเดียวกัน

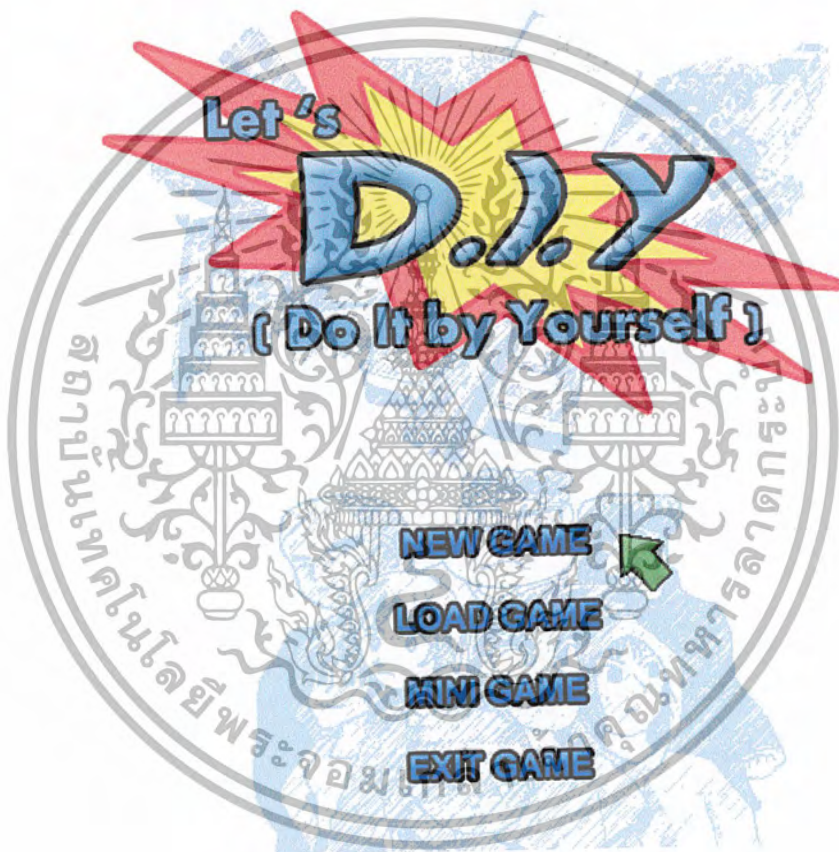
ในการพัฒนาโปรแกรมที่เป็นเกม คำสั่งควบคุมการทำงานทุกอย่างจะเขียนอยู่ภายในรูป เพื่อเป็นการง่ายในการเขียน โปรแกรม และง่ายในการตรวจสอบ แก้ไข หรือเพิ่มเติม จึงได้แบ่งการเขียนโปรแกรมให้ขึ้นอยู่กับแต่ละหน้าจอ โดยการตรวจสอบว่าขณะนี้หน้าจอของผู้เล่นเป็นหน้าจอใด แล้วเขียนโปรแกรมควบคุมการทำงานให้กับหน้านั้น ๆ โดยภายในหนึ่งหน้าจอจะมีการเขียนโปรแกรมควบคุมการทำงานหลัก ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แสดงฉากหลัง รูปภาพและปุ่มต่าง ๆ ของหน้าจอ นั้น
- รอรับอินพุตจากผู้เล่นผ่านทาง การคลิกเมาส์หรือการพิมพ์จากคีย์บอร์ด

3.3.1 หน้าจอ Intro

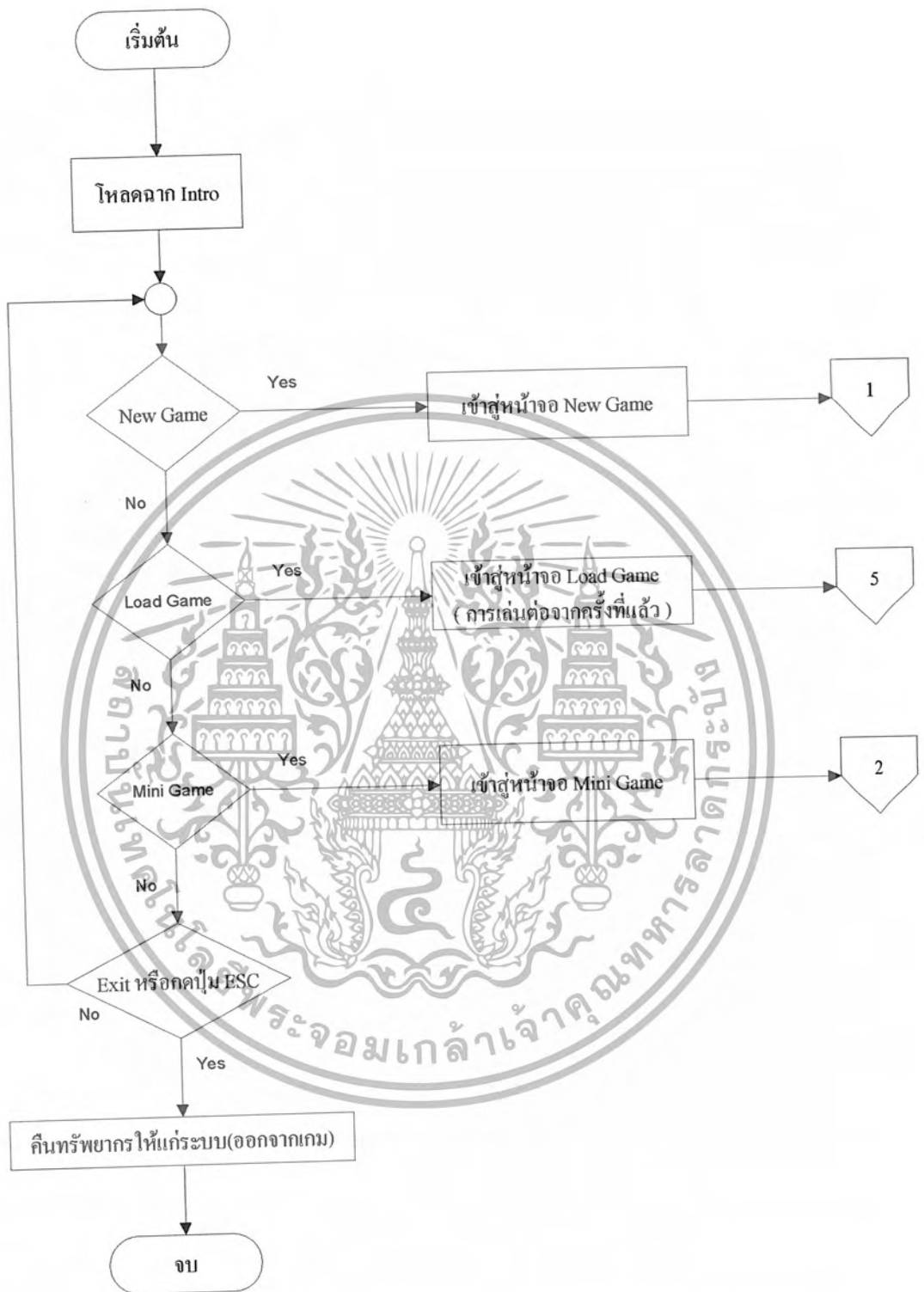
ในหน้าจอ Intro มีปุ่มเอาไว้ให้เลือกทั้งหมด 4 ปุ่ม ได้แก่ ปุ่ม New Game, ปุ่ม Load Game, ปุ่ม Mini Game, ปุ่ม Exit ดังรูป



รูปที่ 3.66 รูปหน้าจอ Intro

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนภาพการทำงานดังนี้

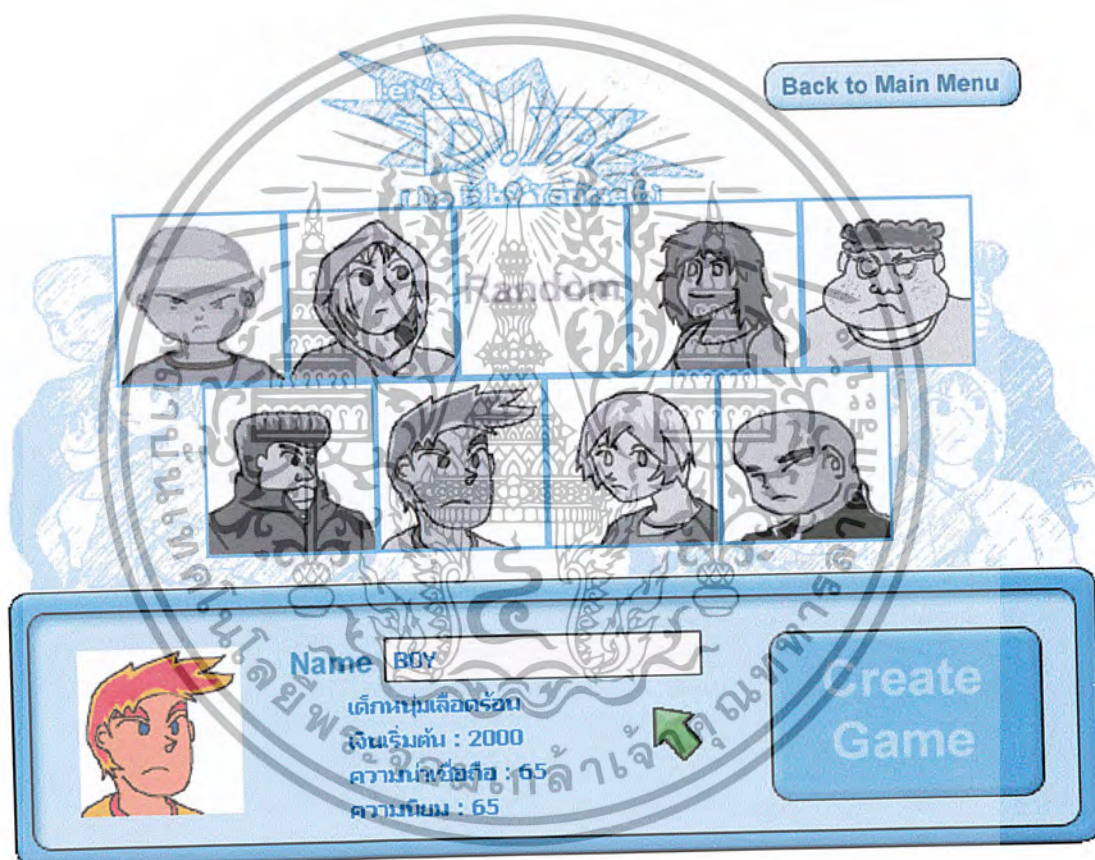


รูปที่ 3.67 แผนภาพการทำงานในหน้าจอ Intro

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

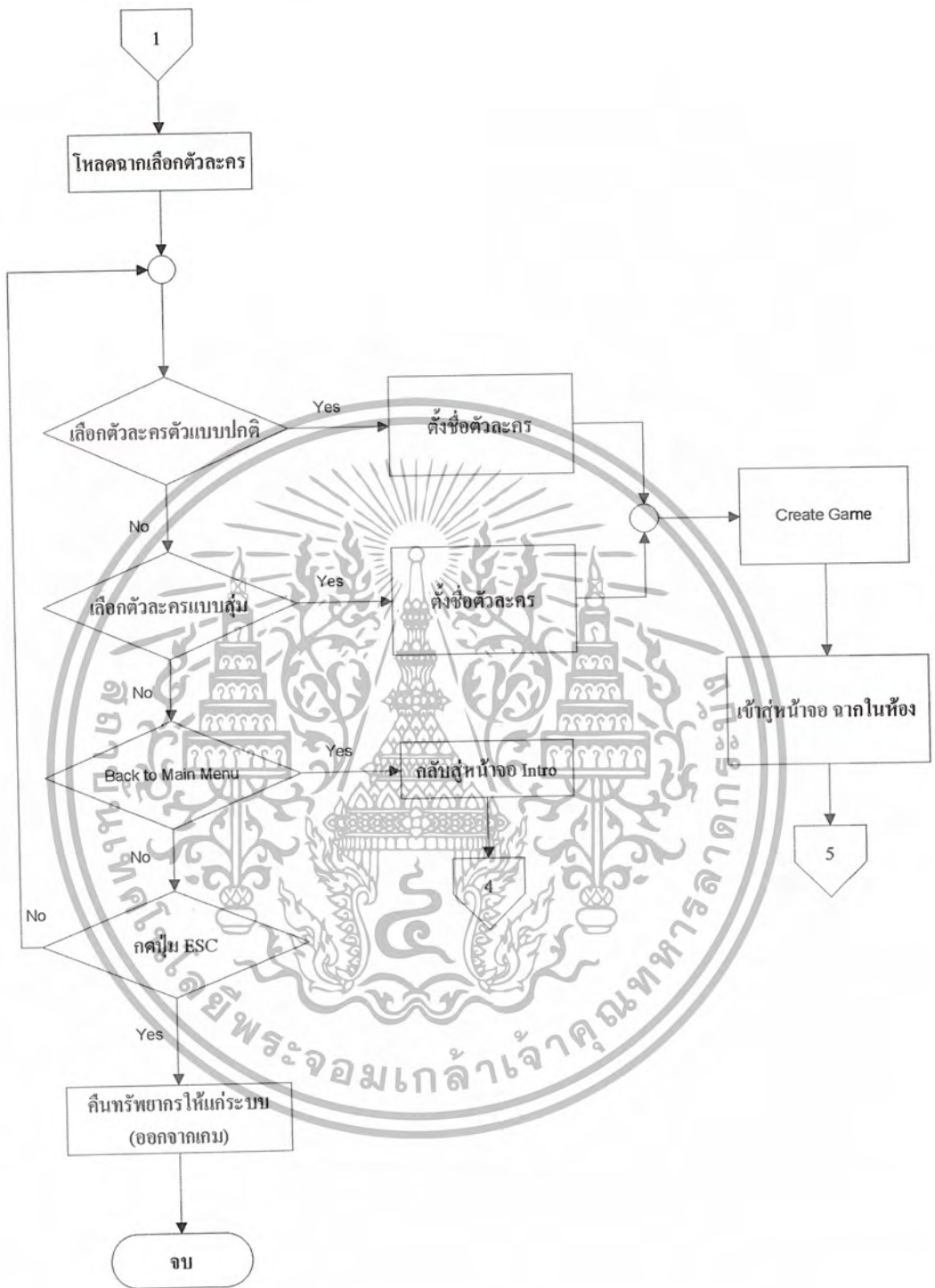
3.3.2 หน้าจอการเลือกตัวละคร

ในการเลือกตัวละครนั้นสามารถทำได้ 2 วิธีคือ วิธีแรกเป็นการเลือกตัวละครแบบปรกติคือเมื่อผู้เล่นนั้นตัดสินใจเลือกตัวละครที่มีลักษณะคล้ายกับตัวผู้เล่นเองหรือมีความชอบส่วนตัว วิธีที่สองเป็นการเลือกตัวละครแบบสุ่มวิธีเป็นการเลือกแบบไม่เจาะจงเพราะว่าผู้เล่นนั้นยังตัดสินใจไม่ได้ว่าจะเลือกเล่นตัวละครแบบไหน เมื่อผู้เล่นเลือกตัวละครได้แล้วนั้นจะต้องทำการตั้งชื่อก่อน โดยการจัดชื่อนั้นจะทำการรับค่ามาจากคีย์บอร์ด เมื่อผู้เล่นตั้งชื่อเสร็จแล้วจะสามารถกดปุ่ม Create Game ได้ หรือถ้าผู้เล่นต้องการกลับไปหน้าจอ Intro ก็สามารถกดที่ Back to Main Menu หรือถ้าผู้เล่นต้องการออกจากเกมก็สามารถกดปุ่ม ECS ได้



รูปที่ 3.68 รูปแสดงหน้าจอการเลือกตัวละคร

มีแผนภาพการทำงานดังนี้

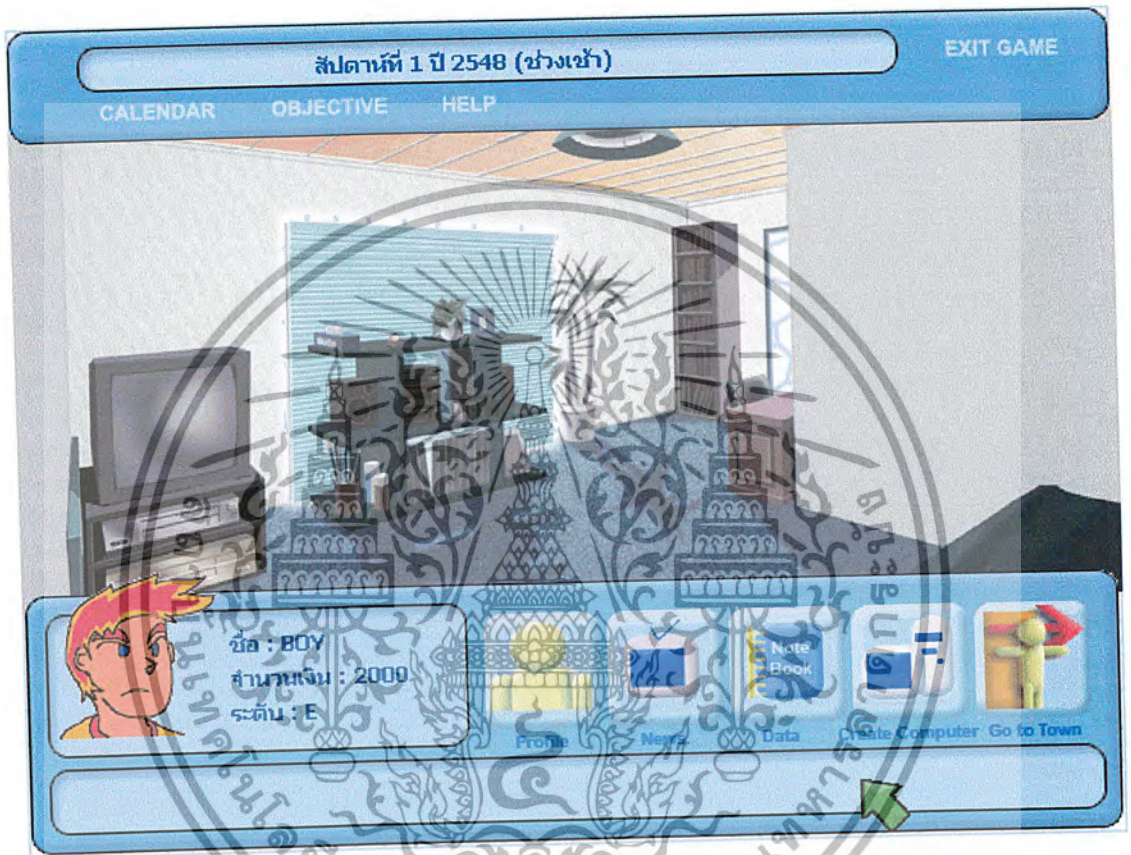


รูปที่ 3.69 แผนภาพการทำงานของหน้าจอการเลือกตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 หน้าจอฉากภายในห้อง

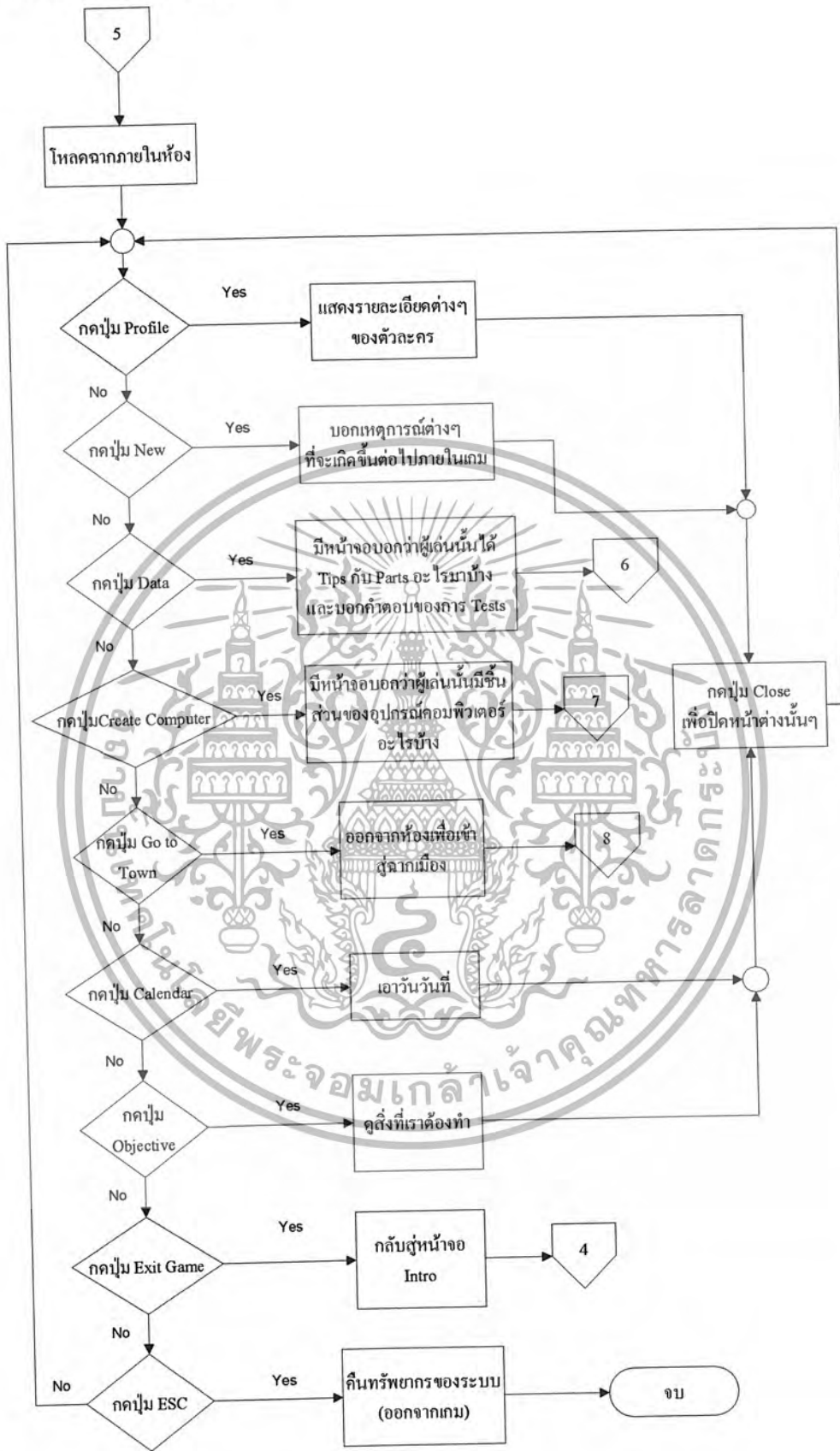
ซึ่งภายในฉากนี้จะประกอบไปด้วย ปุ่มเมนูต่างๆ มากมายเอาไว้ให้กดดู เช่น Profile, New, Data, Create Computer, Go to Town, Calendar, Objective เป็นต้นซึ่งปุ่มต่างๆ นั้นจะมีรายละเอียดต่างๆ เอาไว้ให้ผู้เล่นเกมสามารถดูได้ว่าจะต้องทำอะไรเป็นอันดับแรก และผู้เล่นเกมสามารถดูรายละเอียดต่างๆ ได้ว่าผู้เกมนั้นมีอุปกรณ์อะไรบ้างแล้ว ได้ของอะไรมาบ้างแล้ว



รูปที่ 3.70 รูปแสดงฉากภายในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนภาพการทำงานดังนี้

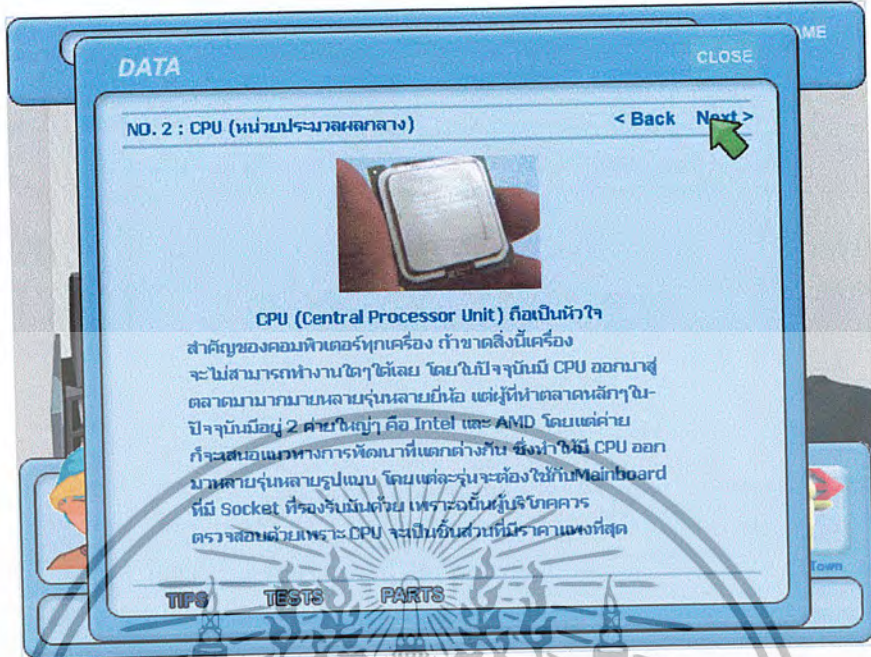


รูปที่ 3.71 แผนภาพการทำงานของฉากภายในห้อง

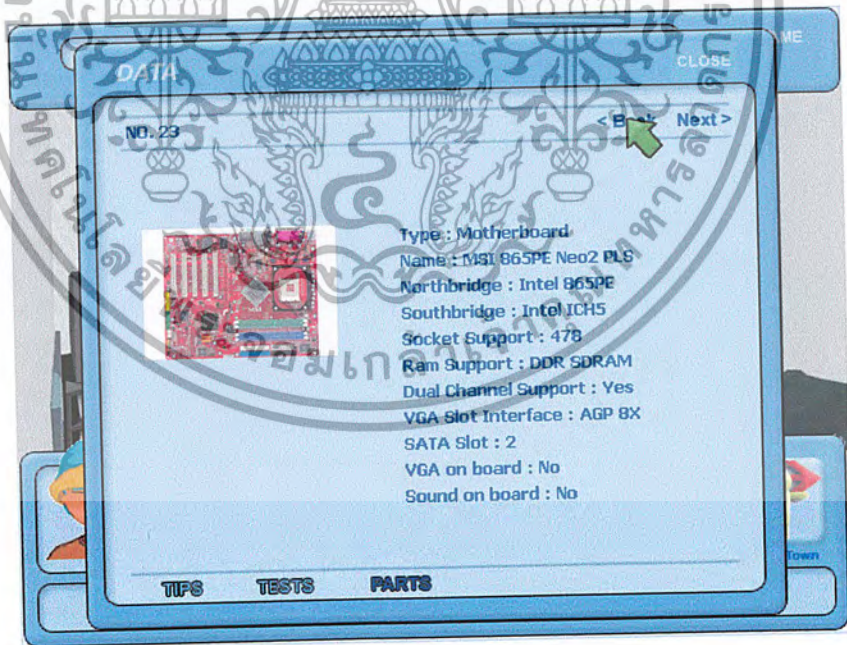
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.1 หน้าจอ Data

