

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมช่วยสร้างเกม 2 มิติ

2D GAME CREATOR



คณาจารย์  
สมโภช ศรีสารสกุล  
อนุพงศ์ เลียงมณีกุล

เลขหมู่.....  
เลขทะเบียน.....59404  
วัน,เดือน,ปี..... 2548

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ 1956มต๑

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## **2D GAME CREATOR**



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULLFILLMENT  
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 2005**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ โปรแกรมช่วยสร้างเกม 2 มิติ  
2D GAME CREATOR

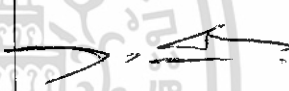

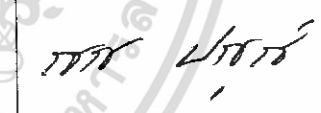
ชื่อนักศึกษา นายธนัญญา เล่งเวหาสถิตย์ 45050481  
นายสมโภช ศรีสารสกุล 45050527  
นายอนุพงศ์ เลี้ยงมณีกุล 45050542

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2548

อาจารย์ที่ปรึกษา ผศ.ดร.กรกช ประชุมรักษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2548

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ผศ.กฤษณา บุศรา	
กรรมการ	อ.วีระ พิกอ่อน	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.ดร.กรกช ประชุมรักษ์	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมช่วยสร้างเกม 2 มิติ		
ชื่อนักศึกษา	นายชนนญา	เล่งเวทาสติชัย	45050481
	นายสมโภช	ศรีสารสกุล	45050527
	นายอนุพงศ์	เลี่ยมฉวีกุล	45050542
ปริญญา	วิทยาศาสตรบัณฑิต		
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2548		
อาจารย์ที่ปรึกษา	ผศ.ดร.กรรช ประชุมรัมย์		

### บทคัดย่อ

โปรแกรมช่วยสร้างเกม 2 มิติ เป็นโปรแกรมที่ใช้สร้างเกมคอมพิวเตอร์ พัฒนาขึ้นมาด้วยภาษาซี พลัส พลัส ไคเร็กเอ็กซ์ เอ็นจิน และ ซีดีเอ็กซ์ เอ็นจิน โดยที่ผู้ใช้งานโปรแกรมไม่จำเป็นต้องเขียนโปรแกรมแต่อย่างใด ประเภทของเกมที่สามารถสร้างได้ ได้แก่ เกมตะลุมดำน เกมต่อสู้ และเกมยานบินตะลุมดำนอวกาศ

<b>Special Project Title</b>	2D GAME CREATOR	
<b>Students</b>	Mr.Thanasada Lengwehasatit	45050481
	Mr.Somphot Srisarnsakul	45050527
	Mr.Anupong Liangmaneekul	45050542
<b>Degree</b>	Bachelor of Science	
<b>Department</b>	Mathematics and Computer Science, Faculty of Science	
<b>Programme</b>	Computer Science	
<b>Academic Year</b>	2005	
<b>Special Project Adviser</b>	Assistant Professor Doctor Korakot Prachumrak	

### ABSTRACT

The 2D Game Creator is an application used for creating 2D computer game. It's developed using C++ language, DirectX engine and CDX engine. Users can create new games using this 2D Game Creator without any programming. These games include an action game, a fighting game and a shooting game.

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง โปรแกรมช่วยสร้างเกม 2 มิติ สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์คือกเตอร์ กรกช ประชุมรัมย์ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณบิดามารดา ที่ได้ให้ความสนับสนุนด้านกำลัง และทุนทรัพย์จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี โดยเฉพาะพี่ ๆ ที่ให้คำปรึกษา ชมรมอินเทอร์เน็ตที่เอื้อเฟื้อสถานที่และคอมพิวเตอร์ในการพัฒนาปัญหาพิเศษ รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคน ที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ  
มีนาคม 2549



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา.....	1
1.4 ขอบเขตของการศึกษา.....	1
1.5 ขั้นตอนการศึกษา.....	2
1.6 การใช้งาน.....	2
1.7 ผลที่คาดว่าจะได้รับ.....	2
1.8 เกมที่สามารถสร้างได้.....	2
บทที่ 2 งานวิจัยที่เกี่ยวข้อง.....	4
2.1 การเขียนโปรแกรมด้วยภาษาซี พลัส พลัส (C++) พื้นฐานโดยใช้โปรแกรม วิชวล ซี พลัส พลัส ( Visual C++ ).....	4
2.1.1 โปรเจกต์เวิร์กสเปซ (Project Work Space) .....	4
2.1.1.1 การจัดการโปรเจกต์เวิร์กสเปซ.....	4
2.1.1.2 การบันทึกและการโหลดโปรเจกต์เวิร์กสเปซ.....	5
2.1.1.3 คอมไพเลอร์และลิงก์เกอร์ (Compiler & Linker) .....	6
2.1.2 การตั้งค่าเอ็มเอฟซี ใน วิน 32 แอปพลิเคชัน.....	6
2.1.3 การเขียนโปรแกรมบน วิน โดวส์.....	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

2.1.4 ระบบข่าวสารในวินโดวส์.....	13
2.2 การใช้งานไดเร็กเอ็กซ์ (Direct X) เบื้องต้น.....	16
2.2.1 ความเป็นมาเกี่ยวกับโปรแกรมไมโครซอฟท์ไดเร็กเอ็กซ์ (Microsoft Direct X).....	16
2.2.2 การสร้างโปรเจคใหม่.....	17
2.2.3 การสร้างหน้าต่างเกมโดยใช้ภาษาซี พลัส พลัส.....	19
2.2.4 การสร้างหน้าต่างเกมโดยใช้ไดเร็กเอ็กซ์.....	21
2.2.4.1 การขอใช้ไลบรารีของไดเร็กเอ็กซ์.....	21
2.2.4.2 ส่วนของชุดคำสั่งโปรแกรม.....	22
2.2.5 การแสดงรูปภาพบนเซอร์เฟส.....	23
2.2.6 การใส่เสียงดนตรี.....	27
2.2.7 การใส่เสียงประกอบ.....	28
2.3 เกมเอ็นจิน ซีดี เอ็กซ์ (Game Engine). ....	29
2.3.1 การทำดับเบิลบัฟเฟอร์ (Double Buffer).....	29
2.3.2 การโต้ตอบกับแป้นพิมพ์.....	30
2.3.3 การโต้ตอบกับเมาส์.....	30
2.4 ทฤษฎีการคำนวณในฟิสิกส์.....	31
<b>บทที่ 3 ขั้นตอนการดำเนินงานวิจัย.....</b>	<b>32</b>
3.1 ขั้นตอนการออกแบบ.....	33
3.1.1 การออกแบบระบบสร้างเกมและฟังก์ชันต่าง ๆ ที่ช่วยสนับสนุนผู้ใช้ในการสร้างเกม.....	33
3.1.2 การออกแบบตัวละครต้นแบบและฉากต่าง ๆ ที่มีให้ผู้ใช้ได้เลือกในโปรแกรม.....	33
3.1.3 การออกแบบคลาสต่างๆ ในระบบ.....	34
3.1.4 โครงสร้างและการทำงานของโปรแกรม.....	38
3.2 การออกแบบอินเตอร์เฟซของโปรแกรม.....	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

### หน้า

3.2.1 ส่วนของโปรแกรมหลัก.....	43
3.2.2 ส่วนของการสร้างพื้นที่เก็บไฟล์เกม.....	44
3.2.3 ส่วนของการเพิ่มและลบวัตถุ.....	44
3.2.3.1 ส่วนของการเพิ่มวัตถุ.....	44
3.2.3.2 ส่วนของลบวัตถุ.....	48
3.2.4 ส่วนของหน้าต่างสร้างเหตุการณ์.....	50
3.2.4.1 ส่วนของเงื่อนไขเกม ( Event ).....	51
3.2.4.2 ส่วนของคำสั่ง( Action ).....	54
3.2.5 ขั้นตอนการเขียนโปรแกรม.....	69
<b>บทที่ 4 การพัฒนาและทดสอบโปรแกรม.....</b>	<b>70</b>
4.1 การพัฒนาชุดคำสั่งเกมเอ็นจิน (Game Engine).....	70
4.2 การพัฒนาโปรแกรมช่วยสร้างเกม 2 มิติ (2D Game Creator).....	76
4.2.1 ค่าคงที่ ตัวแปร และชุดคำสั่งเกี่ยวกับการจัดการข้อมูลในอะเรย์.....	76
4.2.1.1 ประกาศค่าคงที่ไว้อ้างอิงอะเรย์.....	77
4.2.1.2 ประกาศชุดคำสั่งสำหรับจัดการอะเรย์.....	77
4.2.1.3 ตัวอย่างการเรียกใช้ชุดคำสั่งจัดการข้อมูลในอะเรย์.....	80
4.3 การพัฒนาคำสั่งด้านฟิสิกส์.....	83
4.4 การทดสอบโปรแกรม.....	84
4.4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็น.....	84
4.4.2 ผลการทดลอง.....	85
<b>บทที่ 5 สรุปผลวิจัยและข้อเสนอแนะ.....</b>	<b>86</b>
5.1 สรุปผลการดำเนินงาน.....	86
5.2 ข้อจำกัดของโปรแกรม.....	86
5.3 ปัญหาที่เกิดขึ้น.....	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
5.3.1 ขั้นตอนการออกแบบโครงสร้างของโปรแกรม. ....	87
5.3.2 ขั้นตอนการเขียนโปรแกรม.. ....	87
5.3.3 ขั้นตอนการทดสอบโปรแกรม. ....	88
5.4 ผลการเปรียบเทียบโปรแกรมช่วยสร้างเกม 2 มิติ กับ โปรแกรมเกมเมกเกอร์ .....	88
5.5 ข้อเสนอแนะเพื่อเป็นแนวทางสำหรับผู้ที่ต้องการพัฒนาเกมนี้ต่อไป.....	88
<b>บรรณานุกรม.....</b>	<b>90</b>
ภาคผนวก ก การติดตั้งโปรแกรมไดเรกต์เอ็กซ์ เอส ดี เค (Direct X SDK).....	91
ภาคผนวก ข การติดตั้งโปรแกรมซี ดี เอ็กซ์ (CDX).....	96
ภาคผนวก ค การปรับแต่งให้โปรแกรม วิซวล ซี พลัส พลัส (Visual C++) สามารถใช้ งานไดเรกต์เอ็กซ์ (Direct X). ....	100
ภาคผนวก ง การปรับแต่งโปรแกรมวิซวล ซี พลัส พลัส (Visual C++) สามารถใช้งาน ซี ดี เอ็กซ์ (CDX).....	105
ภาคผนวก จ ตัวอย่างการสร้างเกมขั้นพื้นฐาน .....	107
ภาคผนวก ฉ ตัวอย่างการสร้างเกมแบบซับซ้อน.....	117

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3-1 ตารางแสดงรายละเอียดของคลาสเบ็คกราวน์.....	35
ตารางที่ 3-2 ตารางแสดงรายละเอียดของคลาสมิวสิก.....	35
ตารางที่ 3-3 ตารางแสดงรายละเอียดของคลาสอ็อบเจ็ค.. ..	36
ตารางที่ 3-4 ตารางแสดงรายละเอียดของคลาสชวาร์น็อล์ฟเฟ็ค. ....	37
ตารางที่ 3-5 ตารางแสดงการเพิ่มคำสั่งลงตัวแปรอะเรย์ต่างเมื่อมีการเพิ่มวัตถุ.....	47
ตารางที่ 3-6 ตารางแสดงการลบชื่อวัตถุในอะเรย์ต่างเมื่อมีการลบวัตถุ.....	49
ตารางที่ 3-7 ตารางแสดงคำสั่งที่เก็บลงอะเรย์ arr_action เมื่อมีการกดปุ่มเหตุการณ์ใด..	53
ตารางที่ 3-8 ตารางแสดงคำสั่งที่เก็บลงอะเรย์ arr_action เมื่อมีการเลือก การกระทำใดๆ.....	66
ตารางที่ 3-9 แสดงไฟล์ที่เกี่ยวข้องกับหน้าต่าง โปรแกรมทั้งหมด.. ..	67
ตารางที่ 5-1 ตารางแสดงการเปรียบเทียบระหว่างโปรแกรมช่วยสร้างเกม 2 มิติ กับ โปรแกรมเกมเมกเกอร์.....	88
ตารางที่ ฉ-1 ตารางแสดงรายละเอียดของตัวละครในเกม.....	124
ตารางที่ ฉ-2 ตารางแสดงการเพิ่มคำสั่งต่าง ๆ ในหน้าต่างเหตุการณ์ .....	128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
รูปที่ 2-1 การจัดการ โปรเจกต์เวิร์กสเปซ.....	5
รูปที่ 2-2 การตั้งค่าเอ็มเอฟซี ใน วิน 32 แอปพลิเคชัน.....	7
รูปที่ 2-3 แผนภาพแสดงการทำงาน อีเวนต์ลูป ของภาษาซี พลัส พลัส.....	11
รูปที่ 2-4 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 1.....	17
รูปที่ 2-5 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 2.....	17
รูปที่ 2-6 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 3.....	18
รูปที่ 2-7 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 4.....	18
รูปที่ 2-8 แสดงการขอใช้ไลบรารีของโคเร็กเอ็กซ์ ขั้นตอนที่ 1.....	21
รูปที่ 2-9 แสดงการขอใช้ไลบรารีของโคเร็กเอ็กซ์ ขั้นตอนที่ 2.....	21
รูปที่ 3-1 ขั้นตอนการวางแผน.....	32
รูปที่ 3-2 แผนภาพแสดงแอตทริบิวต์และฟังก์ชันของคลาสต่างๆ .....	34
รูปที่ 3-3 แผนภาพแสดงการจัดการข้อมูลภายในโปรแกรม.....	39
รูปที่ 3-4 แสดงแผนภาพการทำงานของโปรแกรม.....	42
รูปที่ 3-5 แสดงหน้าต่างหลักของโปรแกรมช่วยสร้างเกม.....	43
รูปที่ 3-6 แสดงหน้าต่างการเลือกตำแหน่งการเก็บไฟล์เกม.....	44
รูปที่ 3-7 แสดงหน้าต่างการเพิ่มตัวละคร.....	44
รูปที่ 3-8 แสดงหน้าต่างการเพิ่มพื้นหลัง.....	45
รูปที่ 3-9 แสดงหน้าต่างการเพิ่มเสียงประกอบ.....	46
รูปที่ 3-10 แสดงหน้าต่างการเพิ่มเสียงเพลง.....	47
รูปที่ 3-11 แสดงหน้าต่างการลบชื่อตัวละคร.....	48
รูปที่ 3-12 แสดงหน้าต่างการลบชื่อเสียง.....	48
รูปที่ 3-13 แสดงหน้าต่างการลบชื่อเพลง.....	49
รูปที่ 3-14 ภาพแสดงหน้าต่างสร้างเหตุการณ์เกม.....	50
รูปที่ 3-15 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการกดปุ่มเป็นพิมพ์.....	51
รูปที่ 3-16 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการกดคลิกเมาส์.....	52
รูปที่ 3-17 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการชนกันของวัตถุ 2 ตัว.....	52
รูปที่ 3-18 แสดงหน้าต่างการเคลื่อนที่ของตัวละคร.....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3-19 แสดงหน้าตาการย้ายที่ของตัวละคร.....	54
รูปที่ 3-20 แสดงหน้าตาการย้ายตัวละคร.....	55
รูปที่ 3-21 แสดงหน้าตาการกำหนดค่าพลังชีวิตของตัวละคร.....	55
รูปที่ 3-22 แสดงหน้าตาการกำหนดจำนวนชีวิตของตัวละคร.....	56
รูปที่ 3-23 แสดงหน้าตาการกำหนดคะแนนของตัวละคร.....	56
รูปที่ 3-24 แสดงหน้าตาการเพิ่มพลังชีวิตให้กับตัวละคร.....	57
รูปที่ 3-25 แสดงหน้าตาการเพิ่มจำนวนชีวิตตัวละคร.....	57
รูปที่ 3-26 แสดงหน้าตาการเพิ่มคะแนน.....	58
รูปที่ 3-27 แสดงหน้าตาการลดพลังชีวิตของตัวละคร.....	58
รูปที่ 3-28 แสดงหน้าตาการลดจำนวนชีวิตของตัวละคร.....	59
รูปที่ 3-29 แสดงหน้าตาการลดคะแนน.....	59
รูปที่ 3-30 แสดงหน้าตาการแสดงผลพลังชีวิตบนหน้าจอเกม.....	60
รูปที่ 3-31 แสดงหน้าตาการแสดงผลจำนวนชีวิตบนหน้าจอเกม.....	60
รูปที่ 3-32 แสดงหน้าตาการแสดงผลคะแนนบนหน้าจอเกม.....	61
รูปที่ 3-33 แสดงหน้าตาการกำหนดการปรากฏ.....	61
รูปที่ 3-34 แสดงหน้าตาการลบตัวละคร.....	62
รูปที่ 3-35 แสดงหน้าตาการเปลี่ยนรูปตัวละคร.....	62
รูปที่ 3-36 แสดงหน้าตาการเลือกเล่นเสียงเพลงประกอบ.....	63
รูปที่ 3-37 แสดงหน้าตาการเลือกเล่นเสียงดนตรี.....	63
รูปที่ 3-38 แสดงหน้าตาการปิดเสียงเพลง.....	64
รูปที่ 3-39 แสดงหน้าตาการปรับความดังของเสียงดนตรี.....	64
รูปที่ 3-40 แสดงหน้าตาการเปลี่ยนฉาก.....	65
รูปที่ 3-41 แสดงหน้าตาการเลื่อนและกำหนดความเร็วของฉาก.....	65
รูปที่ 3-42 แผนภาพแสดงขั้นตอนการดำเนินงาน.....	69
รูปที่ 4-1 แสดงการรับข้อมูลการสร้างตัวละครตัวใหม่.....	80
รูปที่ ก-1 แสดงขั้นตอนการติดตั้งไคลเอนต์ไอที เอส ดี เก ชั้นตอนที่ 1.....	92
รูปที่ ก-2 แสดงขั้นตอนการติดตั้งไคลเอนต์ไอที เอส ดี เค ชั้นตอนที่ 2.....	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ ก-3 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ชั้นตอนที่ 3.....	93
รูปที่ ก-4 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ชั้นตอนที่ 4.....	93
รูปที่ ก-5 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ชั้นตอนที่ 5.....	94
รูปที่ ก-6 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ชั้นตอนที่ 6.....	94
รูปที่ ก-7 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ชั้นตอนที่ 7.....	95
รูปที่ ข-1 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ชั้นตอนที่ 1.....	97
รูปที่ ข-2 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ชั้นตอนที่ 2.....	97
รูปที่ ข-3 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ชั้นตอนที่ 3.....	98
รูปที่ ข-4 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ชั้นตอนที่ 4.....	98
รูปที่ ข-5 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ชั้นตอนที่ 5.....	99
รูปที่ ค-1 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 1.....	101
รูปที่ ค-2 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 2.....	101
รูปที่ ค-3 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 3.....	102
รูปที่ ค-4 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 4.....	102
รูปที่ ค-5 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 5.....	103
รูปที่ ค-6 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 6.....	103
รูปที่ ค-7 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ชั้นตอนที่ 7.....	104
รูปที่ ง-1 แสดงขั้นตอนการทำให้ วิววลซี พลัส พลัส รูจิก ซี ดี เอ็กซ์ ชั้นตอนที่ 1.....	106
รูปที่ ง-2 แสดงขั้นตอนการทำให้ วิววลซี พลัส พลัส รูจิก ซี ดี เอ็กซ์ ชั้นตอนที่ 2.....	106
รูปที่ จ-1 แสดงภาพที่ใช้ในการทำตัวอย่างเกมชั้นตอนที่ .....	108
รูปที่ จ-2 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 1.....	109
รูปที่ จ-3 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 2.....	109
รูปที่ จ-4 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 3.....	110
รูปที่ จ-5 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 4.....	110
รูปที่ จ-6 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 5.....	111
รูปที่ จ-7 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 6.....	111
รูปที่ จ-8 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 7.....	112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ จ-9 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 8.....	112
รูปที่ จ-10 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 9.....	113
รูปที่ จ-11 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 10.....	113
รูปที่ จ-12 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 11.....	114
รูปที่ จ-13 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 12.....	114
รูปที่ จ-14 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 13.....	115
รูปที่ จ-15 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 14.....	115
รูปที่ จ-16 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 15.....	116
รูปที่ จ-17 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 16.....	116
รูปที่ จ-18 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 17.....	117
รูปที่ จ-19 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 18.....	117
รูปที่ จ-20 แสดงตัวอย่างการสร้างเกม ชั้นตอนที่ 19.....	118
รูปที่ จ-21 แสดงภาพผลลัพธ์จากการประมวลผลเกม.....	118
รูปที่ จ-22 แสดงภาพผลลัพธ์จากการเล่นเกม.....	119
รูปที่ ฉ-1 แสดงเกมตัวอย่างแบบซับซ้อน.....	121
รูปที่ ฉ-2 แสดงรูปตัวละครที่ใช้ในเกม.....	122
รูปที่ ฉ-3 แสดงการเพิ่มวัตถุหรือตัวละครลงในเกม.....	123
รูปที่ ฉ-4 แสดงการเพิ่มฉากหลัง.....	125
รูปที่ ฉ-5 แสดงการเพิ่มเสียงประกอบ.....	126
รูปที่ ฉ-6 แสดงการเพิ่มเสียงดนตรี.....	126
รูปที่ ฉ-7 แสดงหน้าต่างเพิ่มเหตุการณ์และคำสั่ง.....	127
รูปที่ ฉ-8 แสดงการใส่คำสั่งเล่นเสียงดนตรี.....	127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้ อุตสาหกรรมซอฟต์แวร์เกมในประเทศไทยกำลังขยายตัวไปอย่างรวดเร็ว อีกทั้งรัฐบาลและหน่วยงานต่างๆ ยังมีนโยบายสนับสนุน ส่งเสริม ให้มีการผลิตบุคลากรทางด้านการพัฒนาเกม ซึ่งมีผู้สนใจเป็นอย่างมาก แต่หากว่ายังมีผู้ที่สนใจทั้งหลายเหล่านั้นเป็นจำนวนมากที่ยังขาดความรู้และความเข้าใจในแนวทางที่จะเริ่มต้นพัฒนาเกม เราจึงได้ศึกษา ค้นคว้า วิธีการเขียนโปรแกรมช่วยสร้างเกมขึ้นมา เพื่อให้โปรแกรมนี้เป็นโอเพนซอร์สของคนไทยโดยเฉพาะ ผู้ใช้งานจึงไม่จำเป็นต้องเสียเงินซื้อซอฟต์แวร์ช่วยสร้างเกมที่เป็นของต่างประเทศ เนื่องจากเป็นโปรแกรมที่สามารถดาวน์โหลดได้โดยไม่เสียค่าใช้จ่ายใดๆ

โปรแกรมช่วยสร้างเกมเป็นโปรแกรมที่ใช้สำหรับสร้างสรรค์เกมใหม่ ๆ ขึ้นมาได้โดยที่ไม่จำเป็นต้องเขียนโปรแกรมทั้งหมดด้วยตนเอง ผู้ใช้เพียงแต่มีความคิดสร้างสรรค์และมีความรู้ทางด้านคอมพิวเตอร์เพียงเล็กน้อย ก็สามารถสร้างเกมของตนเองออกมาได้

โปรแกรมนี้อจะเป็นอีกทางเลือกหนึ่ง ให้ผู้ที่เริ่มต้นและสนใจศึกษา เรียนรู้ วิธีการทำเกมแบบง่ายๆ ได้เพื่อเป็นแนวทางที่จะพัฒนาเกมในระดับสูงขึ้น

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ศึกษาการเขียนโปรแกรมด้วยวิซวล ซี พลัส พลัส และ ไคเร็กเอ็กซ์ เอ็นจิน เพื่อเป็นแนวทางในการพัฒนาโปรแกรมช่วยสร้างเกมขึ้น
2. เพื่อเป็นประโยชน์สำหรับผู้เริ่มต้นและสนใจที่จะพัฒนาเกมในระดับสูงขึ้น
3. ช่วยเสริมสร้างความรู้ ความคิดสร้างสรรค์ และความบันเทิง ให้กับเยาวชนไทย
4. เพื่อส่งเสริมให้คนไทย ให้ความสำคัญกับการพัฒนาเกมมากยิ่งขึ้น

### 1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา

พัฒนาแอปพลิเคชันที่อำนวยความสะดวกต่อผู้สร้างเกม ให้สร้างเกมได้ง่ายขึ้น โดยการเรียกใช้คอมโพเนนต์ต่าง ๆ เพียงแค่ใช้เมาส์เลือกคอมโพเนนต์ที่ต้องการ ซึ่งโปรแกรมจะสร้างชุดคำสั่งเกมให้โดยอัตโนมัติ

ภายในชุดคำสั่งที่สร้างขึ้นนั้นจะนำเอาทฤษฎีคอมพิวเตอร์กราฟิกโดยใช้ไคเร็กเอ็กซ์ซึ่งเป็นเครื่องมือของบริษัท ไมโครซอฟท์ที่พัฒนาขึ้นมาสำหรับการสร้างเกมโดยเฉพาะ

## 1.4 ขอบเขตของการศึกษา

ศึกษาการเขียนโปรแกรมด้วยภาษาซี พลัส พลัส ,โคเร็กเอ็กซ์ และซีดีเอ็กซ์ เอ็นจิน เบื้องต้น เช่น การสร้างหน้าต่างโปรแกรม การสร้างหน้าต่างเกม การโหลดรูปภาพ การทำ ภาพเคลื่อนไหว การชนกันของวัตถุ การใส่เสียง การคอมไพล์ไฟล์ การควบคุมปุ่มด้วยเมาส์และ แป้นพิมพ์ และการเรียกใช้งานฟังก์ชันในไลบรารีต่างๆ

## 1.5 ขั้นตอนการศึกษา

1. ศึกษาการเขียนโปรแกรมโดยใช้ภาษาซี พลัส พลัส และโคเร็กเอ็กซ์
2. ศึกษาวิธีใช้งานโปรแกรมเกมเมกเกอร์
3. ออกแบบโปรแกรม
4. พัฒนาโปรแกรม
5. ทดสอบการใช้งานโปรแกรม
6. แก้ไขปรับปรุงเพิ่มเติม
7. จัดทำคู่มือการใช้งานและเอกสารประกอบ

## 1.6 การใช้งาน

ผู้ใช้งานสามารถใช้โปรแกรมช่วยสร้างเกม 2 มิติ ในการสร้างเกมขึ้นเองได้โดยง่าย ไม่ สลับซับซ้อน เพราะมีฟังก์ชันที่ใช้ในการสร้างอย่างง่ายไม่ยุ่งยาก เพียงแค่คลิกเท่านั้น

## 1.7 ผลที่คาดว่าจะได้รับ

ผู้ใช้งานจะสามารถสร้างเกมขึ้นเองได้อย่างรวดเร็ว และได้รับความสนุกเพลิดเพลินไปกับการสร้างเกม

## 1.8 เกมที่สามารถสร้างได้

โปรแกรมช่วยสร้างเกมจะสนับสนุนประเภทของเกมที่จะสร้าง 2 ประเภทคือ เกมชู้ตติง (shooting) และเกมแอดเวนเจอร์ (adventure)

ลักษณะของเกมแนวชู้ตติง เช่น เกมยิงยาน เกมพินบอล เป็นเกมที่มีส่วนประกอบหลักคือ ตัวละครฝ่ายเราและฝ่ายศัตรู จาก และมีเหตุการณ์ต่าง ๆ (event) คือ การเคลื่อนที่ การยิง การชน การระเบิด

ลักษณะของเกมแนวแอตเวนเจอร์ เช่น เกมมาริโอ เป็นเกมที่มีส่วนประกอบหลักคือ ตัวละครฝ่ายเราและฝ่ายศัตรู ฉาก สิ่งของ และมีเหตุการณ์ต่าง ๆ (event) คือ การเดิน การวิ่ง การกระโดด การชน การยิง การปล่อยพลัง การตายของตัวละคร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 งานวิจัยที่เกี่ยวข้อง

### 2.1 การเขียนโปรแกรมด้วยภาษาซี พลัส พลัส (C++) พื้นฐานโดยใช้โปรแกรมวิชวล ซี พลัส พลัส ( Visual C++)

#### 2.1.1 โปรเจกต์เวิร์กสเปซ (Project Work Space)

เมื่อเราจะทำการสร้างโปรแกรมโดยใช้ วิชวล ซี พลัสพลัส อันดับแรกที่เราต้องทำก็คือการสร้างโปรเจกต์เวิร์กสเปซซึ่งเป็นการกำหนดพื้นที่ในการเก็บโปรเจกต์ที่ต้องการสร้างและกำหนดตัวเลือกต่าง ๆ การใช้งานจะมีลักษณะการทำงานเหมือนโปรเจกต์ไฟล์

โปรเจกต์ไฟล์ของ วิชวล ซี พลัสพลัส เราจะเรียกว่า “เวิร์กสเปซ” โดยจะใช้นามสกุล คือ อดดับเบิลยู (.dsw) เป็นไฟล์ที่เก็บตัวเลือกต่างๆ ของโปรเจกต์ ประโยชน์ของโปรเจกต์ไฟล์ก็คือ

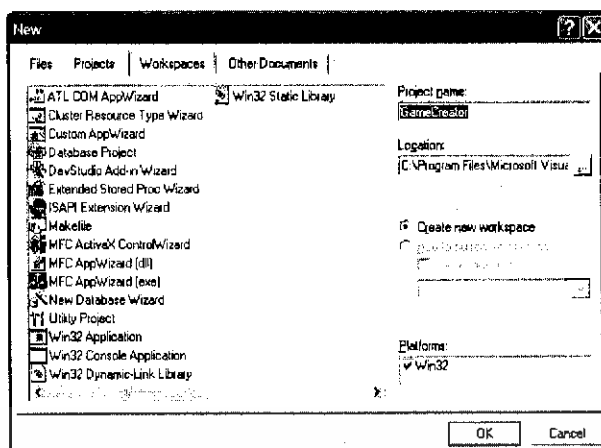
1. โปรเจกต์ไฟล์จะเก็บรายชื่อของไฟล์ที่เป็นซอร์ส โคดโปรแกรมทั้งหมด ที่ใช้ร่วมกันในโปรเจกต์ เช่น ไฟล์นามสกุล เอช (.h) และ ซีพีพี (.cpp) รวมทั้งไฟล์ฐานข้อมูลโปรแกรมที่ใช้ในคลาสวิซาร์ด (Class Wizard) เป็นต้น