จะประกอบไปด้วยทั้งหมด 3 ส่วน คือ Tips, Tests และ Parts



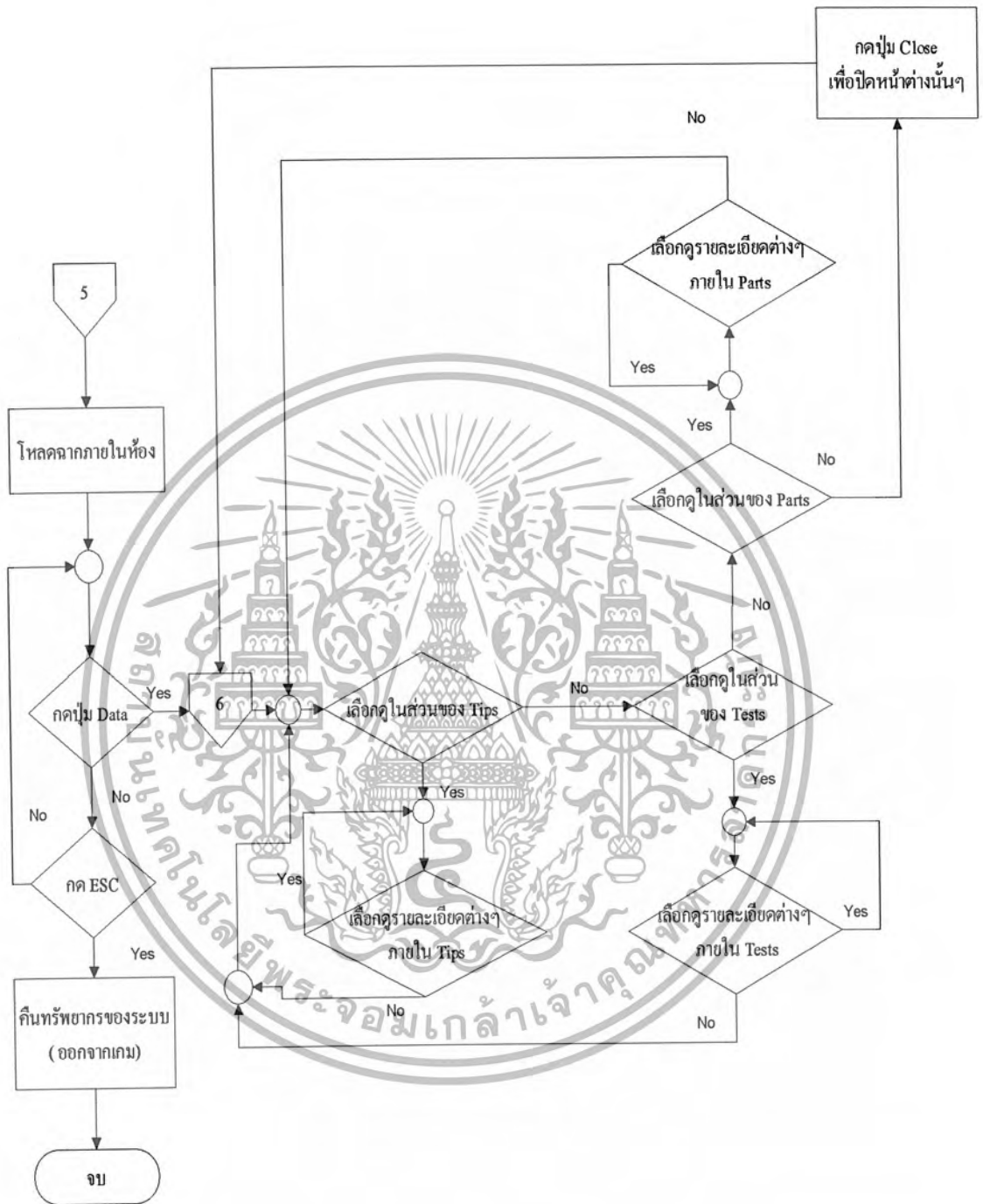
รูปที่ 3.72 รูปแสดงรายละเอียดในส่วน Tips



รูปที่ 3.73 รูปแสดงรายละเอียดในส่วน Parts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

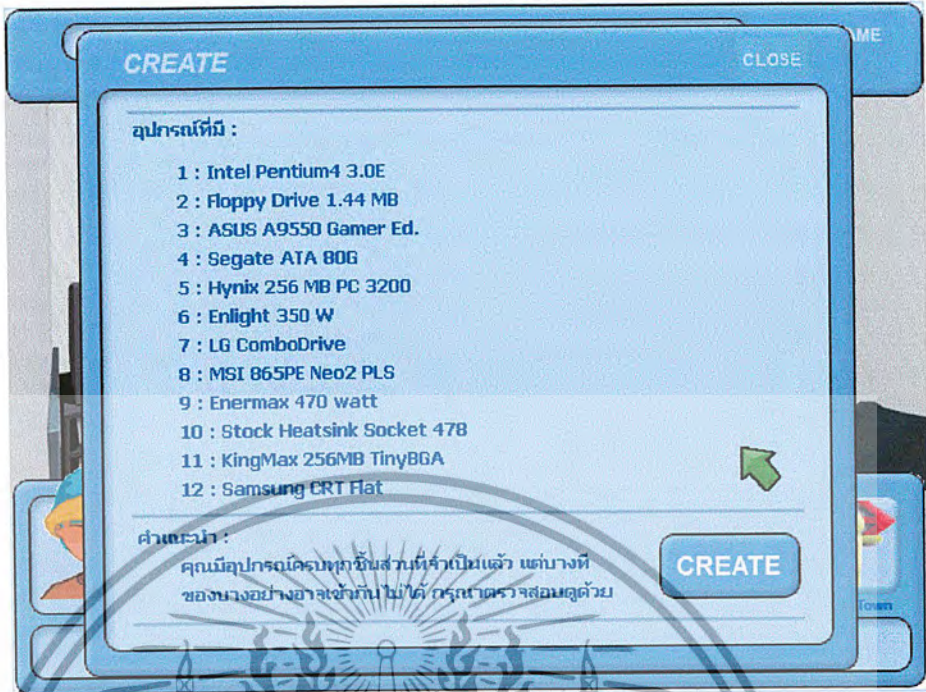
มีแผนภาพการทำงานดังนี้



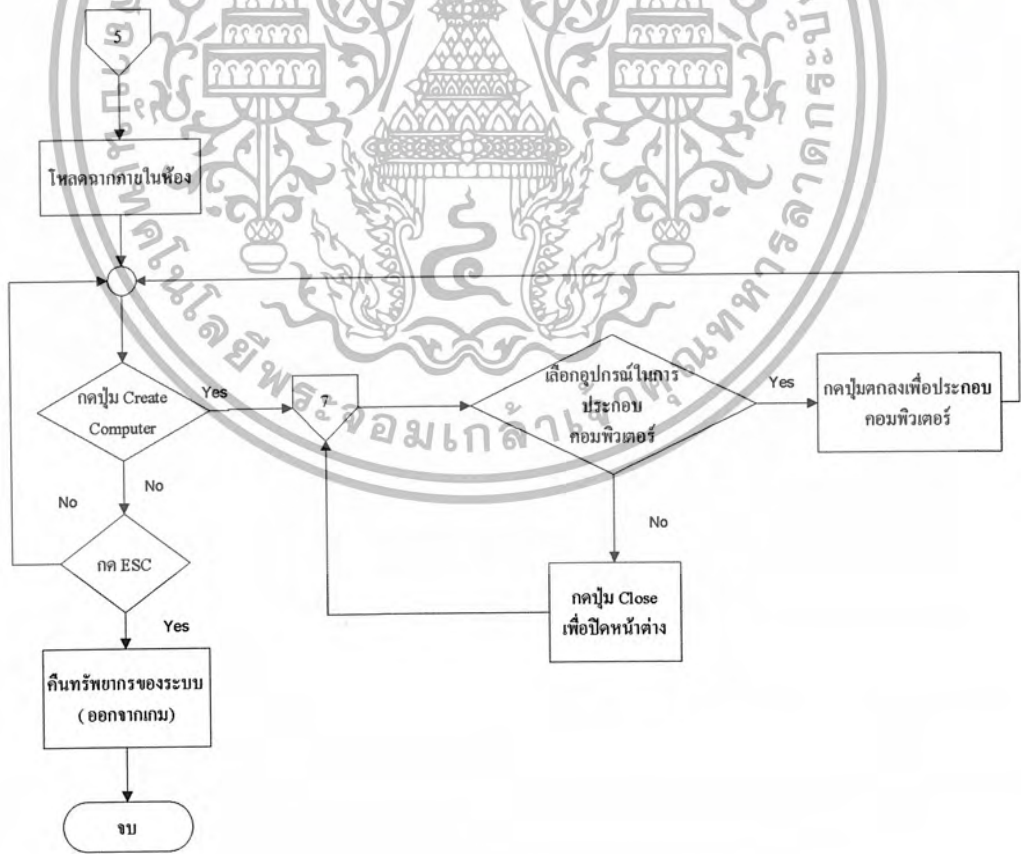
รูปที่ 3.74 แผนภาพการทำงานของปุ่ม Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.2 หน้าจอการ Create Computer



รูปที่ 3.75 รูปหน้าจอการเลือกอุปกรณ์คอมพิวเตอร์ที่จะนำไปประกอบ

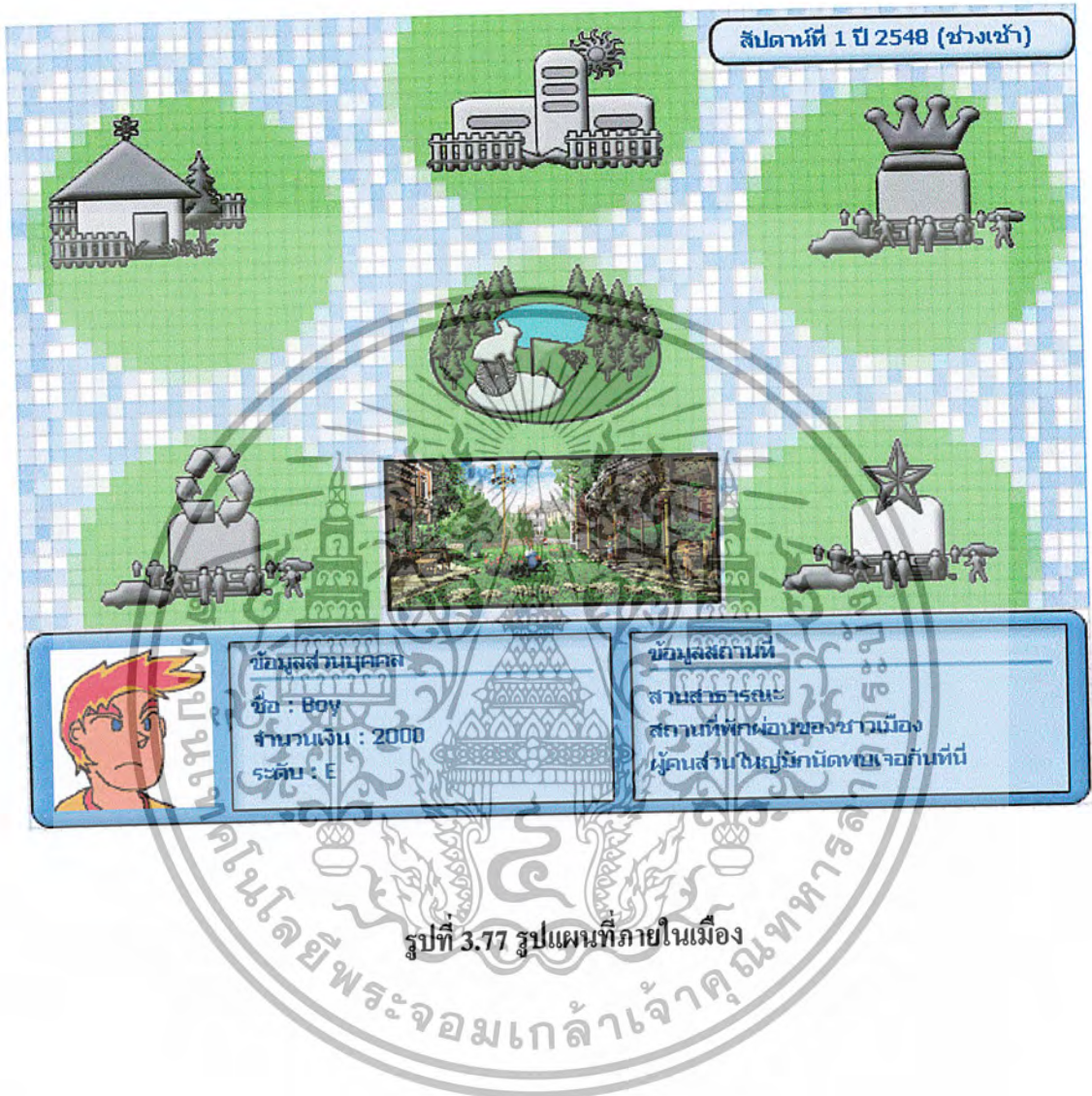


รูปที่ 3.76 แผนภาพการทำงานของปุ่ม Create Computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

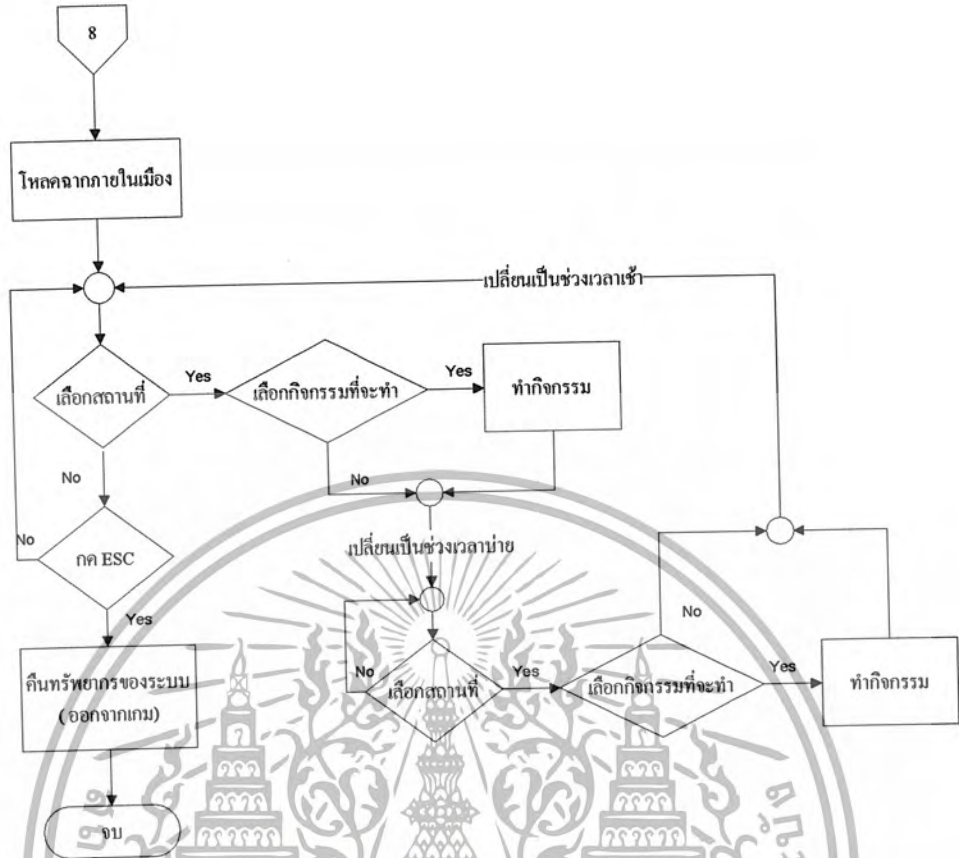
3.3.4 ฉากแผนที่ในเมือง

เป็นสถานที่ที่เราต้องเข้าไปเก็บชิ้นส่วนอุปกรณ์คอมพิวเตอร์ชนิดต่างๆ มาให้ครบถ้วน และต้องรวบรวมความรู้เกี่ยวกับคอมพิวเตอร์มาให้ครบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

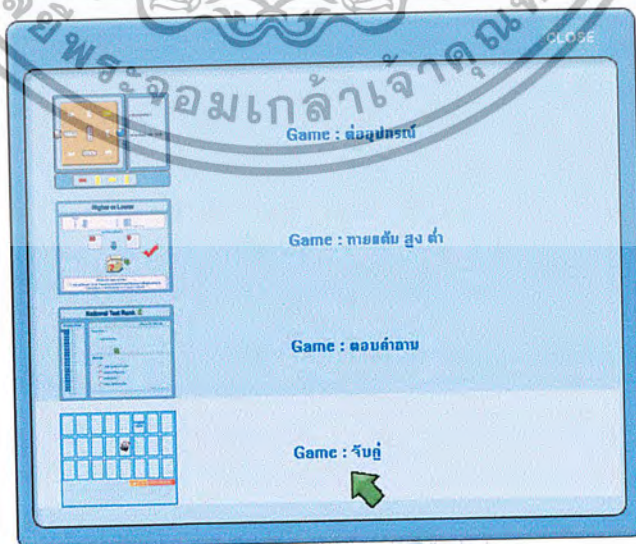
มีแผนภาพการทำงานดังนี้



รูปที่ 3.78 แผนภาพการทำงานของฉากแผนที่

3.3.5 ฉาก Mini Game

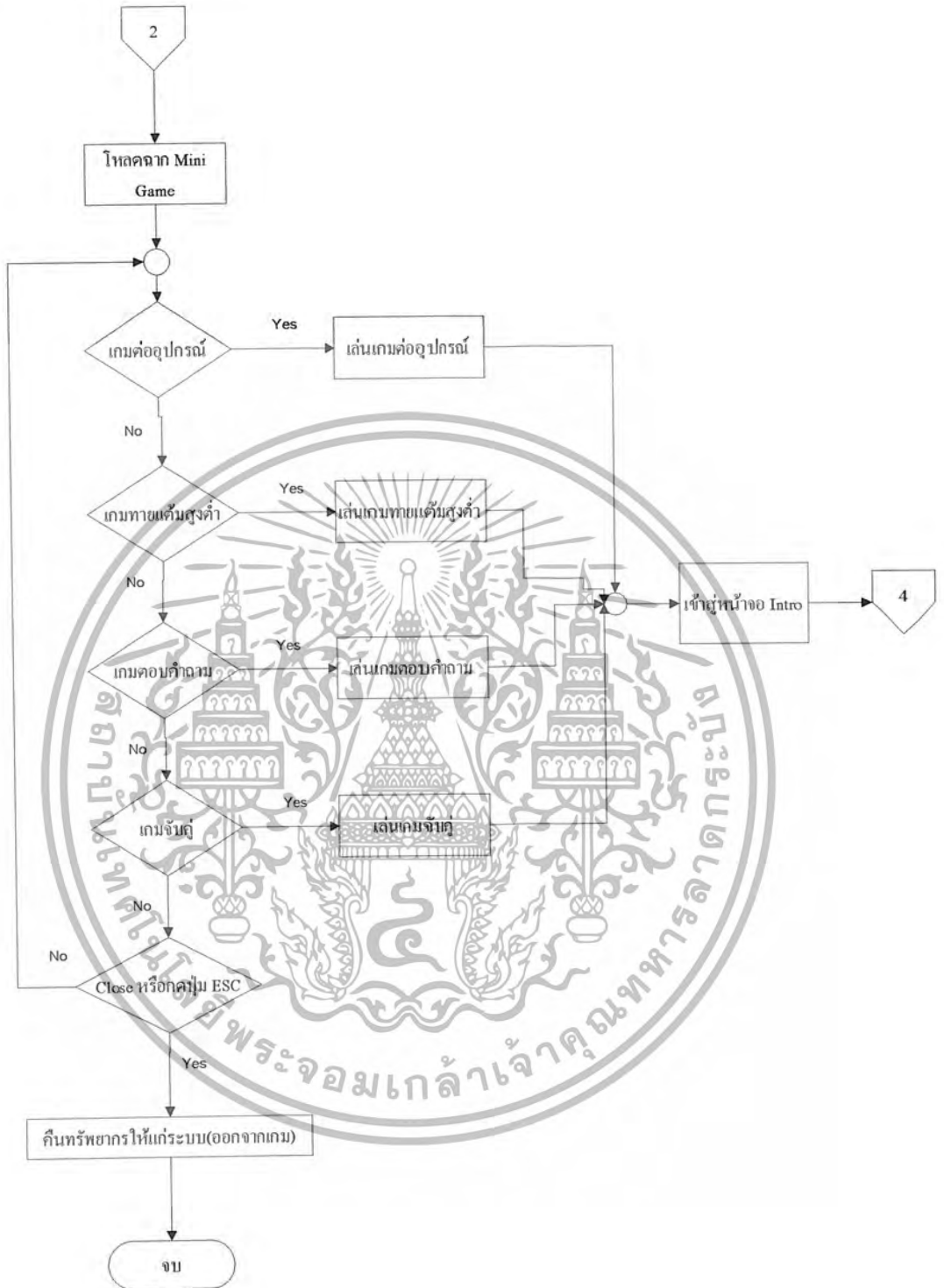
จะมีเกมย่อยทั้งหมด 4 เกมเอาไว้ให้ผู้เล่น



รูปที่ 3.79 รูปแสดงเกมย่อยต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนภาพการทำงานดังนี้

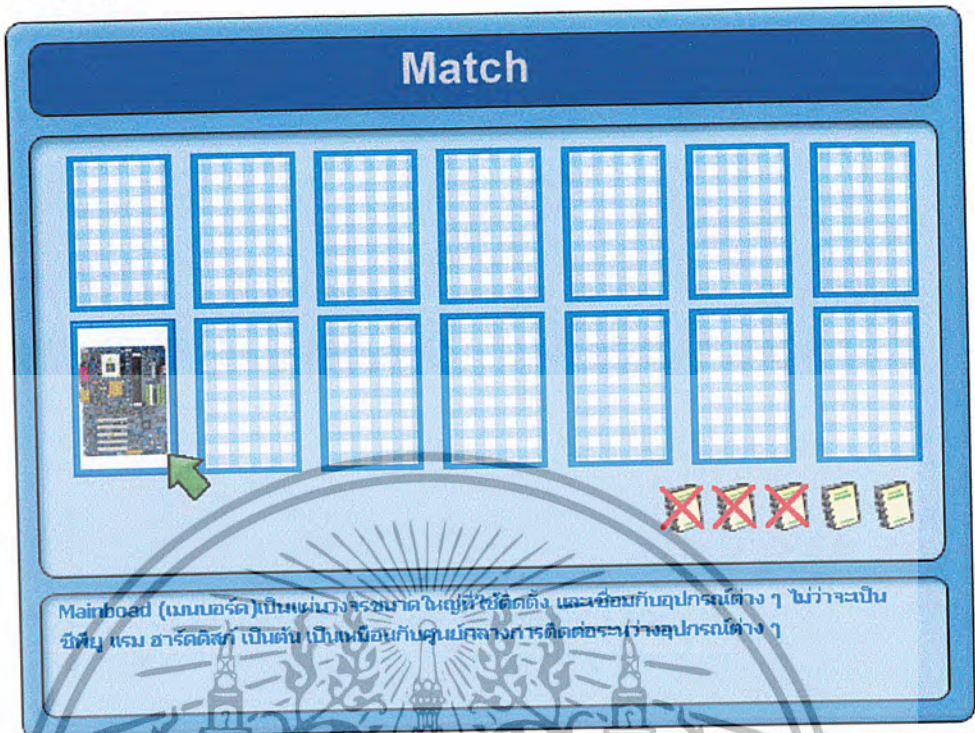


รูปที่ 3.80 แผนภาพการทำงานของฉาก Mini Game

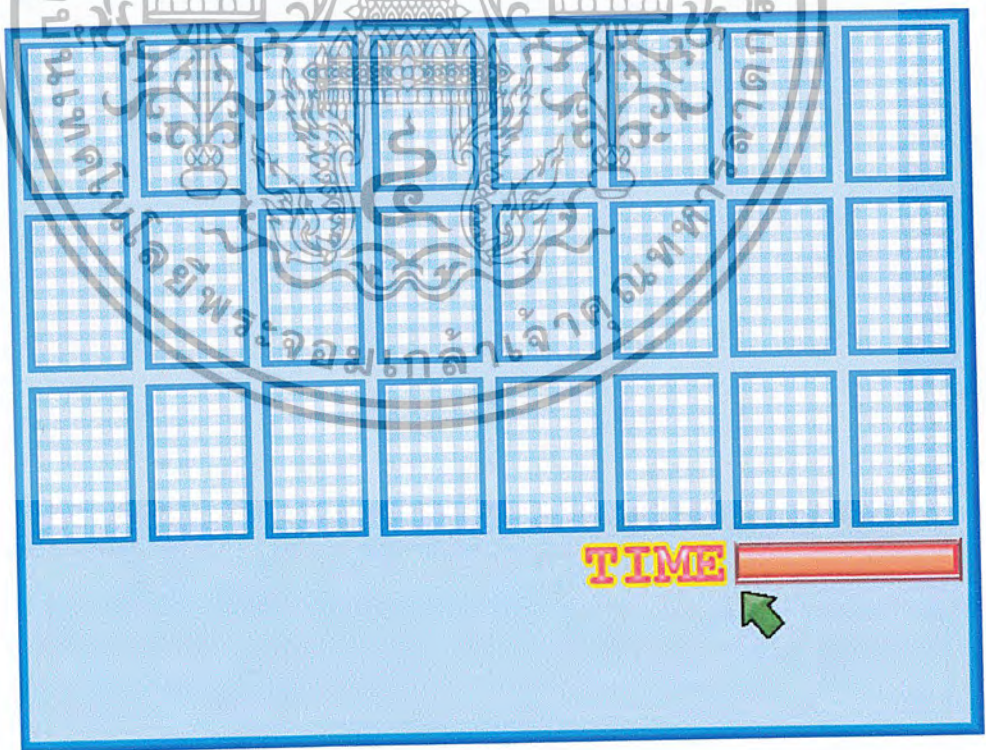
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6 เกมต่างๆ