2. โปรเจกต์ไฟล์จะเก็บค่าตัวเลือกสำหรับการประมวลผลชุดคำสั่งและการลิงค์เอาไว้เพื่อบอกให้รู้ว่าโปรเจกต์นี้จะทำการประมวลผลชุดคำสั่งและลิงค์กับไลบรารีใด หรือมีการสร้างส่วนประกอบอื่น ๆ อีกหรือไม่ เช่น ส่วนประกอบในการดีบั๊ก

3. โปรเจกต์ไฟล์จะเก็บค่าตัวเลือกที่แสดงว่าโปรเจกต์นี้เป็นโปรเจกต์แบบใดเมื่อทำการประมวลผลชุดคำสั่ง เช่น วินโดวส์ แอปพลิเคชัน (.EXE) หรือ ไดนามิค ลิงค์ ไลบรารี (.DLL) เป็นต้น

##### 2.1.1.1 การจัดการโปรเจกต์เวิร์กสเปซ

ในการเริ่มต้นเขียนโปรแกรมด้วย วิชวล ซี พลัส พลัส ทุกครั้ง เราจะต้องทำการสร้างโปรเจกต์เวิร์กสเปซขึ้นมาก่อน สามารถเลือกได้ว่าจะสร้างโปรเจกต์แบบใด ซึ่งมีให้เลือกหลายแบบ เราเลือกคำสั่ง File > New



รูปที่ 2-1 การจัดการโปรเจกต์เวิร์กสเปซ

การสร้างโปรเจกต์เวิร์กสเปซแต่ละแบบจะมีลักษณะที่ต่างกัน หากเราต้องการเขียนโปรแกรมบนวินโดวส์ เราจะต้องเลือกตัวเลือก เอ็มเอฟซี วิศวาร์ด(MFC Wizard) หรือ วิน32 แอปพลิเคชัน (Win32 Application)

ภายในโปรเจกต์เวิร์กสเปซประกอบด้วยแท็บซ้อนกัน 3 แท็บ คือ คลาสวิว(ClassView), ทรัพยากร วิว (Resource View) และ ไฟล์ วิว (FileView) ซึ่งโดยปกติแล้ว ถ้ายังไม่มีการสร้างและแก้ไขโปรเจกต์ใด ๆ จะไม่มีแท็บใดปรากฏเลย

- คลาสวิว จะใช้สำหรับแสดงรายละเอียดภายในของคลาสต่าง ๆ ที่อยู่ในโปรเจกต์นี้ เราสามารถทราบได้ว่ามีคลาสอะไรบ้างในโปรเจกต์ และสมาชิกของคลาสแต่ละตัวเป็นแบบ โพรเท็ค (Protect) หรือ พับลิก (Public) และตัวแปรโกลบอลมีอะไรบ้าง
- ทรัพยากร วิว ใช้สำหรับแสดงถึงทรัพยากรที่มีในโปรเจกต์
- ไฟล์ วิว ใช้สำหรับแสดงรายชื่อไฟล์ที่อยู่ในโปรเจกต์ขณะนั้น โดยจะแยกไฟล์ส่วนหัว นามสกุล เฮช (.h) ไฟล์โปรแกรมนามสกุล ซีพีพี (.cpp) และ ไฟล์ทรัพยากร

### 2.1.1.2 การบันทึกและการโหลดโปรเจกต์เวิร์กสเปซ

เมื่อเราทำการแก้ไขโปรแกรม เราสามารถบันทึกข้อมูลลงดิสก์ได้โดยเลือกคำสั่ง File > Save ซึ่งจะเป็นการบันทึกไฟล์ที่แก้ไขอยู่ หรือเราสามารถเลือกคำสั่ง File>Save Workspace เพื่อบันทึกการแก้ไขทุกอย่างในโปรเจกต์

สำหรับการเปิดโปรเจกต์เก่าขึ้นมาแก้ไข ทำได้โดยเรียกคำสั่ง Open Workspace ให้เปิดไฟล์ที่มีนามสกุลดีเอสดับเบิลยู (.dsw)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.1.3 คอมไพเลอร์และลิงก์เกอร์ (Compiler & Linker)

การพัฒนาโปรแกรมบนวินโดวส์นั้นจะต่างจากการพัฒนาโปรแกรมในระบบดอส (Dos) เช่น การประมวลผลชุดคำสั่งซอร์สโค้ด และการประมวลผลชุดคำสั่งรีซอร์ส เพราะฉะนั้นเครื่องมือหลัก ๆ ที่ใช้ในกระบวนการประมวลผลชุดคำสั่ง (Build) ของ วิซวล ซี พลัสพลัส มีดังนี้

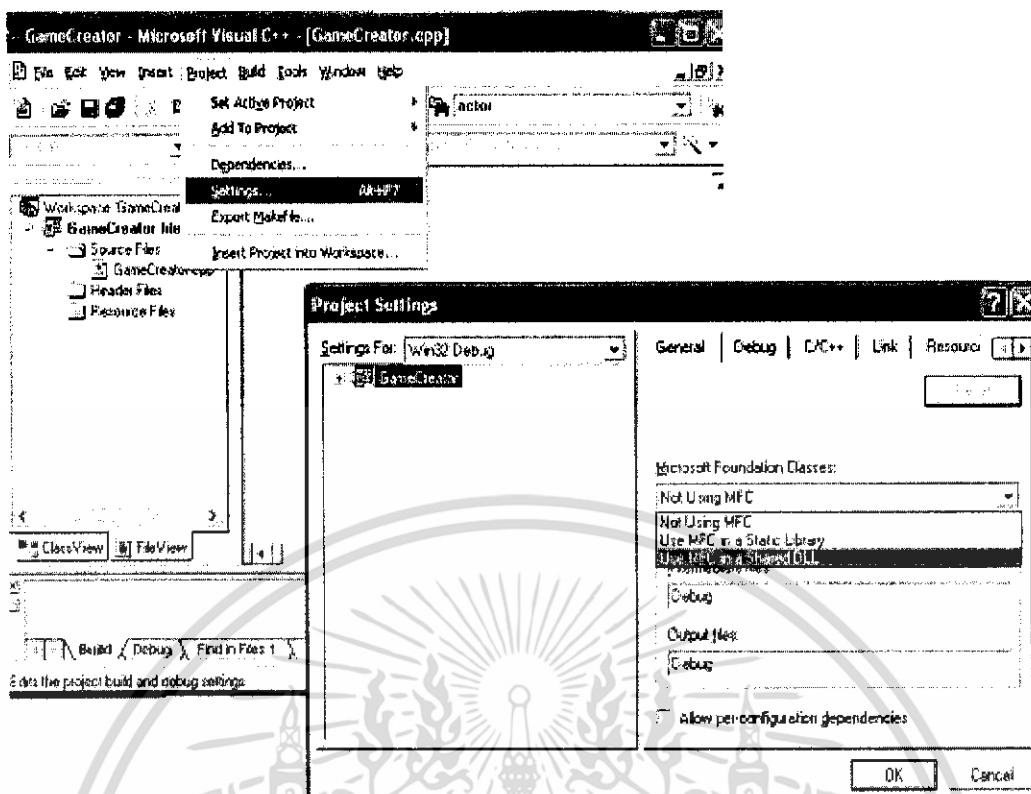
- ซี/ซี พลัส พลัส คอมไพเลอร์ เป็นคอมไพเลอร์ภาษา ซี/ซีพลัสพลัส ของ วิซวล ซี พลัสพลัส ซึ่งจะเป็โปรแกรมตรวจสอบข้อผิดพลาดของซอร์สโค้ด
- รีซอร์สคอมไพเลอร์ เป็นตัวแปลภาษาของรีซอร์สสคริปต์ เพราะในการพัฒนาโปรแกรมบนวินโดวส์ จะต้องมีการสร้างรีซอร์ส เช่น ไอคอน รูปภาพ หรือ ไดอะล็อก รีซอร์สคอมไพเลอร์ตัวนี้จะเป็นโปรแกรมที่ตรวจเช็คความถูกต้องของรีซอร์ส ผลที่ได้จากการประมวลผลชุดคำสั่งก็คือ ออบเจกต์ไฟล์ของรีซอร์ส (.res)
- ลิงก์เกอร์ ลิงก์เกอร์ของ วิซวล ซี พลัส พลัส จะเป็นโปรแกรมที่ทำหน้าที่รวบรวมไฟล์ประเภทออบเจกต์ที่ได้จากการประมวลผลชุดคำสั่งออบเจกต์ของรีซอร์สไฟล์ และไลบรารีไฟล์มาสร้างเป็นไฟล์ที่สามารถใช้งานได้ทันที (Execute File)

ขั้นตอนการประมวลผลชุดคำสั่งโปรแกรมของ วิซวล ซี พลัส พลัส นั้น จะเริ่มต้นที่การประมวลผลชุดคำสั่งรีซอร์สก่อน จากนั้นคอมไพเลอร์จะทำการแปลภาษาและสร้างออบเจกต์ไฟล์ขึ้นมาและลิงก์เกอร์ก็จะทำการเชื่อมโยงไฟล์เหล่านั้นกับไลบรารีไฟล์ให้เป็นไฟล์ที่สามารถใช้งานได้ทันที

### 2.1.2 การตั้งค่าเอ็มเอฟซี ใน วิน 32 แอปพลิเคชัน

ในวิซวล ซี พลัส พลัส มีคลาสสำเร็จรูปที่อำนวยความสะดวกในการสร้างหน้าจอโปรแกรม การจะเรียกใช้งานคลาสเหล่านี้ ผู้ใช้ต้องทำการกำหนดการขอเรียกใช้เอ็มเอฟซี โดยมีวิธีการทำงานดังนี้

ไปที่ Project > Settings จะปรากฏหน้าต่าง Project Settings ในช่อง Microsoft Foundation Classes ให้เลือก Use MFC in a Shared DLL



รูปที่ 2-2 การตั้งค่าเอ็มเอฟซี ใน วิน32 แอปพลิเคชัน

### 2.1.3 การเขียนโปรแกรมบน วินโดวส์

ในส่วนนี้จะอธิบายถึงการเขียนโปรแกรมของวินโดวส์ และพื้นฐานที่ควรรู้สำหรับการเขียนโปรแกรมบนวินโดวส์เพื่อนำไปใช้ในการเขียนเกม

วินโดวส์ เป็นระบบปฏิบัติการแบบอีเวนต์-เบส(event-based) โดยข้อความ (Message) เป็นเหมือนสัญญาณที่จะบอกให้วินโดวส์ ต้องทำอะไร ทำอย่างไร เกิดอะไร อาจกล่าวได้ว่า ทุกสิ่งที่เกิดขึ้นในวินโดวส์คือข้อความนั่นเอง

ส่วนประกอบโดยทั่วไปในการสร้างวินโดวส์แอปพลิเคชันที่ต้องมี ได้แก่

#### 1) วิน 32 ไลบรารี (Win32 Libraries)

เมื่อเราสร้างวินโดวส์แอปพลิเคชันขึ้นมา คอมไพเลอร์จะเป็นตัวจัดการเกี่ยวกับส่วนของไลบรารีให้เองทั้งหมด ยกเว้นเสียแต่ไลบรารีของแอปพลิเคชันที่เราต้องการไม่ได้ถูกตั้งค่าไว้ตั้งแต่เริ่มต้น เมื่อสร้างแอปพลิเคชันที่ใช้ไครีเอทีฟ ผู้ใช้จะต้องเพิ่มไลบรารีของไครีเอทีฟเข้าไปในแอปพลิเคชันนั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2) เดอะ อินคลูดไฟล์ (The Include Files)

ผู้พัฒนาโปรแกรมต้องเรียกใช้ไฟล์ที่จำเป็นสำหรับแอปพลิเคชันนั้นไว้ในตัวโปรแกรมด้วยเสมอ และโดยทั่วไปในวินโดวส์แอปพลิเคชันจะต้องเริ่มต้นด้วย

```
#define win32_LEAN_AND_MEAN

#include <windows.h>

#include<windowsx.h>
```

และอาจเพิ่มไฟล์ที่จำเป็นสำหรับแอปพลิเคชันเข้าไปอีกได้ เช่น “Studio.h” หรือ “Math.h” เป็นต้น

## 3) วินเมน (WinMain)

วินเมน เป็นฟังก์ชันหลักที่วินโดวส์แอปพลิเคชันทุกแอปพลิเคชัน จำเป็นต้องกำหนดไว้คล้าย ๆ กับฟังก์ชัน main() ในภาษาซีและภาษาซีพลัส พลัส การประกาศโปรโตไทป์ของ WinMain() จะเป็นดังนี้

```
int WINAPI WinMain( HINSTANCE hinstance,
                   HINSTANCE hpreinstance,
                   LPSTR lpcmline,
                   Int ncmdshow);
```

## 4) วินโดวส์คลาส (Windows Class)

ผู้พัฒนาโปรแกรมที่ต้องการสร้าง วินโดวส์แอปพลิเคชัน จำเป็นต้องกำหนด วินโดวส์คลาสขึ้นมาด้วย ซึ่งวินโดวส์คลาสจะอธิบายถึงคุณลักษณะโดยทั่วไปของวินโดวส์ทั้งหมด โดยหลังจากที่ผู้พัฒนาโปรแกรมทำการสร้างวินโดวส์คลาสขึ้นมาแล้วผู้พัฒนาโปรแกรมก็จะสามารถสร้างวินโดวส์ได้ตามต้องการ ซึ่งวินโดวส์คลาสมีโครงสร้างดังนี้

```

typedef struct _WINDCLASS
{
    UNIT style;
    WNDPROC lpfnWndProc;
    Int cbClsExtra;
    Int cbWndExtra;
    HANDLE hInstance;
    HICON hIcon;
    HCURSOR hCursor;
    HBRUSH hbrBackground;
    LPCTSTR lpszMenuName;
    LPCSTR lpszClassName;
}
WINDCLASS;
WINDCLASS wndclass

```

#### 5) รีจิสเตอร์ เดอะ วินโดวส์คลาส(Register the Windows Class)

หลังจากที่ทำการสร้างวินโดวส์คลาสขึ้นมาแล้ว ลำดับต่อไปคือการทำการรีจิสเตอร์คลาส การรีจิสเตอร์จะทำให้วินโดวส์รู้ว่ามียินโดวส์คลาสที่สร้างขึ้นมา และสามารถให้ผู้พัฒนาโปรแกรมสร้างวินโดวส์ที่ลักษณะตามวินโดวส์คลาสได้ โดยชื่อแอสกีของคลาสเพียงอย่างเดียว การรีจิสเตอร์คลาส กระทำได้โดย

```

If(RegisterClass(&wndclass)==0)
{
    //error
}

```

#### 6) การสร้างหน้าต่างเกม

การจะสร้างหน้าต่างเกมขึ้นมาต้องทำการเรียกฟังก์ชัน CreateWindow() คำสั่งการสร้างหน้าต่างโปรแกรมมีโครงสร้างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HWND CreateWindow(
    LPCTSTR lpClassName,
    LPCTSTR lpWindowName,
    DWORD dwStyle,
    Int x,
    Int y,
    Int nwidth,
    Int nHeight,
    HWND hWndParent,
    HMENU hMenu,
    HANDLE hInstance,
    LPVOID lpParam
    (NULL)
);

```

#### 7) การแสดงหน้าต่างเกม

หลังจากที่ทำการสร้างหน้าต่างเกมเสร็จแล้ว และ ต้องการแสดงผลออกมา (ในกรณีที่ตั้งค่า พารามิเตอร์ ของ dwStyle ไม่ใช่ WS\_VISIBLE ต้องทำการเรียก ฟังก์ชัน ShowWindow() เสมอ)

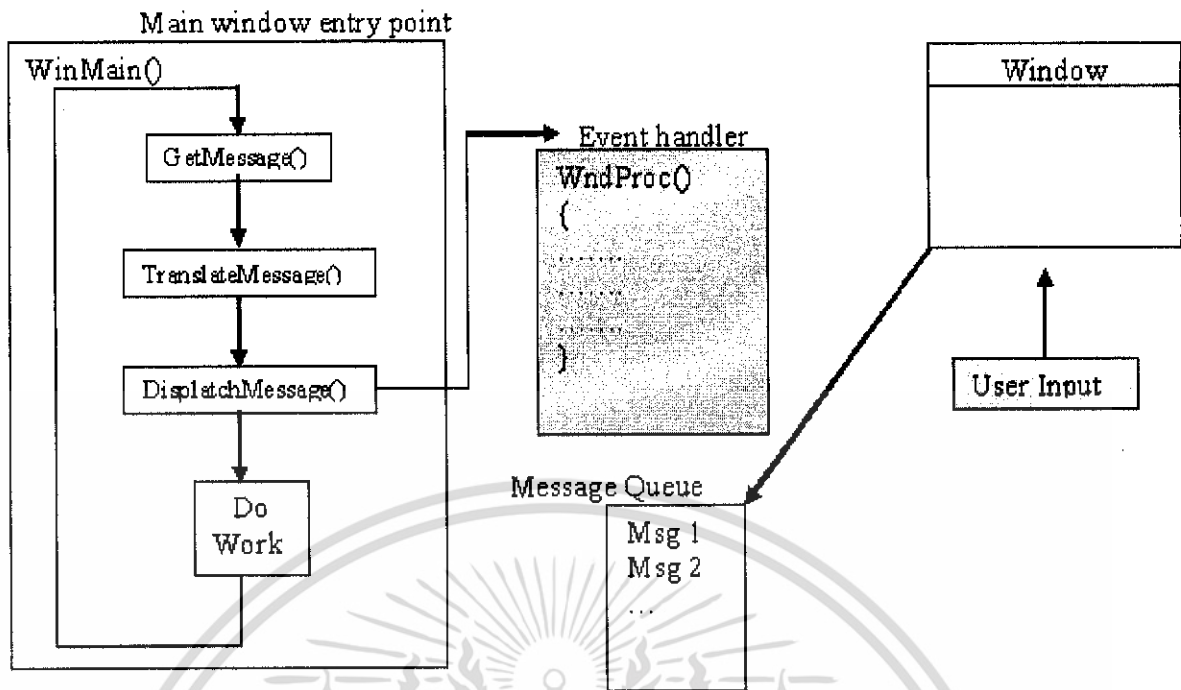
```

BOOL ShowWindow(
    HWND hWnd,
    Int nCmdshow);

```

#### 8) อีเวนต์ลูป (Event Loop)

เมนอีเวนต์ลูป คือ การทำงานแบบวนซ้ำ (Loop) ที่คอยรับข้อความ (Message) ที่เก็บไว้จากชุดลำดับข้อความ (Message Queue) แล้วทำการแปลง แล้วส่งต่อไปยังอีเวนต์แฮนเดอเรอร์ ดังรูป



### รูปที่ 2-3 แผนภาพแสดงการทำงานของอีเวนต์ลูปของภาษาซี พลัส พลัส

ส่วนของ วินเมน มี ฟังก์ชัน ที่ทำหน้าที่ต่างกันอยู่ 3 ฟังก์ชัน ใน อีเวนต์ลูป นี้ ได้แก่

#### - GetMessage

เป็นฟังก์ชันที่ทำหน้าที่รับข้อความจากชุดลำดับข้อความ ถ้ากรณีชุดลำดับข้อความว่าง ฟังก์ชันก็จะทำการรอจนกว่าจะมีข้อความเข้ามา

```

BOOL GetMessage( LPMSG lpMsg,
                 HWND hWnd,
                 UINT wMsgFilterMin,

```

#### - TranslateMessage

เป็น ฟังก์ชัน ที่ใช้แปลง ข้อความอีเวนต์ ให้อยู่ในรูปแบบพื้นฐานของ อีเวนต์ เพื่อจะนำไปใช้ในส่วนของ อีเวนต์แฮนเลอร์

```

BOOL TranslateMessage(CONST MSG*lpmsg);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - DispatchMessage

ทำการส่งข้อความไปยังอีเวนต์แฮนเดอเรอร์เพื่อทำการประมวลผลต่อไป

```
LONG DispatchMessage(CONST MSG*lpmsg);
```

### 9) อีเวนต์แฮนเดอเรอร์ (Event handler)

อีเวนต์แฮนเดอเรอร์ หรือ WinProc คือ ฟังก์ชันที่ต้องเขียนเพื่อจัดการ (Handle) ทุกข้อความที่แอปพลิเคชันนั้นสนใจ วินโดวส์อาจจะส่งข้อความมากมายหลายอย่าง แต่เรามีเพียงข้อความเดียวที่เราสนใจจะจัดการ การประกาศต้นแบบ ทำได้ดังนี้

```
RESULT CALLBACK WindowProc(
    HWND    hwnd,
    UINT    msg,
    WPARAM  wParam,
    LPARAM  lParam);
```

### 10) อีเวนต์ไดร์ฟเวนโปรแกรมมิ่ง (Event Driven Programming)

อีเวนต์ไดร์ฟเวนโปรแกรมมิ่ง คือ โปรแกรมจะทำงานตามอีเวนต์ที่เกิดขึ้น โดยระบบปฏิบัติการจะเป็นตัวคอยบอกโปรแกรมว่ามีอีเวนต์ใดเกิดขึ้นบ้าง หากโปรแกรมได้รับอีเวนต์ที่โปรแกรมสนใจก็จะทำงานตามที่ผู้พัฒนาโปรแกรมได้กำหนดไว้

สำหรับอีเวนต์ในที่นี้ได้รวมถึงทุกอย่างที่เกิดขึ้น ทั้งการได้รับข้อมูลเข้า (Input) เช่น การกดปุ่ม การปล่อยปุ่ม การคลิกเมาส์ เป็นต้น และอีเวนต์เกี่ยวกับวินโดวส์ เช่น การที่วินโดวส์ได้รับโฟกัส สั่งปิดหน้าต่างเกม หรือแม้แต่การวาดหน้าต่างขึ้นใหม่ (Redraw) หน้าต่างเกมดังที่ได้กล่าวมาแล้ว ดังนั้น อีเวนต์ไดร์ฟเวนโปรแกรมมิ่งก็คือการเขียนโปรแกรมที่จะทำงานโดยการตรวจข้อความที่ได้รับไปเรื่อย ๆ แล้วทำงานตามที่กำหนดเมื่อมีข้อความที่โปรแกรมสนใจ

หากเปรียบเทียบกับกรเขียนโปรแกรมโดยทั่วไป การทำงานจะทำงานแบบเรียงลำดับตามผู้เขียนโปรแกรมได้เขียนไว้ แต่สำหรับ อีเวนต์ไดร์ฟเวนโปรแกรมมิ่ง นั่นคือลักษณะที่ผู้ใช้เป็นผู้ควบคุมโปรแกรมมากกว่าที่โปรแกรมจะควบคุมให้ผู้ใช้ทำตาม

### 2.1.4 ระบบข่าวสารในวินโดวส์

โปรแกรมที่ทำงานบนวินโดวส์นั้นจะอยู่ในรูปของหน้าต่างโปรแกรม แต่ละโปรแกรมที่ทำงานอยู่นั้นเป็นวัตถุคนละตัวไม่ยุ่งเกี่ยวกับกัน วินโดวส์มีระบบข้อความ (Message) เพื่อเป็นการส่งข้อมูลให้โปรแกรมต่าง ๆ ทราบถึงเหตุการณ์ที่เกิดขึ้นในระบบ

ข้อความนี้เป็นเพียงแค่ค่าคงที่ ที่ส่งมาเมื่อเกิดเหตุการณ์ต่าง ๆ ขึ้นในวินโดวส์เพื่อใช้ในการอ้างอิงถึงเหตุการณ์ใด ๆ ที่เกิดขึ้นในระบบ ข้อความที่เกิดขึ้นจะถูกส่งเข้าไปยังชุดลำดับข้อความ (Message Queue) โปรแกรมที่ทำงานในระบบ เมื่อต้องการดูว่าข้อความใดที่เป็นของตัวเอง ก็จะมาดักรอที่ชุดลำดับข้อความนี้

ในฟังก์ชัน WindowProc นั้น จะทำการตรวจจับตัวแปร msg ซึ่งเป็นตัวแปรแบบ UINT (Unsigned Integer) ดังนี้

```

LRESULT CALLBACK WindowProc (HWND hwnd,UINT msg, WPARAM wparam, LPARAM lparam)
{
    switch (msg)
    {
        case WM_PAINT :
        {
            }break;
        case WM_DESTROY :
        {
            PostQuitMessage(0);
            }break;
    }
    return (DefWindowProc (hwnd,msg,wparam,lparam));
}

```

ฟังก์ชัน WindowProc นี้เอง เป็นส่วนที่ทุกโปรแกรมที่ทำงานบนวินโดวส์ต่างมีเหมือนกันและฟังก์ชัน WinProc ของแต่ละโปรแกรมต่างจะคอยตรวจจับดูว่าข้อความใดเป็นข้อความของตัวเอง เพราะข้อความที่ถูกส่งเรียงลำดับกันเข้ามาในชุดลำดับข้อความนั้นเกิดจากระบบอินเตอร์เฟสและอุปกรณ์ต่าง ๆ ที่อยู่ในโปรแกรมเป็นตัวสร้าง และส่งข้อความไปโดยอัตโนมัติ

ที่ฟังก์ชัน WindowProc จะต้องรับพารามิเตอร์ 4 ตัวก็เพราะว่าในการตรวจจับข้อความนั้น ชุดลำดับข้อความ (Message Queue) ต้องรู้ว่าข้อความนั้นเป็นของโปรแกรมใด หน้าต่างใด โดยดูจากตัวแปร hwnd

ตัวแปร msg จะอ้างถึงค่าคงที่ของข้อความนั้น ๆ เช่น WM\_SIZE, WM\_CREATE เป็นต้น และตัวแปร wparam กับ lparam นั้นจะบอกถึงค่าที่เปลี่ยนแปลง หรือข้อกำหนดบางอย่างที่อาจจะเกิดขึ้นพร้อมกับข้อความ

### 1) ข้อความของการเคลื่อนที่เมาส์

เมื่อเราเลื่อนเมาส์ไปภายในกรอบหน้าต่างจะเกิดข้อความที่ชื่อ WM\_MOUSEMOVE ขึ้นมา โดยค่า x และ y ของเคอร์เซอร์ที่เลื่อนไปนั้น จะเก็บอยู่ในตัวแปร lparam ซึ่งจะแบ่งเป็น

LOWORD (lparam) เก็บค่าของแกน x

HIWORD (lparam) เก็บค่าของแกน y

```

LRESULT CALLBACK WindowProc (HWND hwnd,UINT msg, WPARAM wparam, LPARAM lparam)
{
    switch (msg)
    {
        case WM_MOUSEMOVE :
        {
            int x = LOWORD(lparam);
            int y = HIWORD(lparam);
        } break;
    }
    return (DefWindowProc (hwnd,msg,wparam,lparam));
}

```

ตัวแปร wparam ต่อไปนี้จะเก็บค่าคงที่ของปุ่ม เพื่อตรวจสอบการทำงาน

- MK\_LBUTTON           ปุ่มซ้ายของเมาส์ถูกกด
- MK\_MBUTTON           ปุ่มกลางของเมาส์ถูกกด
- MK\_RBUTTON           ปุ่มขวาของเมาส์ถูกกด
- MK\_CONTROL           ปุ่ม <ctrl> ของแป้นพิมพ์ถูกกด และมีการคลิกเมาส์
- MK\_SHIFT              ปุ่ม <shift> ของแป้นพิมพ์ถูกกด และมีการคลิกเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2) เมจเซจของการคลิกปุ่มเมาส์

- WM\_LBUTTONDOWN เมื่อปุ่มซ้ายถูกกดลง
- WM\_RBUTTONDOWN เมื่อปุ่มขวาถูกกดลง
- WM\_LBUTTONUP เมื่อปุ่มซ้ายถูกปล่อยขึ้น
- WM\_RBUTTONUP เมื่อปุ่มขวาถูกปล่อยขึ้น
- WM\_LBUTTONDBLCLK เมื่อปุ่มขวาถูกดับเบิลคลิก
- WM\_RBUTTONDBLCLK เมื่อปุ่มซ้ายถูกดับเบิลคลิก

## 3) ข้อความของการกดปุ่มแป้นพิมพ์

เมื่อกดปุ่มบนแป้นพิมพ์จะเกิดข้อความที่ชื่อ WM\_KEYDOWN

```
case WM_KEYDOWN
{
    Switch(wparam)
    {
        Case VK_RIGHT : strcpy(s,"right"); break;
    }
    TextOut(hdc,10,50,s,strlen(s));
}
}
```

- WM\_CHAR เมื่อเราต้องการรู้ว่ากดปุ่มตัวอักษรใด
- WM\_KEYDOWN เมื่อต้องการตรวจสอบว่ากำลังกดปุ่มอยู่หรือไม่
- WM\_KEYUP เมื่อต้องการตรวจสอบว่าปุ่มที่กดอยู่ถูกปล่อยหรือยัง

## 2.2 การใช้งานไคเร็กซ์ (Direct X) เบื้องต้น

### 2.2.1 ความเป็นมาเกี่ยวกับโปรแกรมไมโครซอฟท์ไคเร็กซ์ (Microsoft Direct X)

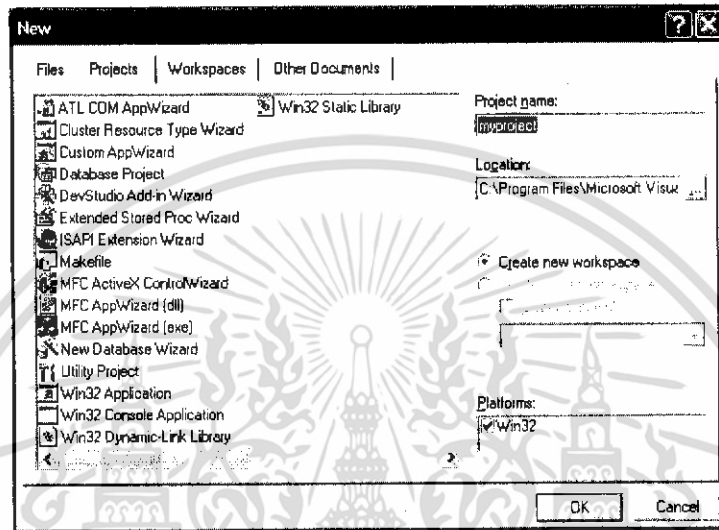
ในการเขียน โปรแกรมติดต่อกับอุปกรณ์ฮาร์ดแวร์ต่าง ๆ โดยตรงแต่เดิมนั้นเป็นสิ่งที่ยุ่งยากและซับซ้อนมาก เนื่องจากผู้พัฒนาเกมต้องศึกษาคำสั่งต่าง ๆ ของอุปกรณ์ฮาร์ดแวร์แต่ละรุ่นแต่ละยี่ห้อ โดยคำสั่งต่าง ๆ ที่ใช้ควบคุมอุปกรณ์ฮาร์ดแวร์นั้นเขียนจากภาษาแอสเซมบลี ซึ่งเป็นภาษาระดับล่างที่มีโครงสร้างของภาษาใกล้เคียงกับภาษาเครื่อง ซึ่งเป็นภาษาที่มีความสามารถในการควบคุมอุปกรณ์ฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ แต่เป็นภาษาที่ยากต่อการทำความเข้าใจของผู้พัฒนาเกม ยิ่งไปกว่านั้นเมื่ออุปกรณ์ฮาร์ดแวร์นั้น ๆ ถูกพัฒนาขึ้น โดยการเพิ่มคำสั่งใหม่ ๆ เข้าไป หรือเปลี่ยนคำสั่งเดิมให้มีประสิทธิภาพมากขึ้น เป็นผลให้ผู้พัฒนาเกมจำเป็นต้องทำการศึกษาคำสั่งต่าง ๆ นั้นใหม่อีกครั้ง ทำให้ผู้พัฒนาเกมเสียเวลาและสร้างความลำบากในการพัฒนาโปรแกรมอีกด้วย