1 เกมจับคู่



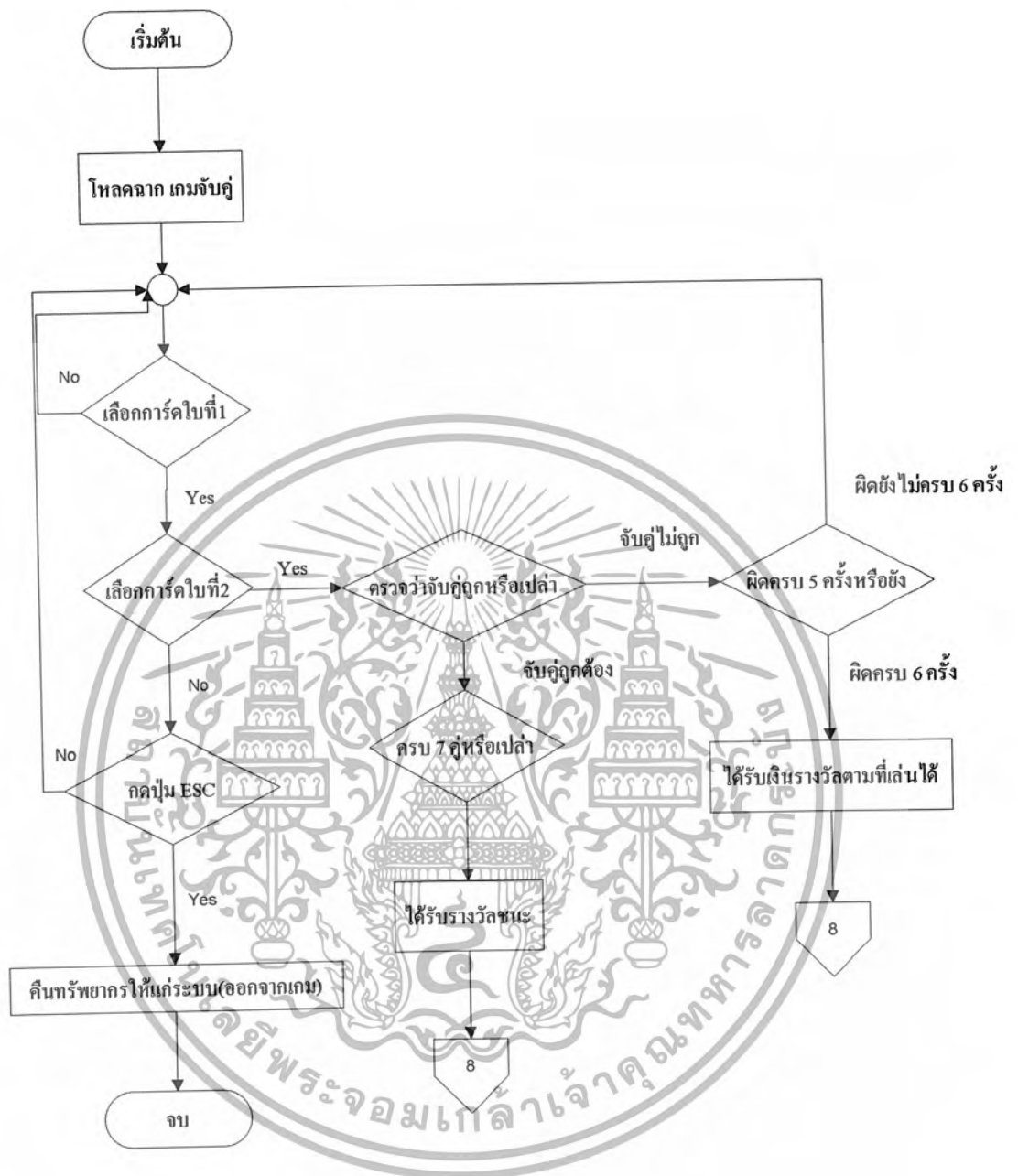
รูปที่ 3.81 รูปของเกมจับคู่ในเกมใหญ่



รูปที่ 3.82 รูปของเกมจับคู่ใน Mini Game

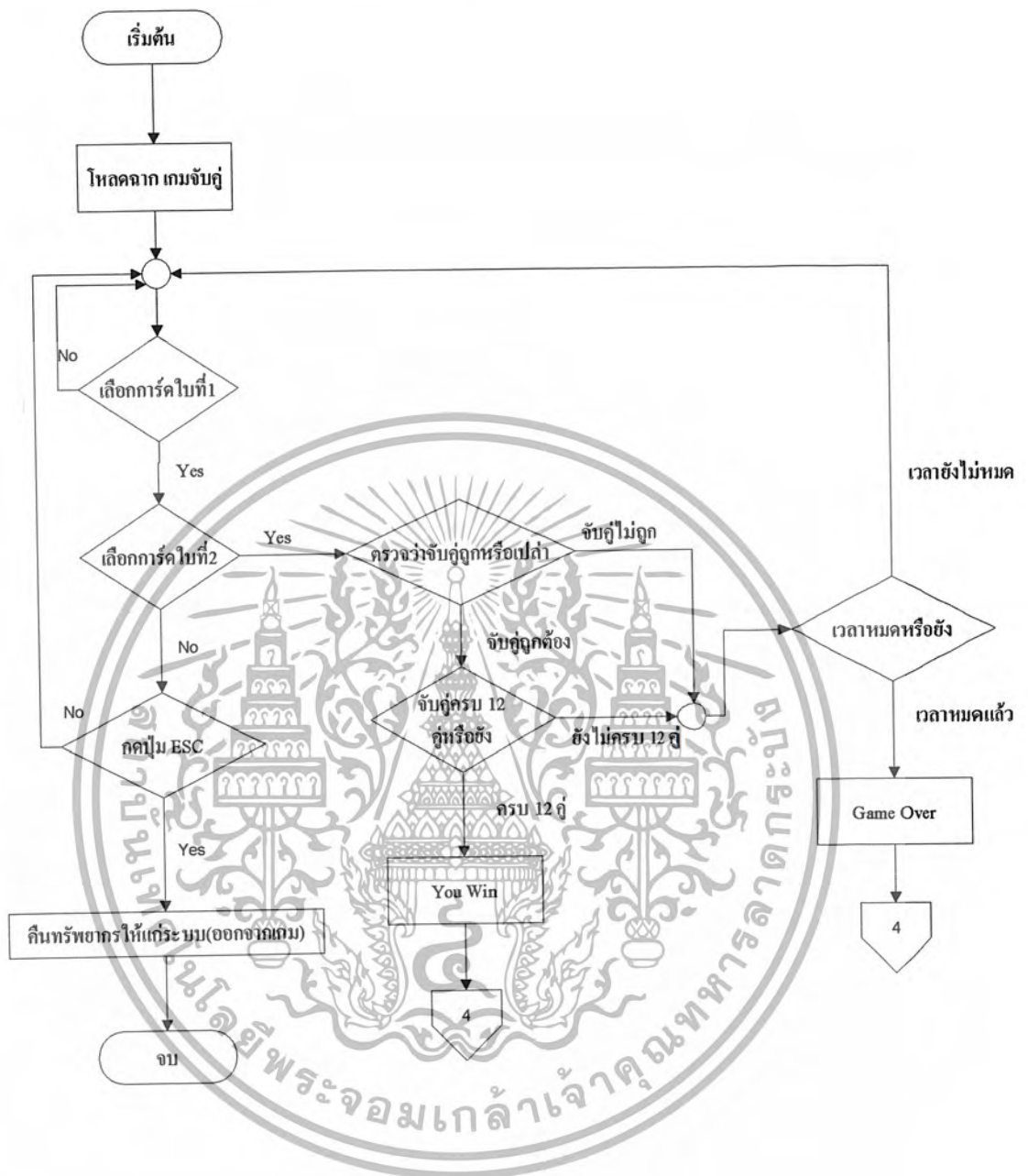
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการทำงานของเกมจับคู่ในเกมใหญ่



รูปที่ 3.83 แผนภาพการทำงานของเกมจับคู่ในเกมใหญ่

แผนภาพการทำงานของเกมจับคู่ใน Mini Game



รูปที่ 3.84 แผนภาพการทำงานของเกมจับคู่ใน Mini Game

2 เกมทายสูงต่ำ

Higher or Lower


Reward	
1.	100 B.
2.	300 B.
3.	600 B.
4.	1,000 B.
5.	1,500 B.
6.	2,100 B.
7.	2,800 B.
8.	3,600 B.
9.	4,500 B.
10.	5,000 B. & Special Item.


ขณะนี้คุณอยู่ที่ข้อที่ 4

7

↓

6





ดีใจด้วยครับ คุณทายถูกต้อง

** ค่าถามมีทั้งหมด 10 ข้อ ถ้าคุณสามารถตอบข้อปัจจุบันในฤกษ์คุณจะมีสิทธิ์เล่นข้อถัดไป ถ้าตอบผิดเกมจะจบทันที และเงินรางวัลจะถูกนำมารั้งหนึ่ง **

รูปที่ 3.85 รูปของเกมทายสูงต่ำในเกมหลัก และ Mini Game

3 เกมตอบคำถาม

National Test Rank E

Answer Sheet

1	2
2	...
3	...
4	...
5	...
6	...
7	...
8	...
9	...
10	...
11	...
12	...
13	...
14	...
15	...
16	...
17	...
18	...
19	...
20	...

Finish

เหลือเวลาอีก 596 นาที

Question 1

ข้อใด ไม่ใช่อินเตอร์เฟซการเชื่อมต่อของฮาร์ดดิสก์(HDD)

Answer

ATA

SATA

SCSI

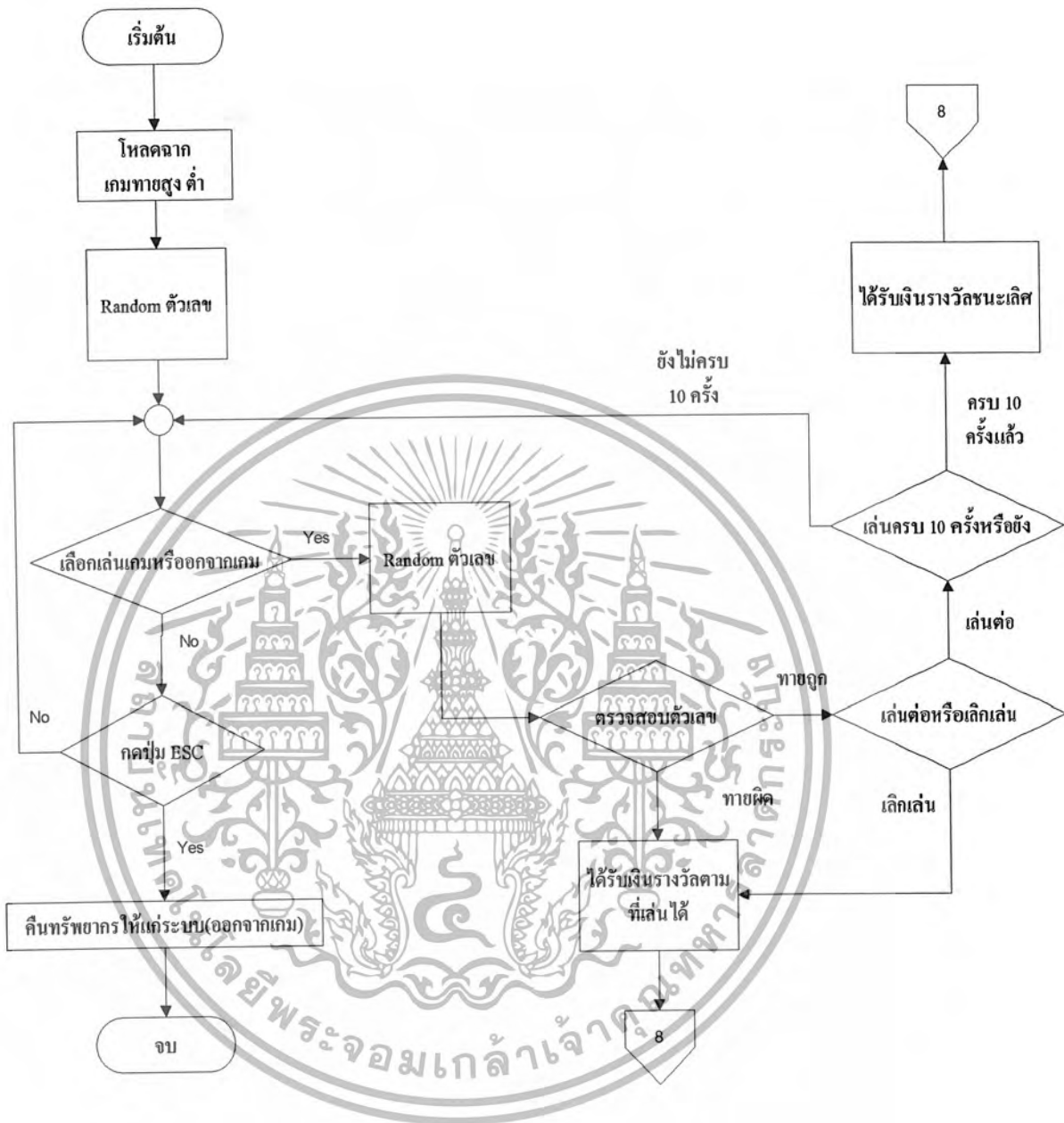
IDE

< Back Next >

รูปที่ 3.86 รูปของเกมตอบคำถามในเกมหลัก และ Mini Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

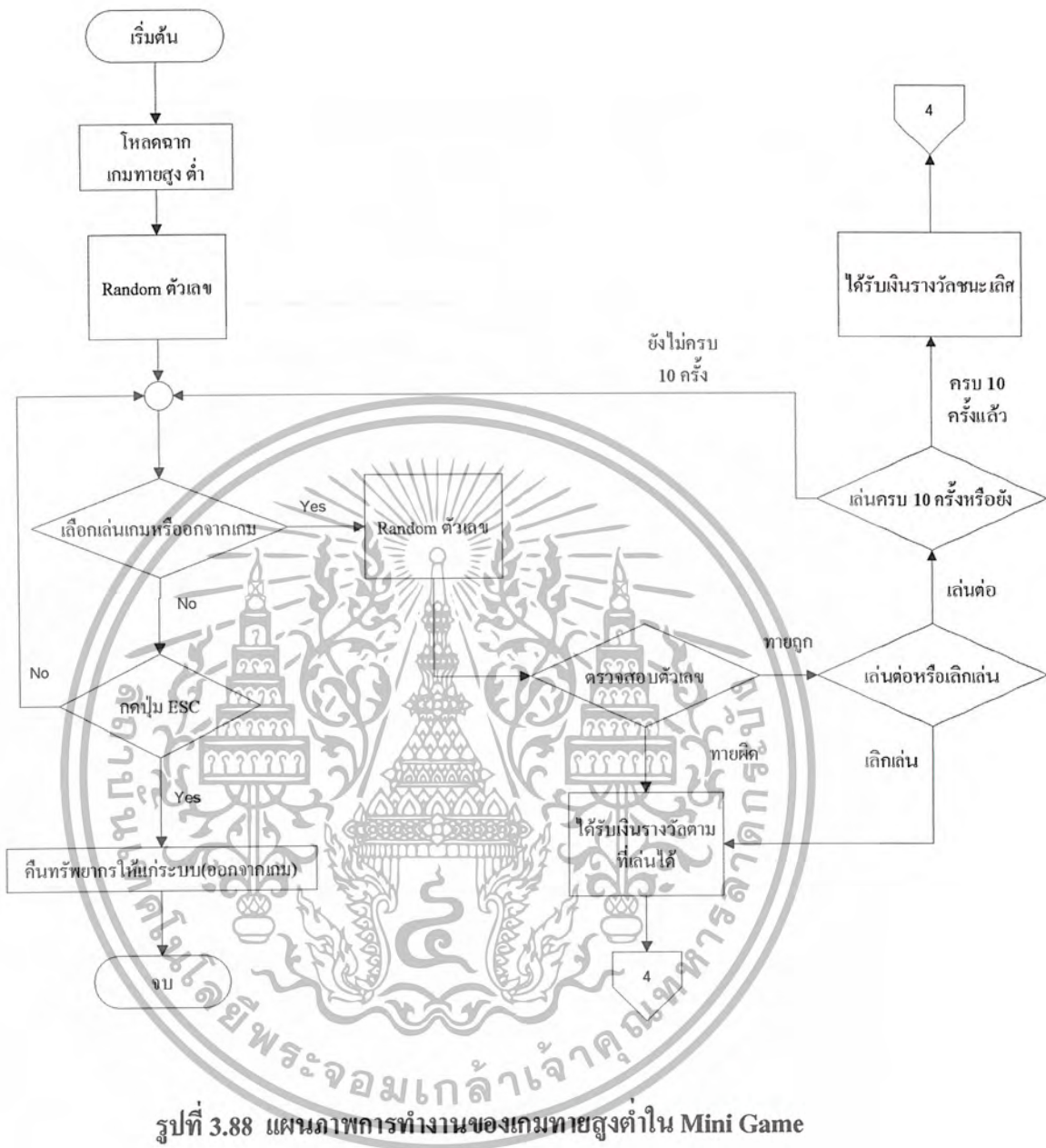
แผนภาพการทำงานของเกมทายสูงต่ำในเกมใหญ่



รูปที่ 3.87 แผนภาพการทำงานของเกมทายสูงต่ำในเกมใหญ่

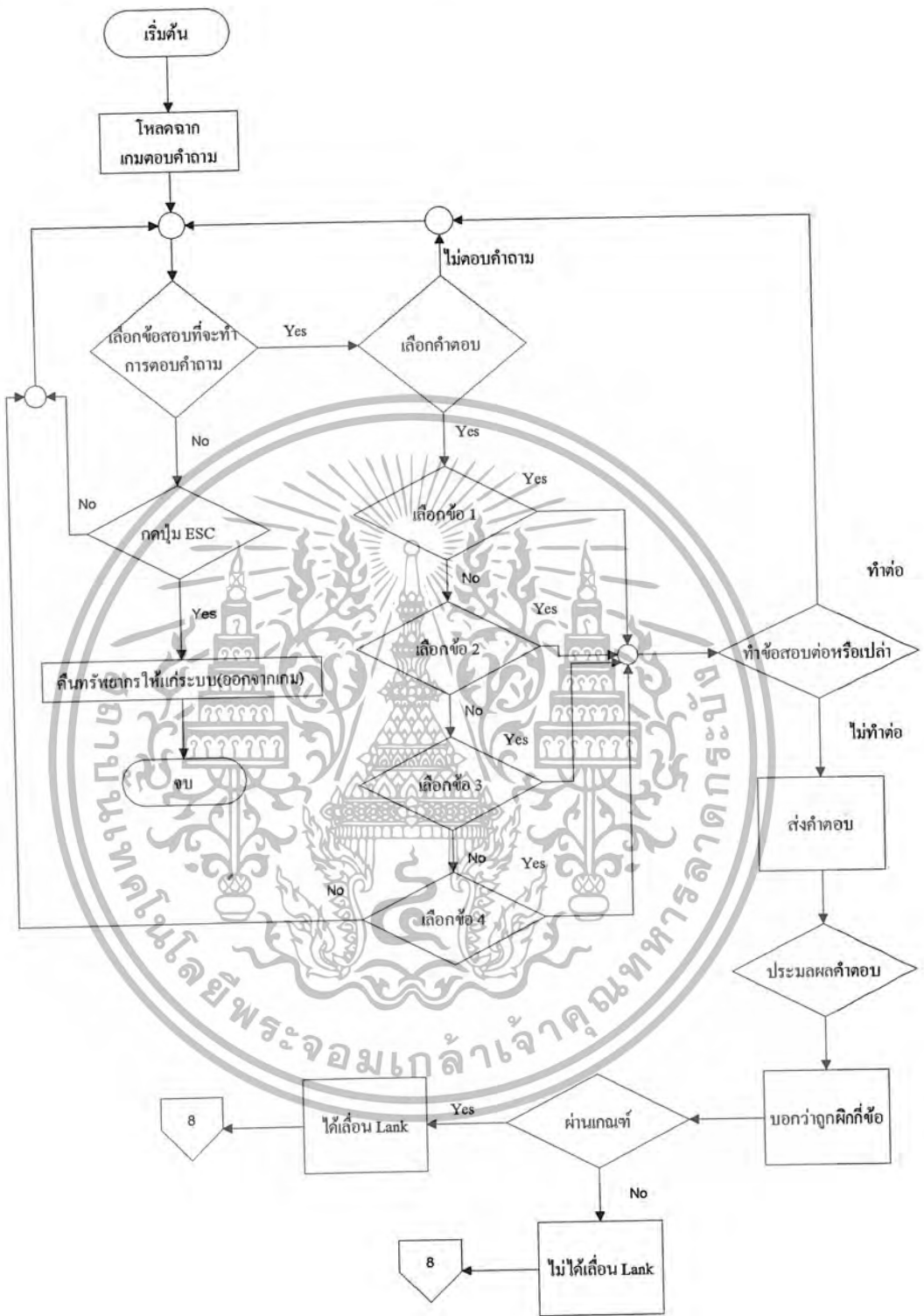
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการทำงานของเกมทายสูงต่ำใน Mini Game



รูปที่ 3.88 แผนภาพการทำงานของเกมทายสูงต่ำใน Mini Game

แผนภาพการทำงานของเกมตอบคำถามในเกมใหญ่



รูปที่ 3.89 แผนภาพการทำงานของเกมตอบคำถามในเกมใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการทำงานของเกมตอบคำถามใน Mini Game



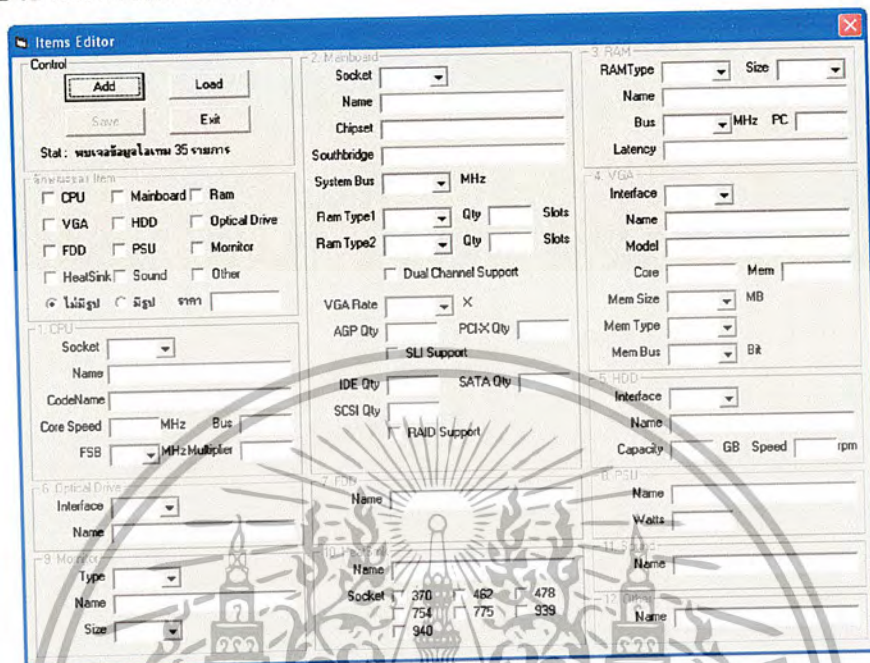
รูปที่ 3.90 แผนภาพการทำงานของเกมตอบคำถาม Mini Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7 โปรแกรมช่วยสร้างข้อมูลต่างๆ

ซึ่งโปรแกรมเหล่านี้จะเป็นโปรแกรมที่ช่วยในการแก้ไขข้อมูลต่างๆ ภายในเกมทำให้ผู้พัฒนานั้นสามารถเพิ่มเติมข้อใหม่เข้าไปในเกมได้เรื่อยๆ

1 โปรแกรม Items Editor



รูปที่ 3.91 หน้าจอโปรแกรม Items Editor

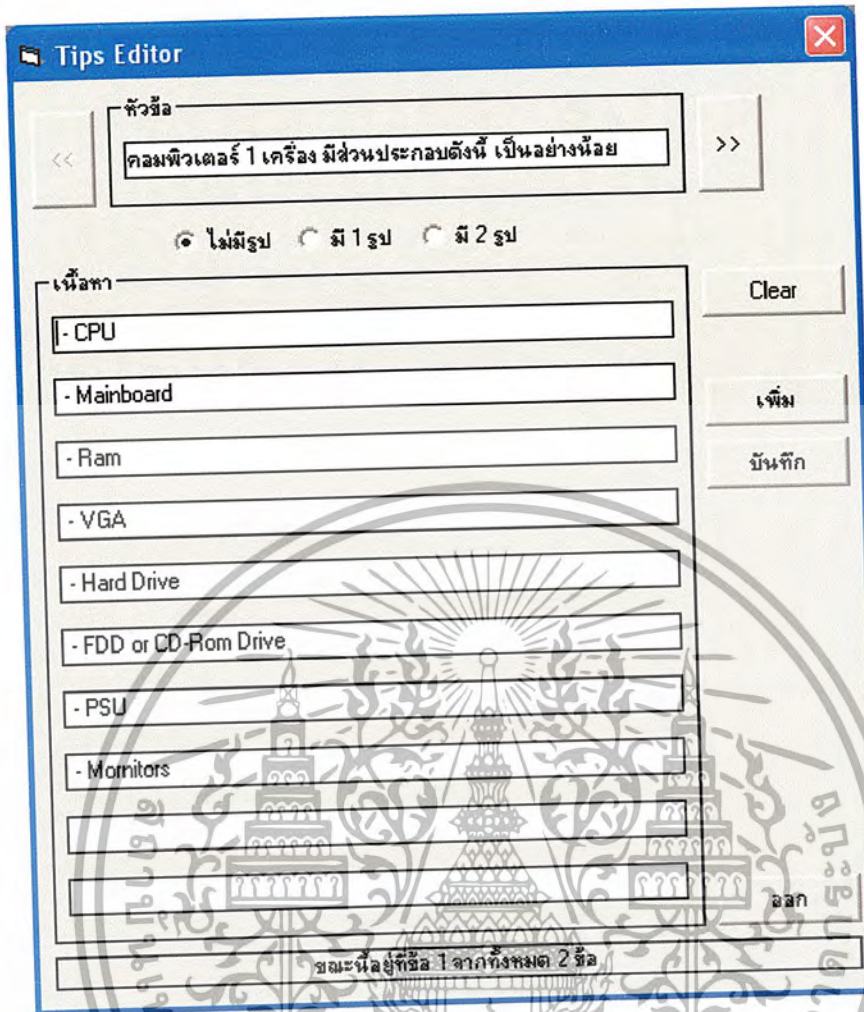
2 โปรแกรม Question Editor



รูปที่ 3.92 หน้าจอโปรแกรม Question Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 โปรแกรม Tips Editor



รูปที่ 3.93 หน้าจอโปรแกรม Tips Editor

3.4 ขั้นตอนการทดสอบเกม

เป็นขั้นตอนสุดท้ายของการทำเกม โดยขั้นตอนนี้เป็นการตรวจสอบความผิดพลาดและประสิทธิภาพโดยรวมของเกมทั้งหมด ว่าสามารถได้ตอบและแสดงผลได้ตามที่ออกแบบไว้หรือไม่ ถ้าตรวจสอบพบความผิดพลาดจะได้นำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขให้มีความถูกต้องมากที่สุด

บทที่ 4

ขั้นตอนการติดตั้งและการใช้งานโปรแกรม

การจะทดสอบขั้นตอนวิธีในเกมที่ได้ออกแบบไว้นั้น ทำได้โดยการสร้างโปรแกรมเกมตามขั้นตอนวิธีที่ได้กล่าวมาแล้วจากบทที่ 3 โดยการเล่นเกม เพื่อตรวจสอบประสิทธิภาพของเกมว่าสามารถเล่นได้ดีเพียงใด

4.1 เครื่องมือที่ใช้ในการทดลอง

คอมพิวเตอร์ที่ใช้ในการทดลองในปัญหาพิเศษนี้ มีลักษณะความสามารถของเครื่อง เป็นดังนี้

1. เครื่องมือที่ใช้ในการพัฒนา

- Microsoft Visual C++ 6.0
- Microsoft DirectX SDK 8.0

2. ลักษณะขั้นต่ำที่เครื่องต้องมี

Input : เมาส์ และ คีย์บอร์ด

Output : หน้าจอขนาด 640 x 480 pixel

OS : Microsoft Windows XP

ความต้องการอื่นๆ : Microsoft DirectX 8.0 Runtime ขึ้นไป

CPU 700 MHZ ขึ้นไป

RAM 128 MB ขึ้นไป

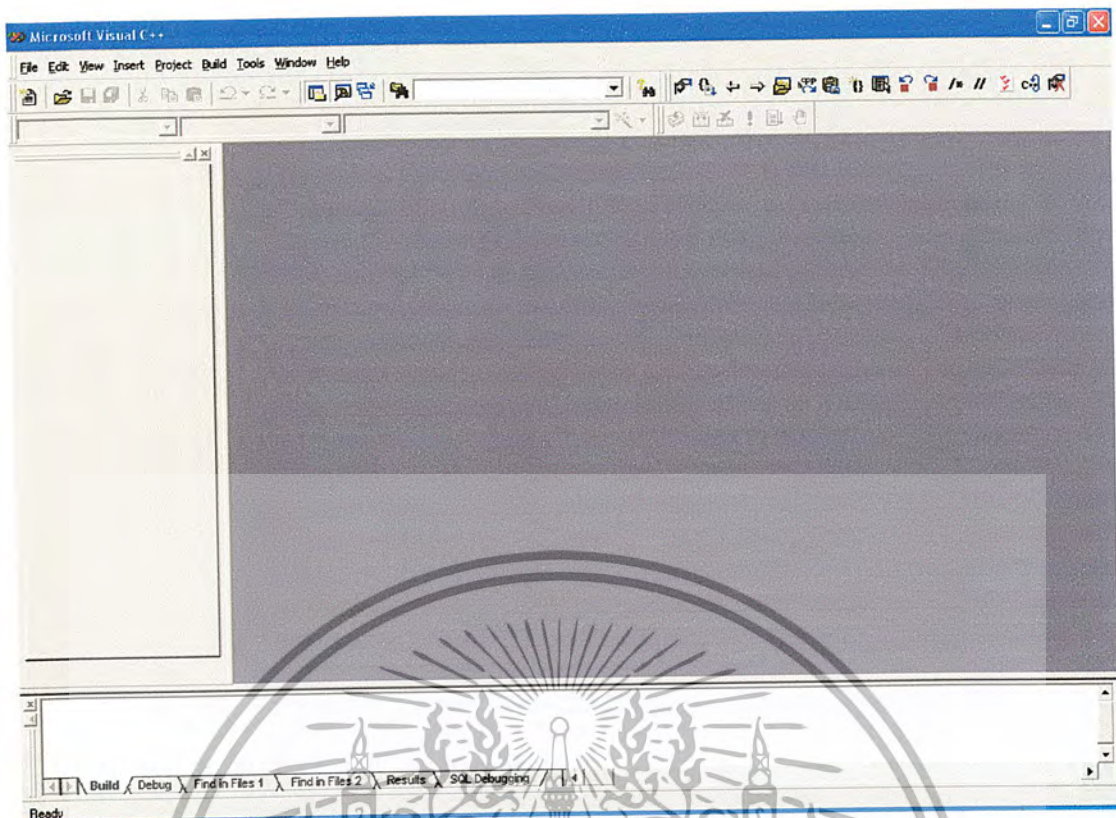
4.1.1 ขั้นตอนการใช้งาน

4.1.1.1 การลงโปรแกรม

การสร้างโปรแกรมเพื่อการทดสอบเกมนั้นจะอ้างอิงตามลักษณะการใช้งานของผู้ใช้ทุกๆ ไปที่ใช้งานกันบนระบบปฏิบัติการ Windows เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่ายและใช้งานได้สะดวกยิ่งขึ้น โดยมีลักษณะการทดสอบดังนี้

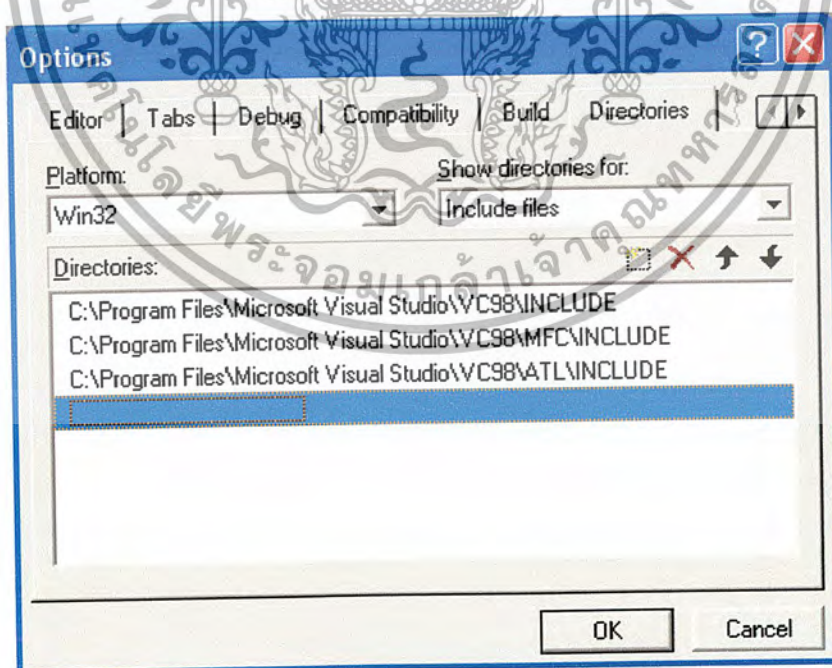
- ให้ทำการลงโปรแกรม Microsoft Visual C++ 6.0 ก่อน
- เมื่อลงโปรแกรม Microsoft Visual C++ 6.0 เสร็จแล้วให้ทำการลง DirectX SDK 8.0
- เมื่อลงโปรแกรมทั้ง 2 เสร็จแล้วเปิด Microsoft Visual C++ 6.0 ขึ้นมาเพื่อทำการปรับแต่งโปรแกรม Microsoft Visual C++ 6.0 ให้รู้จักกับ DirectX SDK ก็เพื่อที่จะให้ Microsoft Visual C++ 6.0 เรียกชุดคำสั่ง DirectX ได้นั่นเอง วิธีการปรับแต่งก็คือ ให้เปิดโปรแกรม Microsoft Visual C++ 6.0 ขึ้นมาก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 รูปหน้าจอของ Microsoft Visual C++

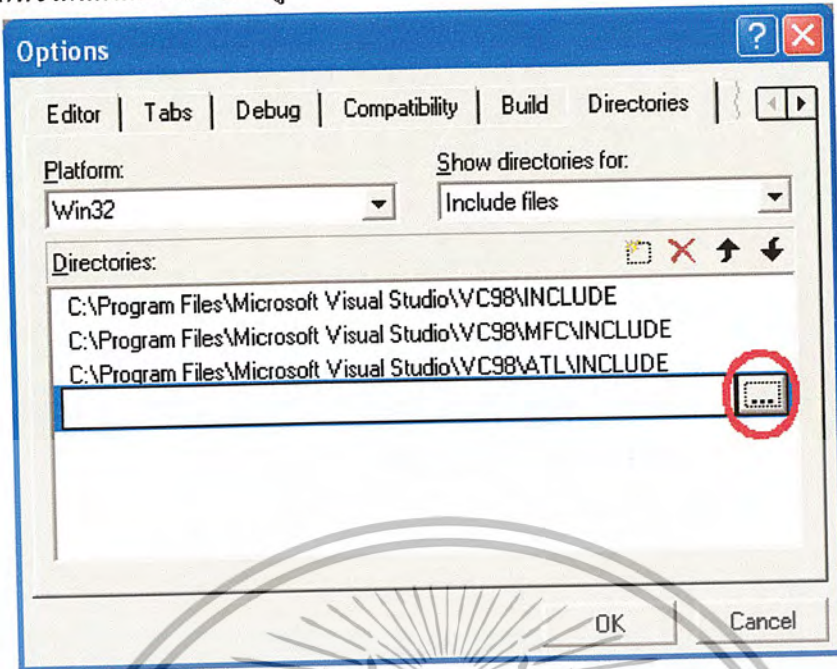
จากนั้นเลือกเมนู Tool > Option > และคลิกไปที่แท็บ Directories ดังรูป



รูปที่ 4.2 รูปหน้าจอของการปรับแต่ง Visual C++

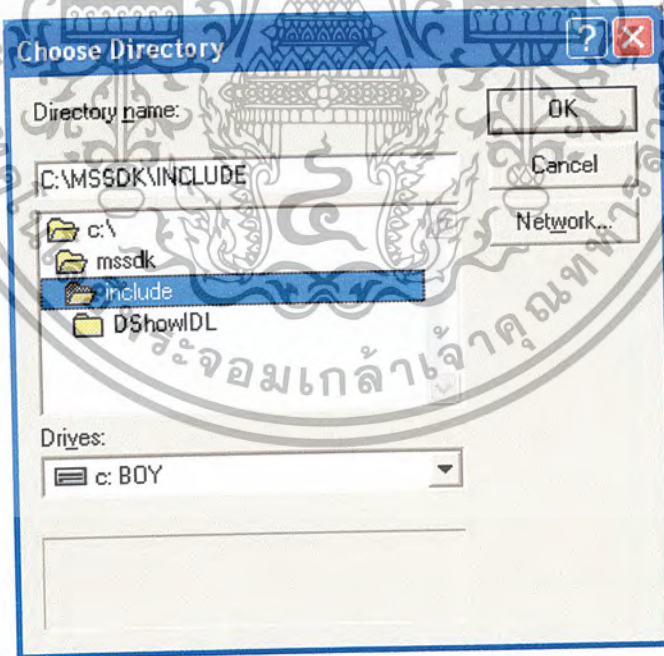
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิกที่วงกลมสีแดงดังรูป



รูปที่ 4.3 รูปหน้าจอของการปรับแต่ง Visual C++ (Include Files)

จากนั้นให้เลือก Directory ไปที่ C:\mssdk\include ดังรูป

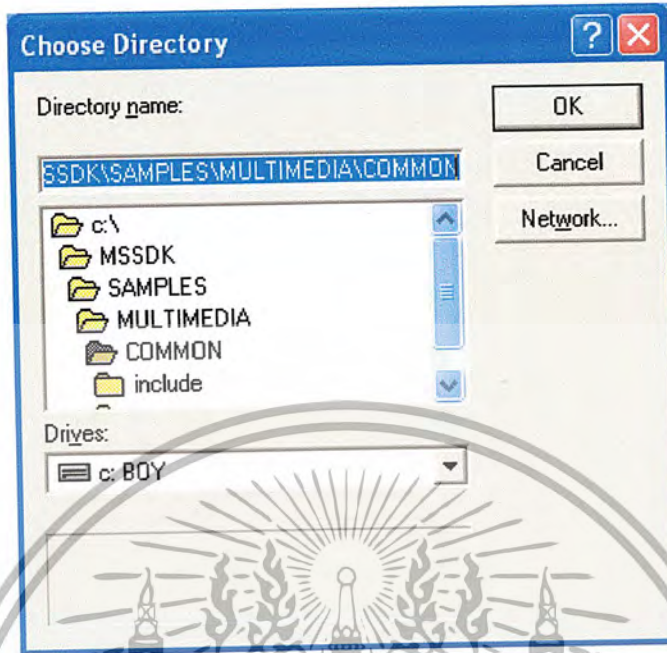


รูปที่ 4.4 รูปหน้าจอของการปรับแต่ง Visual C++ (Choose Directory)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

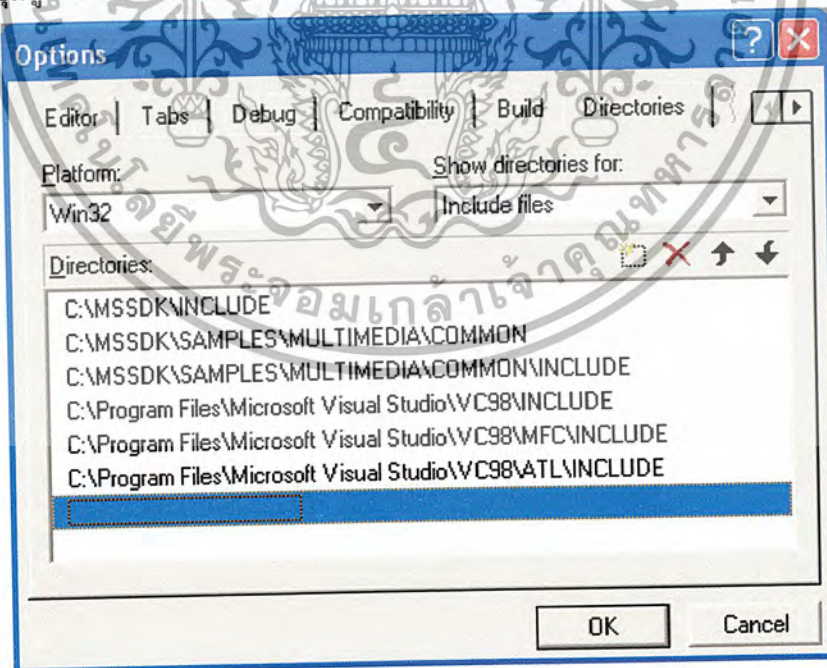
ทำเช่นนั้นอีกครั้ง โดยเลือกไปที่ Directory

C:\MSSDK\SAMPLES\MULTIMEDIA\COMMON ดังรูป



รูปที่ 4.5 รูปหน้าจอของการปรับแต่ง Visual C++ (Choose Directory) ในส่วนของ Samples

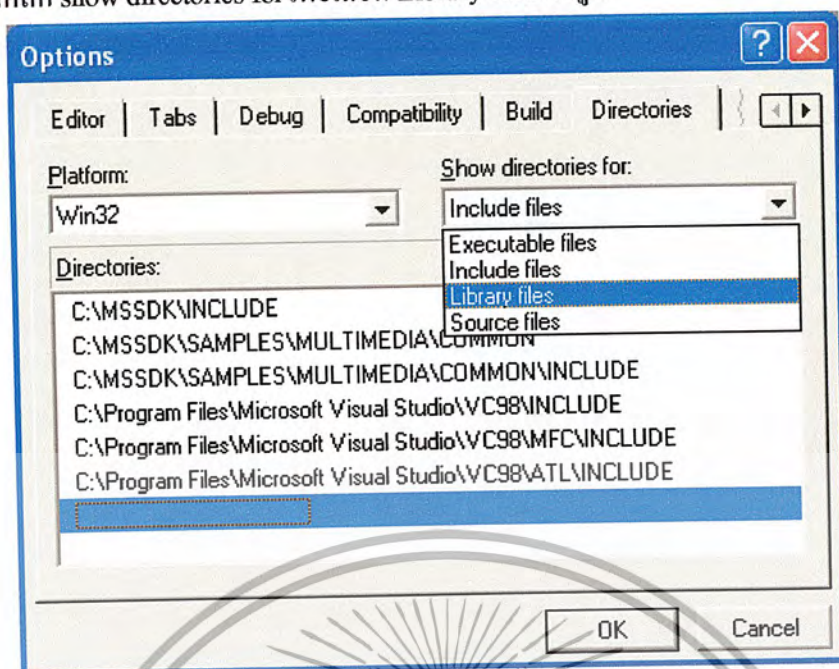
จากนั้นปุ่มถูกตรึงขึ้น เพื่อเลื่อนให้ทั้ง 2 บรรทัดนี้มาอยู่บนสุด ดังรูป



รูปที่ 4.6 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Option ที่มีการจัดแล้ว

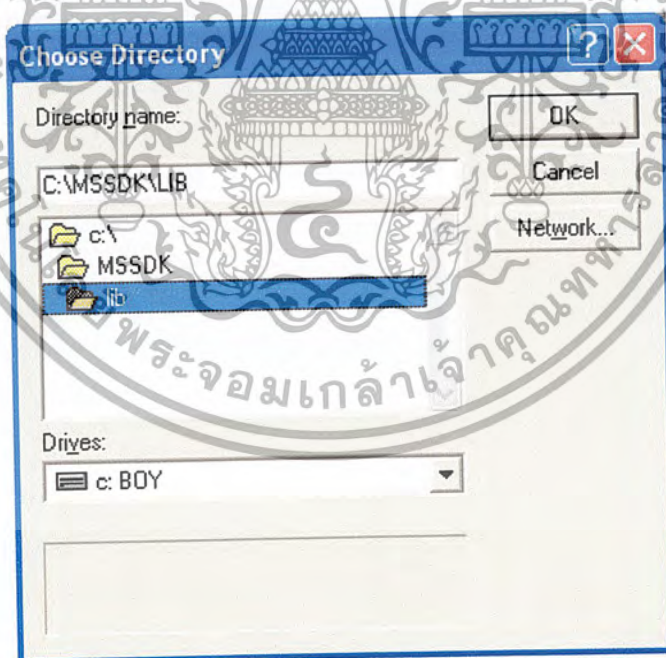
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นคลิก show directories for เพื่อเลือก Library files ดังรูป



รูปที่ 4.7 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library files

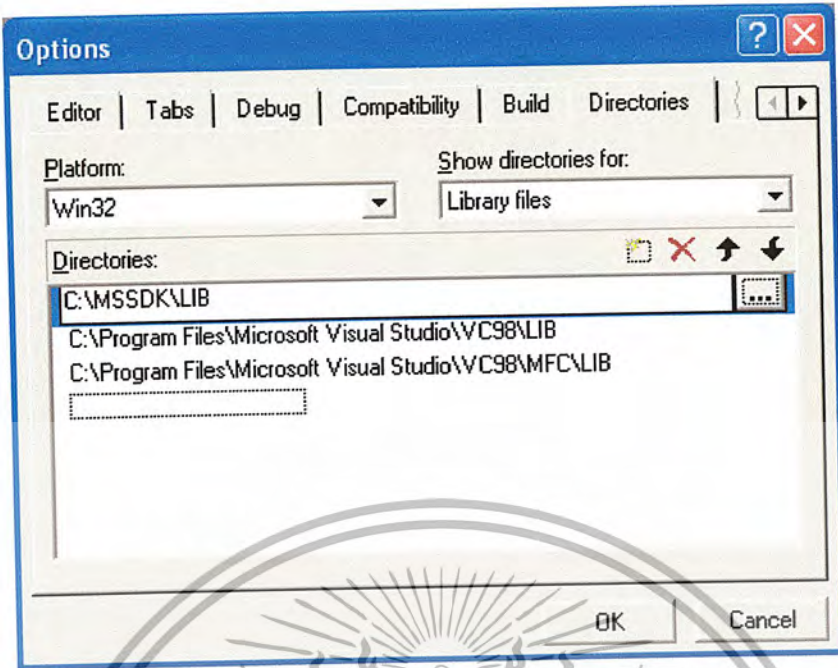
จากนั้นเลือก Directory ไปที่ C:\mssdk\Lib ดังรูป



รูปที่ 4.8 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library files (Choose Directory)

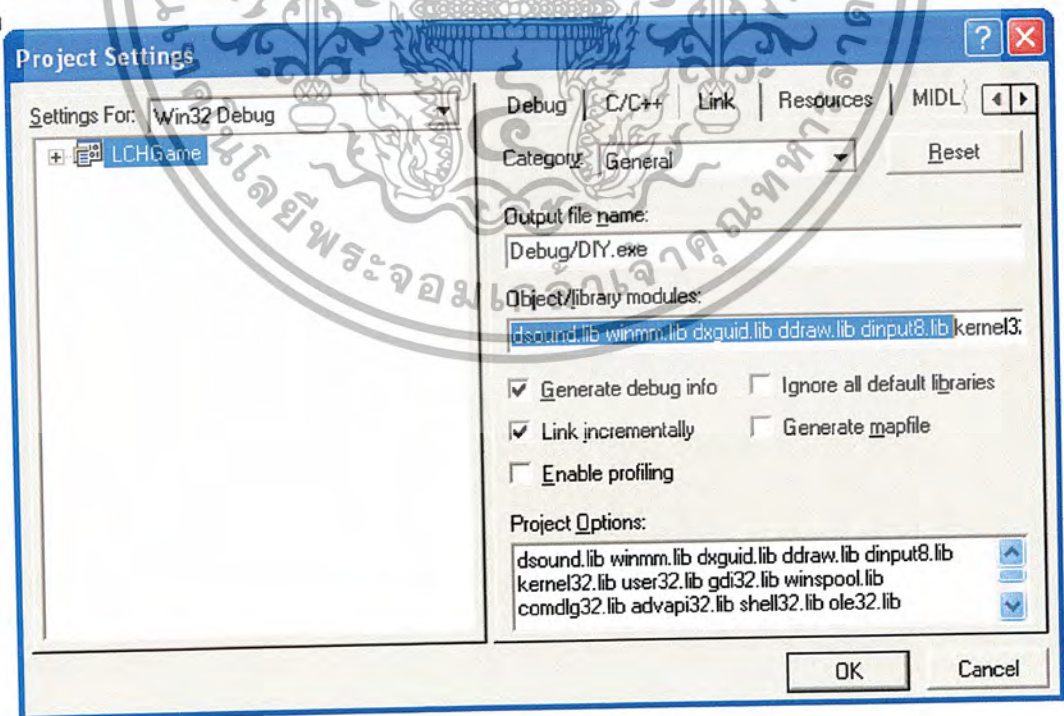
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นปุ่มถูกสร้างขึ้น เพื่อเลื่อนให้มาอยู่ บรรทัดบนสุด ดังรูป



รูปที่ 4.9 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Library ที่จัดเรียงเรียบร้อยแล้ว

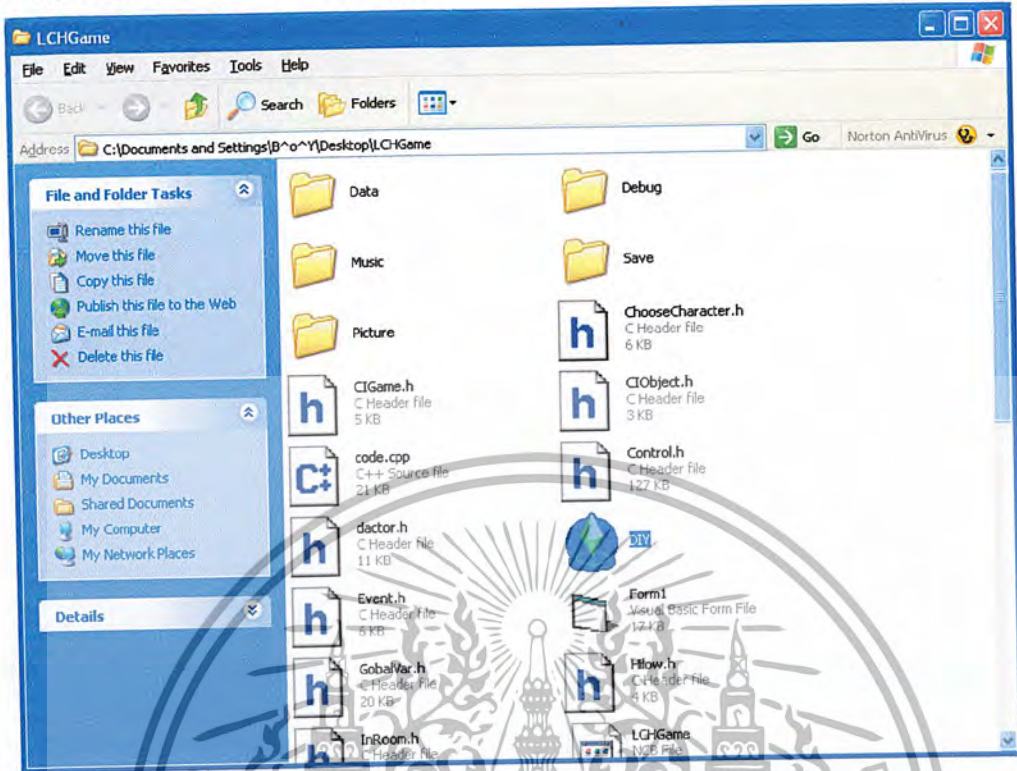
จากนั้นไปที่ project > settings แล้วเลือก Link ใต้ dsound.lib winmm.lib dxguid.lib ddraw.lib dinput8.lib ลงไปในช่องเท่านั้นเราก็สามารถ Run visual C++ ที่ติดตั้ง DirectX เรียบร้อยแล้ว



รูปที่ 4.10 รูปหน้าจอของการปรับแต่ง Visual C++ ในส่วนของ Settings

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากติดตั้งโปรแกรมเรียบร้อยแล้วก็ลองรันโปรแกรมดูก็สามารถเข้าเล่นเกมได้ หรือผู้เล่นสามารถคลิกที่ตัว DIY.EXE ได้ตามดังรูป



รูปที่ 4.11 รูป DIY.EXE

4.1.1.2 คู่มือการเล่นเกม

เป้าหมายของเกมนี้ก็คือต้องการให้คนที่เล่นเกมนั้นมีความรู้เรื่องเกี่ยวกับคอมพิวเตอร์มากขึ้น โดยภายในตัวเกมนั้นจะสอดแทรกความรู้ต่างๆ ให้กับผู้เล่น ซึ่งสิ่งทีผู้เล่นต้องทำนั้นเพื่อให้ภารกิจสำเร็จได้นั้นผู้เล่นจะต้องทำการรวบรวม 3 สิ่งต่อไปนี้ให้ครบ

1. โดยผู้เล่นนั้นจะต้องทำการรวบรวมข้อมูลต่างๆ ที่บอกว่าอุปกรณ์คอมพิวเตอร์แต่ละชิ้นนั้นมีหน้าที่อย่างไร และมีการทำงานอย่างไรจากบุคคลต่างๆ ที่อยู่ภายในเกม ซึ่งผู้เล่นนั้นจะต้องทำการเข้าไปคุยกับบุคคลนั้นๆ หลายๆ ครั้งถึงจะได้ข้อมูลนั้นออกมา
2. ผู้เล่นจะต้องทำการรวบรวมชิ้นส่วนของอุปกรณ์คอมพิวเตอร์ชนิดต่างๆ ให้ครบ เพื่อที่จะนำมาประกอบคอมพิวเตอร์ได้ ซึ่งการหาชิ้นส่วนของอุปกรณ์คอมพิวเตอร์นั้นก็ทำได้ 2 แบบ คือ แบบที่ 1 ผู้เล่นจะต้องเข้าไปเล่นเกมเพื่อชิงของรางวัล แบบที่ 2 ผู้เล่นสามารถหาซื้อชิ้นส่วนของอุปกรณ์คอมพิวเตอร์ได้จากการซื้อ
3. ผู้เล่นต้องทำการเลื่อนระดับขั้นของตัวเองให้ได้ซึ่งการเลื่อนระดับขั้นนั้นผู้เล่นสามารถทำได้โดยการเข้าไปสอบที่โรงเรียนจ๊กให้มีสอบ ซึ่งการสอบนั้นจะจัดทุกๆ 4 สัปดาห์

ระบบของเกมนั้นจะดำเนินเป็นแบบ สัปดาห์ ซึ่งสัปดาห์หนึ่งจะมี 2 ช่วงเวลาคือช่วงเวลา เช้า และช่วงเวลาที่บ่าย เมื่อผู้เล่นทำกิจกรรมอะไรไปแล้ว 1 อย่างช่วงเวลาจะทำการเปลี่ยน ไปโดยทันที และการเล่นเกมเพื่อชิงรางวัลนั้นจะจัดทุกๆ สัปดาห์และจัดทั้งสองช่วงเวลา โดยเกมที่จัดขึ้นนั้นจะ คຸ່ມมาจากเกมทั้งหมดโดยจะเลือกมาเพียงเกมเดียว และจะจัดเพียงสถานที่เดียวเท่านั้นซึ่งผู้เล่นนั้น สามารถดูได้จากเมนูหัวข้อ New ได้ ว่าเกมนั้นจะจัดขึ้นที่ไหน ซึ่งสถานที่ที่มีเกมจัดนั้นจะมี 3 สถานที่คือ ย่านร้านค้าที่ 1, 2, และ 3 ซึ่งมีการเล่นเกมนั้นจะอธิบายตามขั้นตอนดังนี้



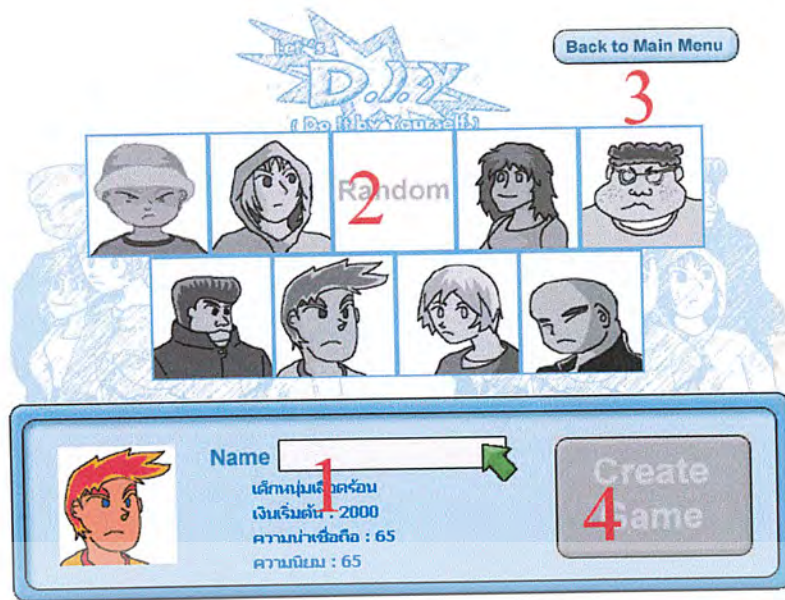
รูปที่ 4.12 แสดงหน้าจอเริ่มต้นของเกมของเกม

โดยแต่ละส่วนจะมีรายละเอียดดังต่อไปนี้

1. New Game เป็นการเริ่มเล่นเกมใหม่ โดยเมื่อเราเข้าสู่หน้าจอนี้แล้วเราจะพบกับหน้าจอที่ให้ผู้เล่นทำการตั้งชื่อผู้เล่น การตั้งชื่อผู้เล่นนั้นเราจะทำการรับคำ จาก Key Board
2. Load Game เป็นการเริ่มเล่นเกมต่อจากครั้งที่แล้ว
3. Mini Game จะมีเกมย่อยให้เลือกเล่น
4. Exit Game เป็นการออกจากเกม

หมายเหตุ - ถ้าผู้เล่นต้องการออกจากเกมนั้นผู้เล่นสามารถกดปุ่ม ESC ได้ตลอดเวลา

เมื่อผู้เล่นเข้าสู่หน้าจอ New Game แล้วจะมีตัวละครให้เราเลือก ซึ่งตัวละครแต่ละตัวนั้นจะมีค่า ต่าง ๆ ไม่เหมือนกัน เช่น เงิน เป็นต้น โดยจะมีตัวละครให้เลือกทั้งหมด 8 ตัวดังนี้