ในช่วงที่ไมโครซอฟท์สร้างระบบปฏิบัติการวินโดวส์ขึ้นมาใหม่นั้น สิ่งที่เป็นเอกลักษณ์ของวินโดวส์ คือ จียูไอ (Graphic User Interface) โดยมีจีดีไอ (Graphic Device Interface) เป็นเครื่องมือสำหรับจัดการทางด้านภาพ ในขณะที่เดียวกันการพัฒนาเกมยังคงพัฒนาอยู่บนระบบปฏิบัติการดอส ดังนั้นบริษัทไมโครซอฟท์จึงเล็งเห็นถึงการพัฒนาเกมบนวินโดวส์ขึ้น ในระบบปฏิบัติการวินโดวส์ตั้งแต่เวอร์ชัน 3.1 ขึ้นไป ได้มีการใช้ไคเร็กซ์ของอุปกรณ์ฮาร์ดแวร์ขึ้น โดยไคเร็กซ์ประกอบด้วยชุดคำสั่งเพื่อที่จะควบคุมการเข้าถึงอุปกรณ์ฮาร์ดแวร์ต่าง ๆ ทำให้การเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ฮาร์ดแวร์ของผู้พัฒนาเกมง่ายและสะดวกขึ้น แต่กระนั้นผู้พัฒนาเกมยังคงคาดหวังว่าระบบปฏิบัติการวินโดวส์รุ่นใหม่ ๆ จะพัฒนาชุดคำสั่งที่ทำหน้าที่เป็นตัวประสานงานระหว่างอุปกรณ์ฮาร์ดแวร์ต่าง ๆ โดยทำผ่านตัวต่อประสานเพียงตัวเดียวได้ ซึ่งจะทำให้การพัฒนาโปรแกรมสะดวกขึ้น เพราะผู้พัฒนาเกมเพียงแต่ทำการศึกษาคำสั่งของตัวต่อประสานตัวนั้นเพียงอย่างเดียวก็เพียงพอต่อการพัฒนาโปรแกรมแล้ว

ด้วยเหตุนี้บริษัทไมโครซอฟท์ได้พัฒนาชุดคำสั่งพัฒนาซอฟต์แวร์ ที่ทำหน้าที่เป็นตัวประสานงานระหว่างผู้พัฒนาเกมกับอุปกรณ์ฮาร์ดแวร์ในการพัฒนาซอฟต์แวร์บนวินโดวส์ เพื่อให้การเข้าถึงอุปกรณ์ฮาร์ดแวร์ทำได้ง่าย โดยตอนแรกได้เน้นเพื่อใช้พัฒนาเกมคอมพิวเตอร์โดยเฉพาะ โดยมีชื่อเรียกว่า “Game Kits” ต่อมาได้เพิ่มความสามารถทางด้านภาพกราฟิก, งานด้านสื่อประสม, งานด้านเน็ตเวิร์ก, งานด้านการรับข้อมูลเข้า และงานภาพกราฟิก 3 มิติ ดังนั้นชุดคำสั่งนี้จึงถูกเรียกใหม่ว่า ไมโครซอฟท์ไคเร็กซ์เอ็กซ์ เอส ดี เค (Microsoft Direct X SDK) และปัจจุบันบริษัทไมโครซอฟท์ได้พัฒนาไคเร็กซ์จนถึงรุ่น 9.0 แล้ว

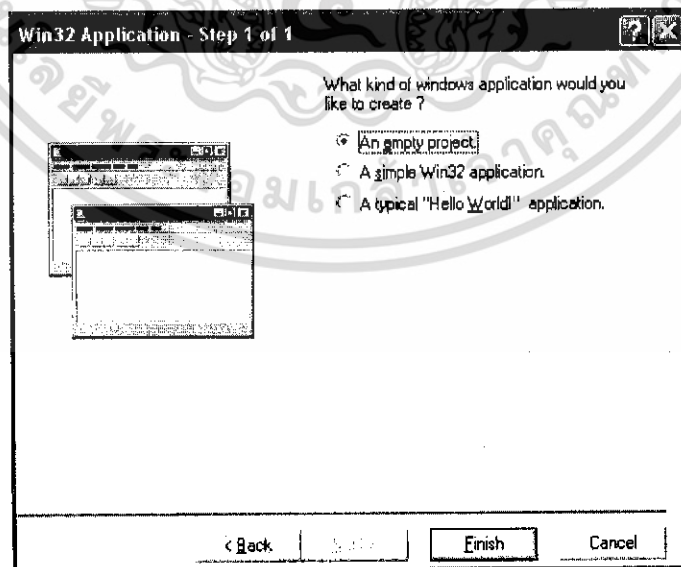
## 2.2.2 การสร้างโปรเจคใหม่

1. เปิดโปรแกรมวิซวล ซี พลัส พลัส ขึ้นมา จากนั้น เลือกเมนู File>New จะปรากฏไดอะล็อกให้เลือกแท็บ Projects จะมีให้เลือกว่าผู้ใช้งานต้องการเขียนโปรแกรมแบบใด ในที่นี้เราใช้เป็นวิน 32 แอปพลิเคชัน จากนั้นตั้งชื่อโปรเจคในช่อง Project name : และเลือกตำแหน่งไฟล์ที่ช่อง Location : ซึ่งอยู่ด้านขวาของหน้าต่าง จากนั้นกดปุ่ม OK



รูปที่ 2-4 แสดงการสร้างโปรเจคใหม่ ขั้นตอนที่ 1

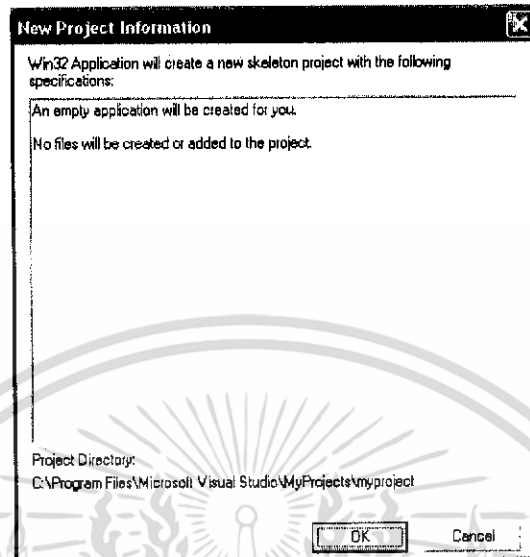
2. เลือก An empty project เพื่อสร้างโปรเจคว่างๆ ไว้ จากนั้นกด Finish



รูปที่ 2-5 แสดงการสร้างโปรเจคใหม่ ขั้นตอนที่ 2

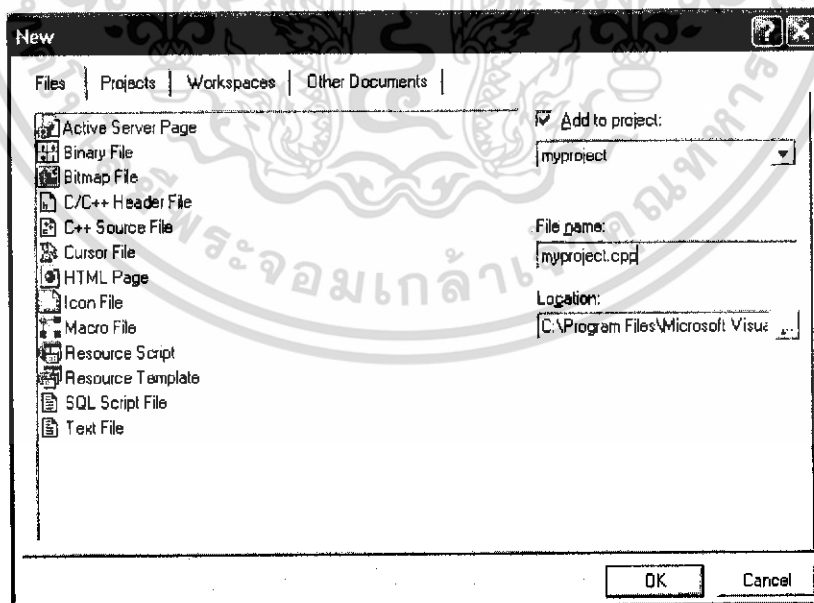
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัด 59404 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. กดปุ่ม OK เพื่อจบการสร้างโปรเจก



รูปที่ 2-6 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 3

4. สร้างไฟล์ใหม่ โดยเลือกเมนู File > New เพื่อเปิดไดอะล็อกอีกครั้ง จากนั้นเลือกแท็บ Files เลือก C++ Source File และตั้งชื่อไฟล์ที่อยู่ในช่อง File Name : ทางขวา จากนั้นกดปุ่ม OK



รูปที่ 2-7 แสดงการสร้างโปรเจกใหม่ ขั้นตอนที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.3 การสร้างหน้าต่างเกมโดยใช้ภาษาซี พลัส พลัส

### 1) ขอใช้ ไดรารี ในการสร้างหน้าต่าง

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

### 2) ส่วนของโปรแกรมหลัก มีการประกาศกำหนดลักษณะและสร้างหน้าต่างโปรแกรม

```
int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE hpreinstance, LPSTR
lpcmdline, int ncmdshow)
{
    WNDCLASSEX winclass;
    winclass.cbSize    = sizeof(WNDCLASSEX);
    winclass.style     = CS_VREDRAW |
    CS_HREDRAW|CS_OWNDC|CS_DBLCLKS;
    winclass.lpfnWndProc    = WindowProc;
    winclass.cbClsExtra    = 0;
    winclass.cbWndExtra    = 0;
    winclass.hInstance    = hinstance;
    winclass.hbrBackground =
    (HBRUSH)GetStockObject(WHITE_BRUSH);
    winclass.lpszClassName = "MyWin";
    winclass.lpszMenuName = NULL;
    winclass.hIcon        = NULL;
    winclass.hCursor      = NULL;
    winclass.hIconSm      = NULL;

    RegisterClassEx(&winclass);

    HWND hwnd;
    MSG msg;
```

```

// สร้างและแสดงหน้าต่างโปรแกรมที่ตำแหน่งจุดที่ (100,100) ถึงจุดที่ (400,400)
hwnd = CreateWindowEx(NULL,"MyWin","Win32
Window",WS_OVERLAPPEDWINDOW,100,100,400,400,NULL,NULL,hinstance,NULL);
ShowWindow(hwnd,SW_SHOW);
UpdateWindow(hwnd);

// วงทำซ้ำการตรวจจัดการส่งข้อความของโปรแกรม
while(1)
{
    if(!GetMessage(&msg,NULL,0,0,PM_NOREMOVE))
    {
        if(!GetMessage(&msg,NULL,0,0))
            return msg.wParam;
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    else
    {
        // game main
    }
}
return (msg.wParam);
}

```

### 3) กำหนดให้การทำงานเมื่อมีข้อความเกิดขึ้น

```

LRESULT CALLBACK WindowProc (HWND hwnd,UINT msg, WPARAM wParam, LPARAM
lparam)
{
    switch (msg)
    {
        // เมื่อมีการปิดหน้าต่างโปรแกรม จะทำการคืนหน่วยความจำ
        case WM_DESTROY :
            { PostQuitMessage(0); } break;
    }
    return (DefWindowProc (hwnd,msg,wparam,lparam));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

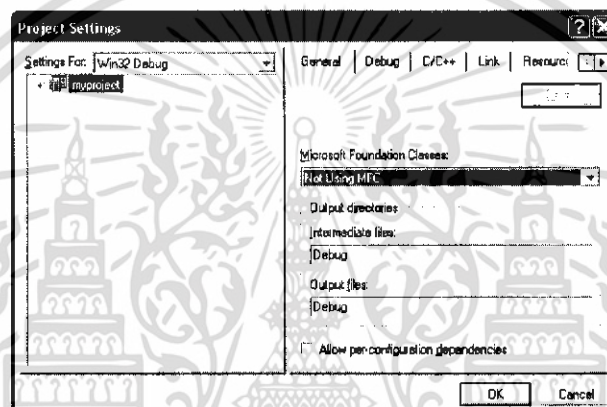
## 2.2.4 การสร้างหน้าต่างเกมโดยใช้โคเร็กเอ็กซ์

### 2.2.4.1 การขอใช้ไลบรารีของโคเร็กเอ็กซ์

การสร้างหน้าต่างโปรแกรมโดยเรียกใช้โคเร็กเอ็กซ์นั้น ต้องทำการเชื่อมโยงไฟล์ไลบรารีต่างๆ เช่น ddraw.lib ใช้ในการวาดภาพและแสดงผลหน้าจอ , dinput.lib ใช้เมื่อมีการใช้งานอุปกรณ์อินพุตเรคทีฟติดต่อกับโปรแกรม

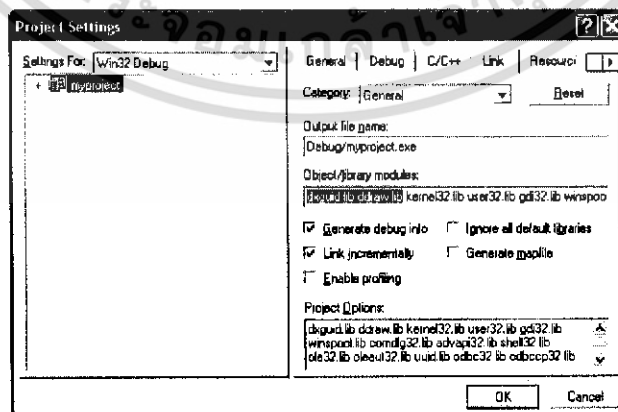
ในที่นี้ ต้องการที่จะสร้างหน้าต่างโปรแกรมเปล่าๆ ขึ้นมา โดยมีไลบรารีที่เกี่ยวข้องคือ dxguid.lib และ ddraw.lib จึงต้องทำการขอใช้งานไลบรารี ตามขั้นตอนต่อไปนี้

#### 1. เลือกที่เมนู Project > Setting...



รูปที่ 2-8 แสดงการขอใช้ไลบรารีของโคเร็กเอ็กซ์ ขั้นตอนที่ 1

2. เลือกแท็บ Link และพิมพ์ dxguid.lib และ ddraw.lib ในช่อง Object/library modules : เพื่อเป็นการขอใช้งานไลบรารีทั้งสองในโปรเจก จากนั้นกดปุ่ม OK



รูปที่ 2-9 แสดงการขอใช้ไลบรารีของโคเร็กเอ็กซ์ ขั้นตอนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.4.2 ส่วนของชุดคำสั่งโปรแกรม

#### 1) ขอบข่ายไลบรารีในการสร้างหน้าต่างในไดเรกต์เอ็กซ์

```
#include "ddraw.h"
```

#### 2) ประกาศใช้งานไดเรกต์ดรอว์ (DirectDraw) และการใช้งานเซอร์เฟส (Surface)

```
LPDIRECTDRAW7 dd = NULL;
LPDIRECTDRAWSURFACE7 primary = NULL;
```

#### 3) ส่วนของโปรแกรมหลักเพิ่มโค้ด (Code) สำหรับแสดงเซอร์เฟส (Surface)

บนหน้าต่างโปรแกรม

```
int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE
hprevinstance, LPSTR lpcmdline, int ncmdshow)
{
.....
ShowWindow(hwnd,ncmdshow);
UpdateWindow(hwnd);

DirectDrawCreateEx(NULL,(LPVOID*)&dd,IID_IDirectDraw
7,NULL);

dd-
>SetCooperativeLevel(hwnd,DDSCL_EXCLUSIVE|DDSCL_FULLSCREEN);

dd->SetDisplayMode(1024,768,16,0,0);

DDSURFACEDESC2 surf;
ZeroMemory(&surf,sizeof(surf));
surf.dwSize = sizeof(surf);
surf.dwFlag=DDSD_CAPS;
surf.ddsCaps.dwCaps=DDSCAPS_PRIMARYSURFACE);

dd->CreateSurface(&surf,&primary,NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(1)
{
    .....
}
}

```

#### 4) ทำการคืนหน่วยความจำเมื่อมีการปิดหน้าต่างโปรแกรม

LRESULT CALLBACK WindowProc (HWND hwnd,UINT msg,  
WPARAM wparam, LPARAM lparam)

```

{
    switch (msg)
    {
        case WM_DESTROY :
        {
            primary->Release();
            primary=NULL;
            dd -> Release();
            dd = NULL;
            PostQuitMessage(0);
        } break;
    }
    return (DefWindowProc (hwnd,msg,wparam,lparam));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.5 การแสดงรูปภาพบนเซอร์เฟส (Surface)

จากโปรแกรมการสร้างหน้าต่างด้วยโคเรียกซ์ให้เพิ่มโค้ดเข้าไป ดังนี้

### 1) สร้างฟังก์ชัน (Function) ใสรูป ชื่อ GetBmp ดังนี้

```
LPDIRECTDRAW7 GetBmp(LPDIRECTDRAW7 directdraw,LPCTSTR filename)
{
    HDC hdc;
    HBITMAP bit;
    LPDIRECTDRAW7 surf;

    bit=(HBITMAP) LoadImage(NULL,filename,IMAGE_BITMAP,0,0,
        LR_DEFAULTSIZE|LR_LOADFROMFILE);

    if (!bit) return NULL;

    BITMAP bitmap;
    GetObject(bit,sizeof(BITMAP),&bitmap);
    int surf_width = bitmap.bmWidth;
    int surf_height = bitmap.bmHeight;

    HRESULT result;
    DDSURFACEDESC2 ddsd;
    ZeroMemory(&ddsd,sizeof(ddsd));
    ddsd.dwSize = sizeof(DDSURFACEDESC2);
    ddsd.dwFlags = DDSD_CAPS|DDSD_WIDTH|DDSD_HEIGHT;
    ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN| DDSCAPS_SYSTEMMEMORY;
    ddsd.dwWidth = surf_width;
    ddsd.dwHeight = surf_height;
    result = directdraw->CreateSurface(&ddsd,&surf,NULL);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (result != DD_OK)
{
    DeleteObject(bit);
    return NULL;
}
else
{
    surf->GetDC(&hdc);
    HDC bit_dc=CreateCompatibleDC(hdc);
    SelectObject(bit_dc,bit);
    BitBlt(hdc,0,0,surf_width,surf_height,bit_dc,0,0,SRCCOPY);
    surf->ReleaseDC(hdc);
    DeleteDC(bit_dc);
}
return surf;
}

```

## 2) การใช้คำสั่งวาดรูป

ในส่วนของลูปเกม เพิ่มคำสั่งวาดรูปลงไป

```

while(1)
{
    if (!GetMessage(&msg,NULL,0,0,PM_NOREMOVE))
    {
        .....
    }
    else
    {
        Primary -> Blt(NULL,GetBmp(dd,"FilePath.bmp"),NULL,DDBLT_WAIT,NULL);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใช้คำสั่ง Bit ซึ่งมีโครงสร้างดังนี้

```
HRESULT Blt(Dest Rect ขนาดพื้นที่กรอบสี่เหลี่ยมที่จะถูกวาดลงไป ,
Surface ที่เก็บภาพไว้ ,
Source Rect ขนาดพื้นที่กรอบสี่เหลี่ยมของภาพต้นแบบ ,
dwFlags ตัวเลือก , NULL);
```

สำหรับพารามิเตอร์ต่างๆ มีรายละเอียด ดังนี้

- Dest Rect และ Source Rect จะใช้กำหนดขนาดของภาพที่ปรากฏใน เซอร์เฟส ซึ่งถ้าเป็น NULL ทั้งสอง ภาพที่ไหลมาจะถูกนำไปแสดงเท่ากับขนาดของจอภาพความละเอียดจอภาพโดยอัตโนมัติ
- Surface ก็คือ เซอร์เฟส ที่เก็บภาพเอาไว้
- Dest Rect เก็บขนาดของพื้นที่กรอบสี่เหลี่ยมที่ Blt วาดลงไป โดยจะวาดอยู่ภายในกรอบนั้น ถ้าใส่ NULL จะหมายถึงใช้พื้นที่ทั้งหมด เซอร์เฟส เป็นพื้นที่วาด ถ้าเราต้องการให้วาดซ้ายบน และความกว้างความยาว Dest Rect เป็นพารามิเตอร์ที่บอกให้ฟังก์ชัน Blt ทราบว่าตำแหน่งไหนใน เซอร์เฟส ที่ถูกวาด และวาดในขนาดเท่าใด เช่น ต้องการวาดรูปจากไฟล์ test.bmp ซึ่งมีขนาด 150x120 พิกเซล ให้ปรากฏที่ตำแหน่ง 100,100 บน ไพรมารีเซอร์เฟส (Primary Surface) จะเขียนได้ดังนี้

```
RECT r ;
r.left = 100;
r.top = 100;
r.right = r.left + 150 ;
r.bottom = r.top + 120;
primary -> Blt(&r,GetBmp(dd,"test.bmp"),NULL,DDBLT_WAIT,NULL);
```

- Source Rect ใช้กำหนดขอบเขตของรูปภาพต้นแบบ หากใส่ค่า NULL จะหมายถึงทั้งรูปภาพ แต่หากกำหนดกรอบ ภาพที่แสดงออกมานั้นจะแสดงเพียงบางส่วนของรูปเท่านั้น เช่น หากว่าต้องการแสดงภาพเพียงครึ่งเดียว ต้องเขียนโปรแกรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RECT src_r;
src_r.left = 1;
src_r.top = 1;
src_r.right = 75;
src_r.bottom = 60;
primary -> Blt(NULL, GetBmp(dd,"test.bmp"),&src_r,DDBLT_WAIT,NULL);

```

### 2.2.6 การใส่เสียงดนตรี

หัวข้อต่อไปนี้เป็นวิธีการใส่เสียงดนตรี ลำดับแรกให้ทำการคัดลอกไฟล์เสียง ในที่นี้คือไฟล์ cave.mid ให้ใส่โค้ดเพิ่มเข้าไปในส่วนการประกาศตัวแปรดังนี้

```

Int      music;
CDXMidi *midi = 0;

```

ตัวแปร itemsshoes เป็นตัวแปรแบบ CDXSound จะเก็บค่าของเสียงสิ่งของเฉพาะไฟล์ .wav หนึ่งไฟล์เท่านั้น สำหรับตัวแปร sound เป็นแบบ CDXSoundBuffer นั้นจะเป็นเหมือนกับตัวเล่นเสียง ถ้าจะเล่นเสียงใดก็เพียงโหลด CDXSound ขึ้นมา และกดปุ่ม Play เสียงก็จะเล่นทันที

เขียนโปรแกรมโหลดไฟล์ใส่เสียง .wav ในฟังก์ชัน cdx\_Init ดังนี้

```

midi = new CDXMidi();
midi -> Init(g_hWnd.NULL);

music = midi->LoadMidi("cave.mid");

```

เล่นเสียง ใช้คำสั่ง

```

midi->Play(music)

```

คืนหน่วยความจำโดยแทรกคำสั่งดังนี้

```

Void cdx_DeInit(void)
{
    SAFEDELETE(midi);
}

```

### 2.2.7 การใส่เสียงประกอบ

หัวข้อต่อไปนี้เป็นวิธีการใส่เสียงประกอบ ลำดับแรกทำการคัดลอกไฟล์เสียง ในที่นี้คือไฟล์ item.wav ให้ใส่โค้ดเพิ่มเข้าไปในส่วนการประกาศตัวแปรดังนี้

```

CDXSound      *itemshoes = 0;
CDXSoundBuffer *sound = 0;

```

ตัวแปร itemshoes เป็นตัวแปรแบบ CDXSound จะเก็บค่าของเสียงไอเท็ม เฉพาะไฟล์ .wav หนึ่งไฟล์เท่านั้น สำหรับตัวแปร sound เป็นแบบ CDXSoundBuffer นั้นจะเป็นเหมือนกับตัวเล่นเสียง ถ้าจะเล่นเสียงใดก็เพียงโหลด CDXSound ขึ้นมา และกดปุ่ม Play เสียงก็จะเล่นทันที

เขียนโปรแกรมโหลดไฟล์ใส่เสียง .wav ในฟังก์ชัน cdx\_Init ดังนี้

```

itemshoes = new CDXSound();
itemshoes -> Create(g_hWnd);

sound = new CDXSoundBuffer();
sound -> Load(itemshoes,"item.wav",1);

```

เล่นเสียง ใช้คำสั่ง

```

Sound -> Play();

```

คืนหน่วยความจำโดยแทรกคำสั่งดังนี้

```

Void cdx_DeInit(void)
{
    SAFEDELETE(sound);
    SAFEDELETE(itemshoes);
}

```

## 2.3 เกมเอ็นจิน ซีดี เอ็กซ์ (Game Engine)

เป็นชุดคำสั่งที่ช่วยอำนวยความสะดวกในการเขียนโปรแกรมให้ง่ายขึ้น เหมือนกับกรอบความยุ่งยากของการเขียนโปรแกรมเอาไว้ และเตรียมฟังก์ชันต่างๆ แบบสำเร็จรูปเอาไว้ให้เราเรียกใช้เพียงอย่างเดียว ผู้นำไปใช้รู้เพียงว่าจะนำเกม เอ็นจิน ไปใช้ได้อย่างไรและมีวิธีการเรียกใช้อย่างไรเท่านั้นก็พอแล้ว ซึ่งในโปรแกรมเกมนี้ก็ได้พัฒนาโดยใช้เกมเอ็นจินคือ ซีดีเอ็กซ์ มันจะเตรียมโค้ดบางส่วนในการเขียนโปรแกรมไว้ให้เป็นฟังก์ชัน ซึ่งทำให้ผู้เขียนโปรแกรมสามารถใช้งานได้สะดวก ไม่ต้องเขียนโปรแกรมมากเท่า ไฉนเอ็กซ์

เราสามารถสร้างโปรเจกต์โดยใช้ ซี ดี เอ็กซ์ ได้โดย เปิดโปรแกรมวิซวล ซี พลัส พลัส ขึ้นมา จากนั้น เลือกเมนู File>New จะปรากฏไดอะล็อกให้เลือกแท็บ Projects จะมีให้เลือกว่าผู้ใช้ต้องการเขียนโปรแกรมแบบใด ในที่นี้เราใช้เป็น CDX Application Wizard จากนั้นตั้งชื่อโปรเจกต์ในช่อง Project name : และเลือกตำแหน่งไฟล์ที่ชื่อ Location : ซึ่งอยู่ด้านขวาของหน้าต่าง จากนั้น กดปุ่ม OK แล้วทำการเลือกส่วนต่างๆ ที่โปรแกรมถามมาทั้ง 3 หน้าตามที่ต้องการ แล้ว กดปุ่ม Finish เราก็สามารถสร้างโปรเจกต์โดยใช้ ซี ดี เอ็กซ์ ได้ โดยที่มันจะทำการเตรียมโค้ดในส่วนของการสร้างหน้าต่างและฟังก์ชันของเกม(ลูปเกม) ไว้ให้

### 2.3.1 การทำดับเบิลบัฟเฟอร์ (Double Buffer)

ในการทำให้เกิดภาพเคลื่อนไหวต่างๆ ได้ นั้น ดับเบิลบัฟเฟอร์ จะเป็นตัวที่ทำให้ภาพใหม่ที่โหลดขึ้นมาไม่กระตุกจะทำให้การเปลี่ยนภาพนั้นนุ่มนวล ซึ่งใช้หลักของการสลับเซอร์เฟส

เซอร์เฟส เป็นส่วนประกอบที่สำคัญมากของ ไคเร็กซ์ เนื่องจากมีหน้าที่จัดการกับภาพกราฟิกต่างๆ ที่แสดงบนหน้าจอ ซึ่ง 1 เซอร์เฟส จะประกอบไปด้วยภาพกราฟิกมากมายที่มาประกอบกันเป็น 1 ภาพ การทำเซอร์เฟส ก็คือการสลับ เซอร์เฟส นั้นเอง ซึ่งประกอบไปด้วย 2 เซอร์เฟส ดังนี้

#### 1. ฟรอนต์เซอร์เฟส (Front Surface)

ฟรอนต์เซอร์เฟส เป็นเซอร์เฟสที่แสดงภาพกราฟิกอยู่ในหน้าจอปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. เบ็คเซอร์เฟส (Back Surface)

เบ็คเซอร์เฟส เป็นเซอร์เฟสที่เตรียมไว้สำหรับแสดงภาพกราฟิกในลำดับต่อจากฟรอนต์เซอร์เฟส

หากมีการสลับเซอร์เฟส เกิดขึ้น เบ็คเซอร์เฟส ก็จะสลับไปแทนที่ ฟรอนต์เซอร์เฟส ทำให้ไม่เสียเวลาในการโหลดรูปและได้ภาพที่เปลี่ยนไปอย่างนุ่มนวล

### 2.3.2 การโต้ตอบกับแป้นพิมพ์

เป็นการเขียน โปรแกรมควบคุมแป้นพิมพ์ ซึ่งจะต้องเขียนคำสั่งในฟังก์ชัน `cdx_DoFrame()` มีรูปแบบการเขียนดังนี้

```
Input->Update();
if (Input->GetKeyState(CDXKEY_ESCAPE)==CDXKEY_PRESS)
{
```

-ที่เป็นตัวหนาคือค่าคงที่ของปุ่มต่างๆ ซึ่งสามารถดูได้ว่าค่าคงที่นั้นคือปุ่มใดได้ที่ `CDXInput.h`

-ในวงเล็บปีกกา คือ คำสั่งที่ต้องการจะทำให้เกิดอีเวนต์เมื่อมีการกดแป้นพิมพ์ปุ่มนั้นๆ

- `CDXKEY_PRESS` คือ ตรวจสอบการกดแป้นพิมพ์ เพียงครั้งเดียวกดค้างจะไม่มีผล ถ้าตัดส่วนนี้ออกจะสามารถกดค้างได้และเกิดผลตามคำสั่งที่เขียนนั้น

### 2.3.3 การโต้ตอบกับเมาส์

เป็นการเขียน โปรแกรมควบคุมเมาส์ซึ่งจะต้องเขียนคำสั่งในฟังก์ชัน `cdx_DoFrame()` ซึ่ง มีรูปแบบการเขียนดังนี้

```
Input->Update();
if (Input->GetKeyState(CDXKEY_MOUSEMIDDLE)==CDXKEY_PRESS)
{
```

-ที่เป็นตัวหนาคือค่าคงที่ของคีย์ ต่างๆ ซึ่งสามารถดูได้ว่าค่าคงที่นั้นคือปุ่มใดได้ที่ `CDXInput.h`

- ในวงเล็บปีกกา คือ คำสั่งที่ต้องการจะทำให้เกิดอีเวนต์เมื่อมีการคลิกเมาส์นั้นๆ
- CDXKEY\_PRESS คือ เช็การคลิกเมาส์เพียงครั้งเดียว คลิกค้างจะไม่มีผล ถ้าตัดส่วนนี้ออกจะสามารถคลิกค้างได้และเกิดผลตามคำสั่งที่เขียนนั้น

## 2.4 ทฤษฎีการคำนวณในฟิสิกส์

การเคลื่อนที่ในฟิสิกส์จะทำการคำนวณค่าพิกัด  $xy$  ( $x,y$ ) , ความเร็วแกน  $xy$  ( $v(x),v(y)$ ) และ ความเร่งแกน  $xy$  ( $a(x),a(y)$ ) เพื่อเปลี่ยนแปลงค่าตลอดเวลา โดยใช้สูตรการเคลื่อนที่ ดังนี้

$$\text{ความเร็ว} \quad v = u + at$$

$$\text{ระยะทาง} \quad s = vt$$

คำนวณทั้ง 2 แกน ได้ดังนี้

$$v(x) = u(x) + a * \text{deltatime}$$

$$v(y) = u(y) + g * \text{deltatime}$$

$$x = v(x) * \text{deltatime}$$

$$y = v(y) * \text{deltatime}$$

หมายเหตุ

- $x,y$  เป็นพิกัดบนจอ
- $v(x),v(y)$  เป็นความเร็วในแนวแกน  $x$  และ แกน  $y$
- $a,g$  เป็นความเร่งในแนวแกน  $x$  และ แกน  $y$
- $\text{deltatime}$  เป็นผลต่างของเวลาในแต่ละเฟรมของโปรแกรมเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## ขั้นตอนการดำเนินงานวิจัย

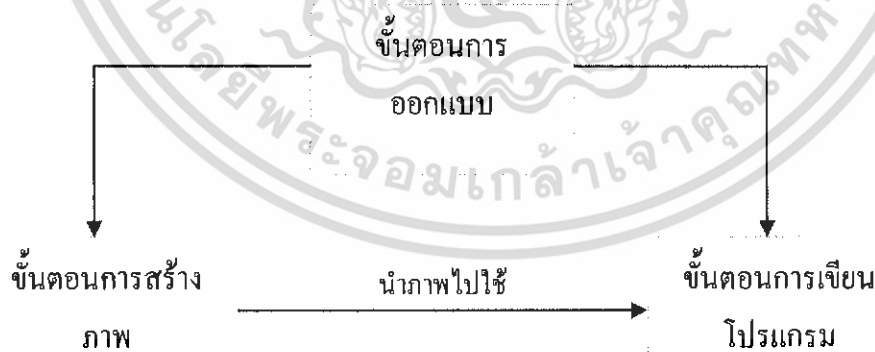
ขั้นตอนทั้งหมดในงานวิจัยจะเกิดขึ้นไม่ได้หากขาดการออกแบบ การวางแผนที่ดีและมีเป้าหมายชัดเจน การดำเนินงานทั้งหมดจะถูกกระทำภายใต้แผนที่วางเอาไว้อย่างรัดกุมเพื่อความสำเร็จตามเป้าหมาย ในแต่ละช่วง แผนงานทั้งหมดมีดังนี้

1. ขั้นตอนการออกแบบ แบ่งได้ 3 ส่วน
  - ออกแบบระบบและ โครงสร้างการทำงานของโปรแกรม
  - การออกแบบอินเตอร์เฟซของโปรแกรม
  - การออกแบบตัวละครต้นแบบที่มีให้ผู้ใช้ได้เลือกในโปรแกรม
2. ขั้นตอนการเขียนโปรแกรม

ขั้นตอนนี้ใช้เวลานานที่สุดในตลอดระยะเวลาทำงาน เพราะจำเป็นต้องศึกษารูปแบบวิธีการเขียนเกมและโปรแกรมจัดการเกี่ยวกับการใส่ฟังก์ชันต่าง ๆ เพื่อการสร้างเกม และการเขียนโปรแกรมเชิงวัตถุ เพื่อให้ได้โปรแกรมที่กระชับรัดและทำงานได้ถูกต้อง เข้าใจง่าย

3. ขั้นตอนการสร้างตัวละคร สร้างฉาก และการสร้างรูปภาพต่าง ๆ ที่ใช้ในเกม

เราสามารถสรุปขั้นตอนการวางแผนเป็นแผนภาพได้ดังนี้



รูปที่ 3-1 ขั้นตอนการวางแผน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 ขั้นตอนการออกแบบ

ขั้นตอนการออกแบบมีรายละเอียดดังนี้

#### 3.1.1 การออกแบบระบบสร้างเกมและฟังก์ชันต่างๆ ที่ช่วยสนับสนุนผู้ใช้ในการสร้างเกม ระบบสร้างเกมออกแบบเป็น 2 ส่วนหลักๆ

- ส่วนของการชุดคำสั่งสำหรับประมวลผลออกมาเป็นเกม ประกอบด้วยคลาสเกมที่มีกลุ่มคำสั่งต่างๆ ที่สนับสนุนในการเล่นเกมนั้น ได้แก่
  - คลาสอ็อบเจกต์ (Object) จัดการเกี่ยวกับตัวละคร
  - คลาสแบ็คกราวด์ (Background) จัดการเกี่ยวกับฉาก
  - คลาสซาวนด์เอฟเฟ็ค (SoundEffect) จัดการเกี่ยวกับการเล่นเสียงประกอบ
  - คลาสมิวสิค (Music) จัดการเกี่ยวกับการเล่นเพลง

ตัวอย่างการทำงานพื้นฐานของคลาสเหล่านี้ เช่น

- การเคลื่อนที่
- การเปลี่ยนรูป
- การเพิ่ม/ลด พลังชีวิต
- การตรวจสอบการชนกันของวัตถุ
- การตรวจสอบการตาย
- การเปลี่ยนฉาก

- ส่วนของโปรแกรมมีหน้าที่ในการรับข้อมูลจากผู้ใช้งานโปรแกรมผ่านหน้าต่างโปรแกรม และทำการสร้างชุดคำสั่งสำหรับประมวลผลออกมาเป็นเกม ตามที่ผู้ใช้ต้องการ โดยชุดคำสั่งเหล่านั้นจะถูกนำไปบันทึกลงไฟล์เพื่อนำไปประมวลผลเป็นเกมที่เล่นได้จริง

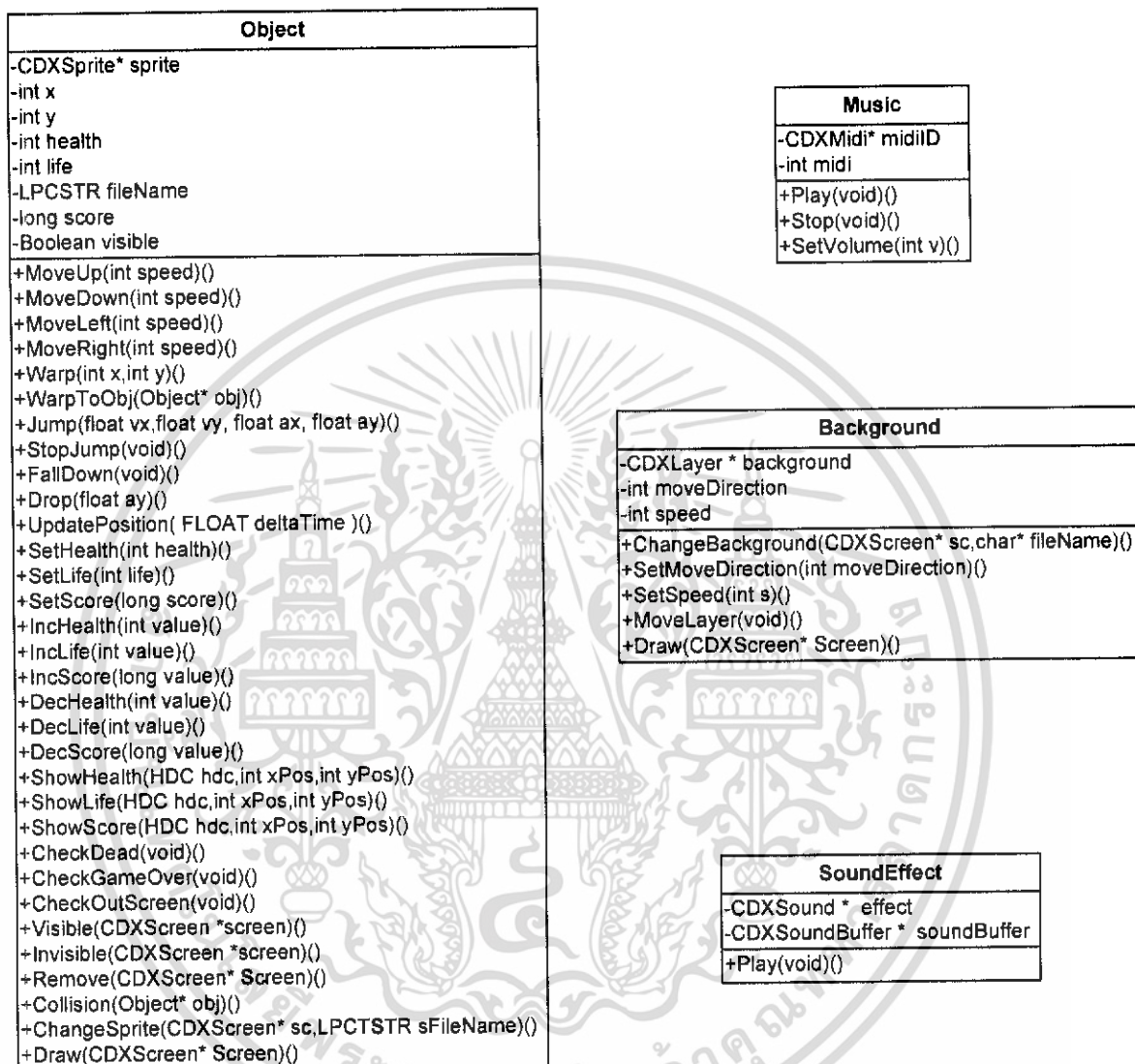
#### 3.1.2 การออกแบบตัวละครต้นแบบและฉากต่าง ๆ ที่มีให้ผู้ใช้ได้เลือกในโปรแกรม

ในบางครั้งผู้สร้างเกมไม่ถนัดทางด้านกราฟิก โปรแกรมช่วยสร้างเกมจึงมีตัวละครต้นแบบและฉากสำเร็จรูปเตรียมไว้ให้จำนวนหนึ่ง

การออกแบบตัวละครและฉากนั้น เราจะใช้โปรแกรมโฟโตช้อพ 7.0 (Photoshop 7.0) ซึ่งอำนวยความสะดวกและมีลูกเล่นมากมายให้ได้เลือกใช้ สามารถสร้างตัวละครและฉากได้ตามต้องการ

### 3.1.3 การออกแบบคลาสต่างๆในระบบ

ในส่วนการออกแบบคลาสต่าง ๆ นี้ จะแสดงคลาสหลัก ๆ ที่มีในระบบ ซึ่งมีดังนี้



รูปที่ 3-2 แผนภาพแสดงแอตทริบิวต์และฟังก์ชันของคลาสต่าง ๆ

คลาสไดอะแกรมนี้แสดงให้เห็นถึงความสัมพันธ์กับคลาสหลักในโปรแกรม ซึ่งประกอบไปด้วยคลาสดังต่อไปนี้

คลาสแบ็กกราวนด์ (Background)

คลาสมิวสิก (Music)

คลาสอ็อบเจ็ค (Object)

คลาสซาวนด์เอฟเฟ็ค (SoundEffect)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของฟังก์ชันในคลาส มีดังนี้

### คลาสแบ็กกราวนด์ (Background)

เป็นคลาสที่ใช้ในการจัดการเกี่ยวกับภาพพื้นหลัง เช่น การโหลดภาพพื้นหลัง การทำให้ภาพพื้นหลังเลื่อนไปตามทิศทางไหนด้วยความเร็วเท่าไรซึ่งกำหนดเองได้ การเปลี่ยนภาพพื้นหลัง การวาดภาพพื้นหลัง โดยมีรายละเอียดดังนี้

เมมเบอร์	รายละเอียด
CDXLayer * background	กำหนดอ็อบเจ็กต์ของภาพพื้นหลัง
int moveDirection	ทิศทางที่เลื่อนภาพพื้นหลัง
int speed	ความเร็วในการเลื่อนภาพพื้นหลัง
ฟังก์ชัน	รายละเอียด
ChangeBackground(CDXScreen* sc,char* fileName)	เปลี่ยนภาพพื้นหลัง
SetMoveDirection(int moveDirection)	กำหนดทิศทางที่เลื่อนภาพพื้นหลัง
SetSpeed(int s)	กำหนดความเร็วในการเลื่อนภาพพื้นหลัง
MoveLayer(void)	ให้ภาพพื้นหลังเลื่อนไปในทิศที่กำหนด
Draw(CDXScreen* Screen)	วาดภาพพื้นหลัง

### ตารางที่ 3-1 ตารางแสดงรายละเอียดของคลาสแบ็กกราวนด์

### คลาสมิวสิก (Music)

เป็นคลาสที่ใช้ในการจัดการเกี่ยวกับเสียงดนตรีในเกม เช่น การโหลดเสียงดนตรี การเล่นเสียงดนตรี การหยุดเสียงดนตรี การปรับระดับความดังของเสียงดนตรี โดยมีรายละเอียดดังนี้

เมมเบอร์	รายละเอียด
CDXMidi * midiID	กำหนดอ็อบเจ็กต์ของเสียงดนตรี
int midi	รหัสเสียงดนตรี
ฟังก์ชัน	รายละเอียด
Play(void)	เล่นเสียงดนตรี
Stop(void)	หยุดเล่นเสียงดนตรี
SetVolume(int v)	ปรับความดังของเสียงดนตรี

### ตารางที่ 3-2 ตารางแสดงรายละเอียดของคลาสมิวสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คลาสอ็อบเจกต์ (Object)

เป็นคลาสที่ใช้ในการจัดการเกี่ยวกับอ็อบเจกต์ เช่น การกำหนดอ็อบเจกต์ต่างๆ การเปลี่ยนอ็อบเจกต์ การกำหนดตำแหน่งของอ็อบเจกต์ การกำหนดการเคลื่อนที่และความเร็วของอ็อบเจกต์ การจัดการเกี่ยวกับพลัง การจัดการเกี่ยวกับชีวิต การจัดการเกี่ยวกับคะแนน การจัดการเกี่ยวกับการมองเห็น การตรวจจับการชนกันของอ็อบเจกต์ การตรวจจับวัตถุออกนอกจอโดยมีรายละเอียดดังนี้

เมมเบอร์	รายละเอียด
CDXSprite * sprite	กำหนดอ็อบเจกต์ของตัวละคร
int x	ตำแหน่งบนแนวนอนของตัวละคร
int y	ตำแหน่งบนแนวตั้งของตัวละคร
int health	พลังของตัวละคร
int life	ชีวิตของตัวละคร
LPCSTR fileName	เก็บสตริงลงไฟล์
long score	คะแนนของตัวละคร
Boolean visible	การมองเห็นตัวละคร
<b>ฟังก์ชัน</b>	<b>รายละเอียด</b>
MoveUp(int speed)	เคลื่อนที่ตัวละครขึ้นบน
MoveDown(int speed)	เคลื่อนที่ตัวละครลงล่าง
MoveLeft(int speed)	เคลื่อนที่ตัวละครไปทางซ้าย
MoveRight(int speed)	เคลื่อนที่ตัวละครไปทางขวา
Warp(int x,int y)	ย้ายตำแหน่งตัวละคร
WarpToObj(Object* obj)	ย้ายตัวละครไปยังตำแหน่งของตัวละครที่อ้างอิง
Jump(float vx,float vy, float ax, float ay)	ตัวละครกระโดดแบบโปรเจกไทล์
StopJump()	ตัวละครหยุดอยู่กับที่หลังจากที่กระโดด
FallDown()	ตัวละครกระโดดชนเพดานแล้วตกลงมา
Drop(float ay)	ตัวละครตกลงมาด้วยแรงโน้มถ่วง
UpdatePosition( FLOAT deltaTime )	ใช้คำนวณค่าของฟิสิกส์
SetHealth(int health)	กำหนดค่าพลังของตัวละคร
SetLife(int life)	กำหนดค่าชีวิตของตัวละคร
SetScore(long score)	กำหนดคะแนนของตัวละคร
IncHealth(int value)	เพิ่มค่าพลังของตัวละคร
IncLife(int value)	เพิ่มค่าชีวิตของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูงาน เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการศึกษา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IncScore(long value)	เพิ่มคะแนนของตัวละคร
DecHealth(int value)	ลดค่าพลังของตัวละคร
DecLife(int value)	ลดค่าชีวิตของตัวละคร
DecScore(long value)	ลดคะแนนของตัวละคร
ShowHealth(HDC hdc,int xPos,int yPos)	แสดงค่าพลังของตัวละคร
ShowLife(HDC hdc,int xPos,int yPos)	แสดงค่าชีวิตของตัวละคร
ShowScore(HDC hdc,int xPos,int yPos)	แสดงคะแนนของตัวละคร
CheckDead(void)	ตรวจค่าพลัง
CheckGameOver(void)	ตรวจค่าชีวิต
CheckOutScreen(void)	ตรวจจับวัตถุออกนอกจอ
Visible(CDXScreen *screen)	กำหนดให้มองเห็นตัวละคร
Invisible(CDXScreen *screen)	กำหนดให้มองไม่เห็นตัวละคร
Remove(CDXScreen* Screen)	ลบตัวละครออก
Collision(Object* obj)	ตรวจจับการชนกันของวัตถุ
ChangeSprite(CDXScreen* sc,LPCWSTR sFileName)	เปลี่ยนภาพตัวละคร
Draw(CDXScreen* Screen)	วาดภาพตัวละคร

### ตารางที่ 3-3 ตารางแสดงรายละเอียดของคลาสอ็อบเจ็กต์

#### คลาสซาวน์เอฟเฟ็กต์ (SoundEffect)

เป็นคลาสที่ใช้ในการจัดการเกี่ยวกับเสียงเอฟเฟ็กต์ เช่น การโหลดเสียงเอฟเฟ็กต์ การเล่นเสียงเอฟเฟ็กต์ การหยุดเสียงเอฟเฟ็กต์ โดยมีรายละเอียดดังนี้

เมมเบอร์	รายละเอียด
CDXSound * effect	กำหนดอ็อบเจ็กต์ของเสียงเอฟเฟ็กต์
CDXSoundBuffer * soundBuffer	กำหนดอ็อบเจ็กต์ของบัฟเฟอร์เสียงเอฟเฟ็กต์
ฟังก์ชัน	รายละเอียด
Play(void)	เล่นเสียงเอฟเฟ็กต์

### ตารางที่ 3-4 ตารางแสดงรายละเอียดของคลาสซาวน์เอฟเฟ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

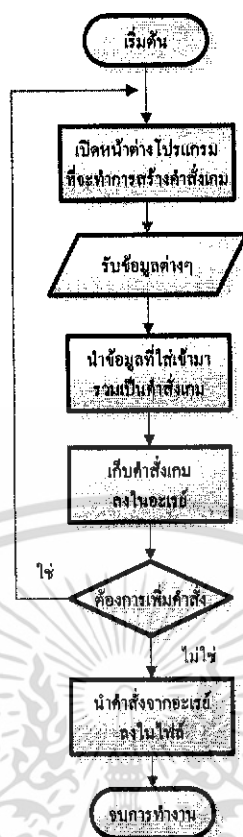
### 3.1.4 โครงสร้างและการทำงานของโปรแกรม

เราแบ่งโครงสร้างของโปรแกรมเป็น 2 ส่วน คือ ส่วนแอฟพลิเคชัน และ ส่วนเกมเอ็นจิน โดยที่ในส่วนของแอฟพลิเคชัน จะมีการให้ผู้ใช้ทำการสร้างที่อยู่ของเกมที่ผู้ใช้ต้องการจะลง มันจะทำการคัดลอกไฟล์เคอร์รี่ซอร์ส (Resource) ซึ่งเป็นส่วนของเกมเอ็นจิน ที่เราเก็บในไดร์ฟซี (C:/) ไปวางไว้ในที่ที่ผู้ใช้ได้สร้างไว้ และเมื่อผู้ใช้ทำการสร้างเกม แอฟพลิเคชันจะมีการเขียนข้อมูลต่างๆที่ผู้ใช้กำหนดไว้ลงในไฟล์นามสกุลเอช (.h) ที่อยู่ในส่วนของเกมเอ็นจิน

#### 1. ส่วนของแอฟพลิเคชัน

หน้าที่หลักของโปรแกรมช่วยสร้างเกม คือ การรับค่าต่างๆ ที่ผู้ใช้ป้อนเข้ามา เช่น ชื่อตัวละคร ไฟล์รูปของตัวละคร ตำแหน่งพิกัดเริ่มต้นที่จะให้ตัวละครปรากฏบนหน้าจอเมื่อเริ่มเกม ค่าต่างๆ นี้จะถูกสร้างออกมาเป็นชุดคำสั่งและเก็บลงในไฟล์เกมที่สามารถนำไปประมวลผลออกมาเป็นเกมจริงได้

การจัดการข้อมูลต่างๆ ของโปรแกรมช่วยสร้างเกม จะใช้ตัวแปรประเภทอะเรย์สำหรับเก็บข้อมูลชนิดสตริง ( CArray < CString , CString > ) โดยในทุกๆ ครั้งที่ผู้ใช้ทำการใส่ค่าผ่านหน้าต่างโปรแกรมเพื่อสร้างคำสั่งใดๆ ชุดคำสั่งที่สร้างขึ้นจะถูกเก็บลงในอะเรย์ หากผู้ใช้เลือกลบหรือแก้ไขข้อมูล โปรแกรมจะทำการลบ แก้ไขข้อมูลในอะเรย์ที่เกี่ยวข้องด้วย เมื่อผู้ใช้งานโปรแกรมจัดการกับข้อมูลต่างๆ เรียบร้อยแล้ว และกดปุ่มสั่งให้โปรแกรมสร้างคำสั่ง โปรแกรมจะทำการนำคำสั่งต่างๆ ที่เก็บในอะเรย์ออกมาบันทึกลงในไฟล์ที่เก็บไว้ในส่วนของเกมเอ็นจิน จากนั้นผู้ใช้จึงนำเกมไปประมวลผลเล่นได้



รูปที่ 3-3 แผนภาพแสดงการจัดการข้อมูลภายในโปรแกรม

ตัวแปรประเภทอะเรย์ที่ใช้เก็บข้อมูล มีทั้งหมด 8 ตัว และมีตัวชี้ (Pointer) ที่ชี้ไปยังตัวแปรประเภทอะเรย์ชื่อ `arr_temp` อีก 1 ตัว หน้าที่เป็นตัวเลือกใช้อะเรย์ต่างๆ ดังนี้

- `arr_object` : เก็บชื่อ Object ที่ผู้ใช้สร้างขึ้น
- `arr_sound` : เก็บชื่อ Sound (Effect) ที่ผู้ใช้สร้างขึ้น
- `arr_bgsound` : เก็บชื่อ BackgroundSound (Music) ที่ผู้ใช้สร้างขึ้น
- `arr_declare` : เก็บคำสั่งการประกาศตัวแปร ก่อนการบันทึกลงไฟล์ "Declare.h"
- `arr_create` : เก็บคำสั่งการสร้างตัวแปร ก่อนการบันทึกลงไฟล์ "Create.h"
- `arr_delete` : เก็บคำสั่งการคืนค่าหน่วยความจำ ก่อนการบันทึกลงไฟล์ "Delete.h"
- `arr_draw` : เก็บคำสั่งการวาดรูป ก่อนการบันทึกลงไฟล์ "Draw.h"
- `arr_action` : เก็บคำสั่งเหตุการณ์ต่างๆ ก่อนการบันทึกลงไฟล์ "GameCondition.h"
- `arr_update` : เก็บคำสั่งคำนวณการเคลื่อนที่ของวัตถุ ก่อนการบันทึกต่อท้ายไฟล์

"GameCondition.h"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ในการจัดการข้อมูลในตัวแปรประเภทอะเรย์ มีดังนี้

void AddLine( int arrayRef ,CString &txt);	เพิ่มข้อมูลต่อท้าย
void InsertLine(int arrayRef , UINT index ,CString &txt);	เพิ่มข้อมูลในแถวที่ index
void DeleteLine( int arrayRef , UINT index );	ลบข้อมูลในแถวที่ index
CString& GetLineData( int arrayRef, UINT index );	เรียกคืนข้อมูลแถวที่ index
INT GetNumLine( int arrayRef );	เรียกคืนขนาดของอะเรย์
void Reset( int arrayRef);	ลบข้อมูลทั้งหมดในอะเรย์
BOOL ReadFileToArray(LPCTSTR filename,int arrayRef);	อ่านข้อมูลจากไฟล์ลงอะเรย์
BOOL WriteArrayToFile(LPCTSTR filename,int arrayRef);	เขียนข้อมูลจากอะเรย์ลงไฟล์

เมื่อ arrRef เป็นชื่อเรียกของอะเรย์ที่ต้องการจัดการที่กำหนดค่าไว้ (define) ในตอนต้นของไฟล์ gameCreator.h

OBJECT	อ้างถึง	arr_object
BGSOUND	อ้างถึง	arr_bgSound
SOUND	อ้างถึง	arr_sound
DECLARE	อ้างถึง	arr_declare
CREATE	อ้างถึง	arr_create
DEL	อ้างถึง	arr_delete
DRAW	อ้างถึง	arr_draw
ACTION	อ้างถึง	arr_action
UPDATE	อ้างถึง	arr_update

ตัวแปรประเภทอะเรย์และชุดคำสั่งสำหรับจัดการข้อมูลในตัวแปรเหล่านี้ทั้งหมด ถูกสร้างไว้ในไฟล์ gameCreator.cpp และต้องประกาศชื่อชุดคำสั่งสำหรับจัดการข้อมูลไว้ในไฟล์ gameCreator.h เพื่อให้ไฟล์อื่นๆ สามารถเรียกใช้งานตัวแปรอะเรย์ได้ จะต้องเพิ่มคำสั่ง #include “gameCreator.h” ไว้ที่ต้นไฟล์นั้นๆ ด้วย

## 2. ส่วนของเกมเอ็นจิน

ในโปรแกรมจะมีการเขียนโปรแกรมเอาส่วนต่าง ๆ ที่ผู้ใช้กำหนดไว้ไอนเทอร์เฟซลงในไฟล์ซึ่งเป็นไฟล์นามสกุลเฮช (.h) เพื่อใช้ในการคอมไพล์ส่วนของไฟล์นั้นออกมาเป็นเกมให้เล่นได้ ซึ่งไฟล์ต่างๆ มีดังนี้

```
Declare.h      : #include "Declare.h"
Create.h       : #include "Create.h"
Delete.h       : #include "Delete.h"
Draw.h         : #include "Draw.h"
GameCondition.h : #include "GameCondition.h"
ไฟล์เหล่านี้จะถูกเรียกใช้โดย โปรแกรม Test.cpp
```

การทำงานของไฟล์ มีดังนี้

### Declare.h

เป็นไฟล์ที่ใช้เก็บตัวแปรต่างๆ ที่ต้องประกาศไว้ ซึ่งจะอยู่ในส่วนของการประกาศตัวแปรแบบโกลบอล

### Create.h

เป็นไฟล์ที่ใช้เก็บคำสั่งในการโหลดภาพพื้นหลังหรืออ็อบเจ็คต่างๆ การโหลดเสียงดนตรี การโหลดเสียงเอฟเฟ็ค ซึ่งจะอยู่ในฟังก์ชัน `cdx_Init()`

### Delete.h

เป็นไฟล์ที่ใช้เก็บคำสั่งการคืนหน่วยความจำของอ็อบเจ็คต่างๆที่ได้โหลดเข้ามาในเกม ซึ่งจะอยู่ในฟังก์ชัน `cdx_DeInit( void )`

### Draw.h

เป็นไฟล์ที่ใช้เก็บคำสั่งในการวาดอ็อบเจ็คต่างๆ แสดงอักษรหรือข้อความต่างๆ ให้ปรากฏในเกม เช่น พลัง ชีวิต คะแนน เป็นต้น ซึ่งจะอยู่ในฟังก์ชัน `cdx_DoFrame()`

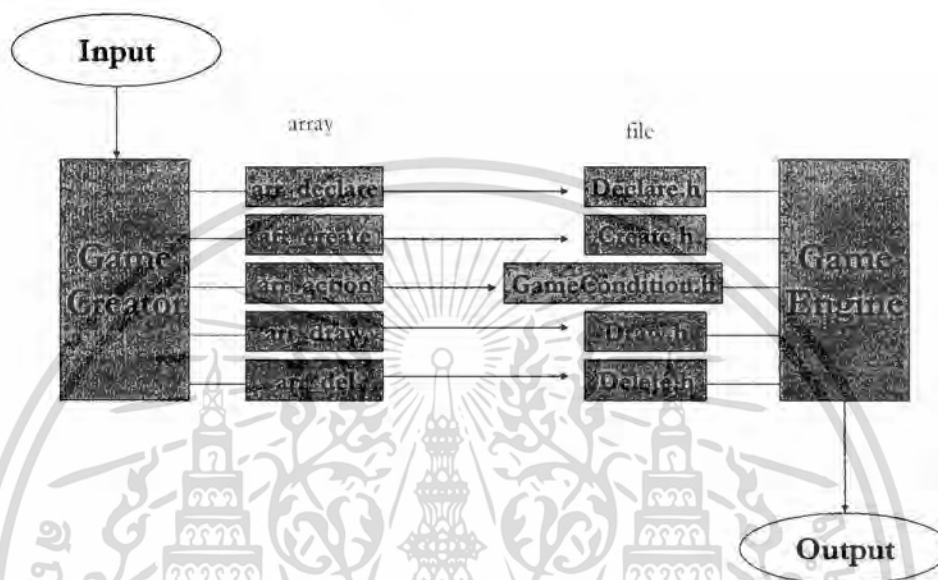
### GameCondition.h

เป็นไฟล์ที่ใช้เก็บพวกคำสั่งในการติดต่อกับคีย์บอร์ดและเมาส์ ซึ่งจะอยู่ในฟังก์ชัน

`cdx_DoFrame()`

### การทำงานของโปรแกรม

โปรแกรมจะรับข้อมูลเกมที่ใช้ป้อนเข้ามา เช่น รายละเอียดตัวละคร ฉาก เสียงเพลง และเงื่อนไขเกม เก็บไว้ในอะเรย์ต่างๆ เมื่อผู้ใช้คลิกปุ่ม "สร้างชุดคำสั่ง" โปรแกรมจะย้ายข้อมูลจากอะเรย์ไปยังไฟล์นามสกุล .h ในส่วนของเกมเอ็นจิน และนำเกมเอ็นจินนี้ไปประมวลผลออกมาเป็นเกมที่ต้องการ