รูปที่ 4.13 แสดงหน้าจอการเลือกตัวละคร

มีรายละเอียดดังต่อไปนี้

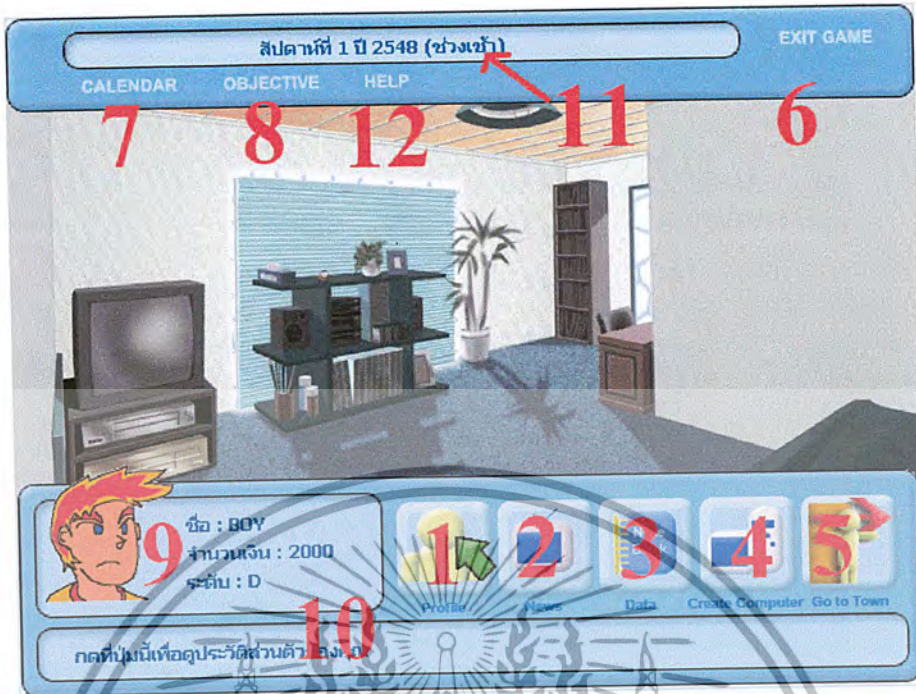
1. ตรง Text Box นั้นจะให้เราทำการตั้งชื่อให้กับตัวละคร โดยการรับค่าจาก Keyboard เมื่อเราตั้งชื่อเสร็จแล้ว เราก็สามารถเข้าเริ่มต้นของเกมได้
2. ตรงคำว่า Random นั้นเป็นการเลือกตัวละครแบบสุ่ม
3. ตรงปุ่มคำว่า Back to Main Menu เป็นการย้อนกลับไปที่เมนูเริ่มต้น
4. ตรงปุ่มคำว่า Create Game นั้นเป็นปุ่มที่ใช้เพิ่มเข้าสู่หน้าจอเกมหลัก สามารถกดปุ่มนี้ได้ ก็ต่อเมื่อมีการตั้งชื่อเรียบร้อยแล้ว



รูปที่ 4.14 แสดงหน้าจอการตั้งชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าเริ่มเล่นเกมนั้น เราจะมาอยู่ในหน้าของตัวละครเอง ซึ่งเป็นสถานที่เอาไว้ดูรายละเอียดต่าง ๆ ดังนี้



รูปที่ 4.15 แสดงหน้าจอในห้อง

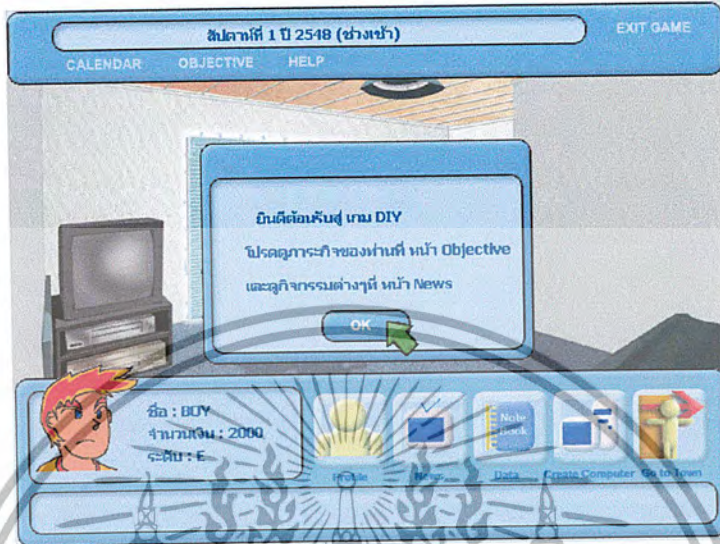
โดยมีรายละเอียดตามหมายเลขดังนี้ดังนี้

1. Profile เป็นปุ่มที่เอาไว้กดเพื่อดูเพื่อแสดงข้อมูลของตัวละคร
2. New เป็นปุ่มที่เอาไว้กดเพื่อดูเพื่อแสดงข่าวสารต่างที่เกิดขึ้นภายในเกม / บอกEvent ที่จะเกิดขึ้นในเกม / บอกว่าจะมีกิจกรรมต่างๆ เกิดขึ้นที่ไหนบ้าง
3. Data เป็นปุ่มเอาไว้กดเพื่อดูเพื่อแสดงข้อมูลที่เราได้มาจากการเหตุการณ์ต่างๆ ที่เคยผ่านมาแล้ว
4. Create Computer เป็นปุ่มที่เอาไว้สร้างคอมพิวเตอร์เมื่อเราสะสมอุปกรณ์คอมพิวเตอร์ครบทุกชิ้นแล้ว
5. Go to Town เป็นปุ่มที่เอาไว้กดเมื่อเราต้องการออกจากห้องเพื่อเข้าสู่เมืองไปทำกิจกรรมต่างๆ ที่เกิดขึ้น
6. Exit Game เอาไว้ย้อนกลับไปหน้าแรก
7. Calendar เอาไว้ดูปฏิทินและดูว่าจะมีเหตุการณ์อะไรเกิดขึ้นที่ไหนบ้าง
8. Objective บอกจุดมุ่งหมายที่จะต้องทำในเกม
9. เอาไว้แสดงรายละเอียดของตัวละคร
10. เมื่อผู้เล่นนำเมาส์ไปวางไว้ที่ปุ่มไหนคนนั้นจะบอกว่าปุ่มนั้นเอาไว้ทำอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

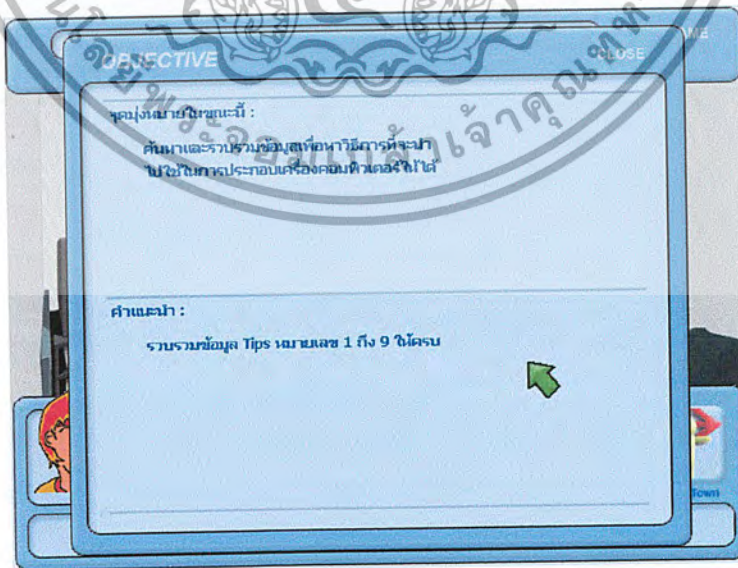
11. เอาไว้บอกผู้เล่นว่าตอนนี้เล่นอยู่ที่สัปดาห์เท่าไรและช่วงเวลา
12. Help เป็นปุ่มที่อธิบายรายละเอียดการเล่นเกมน

เมื่อผู้ที่เริ่มต้นเล่นเกมใหม่นั้นเข้ามาจะเจอหน้าจอดังรูป



รูปที่ 4.16 แสดงหน้าจอในห้องสำหรับผู้ที่เริ่มต้นเล่นเกมใหม่

สำหรับผู้ที่เริ่มต้นเล่นเกมใหม่นั้นเมื่อเข้ามาในส่วนนี้ทางเกม จะบอกว่ายินดีต้อนรับสู่เกม DIY โปรดดูภารกิจของท่านที่หน้าObjective และดูกิจกรรมต่างๆ ที่เกิดขึ้นในหน้าNEW เมื่อคลิกที่ปุ่ม Objective จะมีหน้าจอดังรูป



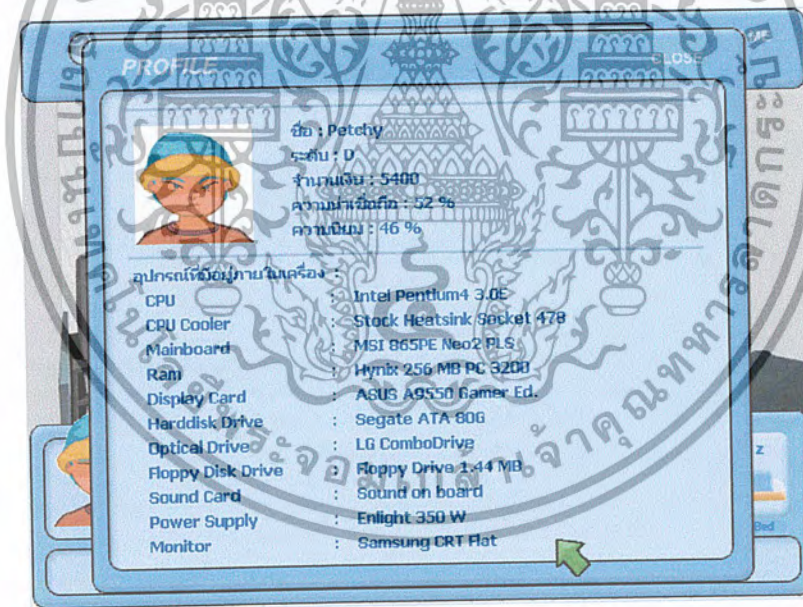
รูปที่ 4.17 หน้าจอ Objective

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เล่นกดที่ปุ่ม Profile จะมีหน้าจอเป็นลักษณะดังรูป



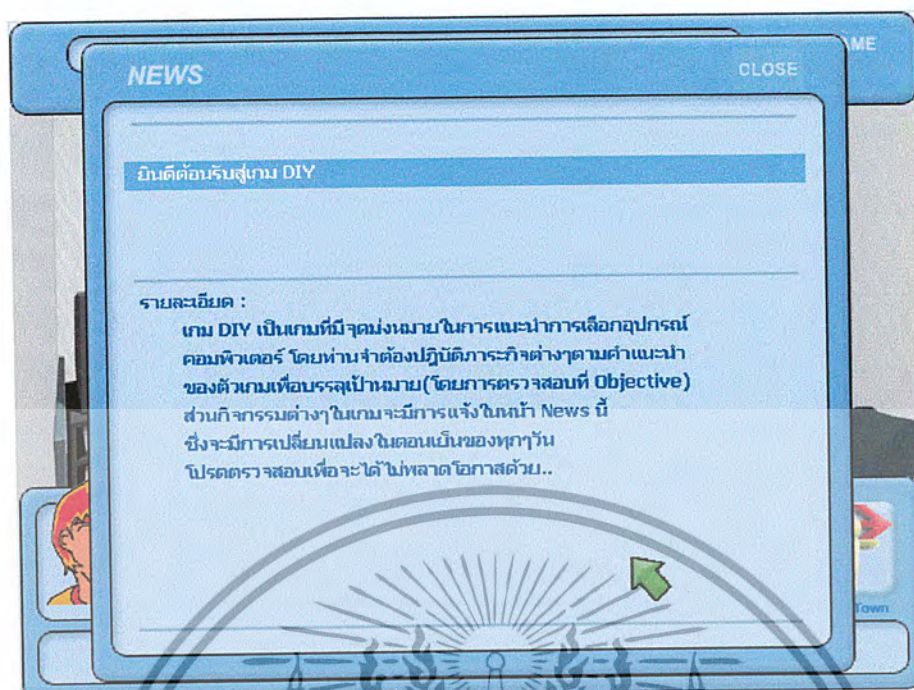
รูปที่ 4.18 หน้าจอ Profile กรณีที่ยังไม่สามารถประกอบคอมพิวเตอร์ได้



รูปที่ 4.19 หน้าจอ Profile กรณีที่สามารถประกอบคอมพิวเตอร์ได้

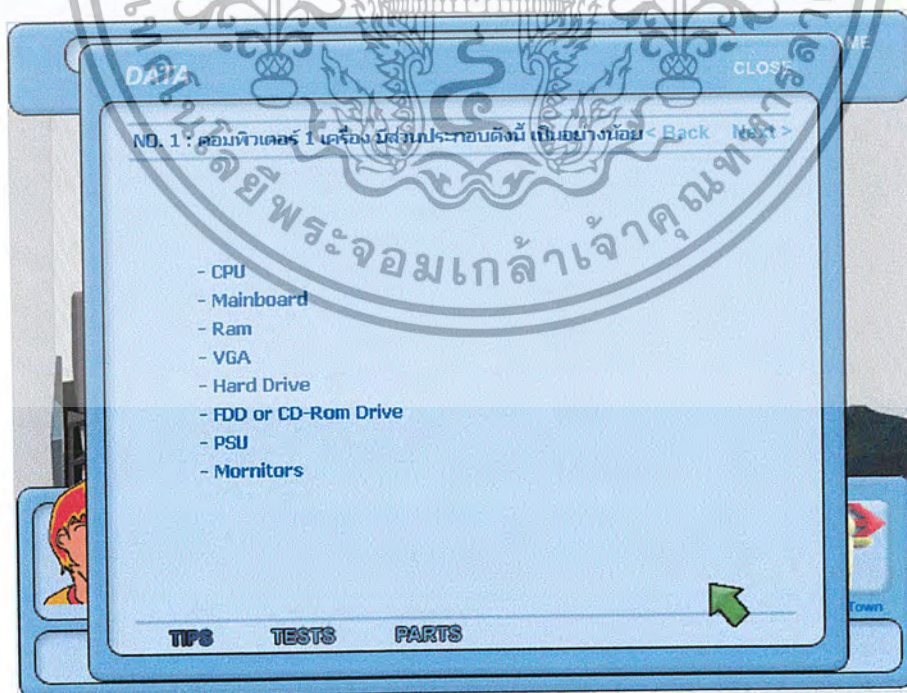
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าสู่หน้าจอ New จะมีลักษณะดังรูป



รูปที่ 4.20 หน้าจอ New

เมื่อเข้าสู่หน้าจอ Data ในส่วนของ Tips จะมีลักษณะดังรูป

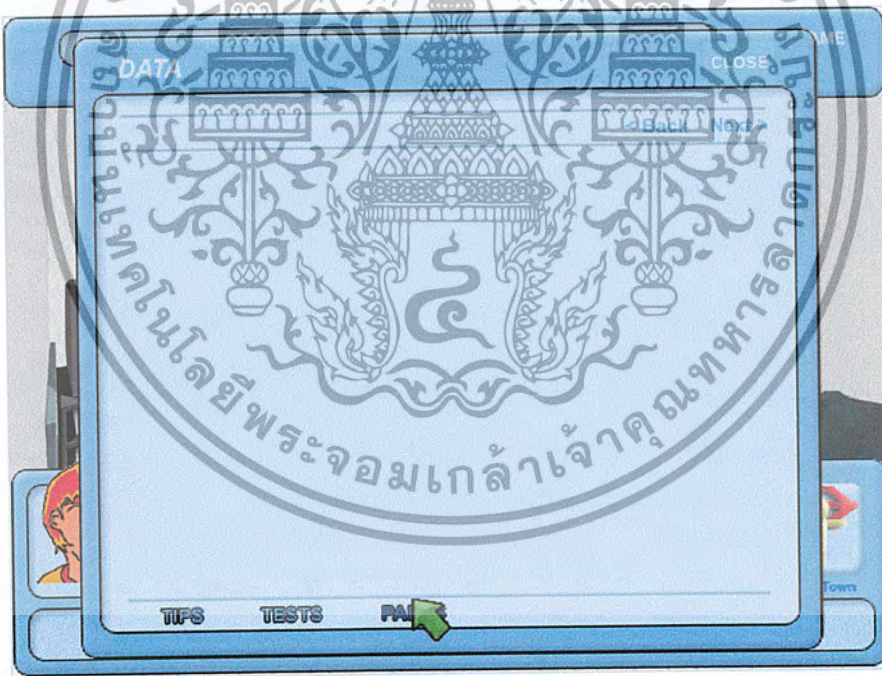


รูปที่ 4.21 หน้าจอ Data ในส่วนของ Tips สำหรับผู้ที่เล่นเกมใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

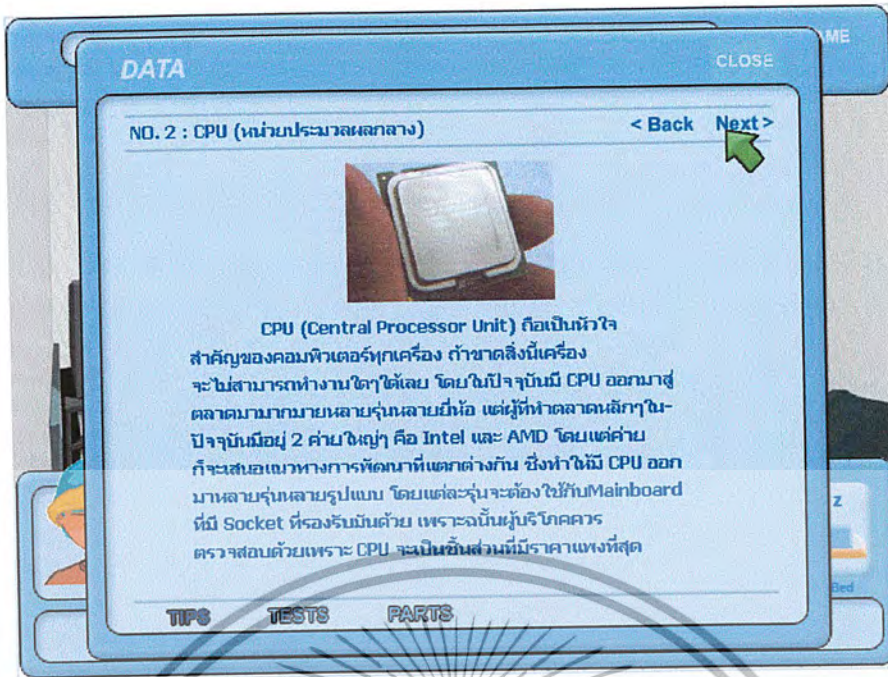


รูปที่ 4.22 หน้าจอ Data ในส่วนของ Tests สำหรับผู้ที่เล่นเกมใหม่

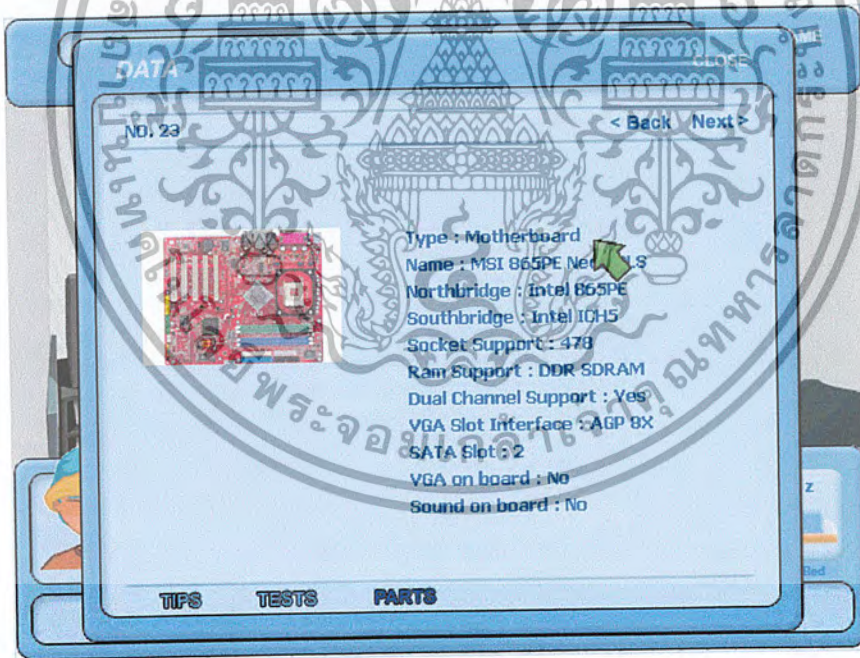


รูปที่ 4.23 หน้าจอ Data ในส่วนของ Parts สำหรับผู้ที่เล่นเกมใหม่

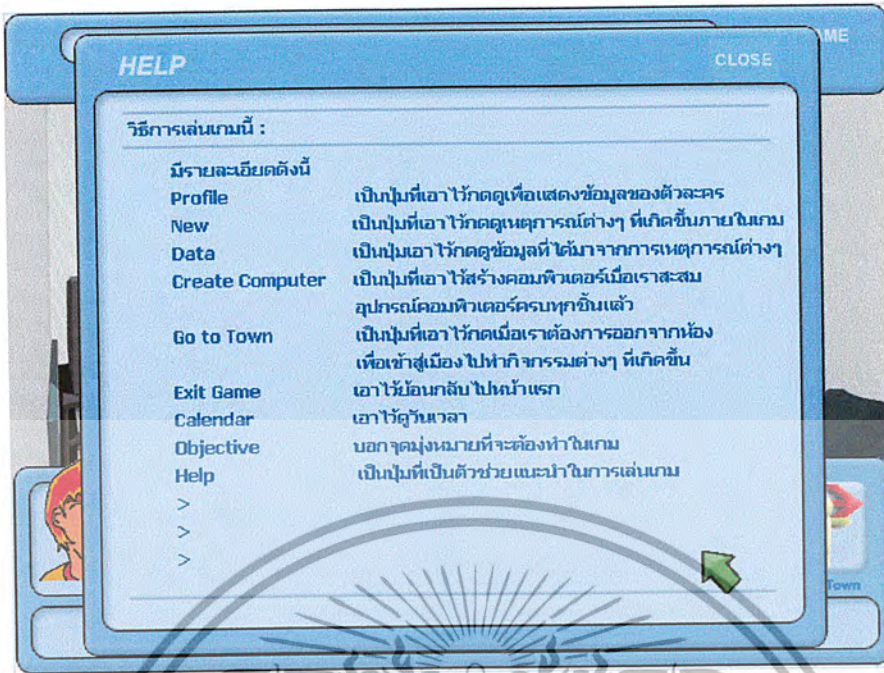
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 หน้าจอ Data ในส่วนของ Tips เมื่อผู้เล่นนั้นได้สะสมข้อมูลต่างๆ แล้ว

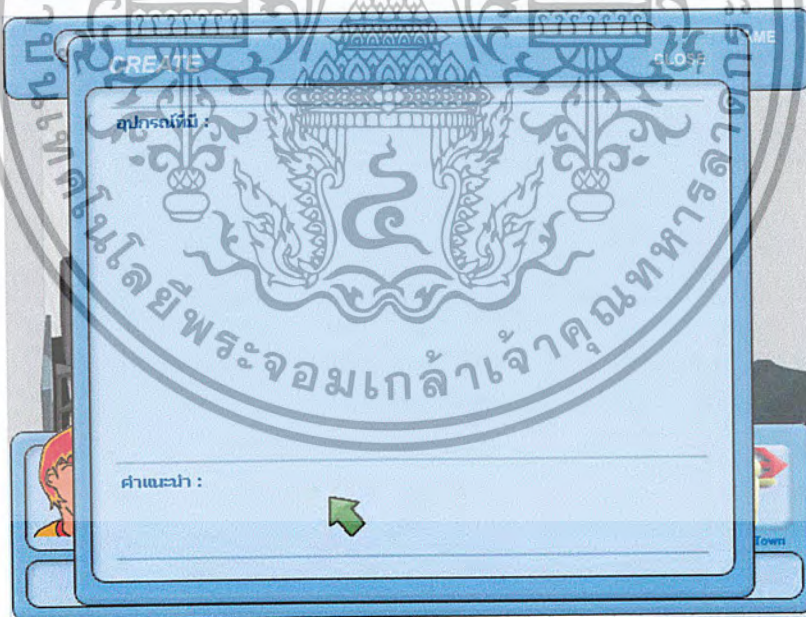


รูปที่ 4.25 หน้าจอ Data ในส่วนของ Tests เมื่อผู้เล่นนั้นมีชิ้นส่วนคอมพิวเตอร์แล้ว



รูปที่ 4.26 หน้าจอ Help

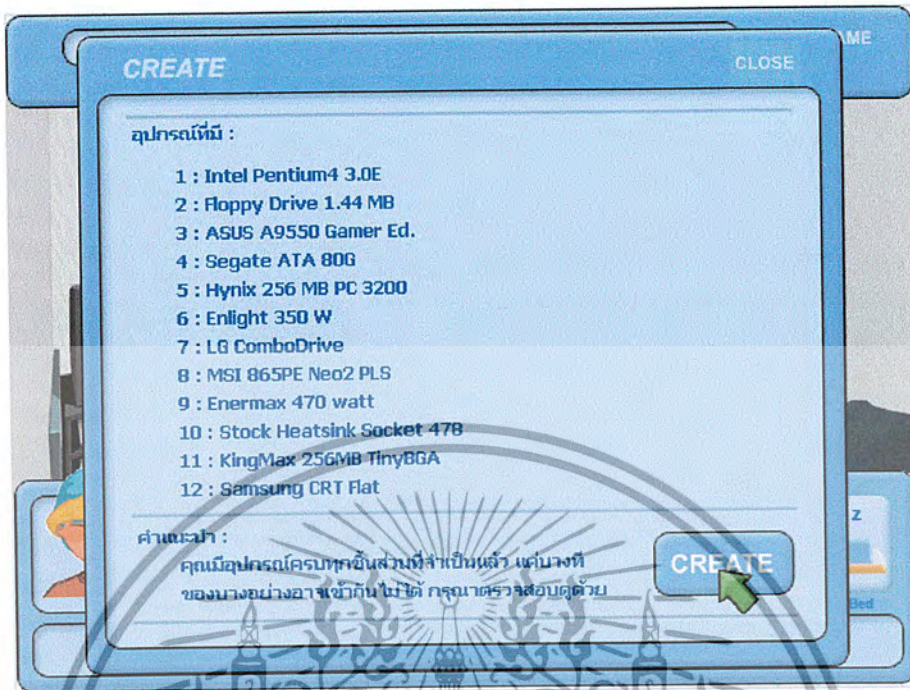
เมื่อกดที่ปุ่ม Create Computer



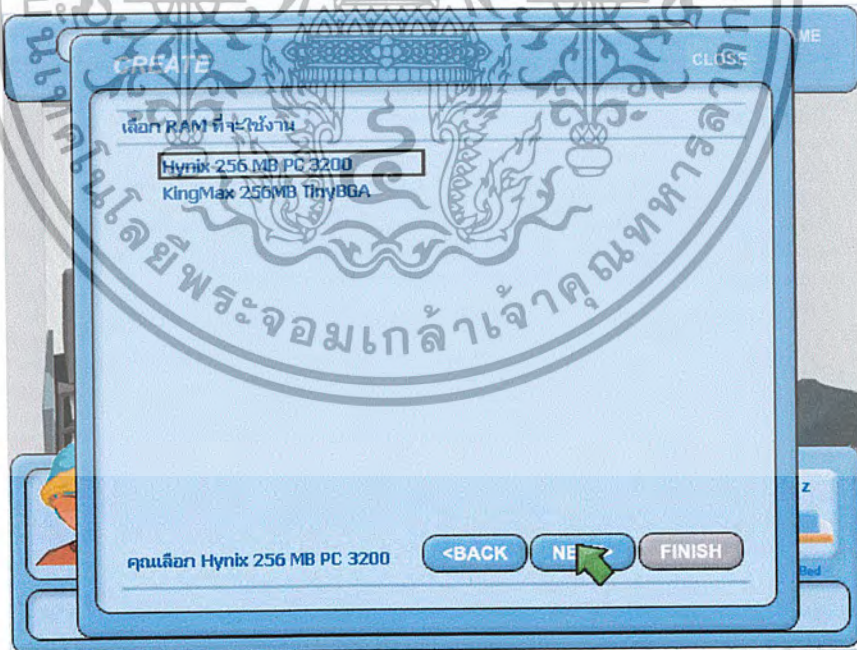
รูปที่ 4.27 หน้าจอการประกอบคอมพิวเตอร์สำหรับผู้เล่นเกมใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เล่นนั้นสะสมอุปกรณ์คอมพิวเตอร์จนสามารถประกอบคอมพิวเตอร์ได้นั้นจะมีหน้าตาดังต่อไปนี้



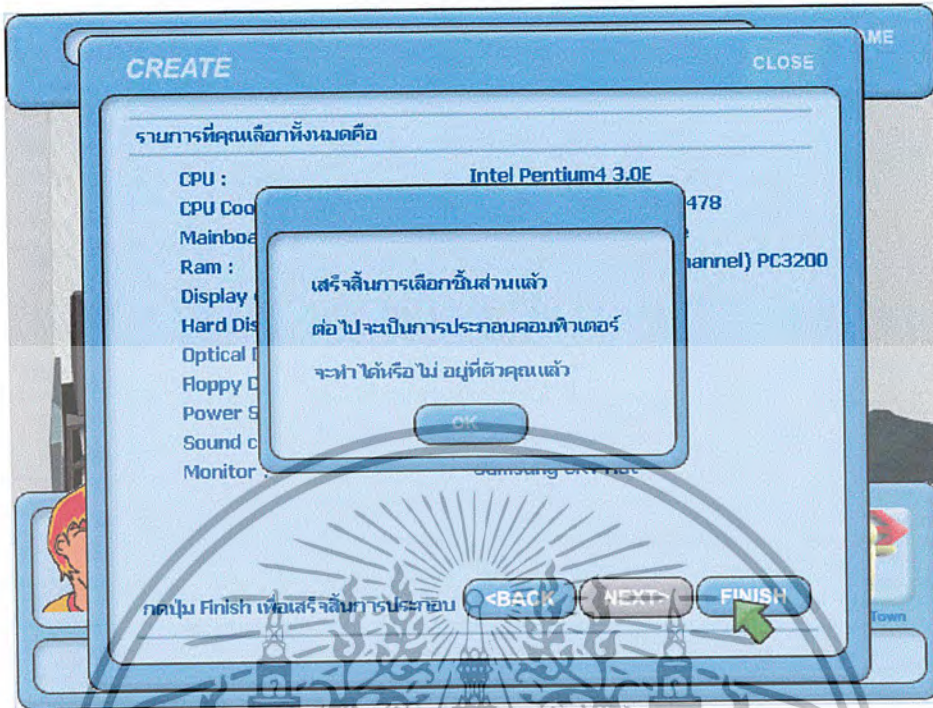
รูปที่ 4.28 รูปหน้าจอการประกอบคอมพิวเตอร์ (กรณีที่มีอุปกรณ์คอมพิวเตอร์ครบแล้ว)



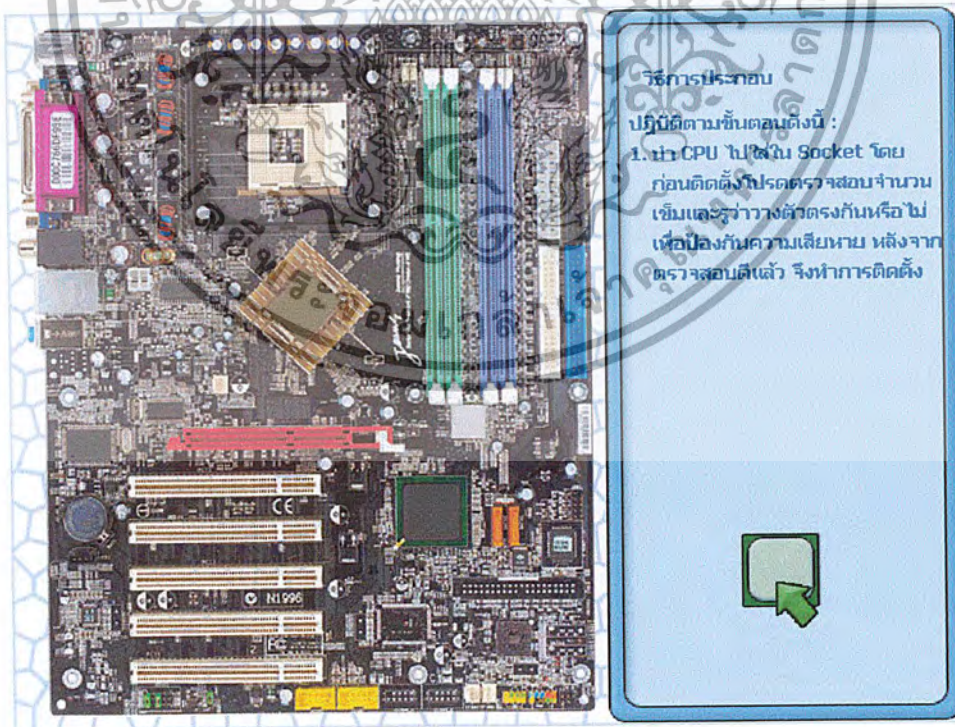
รูปที่ 4.29 หน้าจอการเลือกชิ้นส่วนที่จะประกอบคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกอุปกรณ์คอมพิวเตอร์ที่จะประกอบเสร็จเรียบร้อยแล้วจะมี Message Box ขึ้นมาบอกต่อไปเป็นการประกอบอุปกรณ์คอมพิวเตอร์

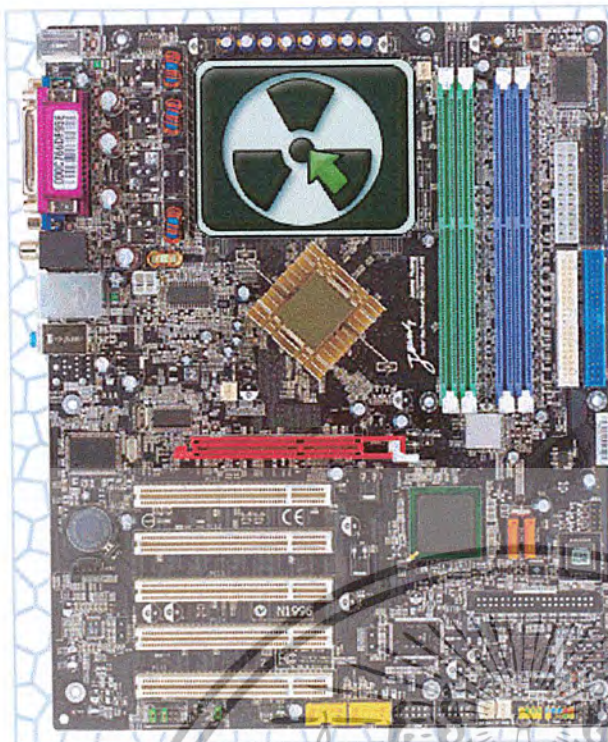


รูปที่ 4.30 หน้าจอบอกว่าเลือกอุปกรณ์ที่จะประกอบเสร็จเรียบร้อยแล้ว



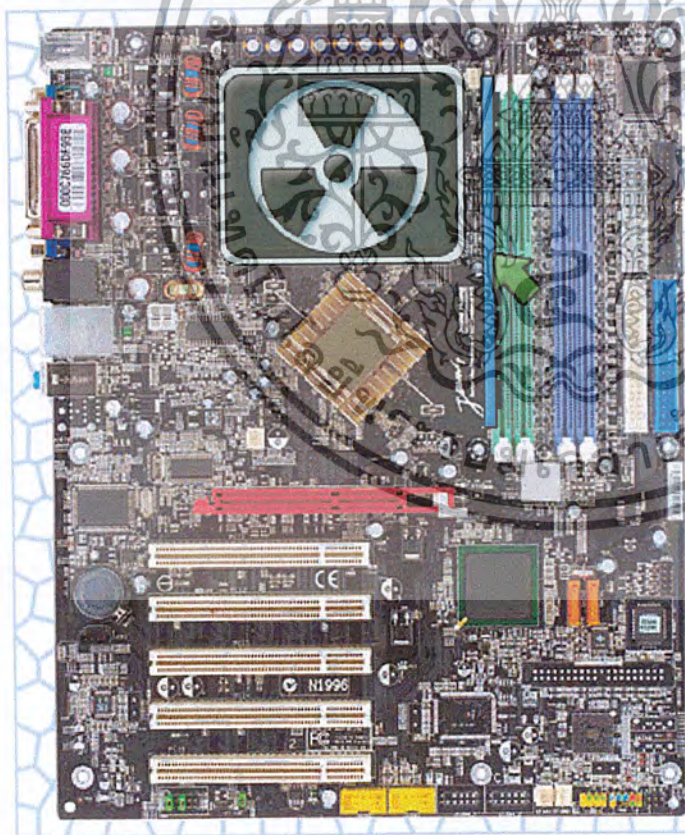
รูปที่ 4.31 หน้าจออธิบายการประกอบ CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วิธีการประกอบ
 ปฏิบัติตามขั้นตอนดังนี้ :
 2. นำ Heatsink วางบน CPU โดยก่อนวางให้หาขั้วลิโคเนหรือติดแผ่นนำความร้อนที่บริเวณ core ของ CPU และขณะวางโปรดตรวจสอบว่านำสัมผัสของ CPU กับ Heatsink สัมผัสกันดี จากนั้นต่อสายพัดลมกับboard

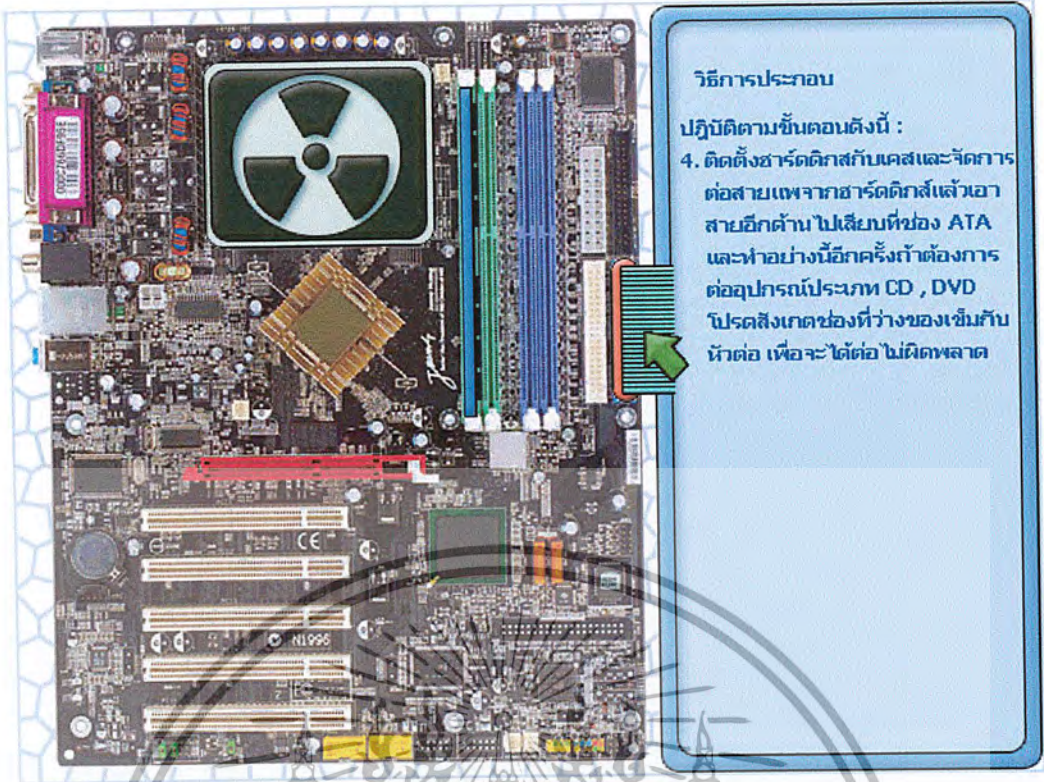
รูปที่ 4.32 หน้าจออธิบายการประกอบ Heat sink



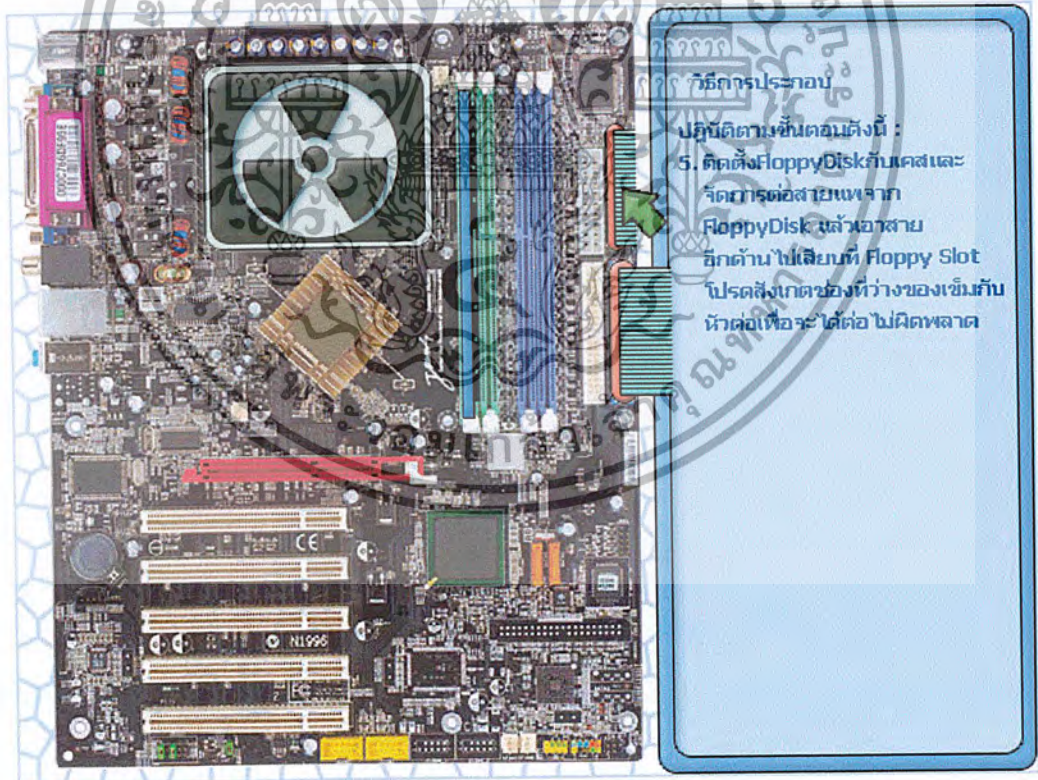
วิธีการประกอบ
 ปฏิบัติตามขั้นตอนดังนี้ :
 3. นำ RAM ที่มี Interface ตรงกัน โดยดูจากตำแหน่งคั่นวางตรงกัน เมื่อตรวจสอบจนแน่ใจว่าสามารถใส่ลงไปได้ จึงทำการเสียบลงไป จากนั้นนำ Mainboard เข้าไปวางในเคสแล้วจัดการขันน็อตยึดครบทุกจุดให้เรียบร้อย

รูปที่ 4.33 หน้าจออธิบายการประกอบ RAM

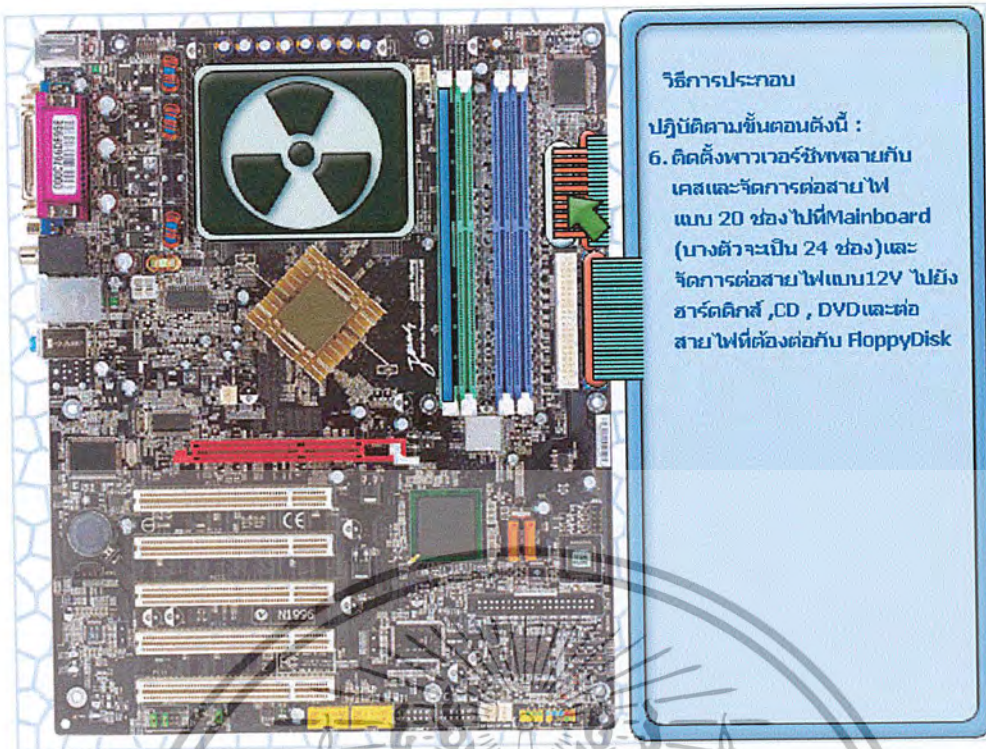
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



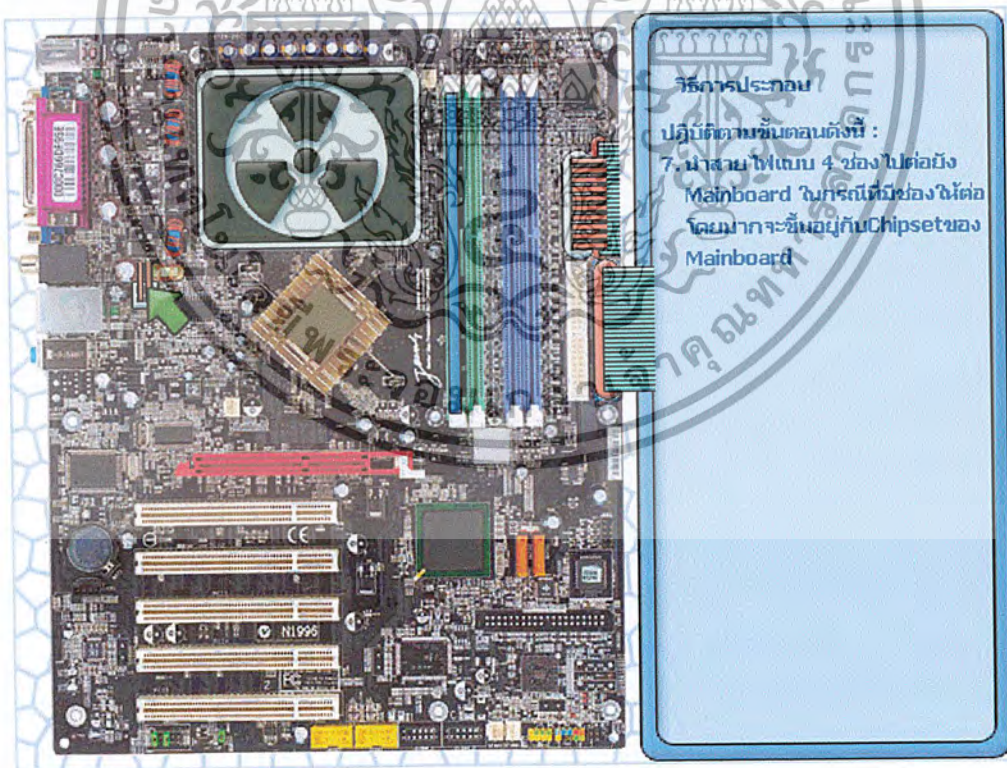
รูปที่ 4.34 หน้าจออธิบายการประกอบสายแพ



รูปที่ 4.35 หน้าจออธิบายการประกอบสาย Floppy Disk

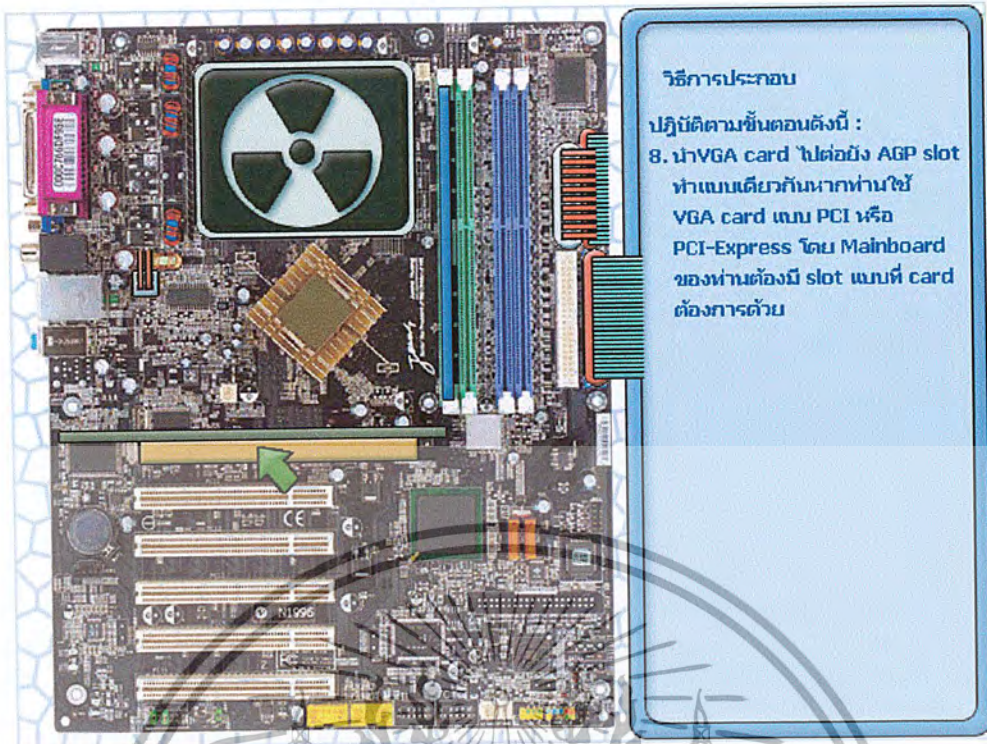


รูปที่ 4.36 หน้าจออธิบายการประกอบสาย Power Supply

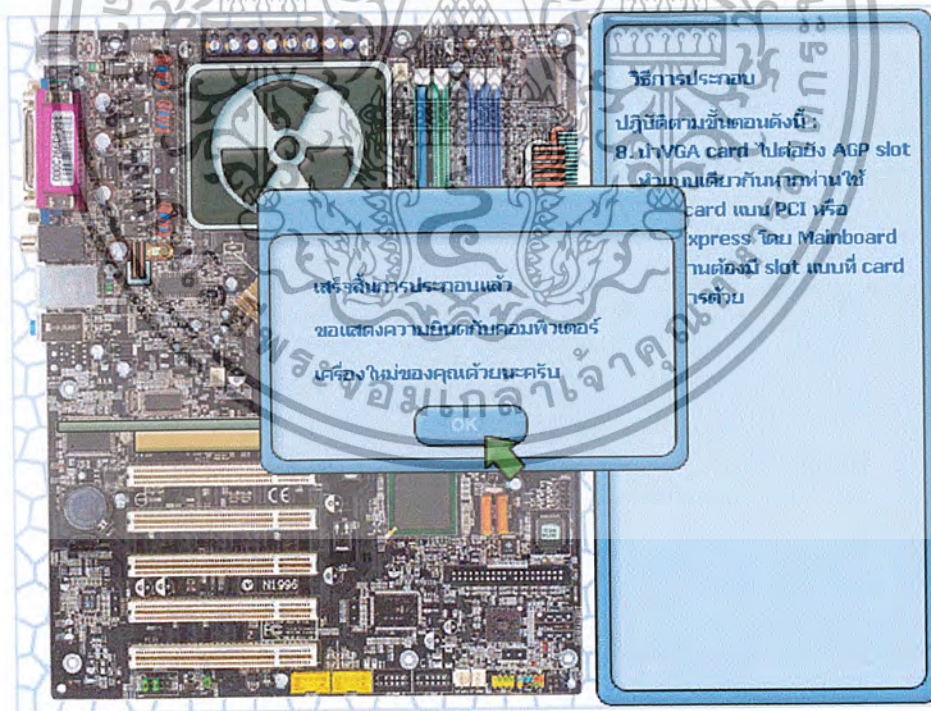


รูปที่ 4.37 หน้าจออธิบายการประกอบสาย Power Supply แบบ 4 pin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

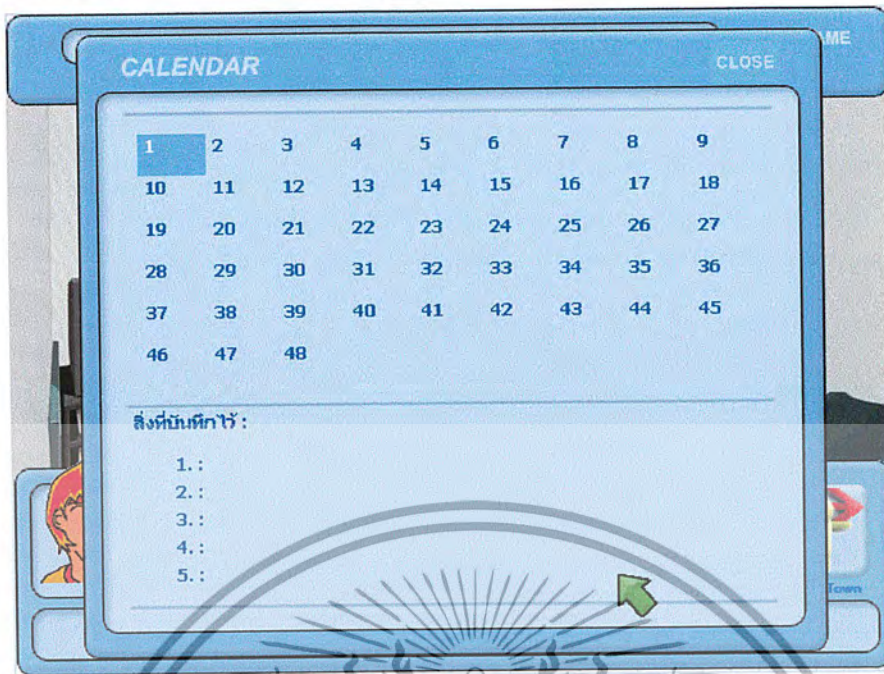


รูปที่ 4.38 หน้าจออธิบายการประกอบสาย VGA



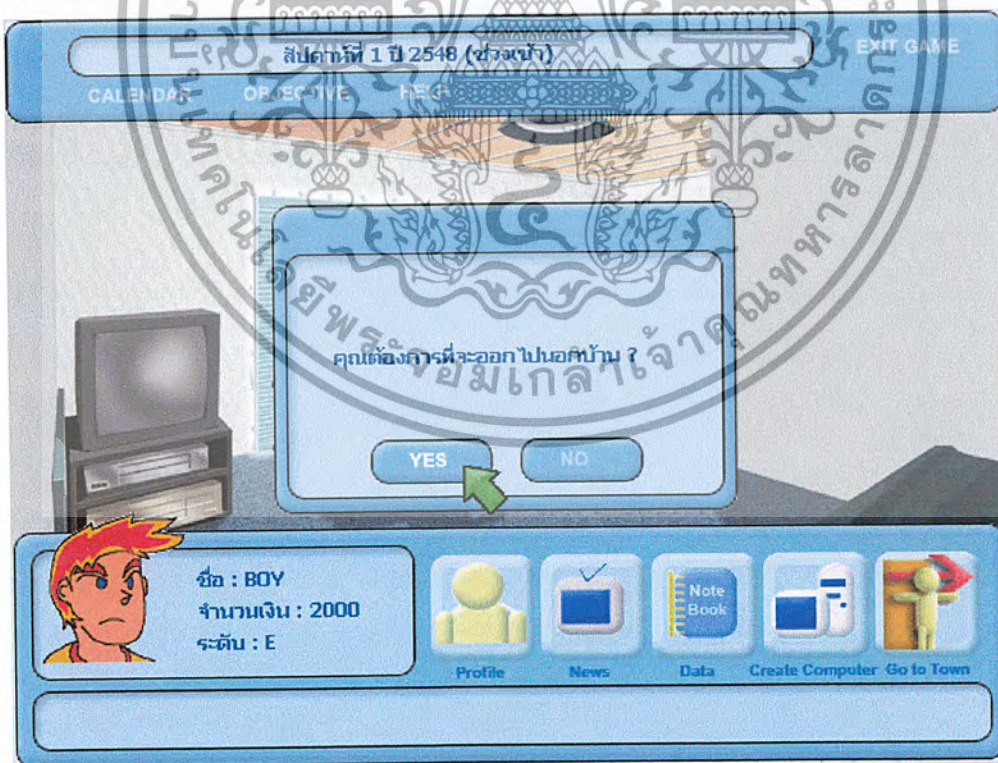
รูปที่ 4.39 หน้าจอบอกว่าประกอบคอมพิวเตอร์เสร็จสิ้นแล้ว