รูปที่ 3-4 แสดงแผนภาพการทำงานของโปรแกรม

### 3.2 การออกแบบหน้าต่างโปรแกรม

อินเตอร์เฟซของโปรแกรมเป็นส่วนสำคัญส่วนหนึ่ง เพราะเป็นส่วนที่ผู้พัฒนาเกมต้องสัมผัสอยู่บ่อยครั้ง การออกแบบอินเตอร์เฟซที่ดีจะช่วยให้การใช้โปรแกรมนั้น เราสามารถเข้าใจง่าย ไม่สับสน ซึ่งหมายถึงการจัดวางตำแหน่งต่างๆ ของเมนู ทูลบาร์ ปุ่มกดต่างๆ ส่วนแสดงหลักและอื่น ๆ

อินเตอร์เฟซของโปรแกรมแบ่งออกเป็น 4 ส่วน ดังนี้

- ส่วนของโปรแกรมหลัก
- ส่วนของการสร้างพื้นที่เก็บไฟล์เกม
- ส่วนของการเพิ่มและลบวัตถุ
- ส่วนของหน้าต่างสร้างเงื่อนไขเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 ส่วนของโปรแกรมหลัก

ประกอบด้วยเมนูเรียกหน้าต่างอื่นๆ ดังนี้

- **เพิ่ม**  
สำหรับเรียกหน้าต่างที่ให้ผู้ใช้งานกำหนดพื้นที่เก็บไฟล์ , เมนูออกจากโปรแกรม
- **เพิ่ม**  
เมนูสำหรับเพิ่มวัตถุต่างๆ ที่ใช้ในเกม ได้แก่ ตัวละคร, ฉากพื้นหลัง , เสียงประกอบ , เสียงดนตรี และเงื่อนไขเกมต่างๆ
- **ลบ**  
เมนูสำหรับลบวัตถุที่ไม่ได้ใช้งานออกจากโปรแกรม

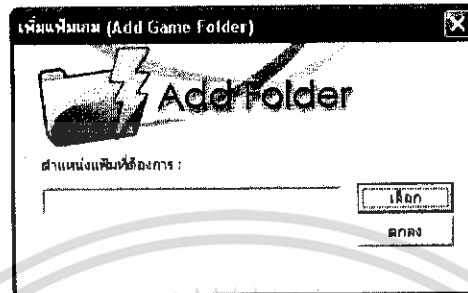


รูปที่ 3-5 แสดงหน้าต่างหลักของโปรแกรมช่วยสร้างเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 ส่วนของการสร้างพื้นที่เก็บไฟล์เกม

เป็นหน้าต่างโปรแกรมแรกที่ใช้จะต้องใช้งานเพื่อสร้างพื้นที่เก็บไฟล์เกม ผู้ใช้จะต้องกดปุ่มเลือกพื้นที่สำหรับเก็บไฟล์เกม เมื่อกดปุ่มตกลง โปรแกรมจะทำการทำสำเนาไฟล์ต่างๆ จากไดเรกทอรี C:\Resource ไปไว้ที่ไดเรกทอรีที่ผู้ใช้เลือกไว้สำหรับเตรียมการสร้างเกมในขั้นตอนถัดไป



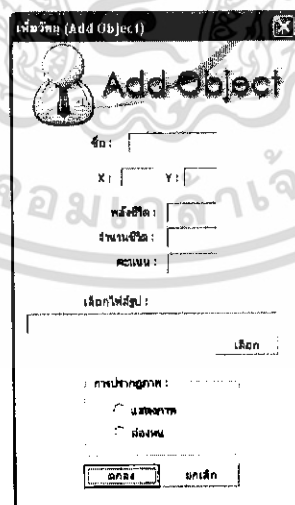
รูปที่ 3-6 แสดงหน้าต่างการเลือกตำแหน่งการเก็บไฟล์เกม

### 3.2.3 ส่วนของการเพิ่มและลบวัตถุ

#### 3.2.3.1 ส่วนของการเพิ่มวัตถุ

##### 3.2.3.1.1 ส่วนของการเพิ่มตัวละคร (Object)

เมื่อกดปุ่มตกลง โปรแกรมจะทำการทำสำเนารูปภาพที่ผู้ใช้เลือกไปไว้ในไดเรกทอรีที่ผู้ใช้เลือกสำหรับเก็บไฟล์เกมในขั้นตอนแรกสุด และเก็บค่าต่างๆ ลงในอะเรย์ต่างๆ ดังนี้



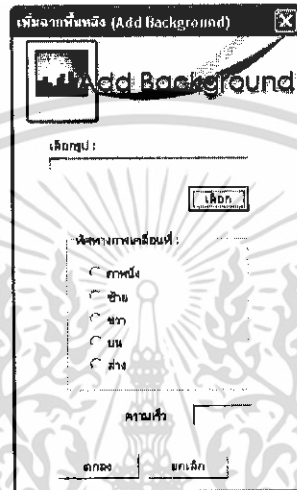
รูปที่ 3-7 แสดงหน้าต่างการเพิ่มตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.1.2 ส่วนของการเพิ่มพื้นหลัง (Background)

เมื่อกดปุ่มตกลง โปรแกรมจะทำการทำสำเนารูปภาพที่ผู้ใช้เลือกไปไว้ในไดเรกทอรีที่ผู้ใช้เลือกสำหรับเก็บไฟล์เกมในขั้นตอนแรกสุด และเก็บค่าต่างๆ ลงในอะเรย์ ดังนี้

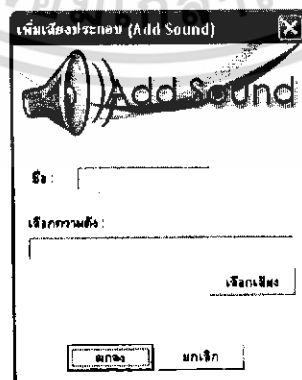
การเพิ่มพื้นหลังนี้จะทำได้เพียงครั้งเดียว หากต้องการแก้ไขรายละเอียดเกี่ยวกับพื้นหลัง ให้แก้ไขในส่วนของการเพิ่มเงื่อนไขของเกม



รูปที่ 3-8 แสดงหน้าต่างการเพิ่มพื้นหลัง

### 3.2.3.1.3 ส่วนของการเพิ่มเสียงประกอบ (Sound)

เมื่อกดปุ่มตกลง โปรแกรมจะทำการทำสำเนาเสียงที่ผู้ใช้เลือกไปไว้ในไดเรกทอรีที่ผู้ใช้เลือกสำหรับเก็บไฟล์เกมในขั้นตอนแรกสุด และเก็บค่าต่างๆ ลงในอะเรย์ ดังนี้

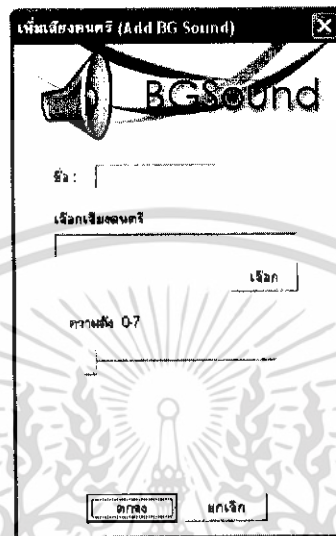


รูปที่ 3-9 แสดงหน้าต่างการเพิ่มเสียงประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.1.4 ส่วนของการเพิ่มเสียงเพลง ( Background Sound )

เมื่อกดปุ่มตกลง โปรแกรมจะทำการทำสำเนาเสียงที่ผู้ใช้เลือกไปไว้ในโคเร็กทอรีที่ผู้ใช้เลือกสำหรับเก็บไฟล์เกมในขั้นตอนแรกสุด และเก็บค่าต่างๆ ลงในอะเรย์ ดังนี้



รูปที่ 3-10 แสดงหน้าต่างการเพิ่มเสียงเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มวัตถุต่างๆ จะมีการเพิ่มคำสั่งต่างๆ ลงในอระเรย์ต่างๆ สรุปได้ดังนี้

หน้าต่างเพิ่มวัตถุ	อระเรย์ที่ถูกเพิ่มคำสั่ง	คำสั่งที่เพิ่มเข้าไปในอระเรย์
เพิ่มตัวละคร (Object)	arr_object	objectName
	arr_declare	Object objectName = NULL;
	arr_create	objectName = Object(CDXScreen* sc, int x, int y, int health, int life, LPCTSTR spriteFileName, long score, bool visible);
	arr_delete	delete objectName;
	arr_draw	objectName->Draw(Screen);
	เพิ่มฉาก (Background)	arr_create
เพิ่มเสียงประกอบ (Sound)	arr_sound	soundName
	arr_declare	Object soundName = NULL;
	arr_create	soundName = SoundEffect(HWND g_hWnd, CDXSoundBuffer* buffer, LPTSTR fileName);
	arr_delete	delete soundName;
เพิ่มเสียง (Background Sound)	arr_BGSound	BGSoundName
	arr_declare	Object BGSoundName = NULL;
	arr_create	BGSoundName = Music(HWND g_hWnd, LPTSTR fileName, int volume);
	arr_delete	delete soundName;

### ตารางที่ 3-5 ตารางแสดงการเพิ่มคำสั่งลงตัวแปรอระเรย์ต่างๆ เมื่อมีการเพิ่มวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.2 ส่วนของลบวัตถุ

#### 3.2.3.2.1 ส่วนของลบตัวละคร ( Object )

เมื่อกดปุ่มเลือกชื่อตัวละครที่ต้องการลบแล้วกดปุ่มตกลงจะลบชื่อตัวละครออกจาก arr\_object



รูปที่ 3-11 แสดงหน้าต่างการลบชื่อตัวละคร

#### 3.2.3.2.2 ส่วนของลบตัวเสียงประกอบ ( Sound )

เมื่อกดปุ่มเลือกชื่อเสียงประกอบที่ต้องการลบแล้วกดปุ่มตกลงจะลบชื่อเสียงออกจาก arr\_sound



รูปที่ 3-12 แสดงหน้าต่างการลบชื่อเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.2.3 ส่วนของลบเสียงเพลง ( Background Sound )

เมื่อกดปุ่มเลือกชื่อเสียงประกอบที่ต้องการลบแล้วกดปุ่มตกลงจะลบชื่อเสียงออกจาก arr\_BGSound



รูปที่ 3-13 แสดงหน้าต่างการลบชื่อเพลง

การลบวัตถุจะทำการลบข้อมูลในอะเรย์ดังนี้

หน้าต่างลบวัตถุ	อะเรย์ที่ถูกลบคำสั่ง
ส่วนของลบตัวละคร ( Object )	arr_object
ส่วนของลบตัวเสียงประกอบ ( Sound )	arr_sound
เสียงเพลง ( Background Sound )	arr_BGSound

ตารางที่ 3-6 ตารางแสดงการลบชื่อวัตถุในอะเรย์ต่างๆ เมื่อมีการลบวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4 ส่วนของหน้าต่างสร้างเหตุการณ์

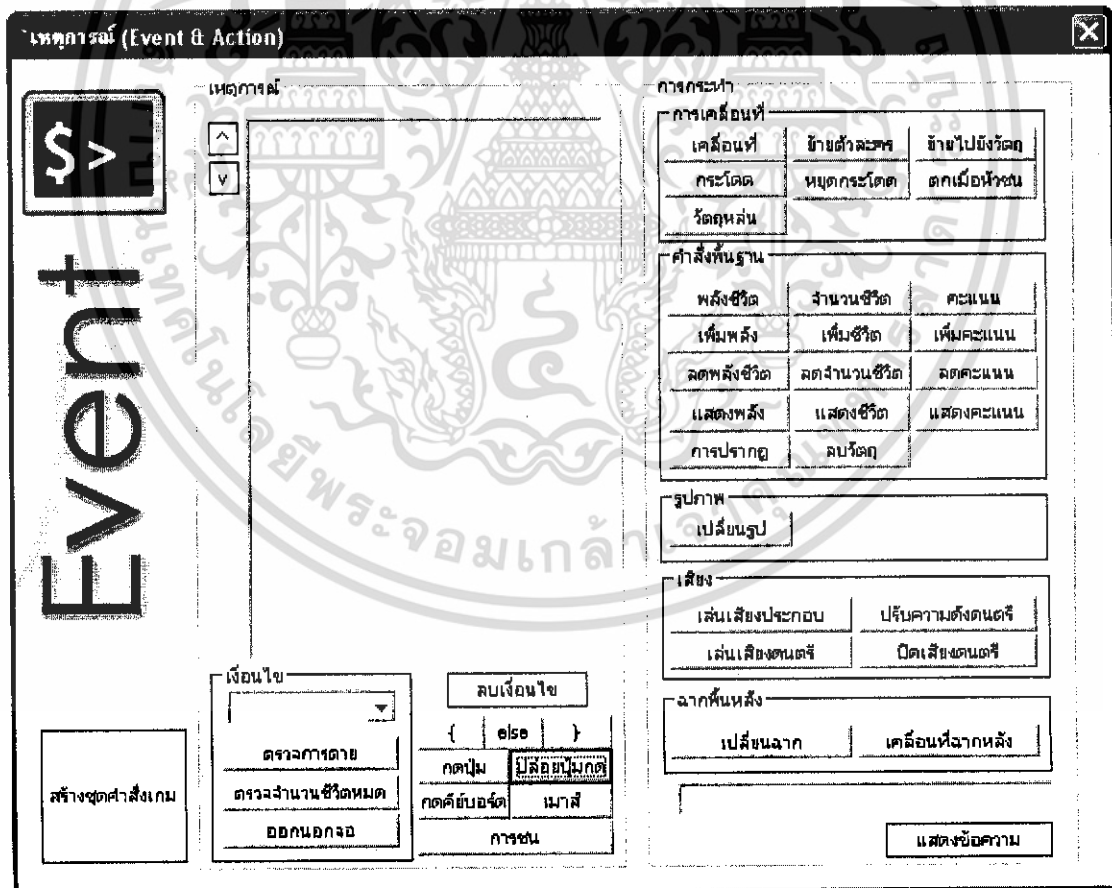
เป็นหน้าต่างสำหรับสร้างเหตุการณ์ของเกมโดยการกดเลือกที่ปุ่มในส่วนของเงื่อนไข และ คำสั่ง(Action) จะเข้าสู่การเพิ่มเงื่อนไขเกมที่เลือกแล้วจะเก็บคำสั่งลงอะเรย์ arr\_action และจะแสดงคำสั่งออกมาในกล่องข้อความ (Event List Box)

ถ้าต้องการให้เงื่อนไขใดๆ มีการทำคำสั่งเกมมากกว่า 1 คำสั่ง ให้เลือกกดเมาส์ที่ปุ่ม { เป็นการเริ่มต้นชุดคำสั่ง และกดปุ่ม } เป็นการจบคำสั่ง หากต้องการลบเงื่อนไขหรือคำสั่งที่ไม่ต้องการก็สามารถทำได้โดยการกดปุ่มลบเงื่อนไขได้

เมื่อต้องการสร้างไฟล์เกม กดปุ่มสร้างชุดคำสั่งเกมทางซ้ายล่างของหน้าต่าง โปรแกรมจะทำการนำข้อมูลทั้งหมดในอะเรย์ที่เกี่ยวข้องกับไฟล์เกมลงไปไฟล์เกม ดังที่กล่าวไว้ในหัวข้อ 3.1.4 โครงสร้างของโปรแกรม ส่วนของแอปพลิเคชัน หน้า 39

หน้าต่างสร้างเหตุการณ์เกม แบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

1. ส่วนของเงื่อนไข (Event)
2. ส่วนของคำสั่ง (Action)



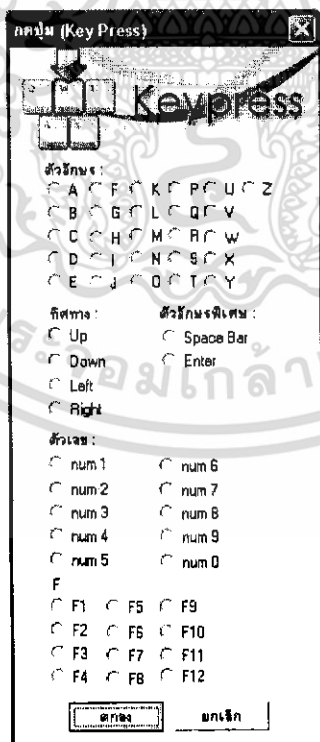
รูปที่ 3-14 ภาพแสดงหน้าต่างสร้างเหตุการณ์เกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.1 ส่วนของเงื่อนไขเกม (Event)

เมื่อผู้ใช้เลือกชื่อตัวละคร จะสามารถกดปุ่มตรวจการตาย (Dead ) และปุ่มตรวจจำนวนชีวิตหมด (Game Over) ได้

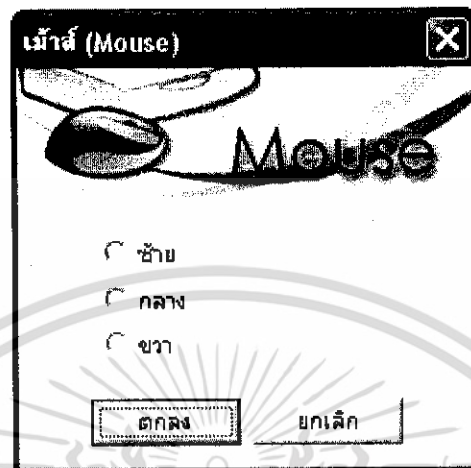
- **ปุ่มตรวจการตาย (Dead )**  
เป็นการสร้างเงื่อนไขว่า ถ้าตัวละครนั้นตาย (พลังชีวิตเป็นศูนย์) จะทำคำสั่ง
- **ปุ่มตรวจจำนวนชีวิตหมด (Game Over )**  
เป็นการสร้างเงื่อนไขว่า ถ้าตัวละครนั้นไม่เหลือจำนวนชีวิตแล้ว จะทำคำสั่ง
- **ปุ่มออกนอกจอ ( Out Screen )**  
เป็นการสร้างเงื่อนไขว่า ถ้าวัตถุอยู่นอกจอแล้ว จะทำคำสั่ง
- **ปุ่มกดคีย์บอร์ด ( Key Board )**  
เป็นการสร้างเงื่อนไข เมื่อมีการกดปุ่มใดๆ บนคีย์บอร์ด



รูปที่ 3-15 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการกดปุ่มคีย์บอร์ด

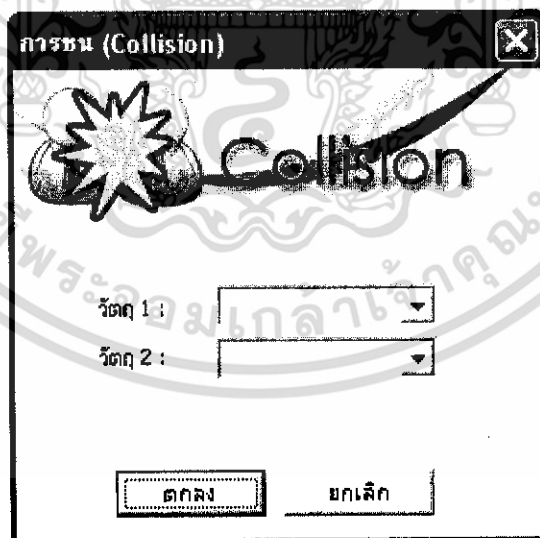
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่มคลิกเมาส์ (Mouse)  
เป็นการสร้างเงื่อนไข เมื่อมีการคลิกเมาส์



รูปที่ 3-16 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการกดคลิกเมาส์

- ปุ่มการชนกันของตัวละคร (Collision)  
เป็นการสร้างเงื่อนไข เมื่อตัวละครชนกัน



รูปที่ 3-17 แสดงหน้าต่างการเลือกเหตุการณ์เมื่อมีการชนกันของวัตถุ 2 ชิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใส่เงื่อนไขต่างๆ จะมีการเพิ่มคำสั่งต่างๆ ลงในอะเรย์ arr\_action ได้ดังนี้

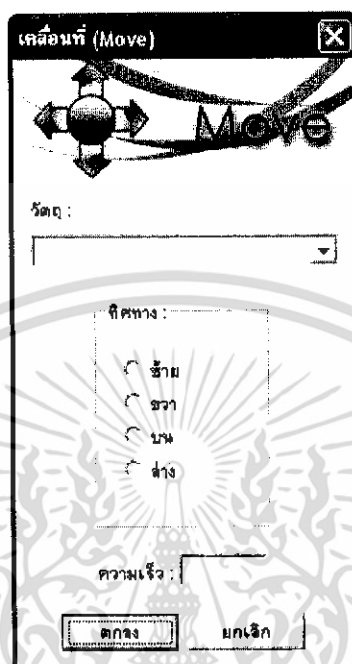
ปุ่มเงื่อนไข	คำสั่งที่ถูกใส่ลง arr_action
ตาย ( Dead )	if ( objectName->CheckDead() )
เกมโอเวอร์( Game Over )	if ( objectName->CheckGameOver() )
การออกนอกจอ ( Out Screen )	if ( objectName->CheckOutScreen() )
กดคีย์เพรส ( KeyPress )	กดปุ่มบน ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_UPARROW) ==CDXKEY_PRESS ) กดปุ่ม Enter ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_ENTER) ==CDXKEY_PRESS ) กดปุ่ม Space ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_SPACE) ==CDXKEY_PRESS ) อื่นๆ เช่น กดปุ่มA ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_A ) ==CDXKEY_PRESS ) กดปุ่ม 0 ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_NUM0 ) ==CDXKEY_PRESS ) กดปุ่ม F1 ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_F1 ) ==CDXKEY_PRESS )
กดคีย์รีลีส ( KeyRelease )	*คำสั่งเหมือนกับกดคีย์เพรสแต่เปลี่ยน CDXKEY_PRESS เป็น CDXKEY_RELEASE
กดคีย์บอร์ด ( KeyBoard )	*คำสั่งเหมือนกับกดคีย์เพรสแต่เปลี่ยน CDXKEY_PRESS เป็น CDXKEY_REPEAT
คลิกเมาส์ ( Mouse )	คลิกซ้าย ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_MOUSELEFT ) ) คลิกกลาง ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_MOUSEMIDDLE ) ) คลิกขวา ใส่คำสั่ง if ( Input->GetKeyState( CDXKEY_MOUSERIGHT ) )
การชน ( Collision )	if ( objectName1->Collision( objectName2 ) )

ตารางที่ 3-7 ตารางแสดงคำสั่งที่เก็บลงอะเรย์ arr\_action เมื่อมีการกดปุ่มเหตุการณ์ใดๆ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2 ส่วนของคำสั่ง (Action)

#### 3.2.4.2.1 การเคลื่อนที่ (Move)

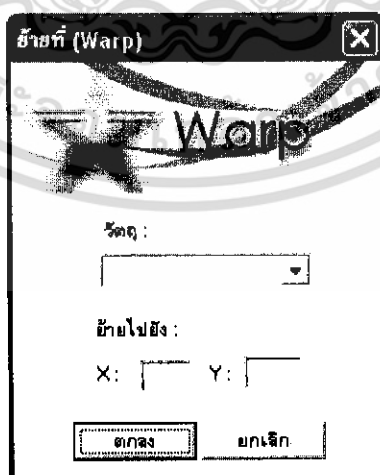
สั่งให้ตัวละครเคลื่อนที่ตามทิศทางและความเร็วที่ต้องการ



รูปที่ 3-18 แสดงหน้าต่างการเคลื่อนที่ของตัวละคร

#### 3.2.4.2.2 ย้ายที่ (Warp)

ให้ตัวละครย้ายที่ไปยังพิกัด x,y ที่กำหนด

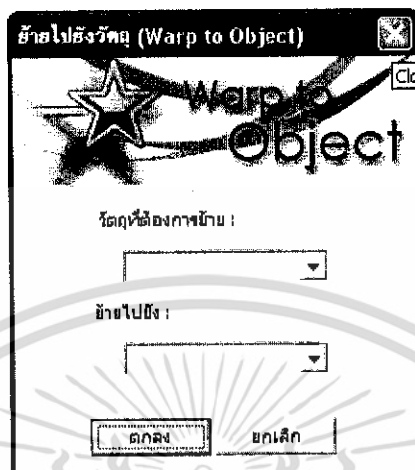


รูปที่ 3-19 แสดงหน้าต่างการย้ายที่ของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.3 ย้ายตัวละคร ( Warp To Object )

ให้ตัวละครย้ายที่ไปอยู่ในตำแหน่งของตัวละครตัวอื่น



รูปที่ 3-20 แสดงหน้าต่างการย้ายตัวละคร

### 3.2.4.2.4 กำหนดพลังชีวิต ( Health )

กำหนดค่าพลังชีวิตให้กับตัวละคร

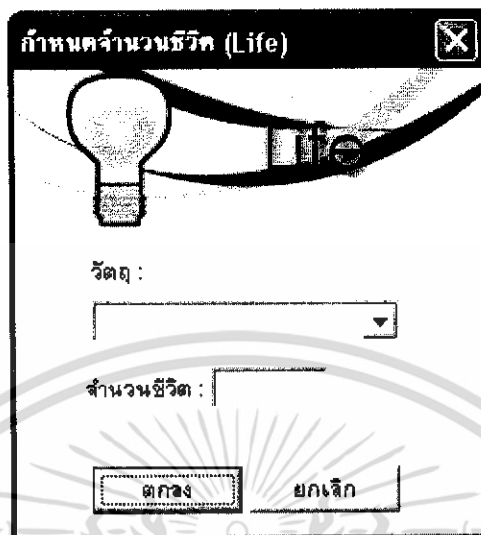


รูปที่ 3-21 แสดงหน้าต่างการกำหนดค่าพลังชีวิตของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.5 กำหนดจำนวนชีวิต ( Life )

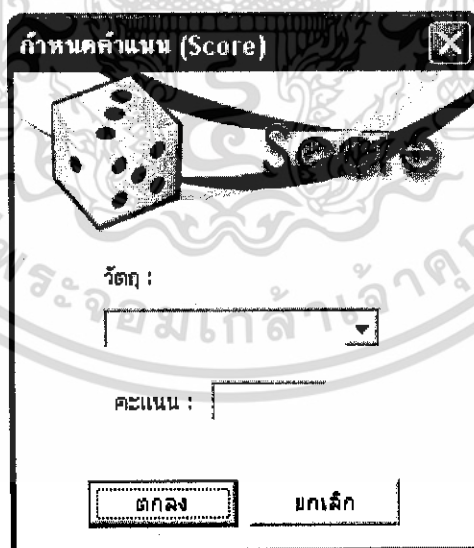
กำหนดจำนวนชีวิตให้กับตัวละคร



รูปที่ 3-22 แสดงหน้าต่างการกำหนดจำนวนชีวิตของตัวละคร

### 3.2.4.2.6 กำหนดคะแนน ( Score )

กำหนดคะแนนให้กับตัวละคร

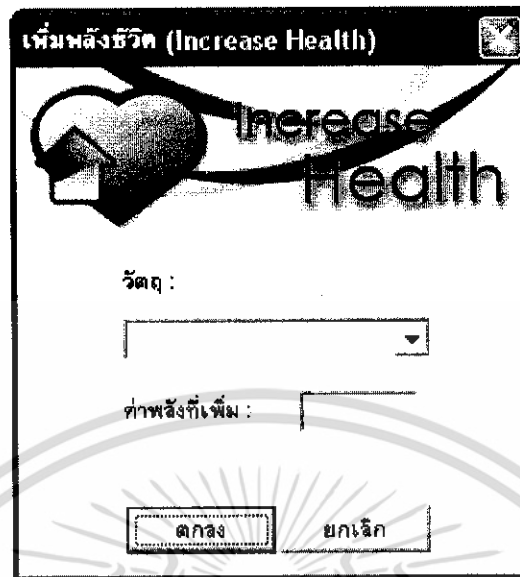


รูปที่ 3-23 แสดงหน้าต่างการกำหนดคะแนนของตัวละคร

### 3.2.4.2.7 เพิ่มพลังชีวิต ( Increase Health )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

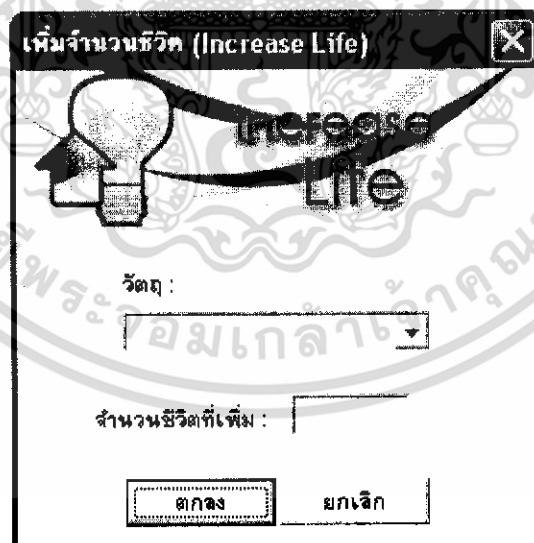
### เพิ่มพลังชีวิตให้กับตัวละคร



รูปที่ 3-24 แสดงหน้าต่างการเพิ่มพลังชีวิตให้กับตัวละคร

#### 3.2.4.2.8 เพิ่มจำนวนชีวิต ( Increase Life )

##### เพิ่มจำนวนชีวิตให้กับตัวละคร

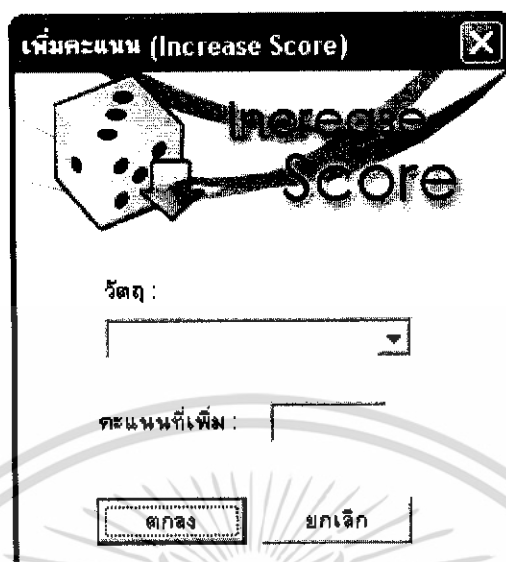


รูปที่ 3-25 แสดงหน้าต่างการเพิ่มจำนวนชีวิตตัวละคร

#### 3.2.4.2.9 เพิ่มคะแนน ( Increase Score )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

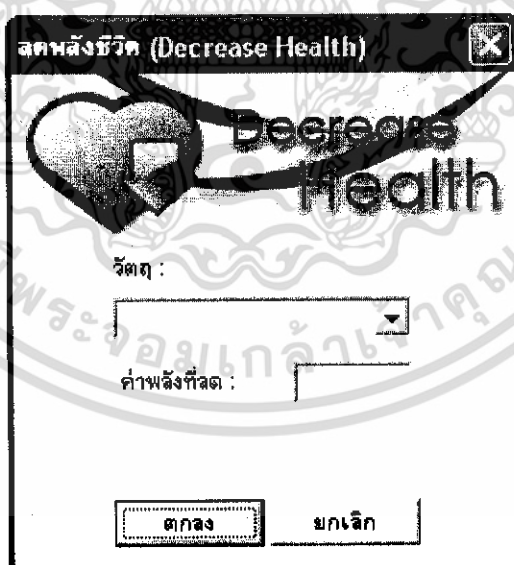
## เพิ่มคะแนน



รูปที่ 3-26 แสดงหน้าต่างการเพิ่มคะแนน

## 3.2.4.2.10 ลดพลังชีวิต (Decrease Health)

ลดพลังชีวิตของตัวละคร

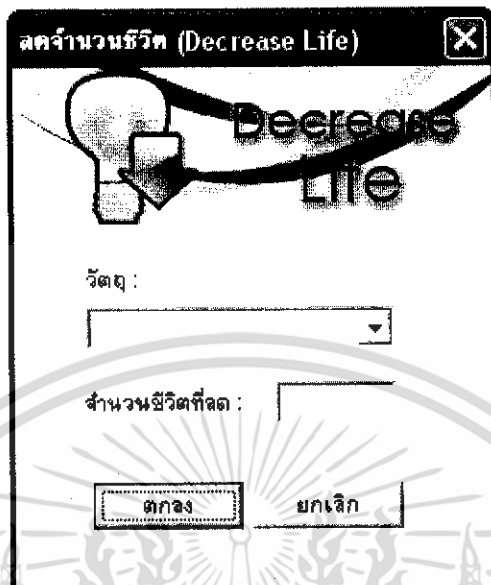


รูปที่ 3-27 แสดงหน้าต่างการลดพลังชีวิตของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.11 ลดจำนวนชีวิต ( Decrease Life )

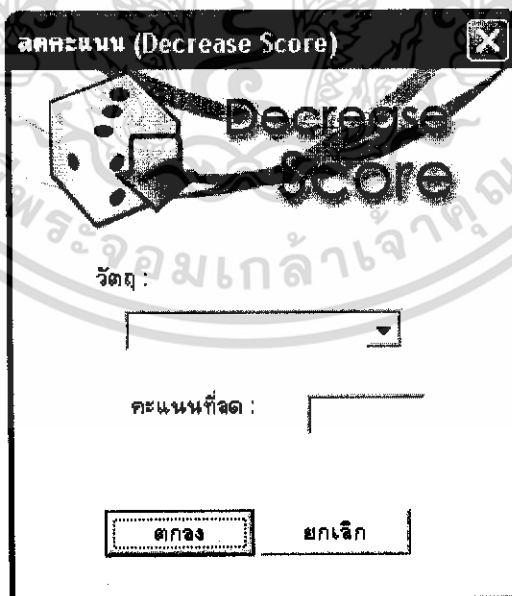
ลดจำนวนชีวิตของตัวละคร



รูปที่ 3-28 แสดงหน้าต่างการลดจำนวนชีวิตของตัวละคร

### 3.2.4.2.12 ลดคะแนน ( Decrease Score )

ลดคะแนน

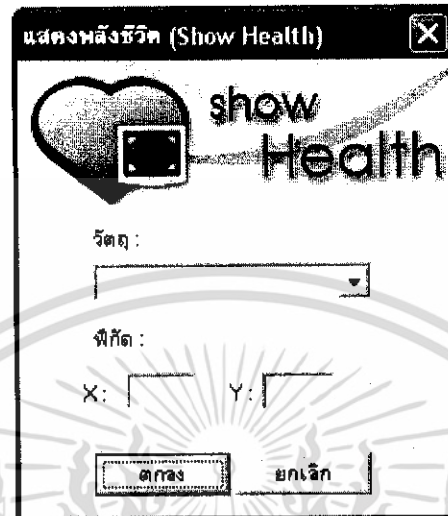


รูปที่ 3-29 แสดงหน้าต่างการลดคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.13 แสดงพลังชีวิต ( Show Health )

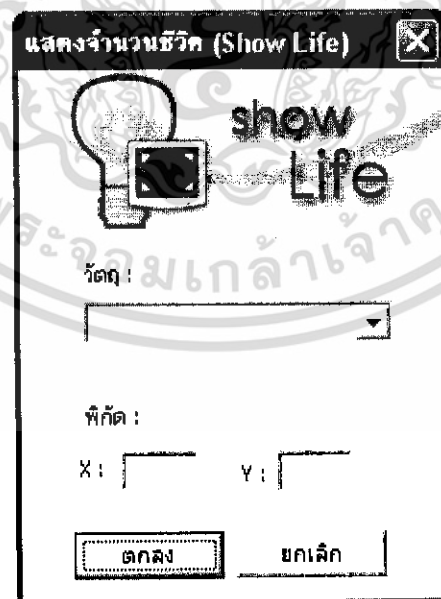
แสดงพลังชีวิตของตัวละครบนหน้าจอเกม



รูปที่ 3-30 แสดงหน้าต่างการแสดงผลพลังชีวิตบนหน้าจอเกม

### 3.2.4.2.14 แสดงจำนวนชีวิต ( Show Life )

แสดงจำนวนชีวิตของตัวละครบนหน้าจอเกม

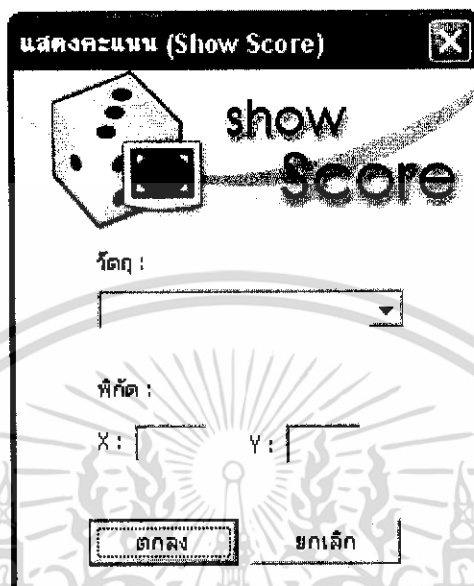


รูปที่ 3-31 แสดงหน้าต่างการแสดงผลจำนวนชีวิตบนหน้าจอเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.15 แสดงคะแนน ( Show Score )

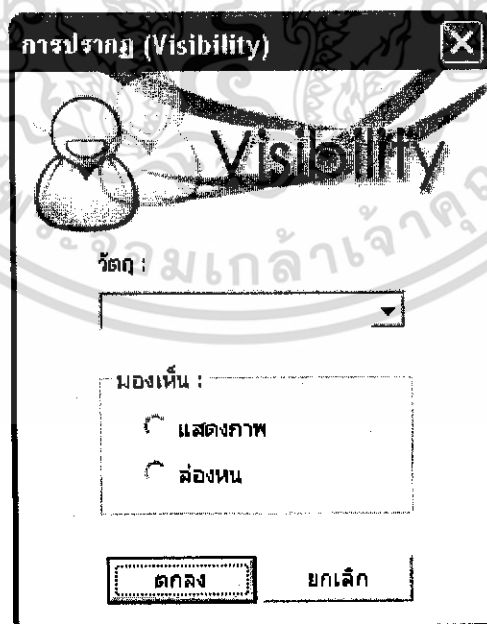
แสดงพลังชีวิตของตัวละครบนหน้าจอเกม



รูปที่ 3-32 แสดงหน้าต่างการแสดงผลคะแนนบนหน้าจอเกม

### 3.2.4.2.16 การปรากฏ ( Visibility )

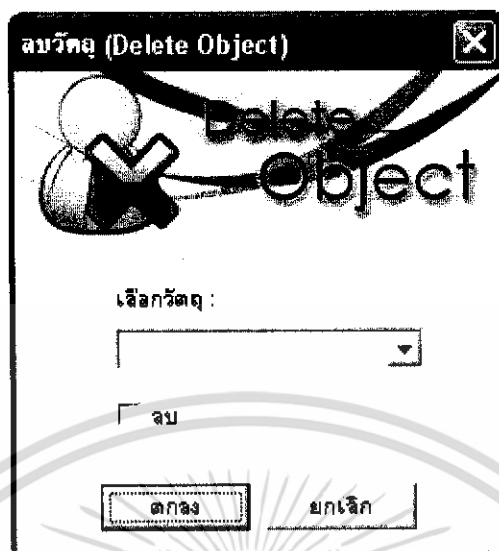
กำหนดให้ตัวละครแสดงภาพออกมาที่หน้าจอหรือให้หายตัวไป



รูปที่ 3-33 แสดงหน้าต่างการกำหนดการปรากฏ

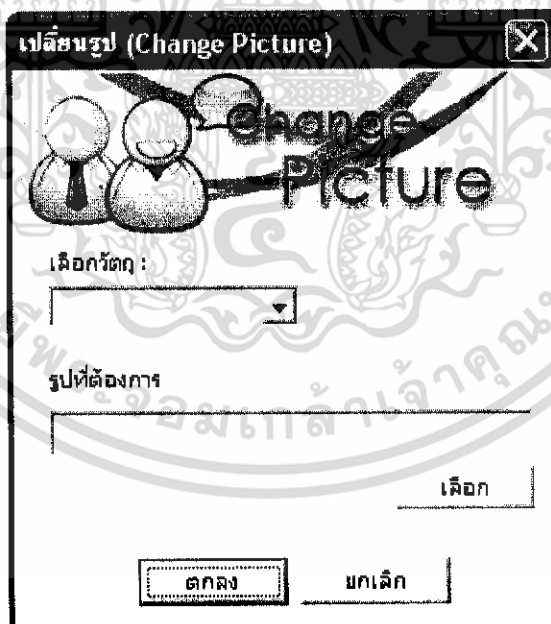
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.17 ลบตัวละคร ( Delete Object )



รูปที่ 3-34 แสดงหน้าต่างการลบตัวละคร

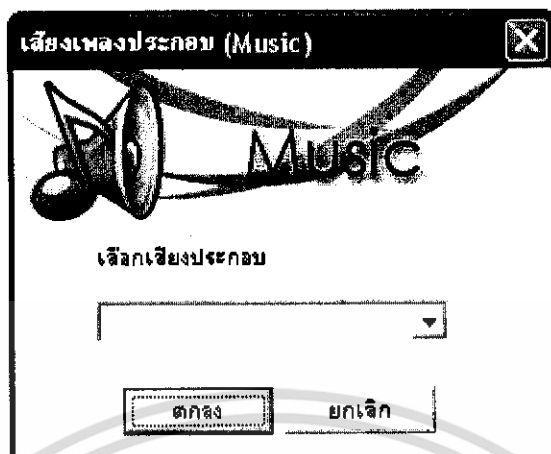
### 3.2.4.2.18 เปลี่ยนรูปตัวละคร ( Change Picture )



รูปที่ 3-35 แสดงหน้าต่างการเปลี่ยนรูปตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.19 เล่นเสียงประกอบ (Music)



รูปที่ 3-36 แสดงหน้าต่างการเลือกเล่นเสียงเพลงประกอบ

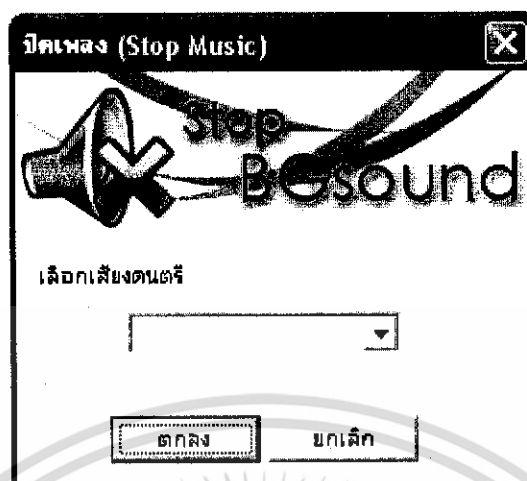
### 3.2.4.2.20 เล่นดนตรี ( Play Music Background)



รูปที่ 3-37 แสดงหน้าต่างการเลือกเล่นเสียงดนตรี

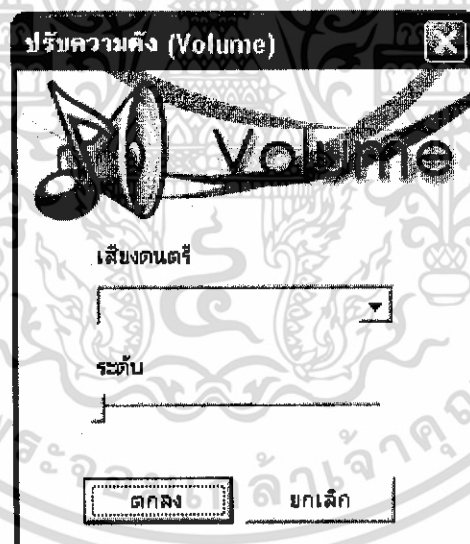
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.21 ปิดเสียงเพลง ( Stop Background Sound )



รูปที่ 3-38 แสดงหน้าต่างการปิดเสียงเพลง

### 3.2.4.2.22 ความดัง ( Volume )

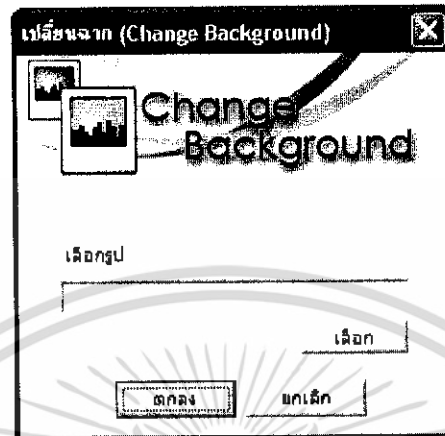


รูปที่ 3-39 แสดงหน้าต่างการปรับความดังของเสียงดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4.2.23 เปลี่ยนฉาก ( Change Background )

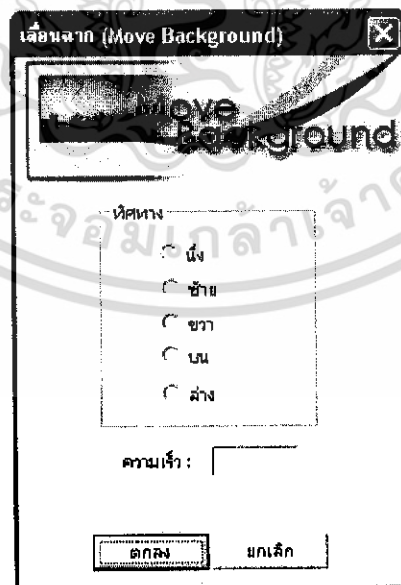
เปลี่ยนฉากพื้นหลัง มีการเลือกฉากใหม่และทำการทำสำเนาภาพเก็บลงในพื้นที่โปรแกรมเกมที่ใช้สร้างขึ้น



รูปที่ 3-40 แสดงหน้าต่างการเปลี่ยนฉาก

### 3.2.4.2.24 เลื่อนและกำหนดความเร็วฉาก ( Move & Set Speed Background )

เปลี่ยนทิศทางการเลื่อนของฉาก และสามารถกำหนดให้ฉากเลื่อนด้วยความเร็วตามที่กำหนด



รูปที่ 3-41 แสดงหน้าต่างการเลื่อนและกำหนดความเร็วของฉาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มคำสั่งต่างๆ จะมีการเพิ่มข้อมูลในอะเรย์ arr\_action สรุปเป็นตารางได้ ดังนี้

หน้าต่างคำสั่ง (Action)	คำสั่งที่เพิ่มลงในอะเรย์ arr_action
เคลื่อนที่ (Move)	เคลื่อนซ้าย : objectName->MoveUp(int speed); เคลื่อนขวา : objectName->MoveDown(int speed); เคลื่อนบน : objectName->MoveLeft(int speed); เคลื่อนล่าง : objectName->MoveRight(int speed);
ย้ายที่ (Warp)	objectName->Warp(int x,int y);
กระโดด (Jump)	objectName->Jump(float vx,float vy, float ax, float ay)
หยุดหลังกระโดด (StopJump)	objectName->StopJump()
ตกจากการชนเพดาน (FallDown)	objectName->FallDown()
ตกด้วยแรงโน้มถ่วง (Drop)	objectName->Drop(float ay)
พลังชีวิต (Health)	objectName->SetHealth(int health);
ชีวิต (Life)	objectName->SetLife(int life);
คะแนน (Score)	objectName->SetScore(long score);
เพิ่มพลังชีวิต (Increase Score)	objectName->IncHealth(int value);
เพิ่มจำนวนชีวิต (Increase Life)	objectName->IncLife(int value);
เพิ่มคะแนน (Increase Score)	objectName->IncScore(long value);
ลดพลังชีวิต (Decrease Health)	objectName->DecHealth(int value);
ลดจำนวนชีวิต (Decrease Life)	objectName->DecLife(int value);
ลดคะแนน (Decrease Score)	objectName->DecScore(long value);
แสดงพลังชีวิต (Show Health)	objectName->ShowHealth(HDC hdc,int xPos,int yPos);
แสดงจำนวนชีวิต (Show Life)	objectName->ShowLife(HDC hdc,int xPos,int yPos);
แสดงคะแนน (Show Score)	objectName->ShowScore(HDC hdc,int xPos,int yPos);
การปรากฏ (Visibility)	ปรากฏ : objectName->Visible(CDXScreen *screen); ล่องหน : objectName->Invisible(CDXScreen *screen);
ลบตัวละคร (Delete)	objectName->Remove(CDXScreen* Screen);
เปลี่ยนรูปตัวละคร (Change Sprite)	objectName->ChangeSprite(CDXScreen* sc,LPCWSTR sFileName);
เล่นเสียงประกอบ (Play Sound)	objectName->Play(void);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เล่นเสียงดนตรี (Play Music)	objectName->Play(void);
หยุดเสียงดนตรี (Stop Music)	objectName->Stop(void);
ความดัง (Volume)	objectName->SetVolume(int v);
เปลี่ยนฉาก (Change Background)	objectName->ChangeBackground(CDXScreen* sc,char* fileName);
เลื่อนและกำหนดความเร็วฉาก ( Move & Set Speed Background)	objectName->SetMoveDirection(int moveDirection); objectName->SetSpeed(int s);
แสดงข้อความ (Show Message)	ShowMessage(HDC hdc,LPCTSTR str);

ตารางที่ 3-8 ตารางแสดงคำสั่งที่เก็บลงอะเรย์ arr\_action เมื่อมีการเลือกการกระทำใดๆ

หน้าต่าง โปรแกรมทั้งหมดจะเขียน สามารถเขียนรวบรวมไว้ในตารางสรุปได้ดังนี้

ชื่อหน้าต่างโปรแกรม	ชื่อไฟล์	ไฟล์ที่เกี่ยวข้อง
สร้างพื้นที่เก็บไฟล์ (AddFolder)	IDD_FOLDER	AddFolder.h , AddFolder.cpp
เพิ่มตัวละคร (AddObject)	IDD_ADD_OBJ	AddObj.h , AddObj.cpp
เพิ่มฉาก (AddBackground)	IDD_ADD_BACKGROUND	AddBackground.h , AddBackground.cpp
เพิ่มเสียงดนตรี (AddSound)	IDD_ADD_SOUND	AddSound.h , AddSound.cpp
เพิ่มเสียงประกอบ (AddMusic)	IDD_ADD_BG_SOUND	AddBackgroundSound.h , AddBackground.cpp
เพิ่มเหตุการณ์ (AddEvent)	IDD_ADD_OBJECT	AddObject.h , AddObject.cpp
ลบตัวละคร (DeleteObject)	IDD_DELETE_OBJ	DeleteObj.h , DeleteObj.cpp
ลบเสียงประกอบ(DeleteSound)	IDD_DELETE_SOUND	DeleteSound.h , DeleteSound.cpp
ลบเสียงดนตรี (DeleteMusic)	IDD_DELETE_BGSOUND	DeleteBGSound.h , DeleteBGSound.cpp
กดคีย์เพรส (KeyPress)	IDD_KEY_PRESS	KeyPress.h , KeyPress.cpp
กดคีย์ลีส (KeyRelease)	IDD_KEY_RELEASE	KeyRelease.h , KeyRelease.cpp
กดคีย์บอร์ด (KeyBoard)	IDD_KEY_KEYBOARD	Keyboard.h , Keyboard.cpp
คลิกเมาส์ (Mouse)	IDD_MOUSE	Mouse.h , Mouse.cpp
การชน (Collision)	IDD_COLISION	Collsion.h , Collision.cpp
เคลื่อนที่ (Move)	IDD_MOVE	Move.h , Move.cpp
ย้ายที่ (Warp)	IDD_WARP	Warp.h , Warp.cpp
ย้ายตัวละคร (WarpToObject)	IDD_WARP_TO_OBJ	WarpToObj.h , WarpToObj.cpp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

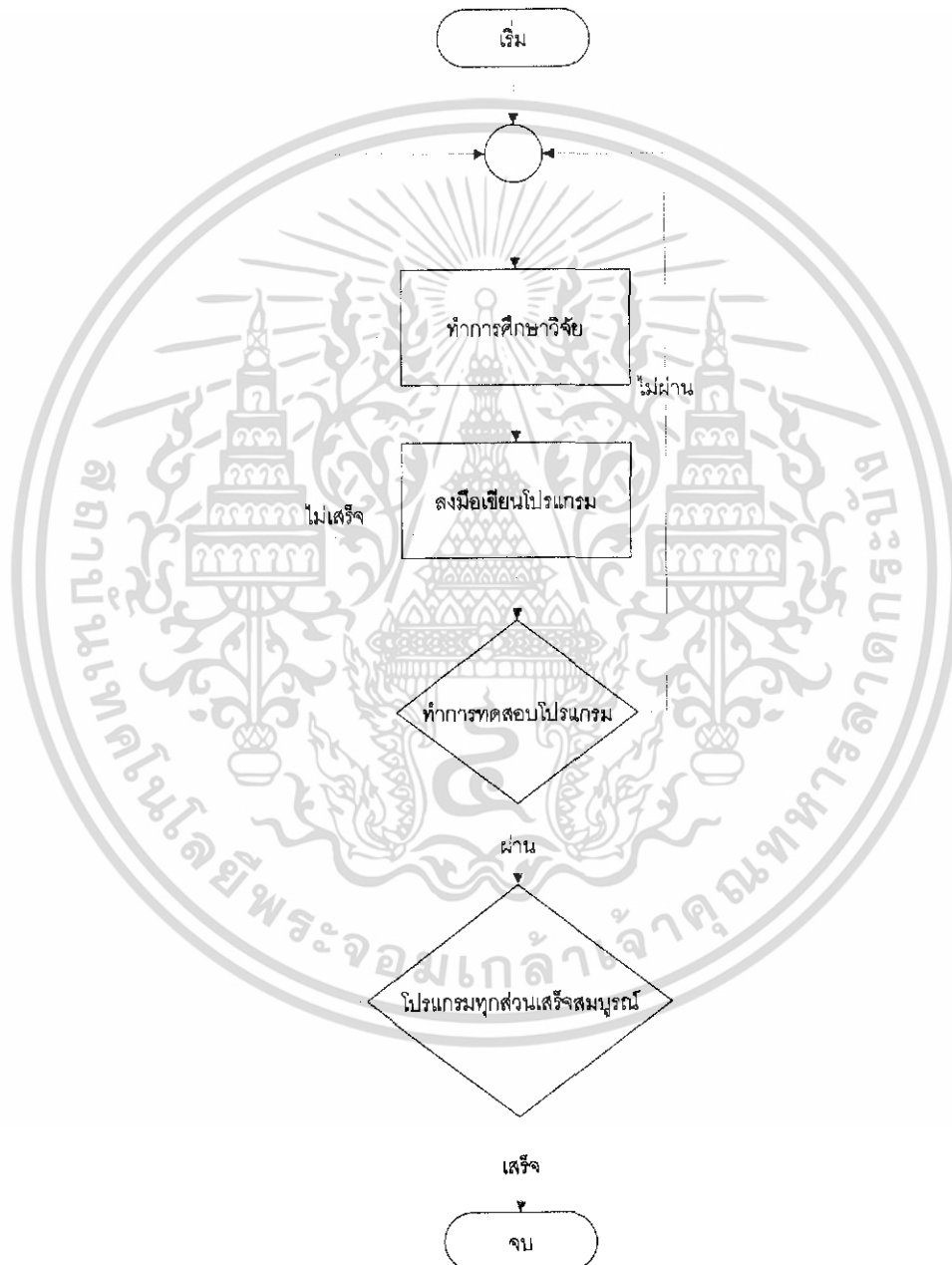
กระโดด (Jump)	IDD_JUMP	Jump.h , Jump.cpp
หยุดหลังกระโดด (StopJump)	IDD_STOPJUMP	StopJump.h , StopJump.cpp
ตกจากการชนเพดาน (FallDown)	IDD_FALLDOWN	FallDown.h , FallDown.cpp
ตกด้วยแรงโน้มถ่วง (Drop)	IDD_DROP	Drop.h , Drop.cpp
กำหนดพลังชีวิต (SetHealth)	IDD_HEALTH	Health.h , Health.cpp
เพิ่มพลังชีวิต (IncreaseHealth)	IDD_INC_HEALTH	IncreaseHealth.h , IncreaseHealth.cpp
ลดพลังชีวิต (DecreaseHealth)	IDD_DEC_HEALTH	DecreaseHealth.h , DecreaseHealth.cpp
แสดงพลังชีวิต (ShowHealth)	IDD_SHOW_HEALTH	ShowHealth.h , ShowHealth.cpp
กำหนดคะแนน (SetScore)	IDD_SCORE	Score.h , Score.cpp
เพิ่มคะแนน (IncreaseScore)	IDD_INC_SCORE	IncreaseScore.h , IncreaseScore.cpp
ลดคะแนน (DecreaseScore)	IDD_DEC_SCORE	DecreaseScore.h , Decrease.cpp
แสดงคะแนน (ShowScore)	IDD_SHOW_SCORE	ShowScore.h , ShowScore.cpp
กำหนดจำนวนชีวิต (SetLife)	IDD_LIFE	Life.h , Life.cpp
เพิ่มจำนวนชีวิต (IncreaseLife)	IDD_INC_LIFE	IncreaseLife.h , IncreaseLife.cpp
ลดจำนวนชีวิต (DecreaseLife)	IDD_DEC_LIFE	DecreaseLife.h , DecreaseLife.cpp
แสดงจำนวนชีวิต (ShowLife)	IDD_SHOW_LIFE	ShowLife.h , ShowLife.cpp
การปรากฏ (Visibility)	IDD_VISIBILITY	Visibility.h , Visibility.cpp
ลบตัวละคร (RemoveObject)	IDD_DELETE	Delete.h , Delete.cpp
เปลี่ยนภาพตัวละคร (ChangeSprite)	IDD_CHANGE_SPRITE	ChangeSprite.h , ChangeSprite.cpp
เปิดเพลงประกอบ (PlaySound)	IDD_PLAY_SOUND	PlayMusicBackground.h, PlayMusicBackground.cpp
ปรับความดัง (SetVolume)	IDD_VOLUME	Volume.h , Volume.cpp
เปิดเสียงดนตรี (PlayMusic)	IDD_PLAY_MUSIC_BG	PlaySound.h , PlaySoundcpp
ปิดเสียงดนตรี (StopMusic)	IDD_STOP_MUSIC_BACKGR OUND	StopMusicBackground.h , StopMusicBackground.cpp
เปลี่ยนภาพพื้นหลัง (ChangeBackground)	IDD_CHANGE_BG	ChangeBg.h , ChangeBg.cpp
เลื่อนฉาก (MoveBackground)	IDD_BG_MOVE	Background.h , Background.cpp

ตารางที่ 3-9 แสดงไฟล์ที่เกี่ยวข้องกับหน้าต่างโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5 ขั้นตอนการเขียนโปรแกรม

ส่วนใหญ่แล้วขั้นตอนต่างๆจะมีความทำงานเป็นเส้นตรง กล่าวคือเมื่อทำงานอย่างหนึ่งเสร็จก็จะทำงานอย่างอื่นต่อ โดยไม่วนมาที่งานนั้นใหม่ แต่ขั้นตอนของการเขียนโปรแกรมนั้นมีความทำงานที่ต้องวนมาศึกษาและทำการเขียนโปรแกรม และหากไม่ผ่านการทดสอบแล้ว ก็ต้องทำการศึกษาเพิ่มเติมใหม่ การเขียนโปรแกรมจะสิ้นสุดได้โปรแกรมในทุกส่วนจะต้องถูกเขียนและผ่านการทดสอบเรียบร้อยแล้ว ดังแผนภาพต่อไปนี้



รูปที่ 3-42 แผนภาพแสดงขั้นตอนการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาและทดสอบโปรแกรม

#### 4.1 การพัฒนาชุดคำสั่งเกมเอนจิน (Game Engine)

เกม เป็นโปรแกรมประเภทหนึ่งที่เน้นความบันเทิงให้กับผู้เล่น โดยมีการทำงานหลักๆ คือ มีการโหลดภาพและเสียง การควบคุมตัวละครด้วยอุปกรณ์ต่างๆ เช่น เมาส์ แป้นพิมพ์ และการใส่เหตุการณ์ที่สร้างความตื่นเต้นให้กับผู้เล่น