เมื่อผู้เล่นกดที่ปุ่ม Calendar จะมีลักษณะดังรูป



รูปที่ 4.40 หน้าจอของ Calendar

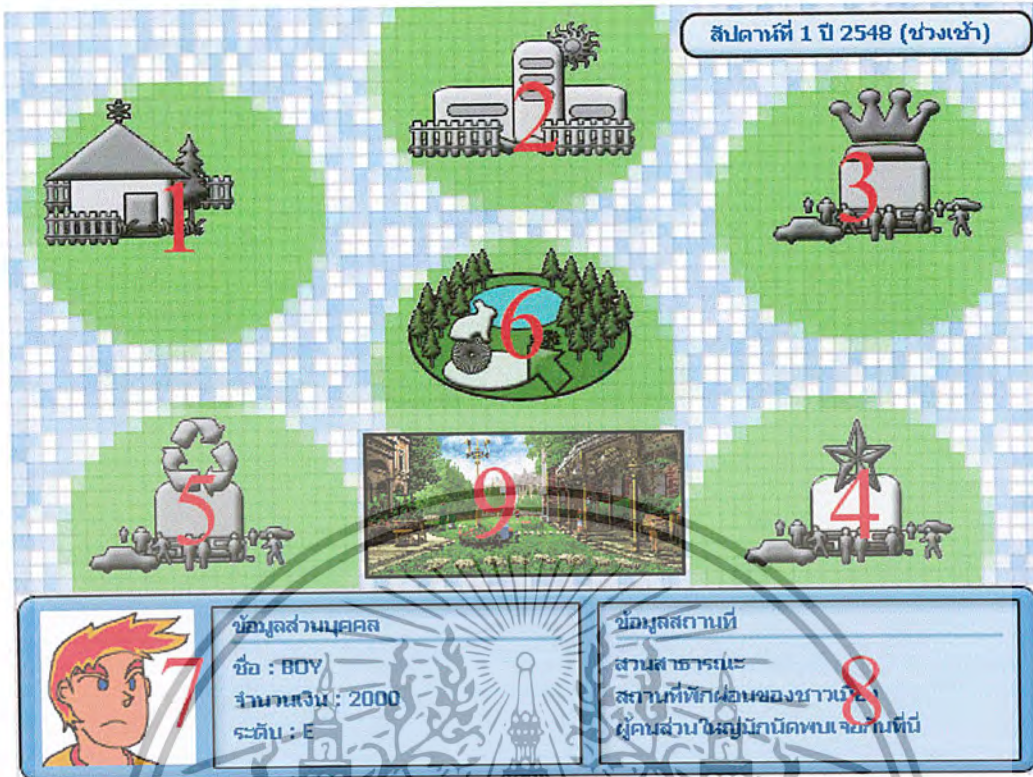
เมื่อผู้เล่นต้องการจะออกจากห้องจะมี Message Box ตามดังรูป



รูปที่ 4.41 รูปแสดง Message Box ตามผู้เล่นก่อนจะออกจากห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราต้องการออกจากฉากบ้านนั้นเราก็ต้องกดปุ่ม Go to town ก็จะเข้าสู่ฉากเมืองดังรูป



รูปที่ 4.42 แสดงหน้าจอฉากภายในเมือง

ซึ่งภายในเมืองนั้นจะประกอบไปด้วยสถานที่ต่าง ๆ ดังนี้

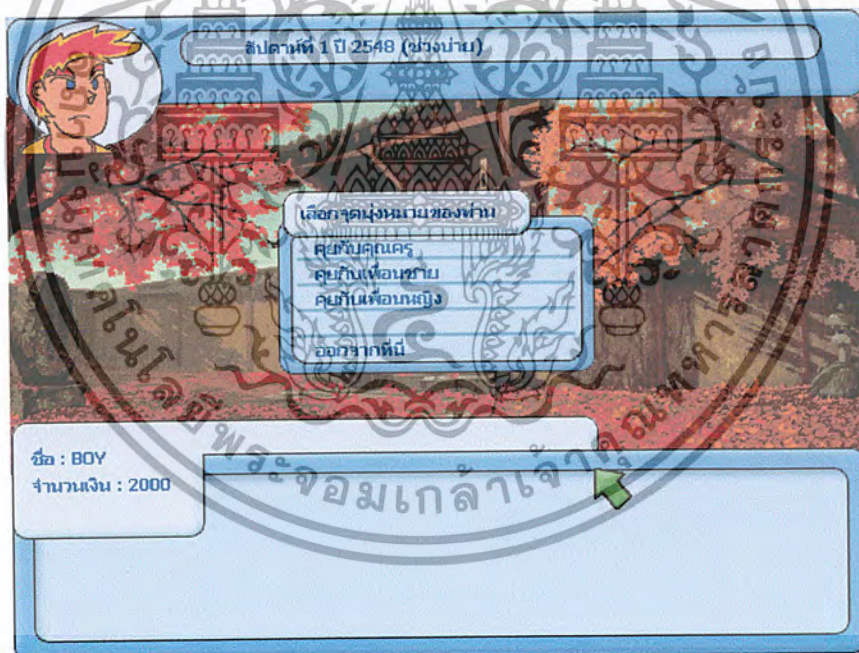
1. บ้าน เป็นสถานที่ที่เราเอาไว้บอกสถานะต่างๆ ของเรา เช่นเอาไว้บอกว่าเรามีอุปกรณ์คอมพิวเตอร์กี่ชิ้นแล้ว, เอาไว้บอกว่าในสัปดาห์นั้นจะมีกิจกรรมอะไรเกิดขึ้นบ้างและเกิดขึ้นที่ไหนบ้าง เป็นต้น ซึ่งถ้าเราเข้าไปในบ้านนั้นเราจะเข้าไปอยู่ในฉากห้องของเราเอง
2. โรงเรียน เป็นสถานที่ที่เราเอาไว้ทำกิจกรรมต่างๆ ที่เกิดขึ้นเช่น การเข้าทดสอบ (การตอบคำถาม), การคุยกับเพื่อน, การคุยกับคุณครู
3. ย่านร้านค้าที่ 1 เป็นสถานที่ที่เราเอาไว้ทำกิจกรรมต่างๆ ที่เกิดขึ้นเช่น เกมทายสูงต่ำ, เกมจับคู่ เป็นต้น ซึ่งเหตุการณ์ต่างๆ นั้นจะเกิดขึ้นแบบ Random ซึ่งสถานที่ใดที่หนึ่งนั้นจะมีเกมเกิดขึ้นเพียงแค่ 1 เกมเท่านั้น และยังมีการขายสินค้าเกิดขึ้นด้วย โดยจะมีร้านค้าให้เลือกเข้าไปซื้อ ซึ่งจะมีร้านค้าให้เอาไว้เลือกซื้อของเพียง 2 ร้าน คือร้านค้าที่ 1 และ ร้านค้าที่ 2
4. ย่านร้านค้าที่ 2 เป็นสถานที่ที่เราเอาไว้ทำกิจกรรมต่างๆ ที่เกิดขึ้นเช่น เกมทายสูงต่ำ, เกมจับคู่ เป็นต้น ซึ่งเหตุการณ์ต่างๆ นั้นจะเกิดขึ้นแบบ Random ซึ่งสถานที่ใดที่หนึ่งนั้นจะมีเกมเกิดขึ้นเพียงแค่ 1 เกมเท่านั้น และยังมีการขายสินค้าเกิดขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะมีร้านค้าให้เลือกเข้าไปซื้อ โดยสถานที่นี้จะมีเพียง 1 ร้านค้าเท่านั้นก็คือ ร้านค้าที่ 3 และ ยังมีร้านซ่อมคอมพิวเตอร์ที่ 1

5. ย่านร้านค้าที่ 3 เป็นสถานที่ที่เอาไว้ทำกิจกรรมต่างๆ ที่เกิดขึ้นเช่น เกมทายสูงต่ำ, เกมจับคู่ เป็นต้น ซึ่งเหตุการณ์ต่างๆ นั้นจะเกิดขึ้นแบบ Random ซึ่งสถานที่ใดที่หนึ่ง นั้นจะมีเกมเกิดขึ้นเพียงแค่ 1 เกมเท่านั้น และยังมีร้านค้าสินค้ามือสองด้วยซึ่งเมื่อเราเข้าไปในร้านขายสินค้ามือสองนั้นเราจะได้ซื้อของในราคาที่ถูกลงกว่าซื้อของในร้านค้าปกติ และก็ยังมีการซ่อมคอมพิวเตอร์ที่ 2 อีกด้วย
6. สวนสาธารณะ จะมีกิจกรรมเกิดขึ้นหลายอย่างเช่น การพูดคุยกับคุณยาย, การพูดคุยกับคุณตา เพื่อเป็นการหาข้อมูลข่าวสารต่างๆ ภายในเกม
7. ข้อมูลของตัวละคร
8. ข้อมูลสถานที่
9. รูปแสดงสถานที่ที่เลือก

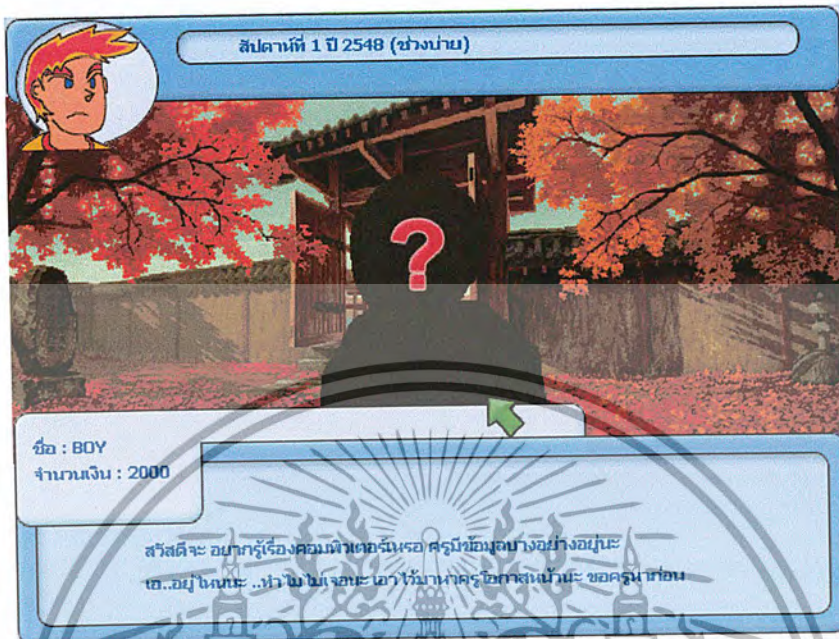
เมื่อเข้าที่โรงเรียนจะมีหน้าจอดังรูป



รูปที่ 4.43 แสดงหน้าจอฉากของโรงเรียน

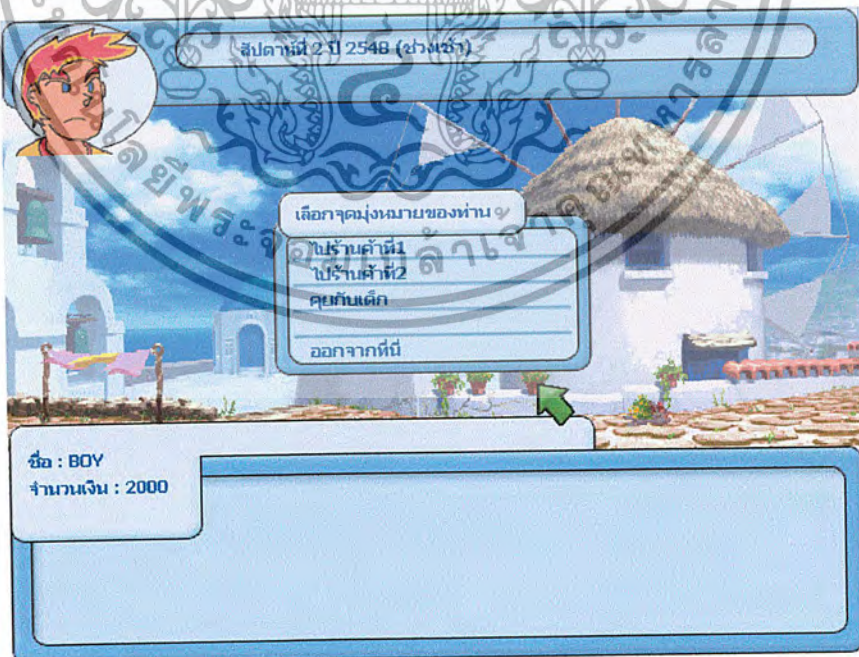
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งผู้เล่นนั้นจะต้องรวบรวมข้อมูลต่างๆ จากทุกสถานที่ โดยการพูดคุยกับบุคคลต่างๆ ที่มีในเกมจนได้ข้อมูลอุปกรณ์คอมพิวเตอร์มานั้น โดยการคุยนั้นอาจจะต้องคุยกับบุคคลเดิมๆ ซ้ำกันหลายๆ ครั้งถึงจะได้ข้อมูลนั้นมา



รูปที่ 4.44 รูปแสดงบทสนทนา

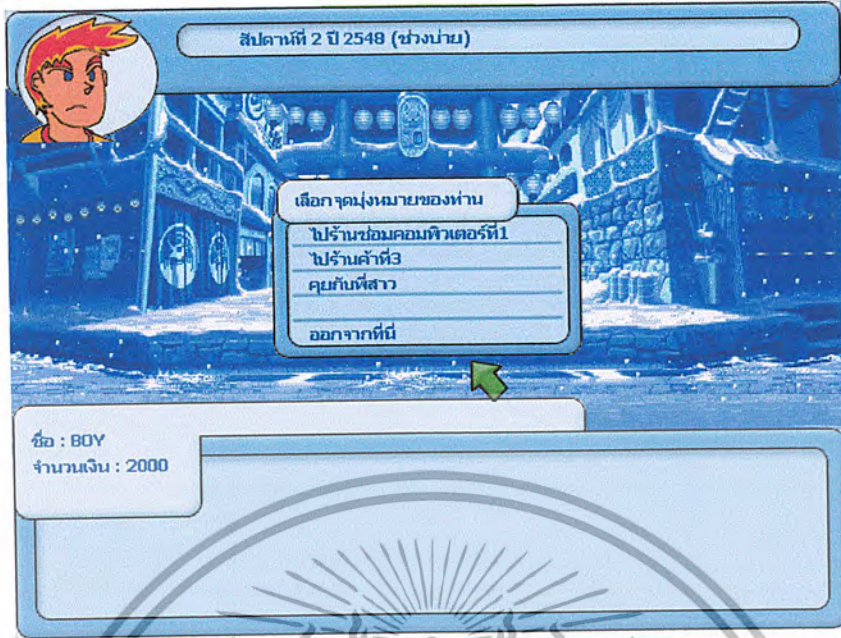
เมื่อเข้าที่ห้างสรรพสินค้า IT ที่ 1



รูปที่ 4.45 แสดงหน้าจอจากของห้างสรรพสินค้า IT ที่ 1

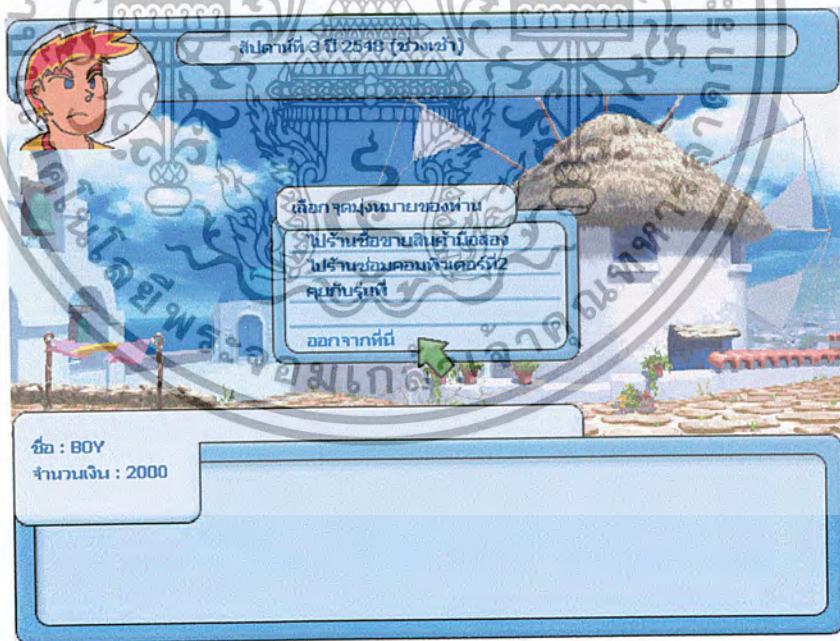
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าที่ห้างสรรพสินค้า IT ที่ 2



รูปที่ 4.46 แสดงหน้าจอฉากของห้างสรรพสินค้า IT ที่ 2

เมื่อเข้าที่ห้างสรรพสินค้า IT ที่ 3



รูปที่ 4.47 แสดงหน้าจอฉากของห้างสรรพสินค้า IT ที่ 3

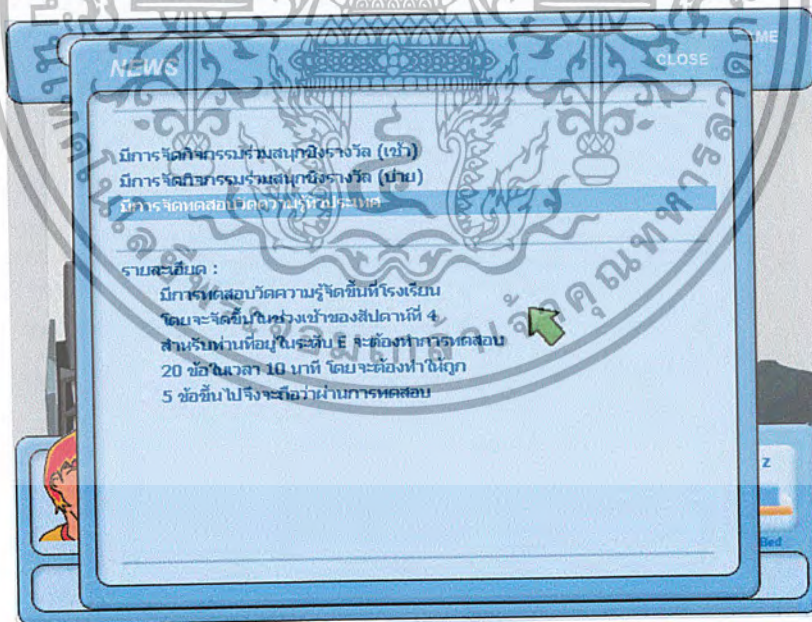
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าที่สวนสาธารณะ



รูปที่ 4.48 แสดงหน้าจอฉากของสวนสาธารณะ

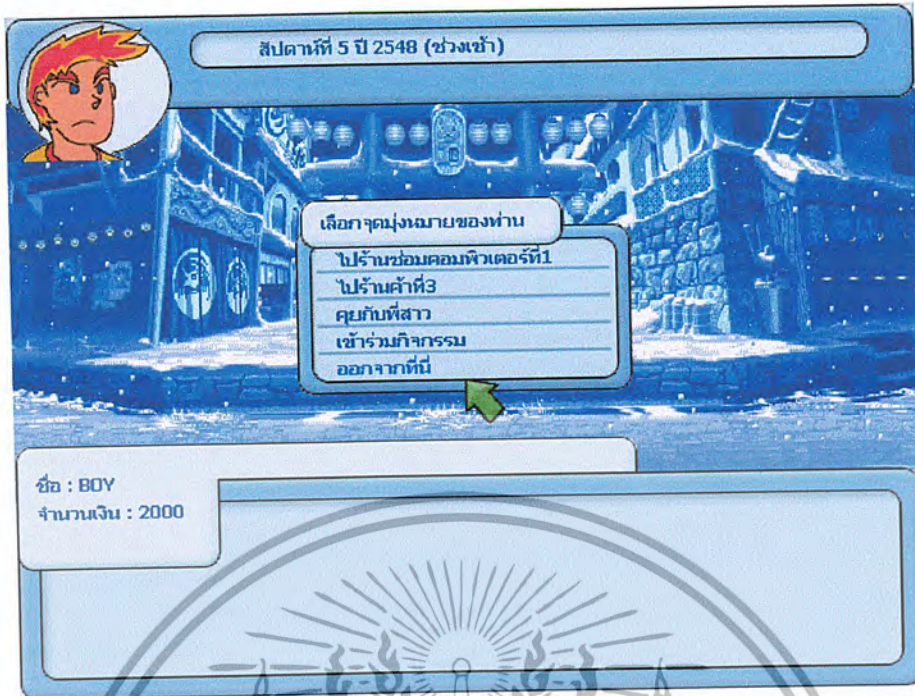
หน้าจอ New บอกว่าจะมีกิจกรรมอะไรเกิดขึ้นบ้าง



รูปที่ 4.49 แสดงหน้าจอ New บอกว่าจะมีอะไรเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงหน้าจอเมื่อมีกิจกรรมให้ร่วมสนุก



รูปที่ 4.50 แสดงหน้าจอเมื่อสามารถเข้าร่วมกิจกรรมได้

โดยกิจกรรมนั้นมีดังต่อไปนี้

ซึ่งกิจกรรมประเภทเกมนั้นจะมีหลายเกมเช่น เกมจับคู่ เป็นต้น เราจะสอดแทรกความรู้ต่างๆ เกี่ยวกับคุณสมบัติต่างๆ ของเครื่องคอมพิวเตอร์เอาไว้ในเกมด้วย

1. เกมจับคู่

เกมจับคู่นั้นเราจะมีรูปอุปกรณ์คอมพิวเตอร์ต่างๆ มากมายให้เลือกจับคู่ โดยเรากำหนดให้มีจำนวนทั้งหมด 48 คู่ด้วยกันทั้งสิ้นเพื่อความหลากหลายในการเล่นในแต่ละครั้ง ซึ่งจะมีอุปกรณ์คอมพิวเตอร์ดังต่อไปนี้

- | | |
|----------------|-----------------|
| - CPU | - Display Card |
| - RAM | - Sound Card |
| - Hard disk | - Mouse |
| - CD-Driver | - Key Board |
| - CD-ROM | - แผ่นดิสก์ |
| - ลำโพง | - DVD-ROM Drive |
| - Power Supply | - Flash Drive |
| - Main board | - Modem |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

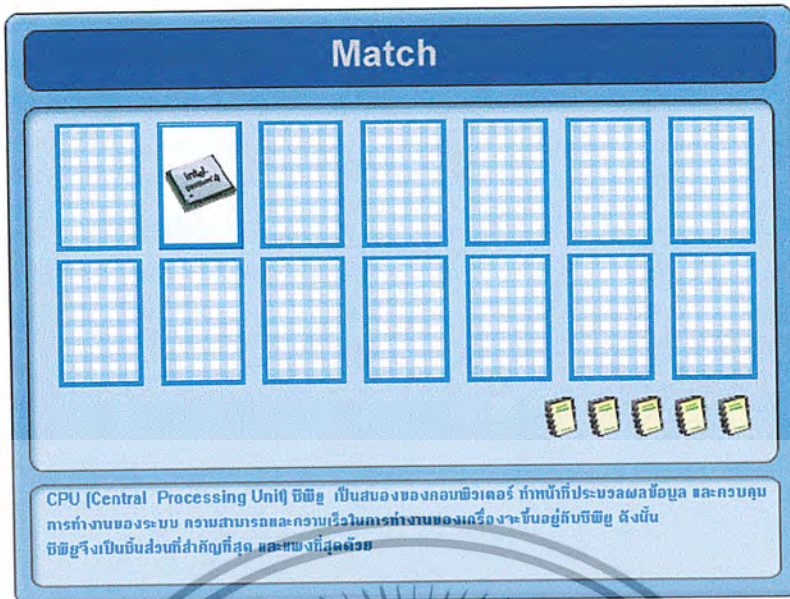
- | | |
|---------------------|-------------------|
| - DVD-ROM Drive | - DVD-Drive |
| - Monitor LCD | - Monitor CRT |
| - UPS | - CD-RW |
| - Floppy Disk Drive | - Scanner |
| - FAN | - Case |
| - Laser Printer | - Ink Jet Printer |

โดยเราจะกำหนดให้ผู้เล่นได้เล่นทั้งหมด 7 คู่จากที่มีทั้งหมด 48 คู่ โดยรูปที่ผู้เล่นจะได้เล่นนั้นเราจะสุ่มขึ้นมาโดยจะไม่มีคู่วินิจฉัยกันเลยซึ่งในการเล่นนั้นผู้เล่นต้องจับคู่ภาพอุปกรณ์คอมพิวเตอร์ให้ตรงกับชื่อรูปอุปกรณ์คอมพิวเตอร์ชนิดนั้น ๆ

โดยเราจะกำหนดให้ผู้เล่นนั้นสามารถผิดได้เพียงแค่ 5 ครั้งเท่านั้นเมื่อผู้เล่นเล่นผิดครบ 5 ครั้งก็จะออกจากเกมทันทีแต่เมื่อผู้เล่นชนะก็จะได้รับรางวัลเป็นอุปกรณ์คอมพิวเตอร์กับเงินรางวัลในการเลือกจับคู่แต่ละครั้งนั้นเมื่อผู้เล่นเปิดได้รูปอุปกรณ์คอมพิวเตอร์แล้วจะมีรายละเอียดของอุปกรณ์คอมพิวเตอร์ชนิดนั้นว่ามีชื่อว่าอะไร ทำหน้าที่ยังไง อยู่ตรงช่องที่เอาไว้แสดงรายละเอียดของอุปกรณ์คอมพิวเตอร์ แต่ถ้าผู้เล่นนั้นไม่สามารถจับคู่ได้ครบตามที่กำหนดก็จะมีเงินรางวัลให้ตามจำนวนคู่ที่เล่นได้ ซึ่งมีรายละเอียดดังต่อไปนี้

จับคู่ 1 คู่	ได้เงินรางวัล 500 บาท
จับคู่ 2 คู่	ได้เงินรางวัล 1000 บาท
จับคู่ 3 คู่	ได้เงินรางวัล 2000 บาท
จับคู่ 4 คู่	ได้เงินรางวัล 3000 บาท
จับคู่ 5 คู่	ได้เงินรางวัล 4000 บาท
จับคู่ 7 คู่	ได้เงินรางวัล 5000 บาท + อุปกรณ์คอมพิวเตอร์ 1 ชิ้น