ในการเขียนโปรแกรมเกมบนวินโดวส์ นิยมเขียนแบบวินโดวส์ แอปพลิเคชัน (Win32 Application) โดยโปรแกรมช่วยสร้างเกมจะทำการเพิ่มคำสั่งในส่วนต่างๆ ซึ่งจัดกลุ่มแต่ละส่วนเป็นไฟล์นามสกุลเอช (.h) ที่เขียนขึ้นเอง เพื่ออำนวยความสะดวกในการเขียนชุดคำสั่ง จะมีโครงสร้างการเขียนโปรแกรมในลักษณะ ดังต่อไปนี้

- ส่วนของการประกาศตัวแปร เรียกใช้ไฟล์ "Declare.h"

```
#include "Declare.h"
```

```
BOOL cdx_Init() {
```

```
...
```

- ส่วนของการสร้างหน้าต่างเกมและวัตถุที่จะใช้ในเกม เรียกใช้ไฟล์ "Create.h"

```
#include "Create.h"
```

```
return TRUE;
```

```
}
```

- ส่วนของการคืนค่าตัวแปรเมื่อจบเกม เรียกใช้ไฟล์ "Delete.h"

```
void cdx_DeInit( void ){
```

```
    #include "Delete.h"
```

```
}
```

- ส่วนของการเล่นเกม

```
void cdx_DoFrame(){
```

```
    Input->Update();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของเงื่อนไขการเล่นเกม เรียกใช้ไฟล์ `.GameCondition.h`

```
#include "GameCondition.h"
```

```
Screen->GetBack()->Fill(0);
```

- ส่วนของการวาดภาพบนหน้าจอ เรียกใช้ไฟล์ `"Draw.h"`

```
#include "Draw.h"
```

```
Screen->Flip();
```

```
}
```

```
static long PASCAL WinProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam){
```

```
...
```

- ส่วนของการตรวจจับข้อมูลต่างๆ เช่น สัญญาณจากอุปกรณ์ภายนอก การจับเวลา

```
switch(message)
```

```
{
```

```
case WM_CLOSE: cdx_DeInit();
```

```
case WM_DESTROY: PostQuitMessage(0);
```

```
}
```

```
}
```

```
static BOOL InitApp(int nCmdShow){
```

- ส่วนของการสร้างหน้าต่างเกม

```
return TRUE;
```

```
}
```

```
int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst, LPSTR
lpCmdLine, int nCmdShow)
```

```
{
```

```
InitApp(nCmdShow);
```

```
cdx_Init();
```

```
while(1){
```

```
if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

...
}
else{
    cdx_DoFrame();
}
}
}
}

```

โปรแกรมช่วยสร้างเกมได้เตรียมส่วนประกอบดังกล่าวมาให้ผู้ใช้งานได้เลือกใช้ โดยอาศัยชุดคำสั่งซีดีเอ็กซ์ เอนจิน (CDX) และ ไคเร็กซ์ เอนจิน (Direct X) ที่มีความสามารถในการสนับสนุนการพัฒนาโปรแกรมเกม การเขียนโปรแกรมโดยเรียกใช้เอนจินซีดีเอ็กซ์ นั้น จำเป็นต้องใส่คำสั่งขอใช้งาน ดังนี้

```

#define CDXINCLUDEALL
#include <cdx.h>

```

ตัวอย่างคำสั่งต่างๆ ที่ใช้บ่อยในการเขียนเกม

### การโหลดรูป

ในเอนจินซีดีเอ็กซ์ (CDX Engine) นั้น มีชุดคำสั่งโหลดรูป ดังนี้  
สร้างตัวแปรพอยน์เตอร์ที่ชี้ไปที่ตัวแปรประเภท CDXScreen ขึ้นมาสำหรับวาดฉาก

```

CDXScreen *Screen = NULL;
Screen = new CDXScreen();

```

สร้างตัวแปรพอยน์เตอร์ที่ชี้ไปที่ตัวแปรประเภท CDXSprite ขึ้นมาสำหรับทำงานต่างๆ เกี่ยวกับตัวละคร

```

CDXSprite *Sprite = NULL;
Sprite = new CDXSprite();

```

เรียกใช้คำสั่ง Create เพื่อสร้าง โหลดรูปให้กับตัวละคร

```

Sprite->Create(Screen ,ตำแหน่งของรูป , ความกว้างของรูป,
              ความสูงของรูป ,0 , CDXMEM_VIDTHENSYS );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การวาดรูป

เรียกใช้คำสั่ง Draw ก็จะทำให้ภาพถูกวาดลงบนหน้าจอทันที

```
Sprite->Draw(Screen->GetBack() ,x , y ,CDXBLT_TRANS);
```

### การเคลื่อนที่ตัวละคร

เขียนโปรแกรมเพิ่มพิกัด X หรือ Y กรณีที่ต้องการให้ภาพเคลื่อนที่ไปทางซ้าย

เขียนคำสั่ง

```
x = x - 1 ;
```

เมื่อทำการสั่งวาดรูปอีกครั้ง ภาพจะเปลี่ยนตำแหน่งไป 1 จุด

### การควบคุมตัวละครด้วยแป้นพิมพ์

สร้างตัวแปรประเภทพอยเตอร์ที่ชี้ไปยังตัวแปรประเภท CDXInput รับค่าจาก

แป้นพิมพ์

```
CDXInput *Input = NULL;
```

```
Input = new CDXInput();
```

เขียนโปรแกรมควบคุมแป้นพิมพ์ กรณีกดปุ่มซ้าย ไล่เป็นเงื่อนไข

```
if ( Input->GetKeyState()==LEFT_ARROW )
```

```
{
```

```
    x = x - 1 ;
```

```
}
```

เพื่อให้เกิดความสะดวกในการเขียนชุดคำสั่ง จึงทำการสร้างคลาส (Class) ขึ้นมาเพื่อให้ง่ายต่อการใช้งาน ตัวอย่างการเขียนคลาสตัวละครอย่างง่าย แสดงได้ดังนี้

```
#define CDXINCLUDEALL
#include <cdx.h>
#include <stdio.h>

Class Object{
    public:
        Object(CDXScreen* Screen, int x , int y , LPCTSTR spriteFileName,
              int imgWidth , int imgHeight) {
            Sprite = new CDXSprite();
            SetX(x);
            SetY(y);
            SetSprite(Screen , spriteFileName , x , y, imgWidth, imgHeight);
        }
        ~Object(void) {
            SAFEDELETE(Sprite);
        }
        void SetX(int xPos) {
            x = xPos;
        }
        void SetY(int y) {
            y = yPos;
        }
        void SetSprite(CDXScreen* Screen,LPCTSTR fileName,
                      int ImgWidth , int ImgHeight) {
            Sprite->Create(sc ,filename , imgWidth , imgHeight , 0 ,
                          CDXMEM_VIDTHENSYS)
            CDXError( sc , "Not found image.");
            sprite->SetColorKey();
        }
    };
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

int GetX(void) {    return x;    }

int GetY(void) {    return y;    }

void Draw(CDXScreen* Screen) {

    GetSprite()->Draw(Screen->GetBack() , GetX() , GetY() ,

        CDXBLT_TRANS);

}

void MoveLeft(void) {

    SetX( GetX()-1 );

}

private :

    int x,y;

    CDXSprite*    Sprite;

}

```

เมื่อต้องการเขียนคำสั่ง จะทำการเขียนไฟล์นามสกุลเฮช (.h) ทำให้โปรแกรมมีความเป็นระเบียบมากยิ่งขึ้น

ตัวอย่าง เมื่อต้องการ สร้างตัวละครชื่อ hanuman เริ่มวาดรูปที่พิกัด (240,360) โดยมีรูปมาจากไฟล์ที่ “C:\Hanuman.bmp” ขนาดความกว้างยาว 50x70 pixel และสามารถบังคับให้ตัวละครเคลื่อนที่ไปทางซ้ายเมื่อกดแป้นพิมพ์ปุ่มซ้าย จะมีการเขียนโปรแกรมลงไฟล์ต่างๆ ดังนี้

```

ไฟล์ “Declare.h”    Object *hanuman = NULL;

ไฟล์ “Create.h”    hanuman = new Object(Screen,240,360,

                    “c:\Hanuman.bmp”, 50, 70);

ไฟล์ “Delete.h”    delete hanuman;

ไฟล์ “Draw.h”    hanuman->Draw(Screen);

ไฟล์ “GameCondition.h” if (Input->GetKeyState()==LEFT_ARROW)

                    Hanuman->MoveLeft();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การพัฒนาโปรแกรมช่วยสร้างเกม 2 มิติ (2D Game Creator)

ในส่วนของ โปรแกรมช่วยสร้างเกม 2 มิติ ผู้ใช้งานจะเลือกคำสั่งได้ตามต้องการ โปรแกรมจะนำเอาค่าที่ผู้ใช้กรอกเข้ามาสร้างเป็นคำสั่งต่างๆ เก็บไว้ในตัวแปรอะเรย์ และเมื่อผู้ใช้ต้องการจะให้โปรแกรมบันทึกคำสั่งเหล่านั้นลงในตัวเกม โปรแกรมจะนำข้อมูลในอะเรย์ต่างๆ ใส่งไฟล์นามสกุลเอช (.h) ที่เหมาะสม

### 4.2.1 คำคงที่ ตัวแปร และชุดคำสั่งเกี่ยวกับการจัดการข้อมูลในอะเรย์

เนื่องจากการประกาศค่าคงที่ ตัวแปร และชุดคำสั่งเกี่ยวกับการจัดการข้อมูลในอะเรย์ ในไฟล์ gameCreator.h และค่าเหล่านี้สามารถถูกเรียกใช้งานได้ในทุกๆ ส่วนของโปรแกรม จึงจำเป็นต้องทำการประกาศเรียกใช้งาน โดยเขียนประกาศ #include "gameCreator.h" ไว้ในไฟล์ทุกๆ ไฟล์ที่มีการเรียกใช้งานอะเรย์ด้วย

#### 4.2.1.1 ประกาศค่าคงที่ไว้อ้างอิงอะเรย์

ประกาศค่าคงที่ไว้อ้างอิงอะเรย์ในไฟล์ gameCreator.h เพื่อความง่ายต่อการใช้งาน

```
#define OBJECT 0
#define BGSOUND 1
#define SOUND 2
#define DECLARE 3
#define CREATE 4
#define DEL 5
#define DRAW 6
#define ACTION 7
#define UPDATE 8
```

ประกาศตัวแปรอะเรย์ พอยต์เตอร์ของอะเรย์ ในไฟล์ gameCreator.cpp  
ดังนี้

```
CArray <CString, CString>    arr_object;
CArray <CString, CString>    arr_bgSound;
CArray <CString, CString>    arr_sound;
CArray <CString, CString>    arr_declare;
CArray <CString, CString>    arr_create;
CArray <CString, CString>    arr_delete;
CArray <CString, CString>    arr_draw;
CArray <CString, CString>    arr_action;
CArray <CString, CString>    arr_update;
CArray <CString, CString>    *arr_temp;
```

#### 4.2.1.2 ประกาศชุดคำสั่งสำหรับจัดการอะเรย์

ประกาศชุดคำสั่งต่างๆ ที่ใช้จัดการกับอะเรย์ในไฟล์ gameCreator.h

- ชุดคำสั่งเรียกใช้งานอะเรย์ที่ต้องการ

```
CArray<CString, CString> * GetArrPtr( int arrayRef );
```

- ชุดคำสั่งเพิ่มข้อมูลในอะเรย์

```
void AddLine( int arrayRef ,CString &txt );
```

- ชุดคำสั่งแทรกคำสั่งในอะเรย์

```
void InsertLine( int arrayRef , UINT index ,CString &txt);
```

- ชุดคำสั่งลบข้อมูลในอะเรย์

```
void DeleteLine( int arrayRef , UINT index );
```

- ชุดคำสั่งเรียกคืนข้อมูลในอะเรย์

```
CString& GetLineData( int arrayRef, UINT index );
```

- ชุดคำสั่งเรียกคืนจำนวนข้อมูล

```
INT GetNumLine( int arrayRef );
```

- ชุดคำสั่งย้ายข้อมูลจากอะเรย์ลงไฟล์

```
BOOL WriteArrayToFile( LPCTSTR filename , int arrayRef );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกาศชุดคำสั่งต่างๆ ที่ใช้จัดการกับอะเรย์ในไฟล์ gameCreator.cpp

- ชุดคำสั่งเรียกใช้งานอะเรย์ที่ต้องการ

```
CArray<CString, CString> * GetArrPtr( int arrayRef ){
    switch(arrayRef){
        case OBJECT    : return &arr_object; break;
        case BGSOUND: return &arr_bgSound; break;
        case SOUND     : return &arr_sound; break;
        case DECLARE   : return &arr_declare; break;
        case CREATE    : return &arr_create; break;
        case DEL       : return &arr_delete; break;
        case DRAW      : return &arr_draw; break;
        case ACTION    : return &arr_action; break;
        case UPDATE    : return &arr_update; break;
        return NULL;
    }
}
```

- ชุดคำสั่งเพิ่มข้อมูลในอะเรย์

```
void AddLine( int arrayRef ,CString &txt ){
    arr_temp = GetArrPtr( arrayRef );
    arr_temp->Add(txt);
}
```

- ชุดคำสั่งแทรกคำสั่งในอะเรย์

```
void InsertLine( int arrayRef , UINT index ,CString &txt){
    arr_temp = GetArrPtr( arrayRef );
    arr_temp->InsertAt( index, txt );
}
```

- ชุดคำสั่งลบข้อมูลในอะเรย์

```
void DeleteLine( int arrayRef , UINT index ){
    arr_temp =GetArrPtr( arrayRef );
    arr_temp->RemoveAt( index );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

- ชุดคำสั่งเรียกคืนข้อมูลในอะเรย์

```
CString& GetLineData( int arrayRef, UINT index ){
    arr_temp = GetArrPtr( arrayRef );
    return arr_temp->GetAt(index);
}
```

- ชุดคำสั่งเรียกคืนจำนวนข้อมูล

```
INT GetNumLine( int arrayRef ) {
    arr_temp = GetArrPtr( arrayRef );
    return (INT)arr_temp->GetSize();
}
```

- ชุดคำสั่งย้ายข้อมูลจากอะเรย์ลงไฟล์

```
BOOL WriteArrayToFile( LPCTSTR filename , int arrayRef
){
    CStdioFile file;
    - สร้างไฟล์ใหม่และเขียนคำสั่งต่อท้ายไฟล์
    if(file.Open( filename, CFile::modeCreate |
CFile::modeNoTruncate | CFile::modeWrite | CFile::typeText)) {
        arr_temp = GetArrPtr( arrayRef );
        CString str;
        for( INT n=0; n<GetNumLine( arrayRef );
n++) {
            file.WriteString( GetLineData(
arrayRef ,(UINT)n );
            if( n<GetNumLine( arrayRef )-1)
                file.WriteString( _T("\n"));
        }
        file.Close();
        return TRUE;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

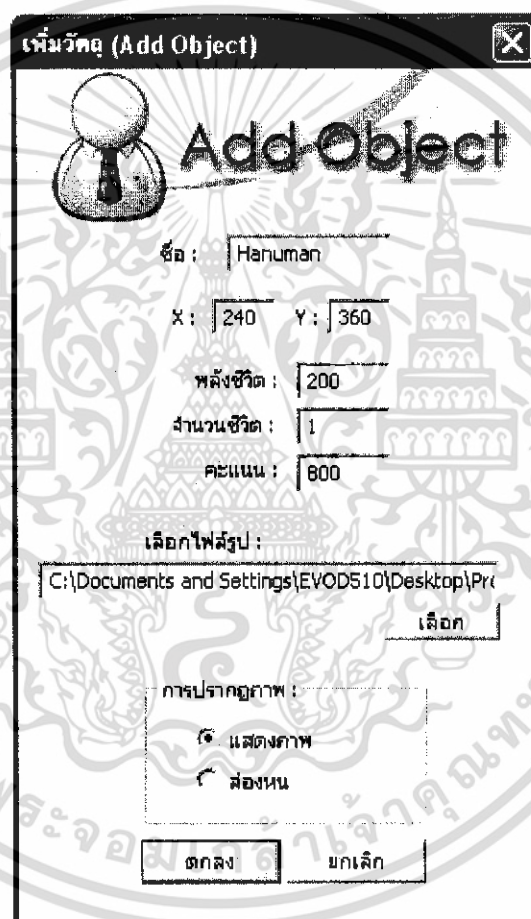
}

#### 4.2.1.3 ตัวอย่างการเรียกใช้ชุดคำสั่งจัดการข้อมูลในอะเรย์

ในที่นี้ จะแสดงเพียงการนำคำสั่งที่เกิดจากการรับค่าจากผู้ใช้งานมาสร้างเป็นคำสั่งของเกมเอนจิน

##### 4.2.1.3.1 ตัวอย่างการใส่ข้อมูลลงในอะเรย์

กำหนดให้ผู้ใช้เลือกหน้าต่างสร้างตัวละครและใส่ค่าต่างๆ ลงไป  
ดังรูป



รูปที่ 4-1 แสดงการรับข้อมูลการสร้างตัวละครตัวใหม่

เมื่อผู้ใช้คลิกปุ่มตกลง จะมีการเขียนคำสั่งลงในอะเรย์ต่างๆ ดังนี้

- เก็บตัวชื่อตัวละครที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร name

CString name;

GetDlgItemText(IDC\_ADD\_OBJ\_NAME,name);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เก็บพิกัด x ที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร x

```
CString x;
```

```
GetDlgItemText(IDC_ADD_OBJ_X,x);
```

- เก็บพิกัด y ที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร y

```
CString y;
```

```
GetDlgItemText(IDC_ADD_OBJ_Y,y);
```

- เก็บพลังชีวิตเริ่มต้นที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร health

```
CString health;
```

```
GetDlgItemText(IDC_ADD_OBJ_HEALTH,health);
```

- เก็บจำนวนชีวิตเริ่มต้นที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร life

```
CString life;
```

```
GetDlgItemText(IDC_ADD_OBJ_LIFE,life);
```

- เก็บคะแนนของตัวละครที่ผู้ใช้ป้อนเข้ามาไว้ที่ตัวแปร score

```
CString score;
```

```
GetDlgItemText(IDC_ADD_OBJ_SCORE,score);
```

- สร้างตัวแปรกำหนดค่าให้กับตัวเลือกของการปรากฏ

```
CButton *visibleButton,*invisibleButton;
```

```
visibleButton=(CButton*)GetDlgItem(IDC_ADD_OBJ_VISIBLE);
```

```
invisibleButton=(CButton*)GetDlgItem(IDC_ADD_OBJ_INVISIBLE);
```

- ตรวจสอบว่าผู้ใช้เลือกการปรากฏแบบใด และใส่ค่านั่นลงตัวแปร

```
CString visibility;
```

```
if ((visibleButton)->GetCheck()==1) {
```

```
    visibility = "true";
```

```
}
```

```
else if ((invisibleButton)->GetCheck()==1) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        visibility = "false";
    }
    else {
        visibility = "true";
    }
    CString str;
    str = "";
    str.Append("Object* "+name+" = NULL;");

```

- เพิ่มคำสั่งลงในอะเรย์ arr\_declare;

```
AddLine( DECLARE , str );
```

```

str = "";
str.Append(name+" = new Object(Screen,"+x+", "+y+
    "+health+", "+life+", "Resource/Sprite/"+GetFileName()+
    "\", "+score+", "+visibility+");");

```

- เพิ่มคำสั่งลงในอะเรย์ arr\_create;

```
AddLine( CREATE , str );
```

```

str = "";
str.Append("delete "+name+");");

```

- เพิ่มคำสั่งลงในอะเรย์ arr\_delete;

```
AddLine( DEL , str );
```

```

str = "";
str.Append(name+"->Draw(Screen);");

```

- เพิ่มคำสั่งลงในอะเรย์ arr\_draw;

```
AddLine( DRAW , str );
```

- เพิ่มคำสั่งลงในอะเรย์ arr\_object;

```
AddLine( OBJECT , name );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.1.3.2 ตัวอย่างการนำข้อมูลในอะเรย์บันทึกลงไฟล์

หลังจากที่โปรแกรมสร้างคำสั่งต่างๆ เรียบร้อยแล้ว ผู้ใช้ต้องกดปุ่มสร้างคำสั่ง โปรแกรมจะนำคำสั่งที่โปรแกรมสร้างขึ้นทั้งหมดบันทึกลงในไฟล์ในส่วนของเกมเอ็นจิน ดังนี้

```
WriteArrayToFile("Declare.h" , DECLARE );
WriteArrayToFile("Create.h" , CREATE );
WriteArrayToFile("Delete.h" , DEL );
WriteArrayToFile("Draw.h" , DRAW );
WriteArrayToFile("GameCondition.h" , ACTION );
```

#### 4.3 การพัฒนาคำสั่งด้านฟิสิกส์

ฟังก์ชันการคำนวณการเคลื่อนที่นี้ เขียนไว้ในฟังก์ชัน

```
void Object::UpdatePosition();
```

เนื่องจากตำแหน่งของ x,y จะต้องทำการคำนวณตลอดเวลา ดังนั้น จึงจำเป็นต้องใส่คำสั่ง UpdatePosition(); ของตัวละครทุกตัวไว้ใน Game Loop ซึ่งในที่นี้คือให้เพิ่มคำสั่งใส่ลงอะเรย์ arr\_update เพื่อเขียนลงไฟล์ "GameCondition.h" ในรูปแบบดังนี้

```
ชื่อตัวละคร->UpdatePosition();
```

ส่วนฟังก์ชันเปลี่ยนค่าความเร็วความเร่งที่ใช้คำนวณการเคลื่อนที่แบบฟิสิกส์ มีดังนี้

กระโดด โดยการใส่ความเร็วความเร่งในแกน x,y เกิดเป็นโปรเจกไทล์

```
void Object::Jump( float vx ,float vy ,float ax ,float ay );
```

หยุดกระโดด เมื่อตัวละครชนกับพื้น

```
void Object::StopJump();
```

ตกลงมาจากการกระแทกเพดาน เมื่อตัวละครกระโดดไปชนเพดาน ตัวละครจะตกลงมา