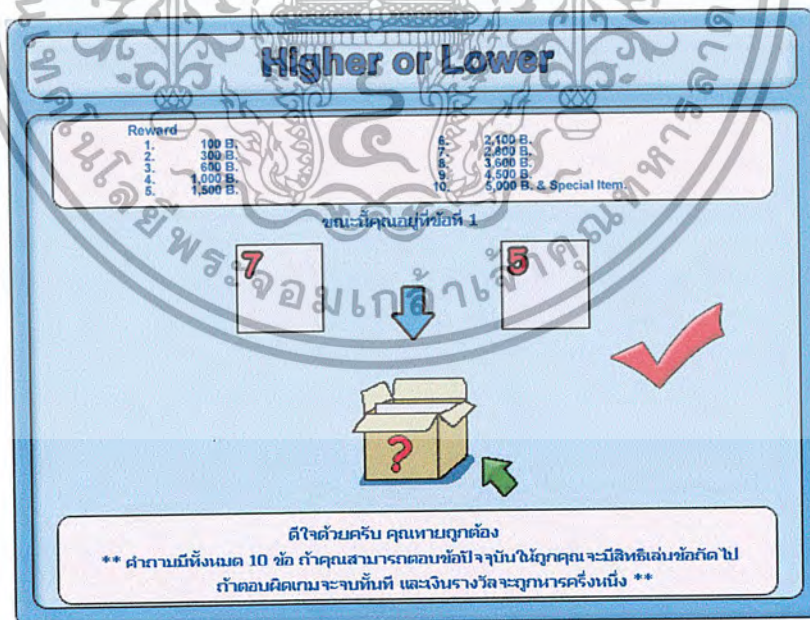
ซึ่งอุปกรณ์คอมพิวเตอร์แต่ละชนิดนั้นอาจมีหลายรูปด้วยกันเพื่อเพิ่มความหลากหลายให้กับผู้เล่น โดยตัวเกมนั้นจะมีลักษณะดังต่อไปนี้



รูปที่ 4.51 แสดงหน้าจอเกมจับคู่

2. เกมทายสูงต่ำ

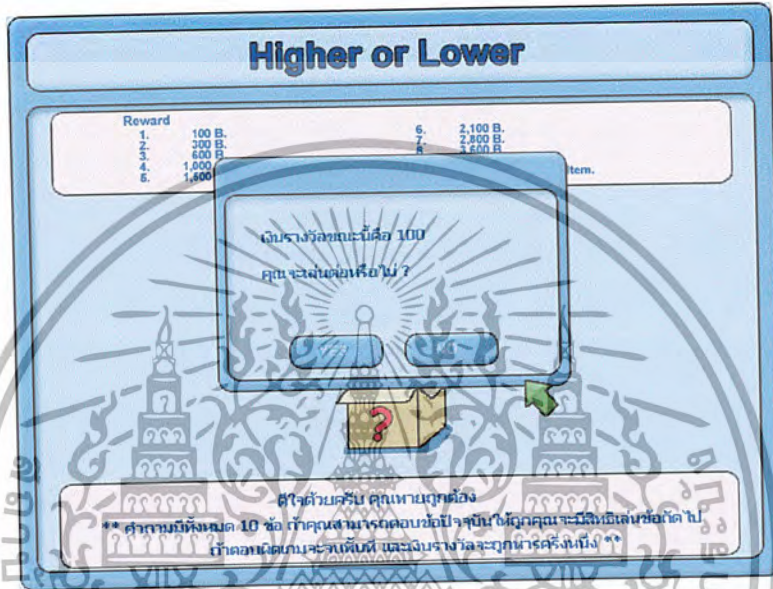
เป็นเกมที่ให้ผู้เล่นต้องทายกับดวงของตัวผู้เล่นเองว่าจะมีดวงมากน้อยแค่ไหนในการทายตัวเลขว่าตัวเลขตัวต่อไปที่จะเกิดขึ้นนั้นเป็นเลขอะไร ซึ่งตัวเลขที่มันนั้นจะมีตั้งแต่ เลข 1 ถึง เลข 9 โดยลักษณะของเกมนั้นมีรูปดังต่อไปนี้



รูปที่ 4.52 แสดงหน้าจอเกมทายสูงต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราจะกำหนดไว้ว่าถ้าตัวเลขที่ Random ออกมาซ้ำกับตัวเลขเริ่มต้นนั้นจะถือว่าเมื่อผู้เล่นทายสูงหรือทายต่ำก็จะถือว่าถูกต้องทั้งนั้น ผู้เล่นก็จะได้ทายเลขต่อไปได้อีกและได้รับเงินรางวัลตามจำนวนครั้งที่ทายถูก โดยเราจะมี Message Box ออกมาถามผู้เล่นว่าจะเล่นเกมต่อหรือเปล่า หรือว่าจะไม่เล่นเกมต่อ ถ้าผู้เล่นนั้นเลือกเล่นเกมต่อก็จะได้เล่นต่อไปถ้าผู้เล่นทายถูกก็จะได้รับเงินรางวัลเพิ่มขึ้น แต่ถ้าผู้เล่นทายผิดก็จะได้รับเงินรางวัลเพียงครั้งเดียวของที่เล่นได้เท่านั้น เมื่อผู้เล่นเล่นครบทั้ง 10 ครั้งแล้วก็จะได้รับรางวัลเป็นอุปกรณ์คอมพิวเตอร์ แต่ถ้าผู้เล่นไม่ต้องการเล่นต่อก็จะได้รับเงินรางวัลตามที่ผู้เล่นเล่นได้ในแต่ละครั้ง



รูปที่ 4.53 แสดง Message Box ถามว่าจะเล่นต่อหรือไม่

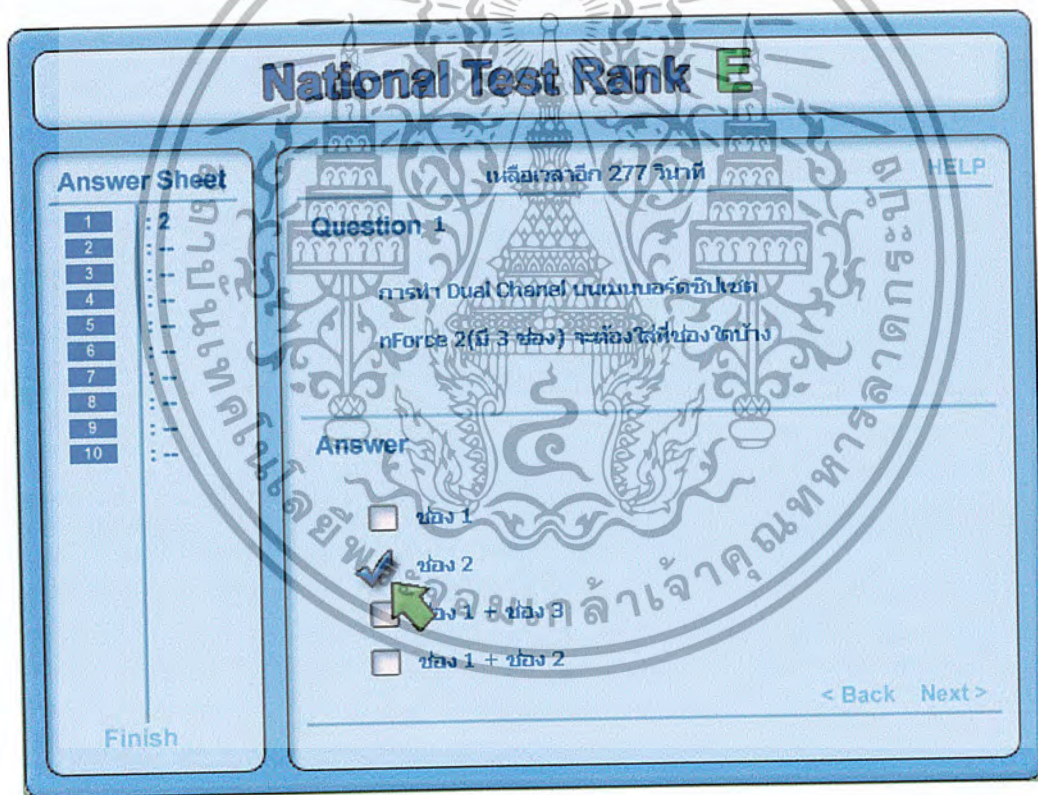
โดยรางวัลที่ได้มีลักษณะดังต่อไปนี้

- | | |
|-----------------|-----------------------------------|
| ทายถูก 1 ครั้ง | ได้รับรางวัล 100 บาท |
| ทายถูก 2 ครั้ง | ได้รับรางวัล 300 บาท |
| ทายถูก 3 ครั้ง | ได้รับรางวัล 600 บาท |
| ทายถูก 4 ครั้ง | ได้รับรางวัล 1000 บาท |
| ทายถูก 5 ครั้ง | ได้รับรางวัล 1500 บาท |
| ทายถูก 6 ครั้ง | ได้รับรางวัล 2000 บาท |
| ทายถูก 7 ครั้ง | ได้รับรางวัล 2800 บาท |
| ทายถูก 8 ครั้ง | ได้รับรางวัล 3600 บาท |
| ทายถูก 9 ครั้ง | ได้รับรางวัล 4500 บาท |
| ทายถูก 10 ครั้ง | ได้รับรางวัล 5000 บาท + ของรางวัล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เกมตอบคำถาม

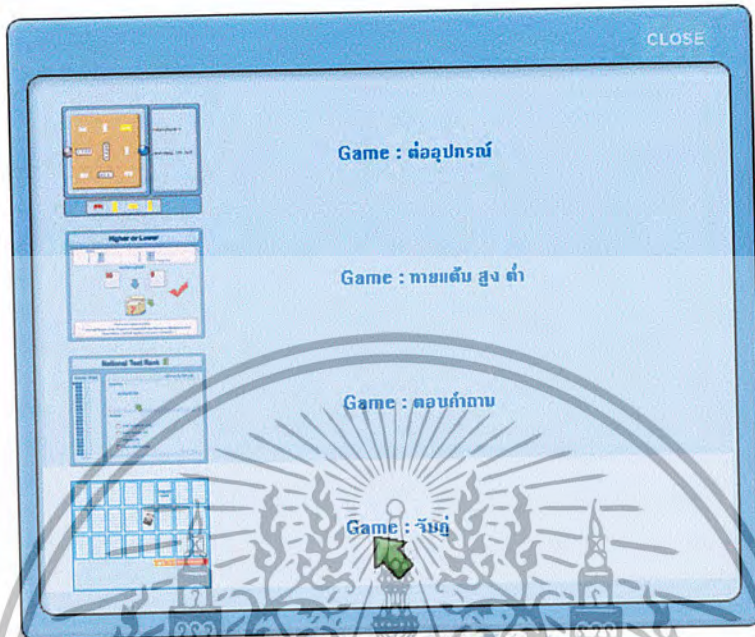
เกมตอบคำถามนั้นเราถือว่าเป็นอีกเกมหนึ่งในเกมใหญ่แต่เราจะแยกเกมตอบคำถามนั้นออกจากเกมอื่นๆ เพราะว่าจะกำหนดให้เกมตอบคำถามนั้นเป็นตัวกำหนด Rank ของผู้เล่น โดยเกมตอบคำถามนั้นเราจะให้เกิดขึ้นที่โรงเรียนที่เดียวเท่านั้นและจะเกิดขึ้นทุกๆ 12 สัปดาห์เท่านั้น โดยเราจะมีคำถามให้ผู้เล่นได้ตอบทั้งหมด 20 คำถาม จากทั้งหมดที่มี (คำถามนั้นเราสามารถเพิ่มเติมได้เรื่อยๆโดยใช้โปรแกรม Edit) คำถามที่ถามนั้นเราจะสุ่มจากคำถามที่มีทั้งหมดในไฟล์โดยเลือกมาเพียง 20 คำถาม และเราจะมีเวลากำหนดให้ผู้เล่นได้เล่นไว้จำนวน 600 วินาที ในการทดสอบเกมนี้นั้นผู้เล่นไม่จำเป็นต้องตอบคำถามทั้งหมดก็ได้ หรือจะตอบคำถามทั้งหมดก็ได้ ถ้าผู้เล่นตอบคำถามถูกทั้งหมดก็จะได้รับรางวัลเป็นอุปกรณ์คอมพิวเตอร์ และได้เลื่อน Rank ขึ้นอีกด้วย หรือถ้าผู้เล่นตอบคำถามไม่ถูกต้องทั้งหมดแต่ผ่านเกณฑ์ที่เราตั้งไว้ก็จะได้เลื่อน Rank แทน



รูปที่ 4.54 แสดงหน้าจอจากเกมตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

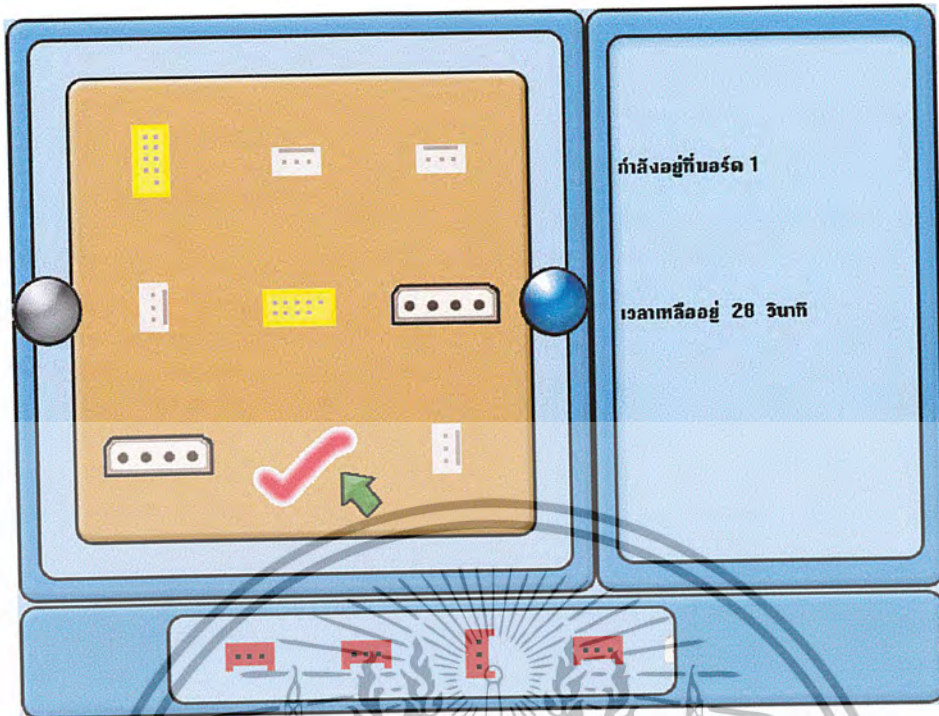
ในส่วนของ Mini Game นั้นในส่วนของ Mini Game นั้นจะมีเกมให้เลือกเล่นทั้งหมด 4 เกมด้วยกัน ซึ่งจะมี 2 เกมที่มีรูปแบบคล้ายในเกมใหญ่ คือเกม ทายเลขสูงต่ำ กับเกมตอบคำถาม และมีอีก 2 เกมดังนี้



รูปที่ 4.55 รูปแสดงหน้าจอการเลือกเล่น Mini Game

1. เกมต่ออุปกรณ์คอมพิวเตอร์

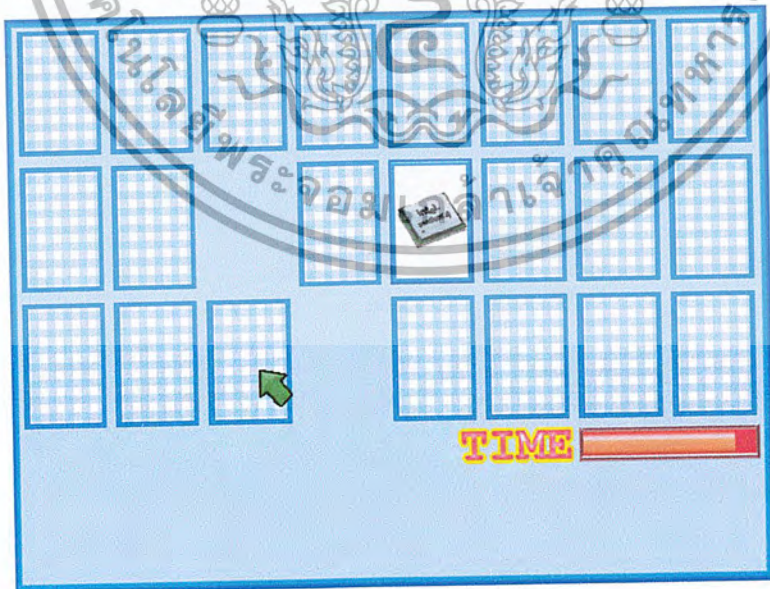
เป็นเกมที่นำ Port ของอุปกรณ์คอมพิวเตอร์มาต่อให้ตรงกันซึ่งเกมนี้เราจะนำไปอยู่ใน Mini Game เพื่อให้ผู้เล่นนั้นได้เลือกเล่นเกมได้ตลอดเวลา โดยจะมีการกำหนดเวลา กำหนดให้มีทั้งหมด 5 บอร์ด ดังนั้นผู้เล่นต้องเล่นให้ชนะให้ทันกับเวลาที่เรากำหนดไว้ถึงจะผ่าน ซึ่งเป็นเกมที่วัดความไวของผู้เล่นว่าสามารถที่จะหมุนรูปแล้วนำไปต่อกับรูปที่กำหนดๆ ไปได้ทันเวลาที่กำหนดไว้หรือเปล่า มีหน้าจอการเล่นดังนี้



รูปที่ 4.56 แสดงหน้าจอเกมต่ออุปกรณ์คอมพิวเตอร์

2. เกมจับคู่

เกมจับคู่นี้มีลักษณะคล้ายกับภายในเกมใหญ่แต่ต่างกันตรงที่ว่าใน Mini Game นี้จะมีเวลาเป็นตัวบังคับ โดยผู้เล่นต้องจับคู่ให้ครบทั้งหมดให้ทันในเวลาที่กำหนด



รูปที่ 4.57 แสดงหน้าจอเกมจับคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบการทดลองเกม

เพื่อให้ทราบผลและสามารถประเมินผลของเกมว่ามีความถูกต้องเพียงใดนั้นจะต้องทดสอบโปรแกรมหลายๆ ครั้ง โดยการลองเล่นเกมหลายๆ รูปแบบ

4.2.1 ลักษณะการคลิกเมาส์

- การคลิกขวา ซึ่งในโปรแกรมเกมนี้ส่วนใหญ่จะใช้การคลิกเมาส์ขวาส่วนใหญ่ ก็เลยลองเปลี่ยนมาคลิกเมาส์ซ้ายแทนว่าจะเกิดปัญหาขึ้นหรือเปล่า

- การคลิกซ้าย ซึ่งในโปรแกรมเกมนั้นจะมีบางเกมที่ใช้การคลิกซ้าย ก็เลยลองเปลี่ยนมาคลิกเมาส์ซ้ายแทนว่าจะเกิดปัญหาขึ้นหรือเปล่า

- การคลิกขวากดคลิกซ้ายพร้อมกัน ดูว่าจะเกิดปัญหาหรือเปล่า

- การคลิกเมาส์ในส่วนที่ไม่ได้กำหนดให้คลิก ดูว่าเกิดปัญหาขึ้นหรือเปล่า

4.2.2 ลักษณะการกดคีย์บอร์ด

- ดูว่าช่วงไหนกำหนดให้กดคีย์บอร์ดแล้วสามารถกดได้หรือเปล่า

- ดูว่าช่วงที่ให้ใช้เมาส์นั้นแล้วถ้าเรากดคีย์บอร์ดจะเกิดอะไรขึ้น

4.3 ผลการทดลอง

เป็นขั้นตอนสุดท้ายของการทำเกม โดยขั้นตอนนี้เป็นการตรวจสอบความผิดพลาดและประสิทธิภาพโดยรวมของเกมทั้งหมด ว่าสามารถโต้ตอบและแสดงผลได้ตามที่ออกแบบไว้หรือไม่ ถ้าตรวจสอบพบความผิดพลาดจะได้นำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขให้มีความถูกต้องมากที่สุด ซึ่งจากการทดลองตามลักษณะดังที่ได้กล่าวมาแล้วนั้นปรากฏว่าไม่มีเหตุการณ์ผิดปกติเกิดขึ้น

สรุปผลการทดลองและข้อเสนอแนะ

การทำงานโครงการนี้มีการวางแผนขั้นตอนการทำงานเป็นอย่างดี แต่ถึงแม้จะได้วางแผนแต่ละขั้นตอนไว้อย่างดีก็ยังมีปัญหาบางขั้นตอน เนื่องจากตอนแรกนั้นเราได้กำหนดการทำงานไว้สองส่วนคือส่วนสร้างไฟล์รูปภาพ กับ ส่วนของเกม ซึ่งเราได้ทำในส่วนของไฟล์รูปภาพไปแล้วแต่ปรากฏว่าไม่ได้ใช้เลยทำให้เสียเวลา และอีกอย่างหนึ่งก็คือ เนื่องจากแยกกันทำคนละส่วนและต่างก็ใช้ตัวแปรเหมือนกันเลยทำให้เวลานำโปรแกรมมารวมกันนั้นเกิดปัญหาขึ้น

โดยขั้นตอนต่างๆ สรุปได้ดังนี้

- ขั้นตอนการออกแบบ Story Board
- ขั้นตอนการเขียนโปรแกรม
- ขั้นตอนการสร้างตัวละครและภาพต่างๆ

5.1 ขั้นตอนการออกแบบ Story Board

เป็นขั้นตอนที่ดำเนินไปค่อนข้างช้าเนื่องจากว่าความคิดเห็นของผู้พัฒนานั้นมีความคิดเห็นไม่ตรงกันเลยทำให้เกิดความล่าช้าเพราะต่างคนก็ต้องการให้เกมนั้นเป็นไปในลักษณะที่ตนเองได้ออกแบบไว้ และการออกแบบเกมที่จะช่วยเป็นสื่อการสอนได้นั้นผู้พัฒนาคนข้างจะทำให้มันมีลักษณะช่วยสื่อการสอนได้นั้นทำได้ยาก

5.2 ขั้นตอนการเขียนโปรแกรม

เป็นขั้นตอนที่ช้าที่สุดเพราะว่าต่างคนก็มีสไตล์การเขียนโปรแกรมไม่เหมือนกันและแต่ละคนก็มักจะใช้ตัวแปรที่เคยใช้กันบ่อย ๆ เช่น I, j เป็นต้นพอนำโปรแกรมของแต่ละคนนั้นมารวมกันก็มักจะเกิดปัญหาที่มีการประกาศตัวแปรซ้ำ และก็รวมกันไม่ได้เพราะว่าการเขียนนั้นไม่ได้เขียนเป็นฟังก์ชันเลยทำให้แต่ละคนนั้นต้องปรับ Source code ให้เป็นในรูปแบบที่เหมือนกันเพื่อช่วยต่อการรวมเป็นโปรแกรมใหญ่

5.3 ขั้นตอนการสร้างตัวละคร

เป็นขั้นตอนที่ค่อนข้างช้าพอสมควรเนื่องจากผู้พัฒนาแต่ละคนนั้นมีลักษณะการใช้สีสັນไม่ค่อยเหมือนกันพอนำภาพมาใช้ในเกมแล้ว ดูเกมมันแปลกๆ เพราะจะไปกันคนละทิศละทาง เลยทำให้ต้องคอยปรับปรุงเปลี่ยนแปลงภาพอยู่ตลอดเวลา และการสร้างภาพในแต่ละภาพนั้นค่อนข้างใช้เวลานานพอสมควร ผู้พัฒนาเลือกรูปภาพมาจากที่อื่นด้วย และรูปภาพที่จะเอามาจากที่อื่นนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่อนข้างเสียเวลาหาพอสมควรเพราะว่าต้องนำรูปภาพที่ตรงกับความต้องการของเกมจริง ๆ เพื่อจะดูให้สมจริง

5.4 ข้อเสนอแนะ และแนวทางในการพัฒนา

ขั้นตอนการสร้างเกมในปัญหาพิเศษนี้ เป็นเพียงต้นแบบแนวทางหนึ่งในการเขียนเกม 2D ซึ่งเป็นแนวสื่อการสอน อาจมีข้อผิดพลาดบางประการ ซึ่งจะนำไปพัฒนา ปรับปรุงให้ดีขึ้นได้ต่อไป

ข้อเสนอแนะในการพัฒนาขั้นตอนการเขียนเกมนั้นมีดังนี้

1. เนื่องจากข้อมูลของ สินค้า, คำแนะนำ รวมถึง คำถามที่มีใน โครงงานยังเป็นแค่ตัวอย่างซึ่งจำนวนน้อย และข้อมูลที่มีอาจมีการผิดพลาดได้บ้าง ดังนั้นทางผู้พัฒนาจึงได้เสนอให้มีการสร้างเครื่องมือที่จะนำไปใช้ในการ เพิ่มเติมและปรับแต่งแก้ไขได้ข้อมูลที่ใช้ในเกมได้ โดยในโปรแกรมที่ได้พัฒนาได้เปิดโอกาสให้สามารถทำในจุดนี้ได้
2. เหตุการณ์ในเกมที่ได้ออกแบบมาเพื่อนำเสนอเป็นเพียงเหตุการณ์ที่มีเงื่อนไขง่าย ๆ ดังนั้นหากต้องการพัฒนาให้สมจริงควรใช้เหตุการณ์ที่มีเงื่อนไขหลากหลาย สามารถเปลี่ยนแปลงได้ตามสถานการณ์แล้วแต่การกระทำของผู้เล่น
3. การบันทึกข้อมูลภายในเกมที่น่าเสนอสามารถทำได้เพียง 1 ผู้เข้าเล่นเท่านั้น ทำให้มีปัญหาในการบันทึกซ้ำซ้อน ดังนั้นจึงเสนอให้มีการสร้างการบันทึกได้หลากหลายผู้เข้าเล่นมากขึ้น
4. จุดมุ่งหมายภายในเกมที่น่าเสนอมีเพียง 3 ชื่อเท่านั้น ดังนั้นหากมีการเพิ่มจุดมุ่งหมายที่มากขึ้นจะทำให้เกมสมจริงมากขึ้น
5. ควรมีการสร้างเครื่องมือเพื่อจัดการกับเหตุการณ์หลักภายในเกม(เช่นการกำหนดการเกิดเกมย่อย) ให้สามารถปรับปรุงแก้ไขได้เพื่อให้สามารถปรับความเหมาะสมของเกมได้

บรรณานุกรม

ยุทธนา ลีลาศวัฒนกุล . 2544 . คู่มือการเขียนโปรแกรมด้วย Visual C++ 6.0 ฉบับโปรแกรมเมอร์ .

กรุงเทพฯ : อินโฟเพรส.

นิรุช อำนวยศิลป์ . 2544 . คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0

กรุงเทพฯ : อินโฟเพรส.

นิรุช อำนวยศิลป์ . 2545 . เขียนเกมอย่างมืออาชีพด้วย Visual C++ และ Direct X .

กรุงเทพฯ : อินโฟเพรส.

ร.ศ. มัณฑนา ปราการสมุทร . 2545 . การเขียนคำสั่งชุดภาษาซี .

กรุงเทพฯ : ดวงกมลสมัย.

พีรภัทร์ สว่างเพ็ชร . 2545 . เทคนิคการเขียนโปรแกรมและเกมด้วย Visual C++ .

กรุงเทพฯ : ซีเอ็ดดูเคชั่น

กิตติ ภัคดีวัฒนะกุล และ จำลอง กรูอดุทธาหะ . 2545 . Visual Basic 6 ฉบับโปรแกรมเมอร์ .

กรุงเทพฯ : DEV BOOK.

สมภพ ตั้งจิตต์วัฒนา . “เกมวางแผนการรบระบบ Interactive” วิทยานิพนธ์วิทยาศาตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ

ทหารลาดกระบัง . 2542 .