```
void Object::FallDown();
```

ของหล่นลงมาด้วยความหน่วง ay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void Object::Drop( float ay );

#### 4.4 การทดสอบโปรแกรม

การทดสอบที่กล่าวถึงในบทนี้ จะเป็นขั้นตอนการทดสอบและแสดงผลการทดสอบในแต่ละขั้น โดยผลการทดสอบนี้จะถูกนำไปวิเคราะห์ถึงสาเหตุของการเกิดปัญหาและแนวทางในการพัฒนาต่อไปในอนาคต ซึ่งจะกล่าวถึงในบทที่ 5

คุณสมบัติของระบบที่นำมาทดสอบ

- ระบบปฏิบัติการ ไมโครซอฟท์ วินโดวส์ เอ็กซ์พี (Microsoft Windows XP)
- ติดตั้งชุดคำสั่ง ไมโครซอฟท์ ไคเร็กเอ็กซ์ เอส ดี เค 8.0 (Microsoft DirectX SDK 8.0)
- เครื่องคอมพิวเตอร์ที่มีซีพียูความเร็ว 1.8 กิกะเฮิร์ต
- หน่วยความจำหลักขนาด 256 เมกะไบต์

ขั้นตอนการดำเนินการทดสอบ

- ติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็นต้องการประมวลผล
- รันและประมวลผลภายใต้ระบบที่กำหนด
- ตรวจสอบการใช้คำสั่งของต่างๆของโปรแกรม
- ตรวจสอบการหาความผิดพลาดของโปรแกรม

การประเมินผล

- หลังจากติดตั้งโปรแกรมแล้ว ผู้ใช้สามารถใช้โปรแกรมได้ทันที
- การประมวลผลของโปรแกรมจะต้องทำภายใต้ความเร็วในระดับที่ยอมรับได้
- การใช้คำสั่งของโปรแกรมเป็นไปตามรูปแบบที่นักเขียนโปรแกรมกำหนดไว้
- จำนวนความผิดพลาดของเกมที่เกิดขึ้นต้อง ไม่มีเลย หรือมีน้อยที่สุด

##### 4.4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและซอฟต์แวร์ที่จำเป็น

1. ติดตั้งโปรแกรมวิซวลซีพลัสพลัส 6.0 (Visual C++ 6.0)
2. ติดตั้งไคเร็กเอ็กซ์ เอสดีเค 8(Direct X SDK8)
3. ติดตั้งซีดีเอ็กซ์ เอนจิน3.1(CDX Engine3.1)
4. ปรับแต่งวิซวลซีพลัสพลัส ให้สามารถใช้ ไคเร็กเอ็กซ์ และ ซีดีเอ็กซ์ ได้
5. คัดลอกโฟลเดอร์ ResourceGame ไว้ที่ไดรฟ์ ซี (C:/)
6. ใช้โปรแกรมช่วยสร้างเกม 2 มิติ (Game Creator 2D)
7. เปิดโปรเจกต์ที่โปรแกรมช่วยสร้างเกม 2 มิติ สร้างขึ้นและประมวลผล

ทุกขั้นตอนที่กล่าวมาแล้วนั้นสามารถดูรายละเอียดต่างๆได้ในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.2 ผลการทดลอง

การทดสอบโปรแกรมช่วยสร้างเกม 2 มิติ (2D Game Creator ) พบว่า การใช้งานโปรแกรมมีความยืดหยุ่นพอสมควร สามารถสร้างเกมง่ายๆ ที่มีชุดคำสั่ง ( Engine) ไม่ซับซ้อนได้อย่างรวดเร็ว โดยที่ไม่ต้องเขียนโปรแกรม เพียงคลิกเมาส์และป้อนข้อมูลลงในช่องรับข้อมูลในหน้าค่างโปรแกรม และสามารถลบชุดคำสั่งที่ไม่ต้องการได้

ทั้งนี้ โปรแกรมสามารถสร้างเกมได้จริงแต่ยังไม่สามารถรันเกมได้โดยอัตโนมัติ ยังคงต้องใช้โปรแกรมวิซวล ซี พลัส พลัส ช่วยในการคอมไพล์เกม จึงจะสามารถประมวลผลออกมาเป็นเกมได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินงาน

1. ผู้ใช้สามารถสร้างเกมสองมิติอย่างง่ายได้จากโปรแกรมช่วยสร้างเกมสองมิติ
2. ผู้ใช้ไม่จำเป็นต้องทราบถึงกระบวนการของไดเรกเอ็ทซ์ในการสร้างเกมมาก่อน เพราะระบบได้สร้างชุดคำสั่งพื้นฐานที่จำเป็นไว้ให้เรียบร้อยแล้ว
3. ผู้ใช้สามารถตรวจสอบชุดคำสั่งที่ระบบทำการสร้างและเก็บลงไฟล์ได้จากหน้าต่างโปรแกรมและจากไฟล์ .h ในเพิ่มข้อมูลที่ผู้ใช้กำหนด
4. เหตุการณ์ต่าง ๆ (Event) ที่มีในโปรแกรมครอบคลุมการสร้างเกมสองมิติได้โดยที่ผู้ใช้สามารถเรียนรู้และฝึกหัดได้อย่างไม่ยากเย็น

#### 5.2 ข้อจำกัดของโปรแกรม

1. สามารถสร้างเกมสองมิติได้เท่านั้น ยังไม่สามารถสร้างเกมสามมิติได้ แต่สามารถทดแทนโดยการใช้รูปที่สร้างจากโปรแกรมสร้างภาพสามมิติ เพื่อทำเป็นเกมเสมือนสามมิติได้
2. รูปตัวละครไม่สามารถสร้างเป็นภาพเคลื่อนไหวได้ แต่สามารถใช้การกำหนดเหตุการณ์ (Event) เพื่อเปลี่ยนรูปแทนได้
3. ไม่สามารถทำการบันทึกการกระทำและเหตุการณ์ต่าง ๆ ได้ แต่เพิ่มข้อมูลที่ผู้ใช้สร้างนั้นได้ถูกบันทึกไว้อย่างเรียบร้อยแล้ว ผู้ใช้สามารถแก้ไขและประมวลผลโปรแกรมโดยใช้โปรแกรมวิซวลซีพลัสพลัส 6.0 ได้
4. ไม่สามารถสร้างและเรียกใช้ตัวแปรประเภทตรรกะ (Boolean) ได้

#### 5.3 ปัญหาที่เกิดขึ้น

การทำงานโครงการนี้ได้มีการวางแผนทำงานในทุก ๆ ขั้นตอนตามแบบแผนไว้เป็นอย่างดี เริ่มตั้งแต่ การตั้งปัญหา ขอบเขตของปัญหา วิเคราะห์ปัญหา การออกแบบ การโปรแกรม การตรวจสอบโปรแกรม ทดสอบ และบำรุงรักษา แต่ถึงอย่างไรก็ยังคงมีปัญหาเกิดขึ้นกับขั้นตอนการทำงานบางขั้นตอน อาทิเช่น ปัญหาที่เกิดขึ้นจากความล่าช้าในการคิดระเบียบวิธีการโปรแกรมที่ยากและซับซ้อน ซึ่งจำเป็นต้องศึกษาจากตำราต่างประเทศ เนื่องจากยังไม่มีตำราภาษาไทยในเรื่องนั้น ๆ อย่างจำเพาะเจาะจง บ่อยครั้งที่ต้องค้นคว้าจากแหล่งต่าง ๆ ที่นอกเหนือจากในตำรา เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ต จากเว็บไซต์ที่เกี่ยวข้องกับการ โปรแกรมด้วยภาษาซีพลัสพลัสและไคเร็กเอ็กซ์ และการปรึกษาผู้มีประสบการณ์การเขียนเกม โดยสามารถแบ่งออกเป็นขั้นตอนต่างๆ ได้ 3 ขั้นตอน ได้แก่

### 5.3.1 ขั้นตอนการออกแบบโครงสร้างของโปรแกรม

ขั้นตอนการออกแบบเป็นขั้นตอนที่ต้องใช้ความคิดของทั้งกลุ่มเพื่อหาแนวทางที่ดีที่สุดในการพัฒนาโปรแกรม เป็นขั้นที่ใช้เวลารวบรวมความคิด แนวคิด จนได้มาเป็นโครงสร้างที่เหมาะสมบนแผ่นกระดาษ ก่อนนำมาสร้างจริงบนเครื่องคอมพิวเตอร์ แต่ถึงอย่างไรก็ตาม การออกแบบก็ยังคงต้องมีการเปลี่ยนแปลงในระหว่างปฏิบัติอยู่เสมอเพื่อให้ได้โครงสร้างที่ดีที่สุดในการปฏิบัติ

อีกปัญหาหนึ่ง คือ การออกแบบที่ไม่ชัดเจนตั้งแต่เริ่มต้น การทำงานจึงเกิดความสับสนคลุมเครือ ไม่เป็นระเบียบและเกิดข้อผิดพลาดในการเขียนโปรแกรมตามมาในที่สุด

### 5.3.2 ขั้นตอนการเขียนโปรแกรม

ขั้นตอนนี้สามารถทำควบคู่ไปได้กับขั้นตอนการออกแบบหน้าจอและการตกแต่งหน้าจอ ขั้นตอนนี้เป็นขั้นตอนที่ใช้เวลานานที่สุด เนื่องจากผู้วิจัยต้องทำการศึกษารายละเอียดของโปรแกรมภาษาซีพลัสพลัสที่ต้องติดต่อกับ ไคเร็กเอ็กซ์ และ ซีดีเอ็กซ์ เอ็นจิน ซึ่งเป็นเรื่องใหม่สำหรับผู้วิจัย เมื่อศึกษาเรียบร้อยแล้วจึงลงมือเขียนโปรแกรมโดยการทำงานในส่วนนี้มีซอฟต์แวร์ที่เกี่ยวข้องที่ใช้ในการพัฒนา คือ ไมโครซอฟท์วิซวลซีพลัสพลัสดอทเน็ต (Microsoft Visual C++.NET)

ภาษาซีพลัสพลัส มีรูปแบบการเขียนโค้ดหลากหลายรูปแบบ จึงทำให้เกิดความสับสนในการพัฒนาโปรแกรม ผู้พัฒนาโปรแกรมต้องเรียนรู้และจดจำไวยากรณ์และรูปแบบในการเขียนโค้ดอย่างเหมาะสมกับ โปรแกรมที่ใช้พัฒนา โดยศึกษาจากแหล่งข้อมูลต่างๆ

ซีดีเอ็กซ์ เอ็นจิน เป็นเกมเอ็นจินที่หยุดการพัฒนาแล้ว หาดาวนโหลดได้ยาก ฟังก์ชันบางฟังก์ชันไม่สามารถใช้ได้จริง และไม่สนับสนุนการดีบั๊กแบบ ดีบั๊กโหมด จึงสร้างความสับสนในการใช้งาน มีความซับซ้อนในการเรียนรู้และทำความเข้าใจ ทางคณะผู้จัดทำขอแนะนำว่าหากมีโอกาสศึกษาเอ็นจินตัวอื่น น่าจะเป็นอีกทางเลือกหนึ่งในการพัฒนาเกมโปรแกรมนิ่ง

เนื่องจากคณะผู้จัดทำไม่คุ้นเคยกับการเขียน โปรแกรมภาษาซีพลัสพลัส จึงออกแบบโปรแกรมในขั้นต้นได้ไม่ดึ้นัก ทำให้พบปัญหาในการพัฒนาโปรแกรมจนต้องมีการเปลี่ยนโครงสร้างบางส่วนจากการจัดการข้อมูลด้วยไฟล์เป็นการจัดการข้อมูลด้วยอะเรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.3 ขั้นตอนการทดสอบโปรแกรม

โปรแกรมวิชวล ซี พลัส พลัส คอท เน็ต (Visual C++.NET) ที่ใช้งาน มีปัญหาการใช้งานภาษาไทย ทำให้การแก้ไขโปรแกรมให้เป็นภาษานั้นทำได้ลำบาก บางครั้งเมื่อแก้ไขให้เป็นภาษาไทยแล้วโปรแกรม วิชวล ซี พลัส พลัส คอท เน็ต เกิดข้อผิดพลาด ทำให้โปรแกรมที่แก้ภาษา ถูกแปลงให้กลายเป็นภาษาอื่นที่ไม่สามารถสื่อสารได้ ต้องทำการเปลี่ยนภาษาใหม่หลายรอบ

### 5.4 ผลการเปรียบเทียบโปรแกรมช่วยสร้างเกม 2 มิติ กับ โปรแกรมเกมเมกเกอร์

ข้อเปรียบเทียบ	โปรแกรมช่วยสร้างเกม 2 มิติ	โปรแกรมเกมเมกเกอร์
1. สามารถสร้างเกมง่ายๆได้	สามารถทำได้	สามารถทำได้
2. โอเพนซอร์ส	ใช่	ไม่ใช่
3. ลักษณะการใช้งาน	ไม่ซับซ้อน ใช้งานง่าย	ค่อนข้างซับซ้อนแต่มีฟังก์ชันให้เรียกใช้มากกว่า
4. ลิขสิทธิ์	โดยคนไทย	โดยชาวเนเธอร์แลนด์

ตารางที่ 5-1 ตารางแสดงการเปรียบเทียบระหว่างโปรแกรมช่วยสร้างเกม 2 มิติ กับ โปรแกรมเกมเมกเกอร์

### 5.5 ข้อเสนอแนะเพื่อเป็นแนวทางสำหรับผู้ที่ต้องการพัฒนาเกมนี้ต่อไป

1. พัฒนาอัลกอริทึมในคำนวณเรื่องพื้นฐานของเกม เช่น การเดิน การกระโดด การชน เพื่อให้ตัวโปรแกรมมีความฉลาดมากขึ้นและคำนวณทุกกรณีที่เป็นไปได้
2. พัฒนาให้สามารถสร้างเกมสามมิติได้ และแบบเล่นหลายคนได้
3. รวมคอมพิวเตอร์เข้ากับตัวโปรแกรมเพื่อให้สามารถเล่นเกมได้ทันที
4. พัฒนารูปแบบหน้าต่างโปรแกรมให้มีความสวยงามน่าใช้มากขึ้น
5. พัฒนาให้สามารถเล่นได้บนเครื่องคอมพิวเตอร์ได้ทุกเครื่องโดยที่ไม่ขึ้นกับความละเอียดของหน้าจอ
6. พัฒนาระบบการสร้างตัวละครและจัดฉาก ให้สามารถใช้เมาส์คลิกแล้วลากตัวละครมายังตำแหน่งบนฉากได้
7. เพิ่มระบบปัญญาประดิษฐ์ ทำให้ศัตรูมีความฉลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. พัฒนาให้โปรแกรมสามารถเพิ่มตัวแปรประเภทตรรกะ (boolean) เพื่อสร้างเหตุการณ์ เกม และการตรวจสอบตำแหน่งพิกัดของตัวละคร ซึ่งเป็นจะผลทำให้ผู้ใช้งาน โปรแกรมสามารถ สร้างเกมได้หลากหลายรูปแบบมากขึ้น

ในการทำงานวิจัยเพื่อการสร้าง โปรแกรมสร้างเกมบนคอมพิวเตอร์แบบ 2 มิติ การทำงาน ต่างๆ ตลอดจนปัญหาที่เกิดขึ้น ทำให้ผู้วิจัยสามารถเข้าใจถึงการเขียน โปรแกรมเกมโดยใช้ภาษา ซีพลัสพลัส และ ทราบถึงแนวทางในการพัฒนาเกม 2 มิติ มากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

นิรุช อำนวยศิลป์ . 2542 . คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0 ฉบับใช้งานจริง .

กรุงเทพฯ : ชักเชส มิเดีย.

นิรุช อำนวยศิลป์ . 2545 . สร้างเกมด้วย Visual C++ และ DirectX ฉบับสมบูรณ์ . นนทบุรี : อินโฟเพรส.

นิรุช อำนวยศิลป์ . 2546 . สร้างเกมด้วยเกมเอ็นจิน CDX และ DirectX ฉบับมืออาชีพ .

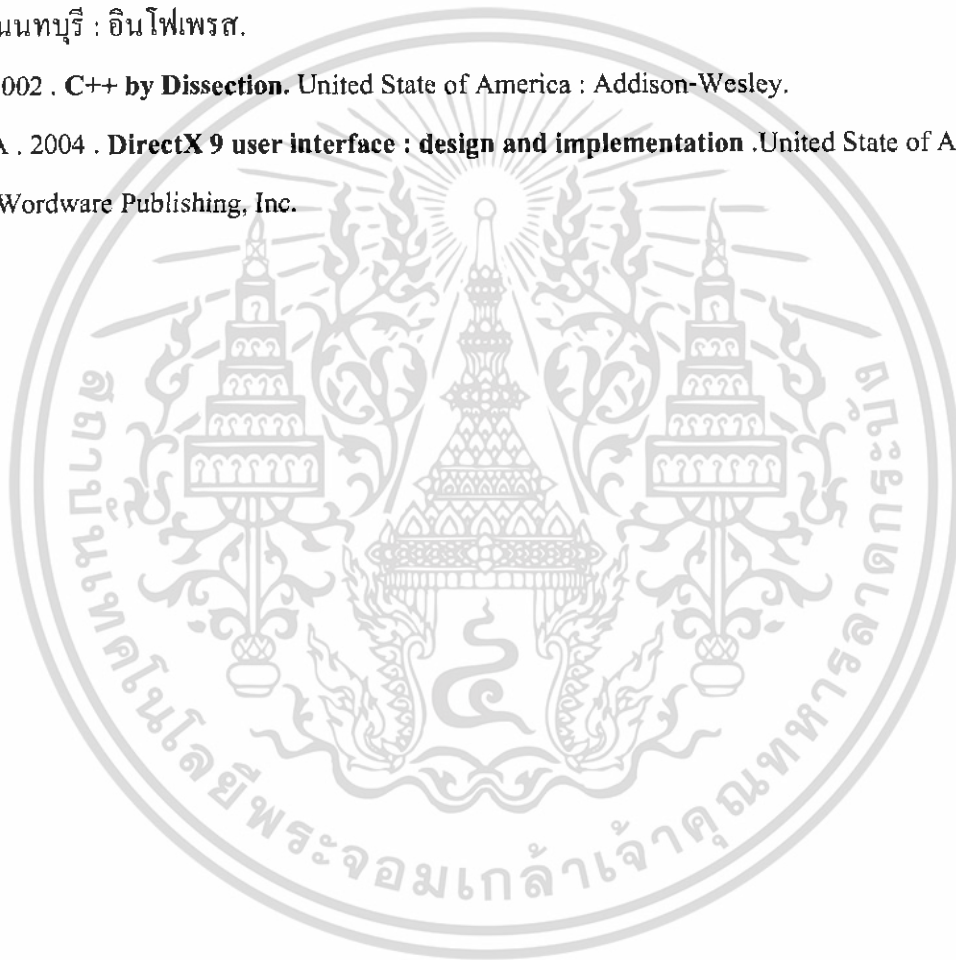
นนทบุรี : อินโฟเพรส.

ยุทธนา ลีลาควัฒนกุล . 2546 . คู่มือการเขียนโปรแกรมและใช้งาน Visual C++.NET ฉบับสมบูรณ์ .

นนทบุรี : อินโฟเพรส.

Ira , P . 2002 . C++ by Dissection. United State of America : Addison-Wesley.

Thorn , A . 2004 . DirectX 9 user interface : design and implementation .United State of America :  
Wordware Publishing, Inc.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

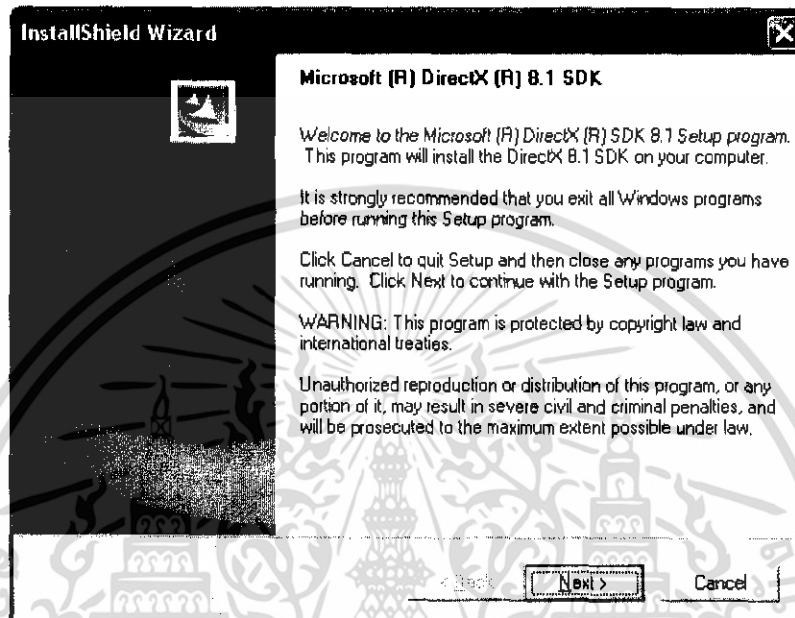


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การลงโปรแกรมไคเร็กเอ็กซ์ เอส ดี เค (Direct X SDK)

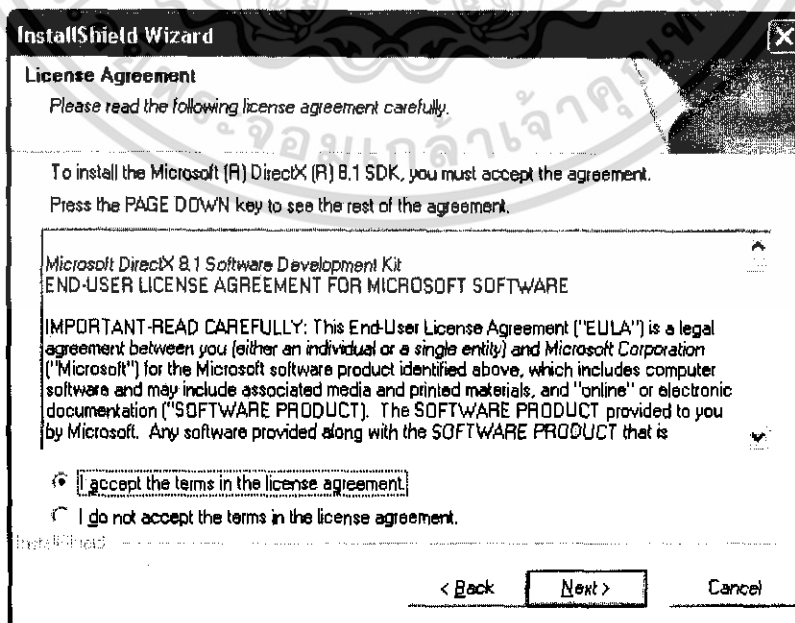
1. ดับเบิลคลิกที่ไอคอนชื่อ SETUP เพื่อเปิดหน้าต่างการลงไคเร็กเอ็กซ์เอสดีเคและคลิก

ปุ่ม Next



รูปที่ ก-1 แสดงขั้นตอนการติดตั้งไคเร็กเอ็กซ์ เอส ดี เค ขั้นตอนที่ 1

2. เลือกยอมรับเงื่อนไขของผู้ผลิตไคเร็กเอ็กซ์ และคลิกปุ่ม Next



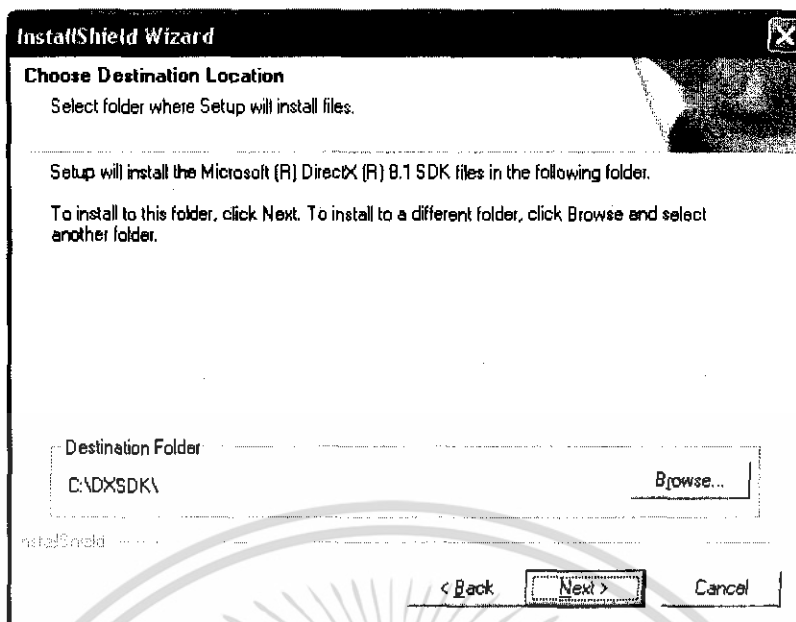
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปที่ ก-2 แสดงขั้นตอนการติดตั้งไคเร็กซ์ เอส ดี เค ขั้นตอนที่ 2

3. ใส่ User Name และ Company Name ของผู้ใช้ และกดปุ่ม Next

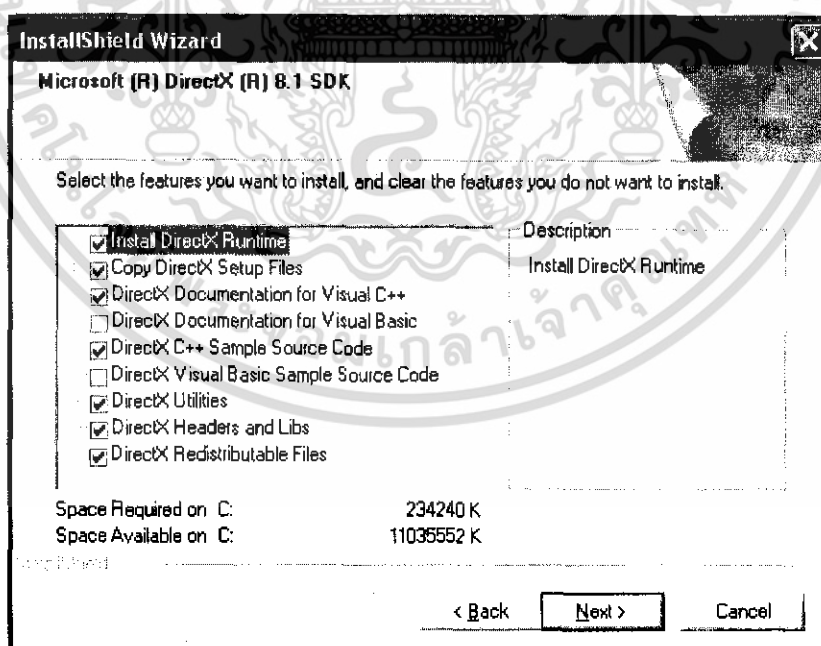
### รูปที่ ก-3 แสดงขั้นตอนการติดตั้งไคเร็กซ์ เอส ดี เค ขั้นตอนที่ 3

4. เลือกตำแหน่งที่จะลงไคเร็กซ์เอสดีเค โดยทั่วไปโปรแกรมจะกำหนดให้ลงที่ C:\DXSDK\ ผู้ใช้ไม่จำเป็นต้องเปลี่ยนแปลงตำแหน่งที่จะลงก็ได้ เพื่อป้องกันความไม่สับสนในการพัฒนาโปรแกรมในขั้นต่อไป



รูปที่ ก-4 แสดงขั้นตอนการติดตั้งไดเรกเอ็กซ์ เอส ดี เค ขั้นตอนที่ 4

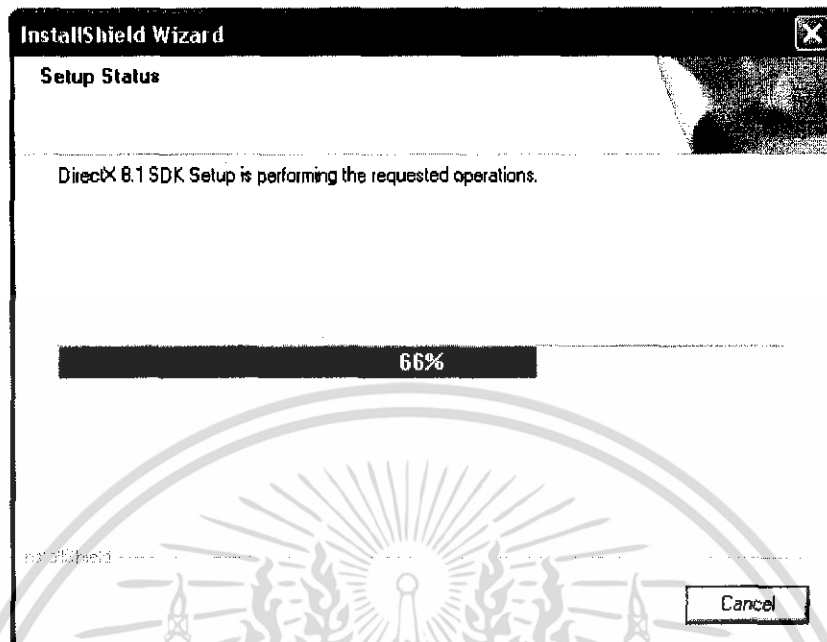
5. เลือกส่วนของโปรแกรมที่ต้องการ เราใช้ไดเรกเอ็กซ์กับโปรแกรมวิซวล ซี พลัส พลัส แต่ไม่ได้ใช้กับวิซวล เบสิก ดังนั้น ไม่จำเป็นต้องเลือกลงไดเรกเอ็กซ์สำหรับวิซวล เบสิก คลิกเลือกเอาเครื่องหมายถูกหน้าหัวข้อที่เกี่ยวกับวิซวล เบสิก ออกได้



รูปที่ ก-5 แสดงขั้นตอนการติดตั้งไดเรกเอ็กซ์ เอส ดี เค ขั้นตอนที่ 5

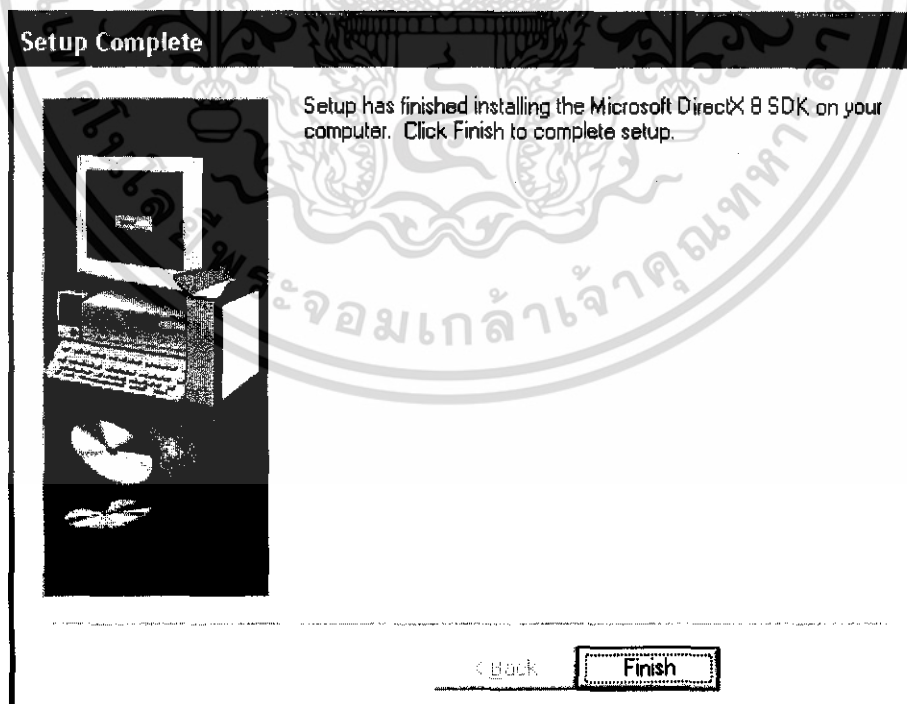
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. รอให้ลงโปรแกรมเสร็จสิ้น



รูปที่ ก-6 แสดงขั้นตอนการติดตั้งไดเรกเอ็กซ์ เอส ดี เค ขั้นตอนที่ 6

## 7. เมื่อโปรแกรมสมบูรณ์ คลิกปุ่ม Finish



รูปที่ ก-7 แสดงขั้นตอนการติดตั้งไดเรกเอ็กซ์ เอส ดี เค ขั้นตอนที่ 7

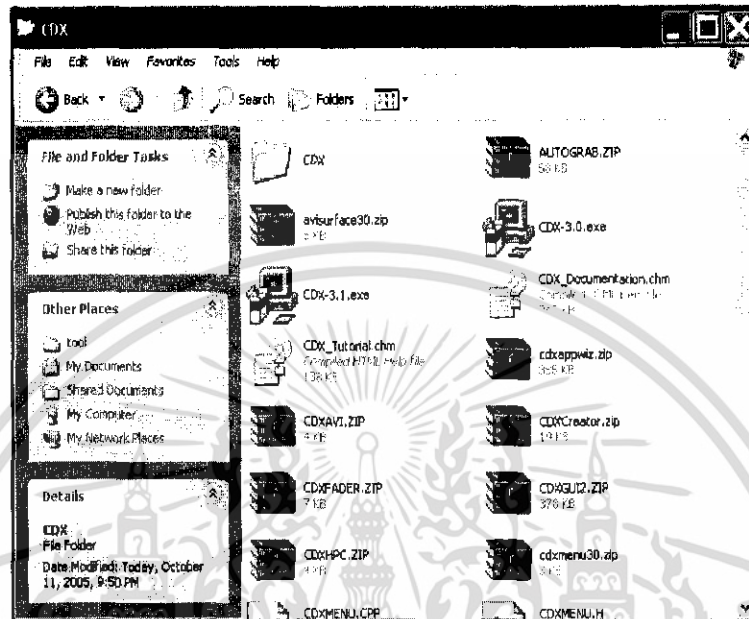
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

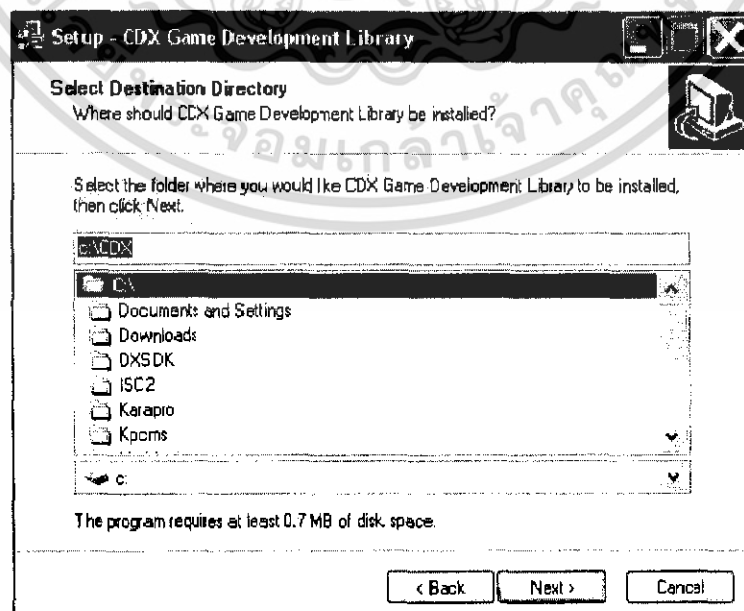
## การติดตั้งโปรแกรมซี ดี เอ็กซ์ (CDX)

### 1. Double Click ไอคอน CDX-3.1.exe



### รูปที่ ข-1 แสดงขั้นตอนการติดตั้งซี ดี เอ็กซ์ ขั้นตอนที่ 1

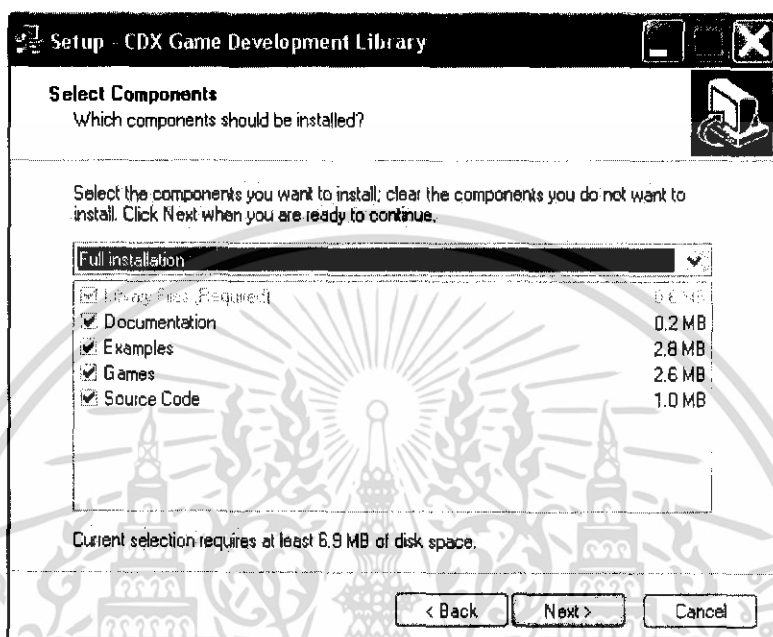
2. เลือกปลายทางที่จะติดตั้ง
3. คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปที่ ข-2 แสดงขั้นตอนการติดตั้งซี ดี เอกซ์ ขั้นตอนที่ 2

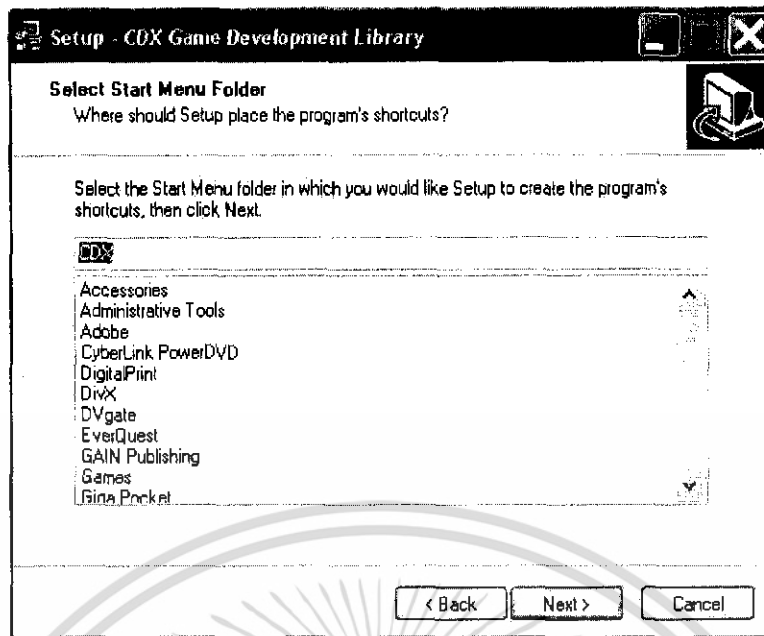
4. เลือกองค์ประกอบต่างๆจากรายการตัวเลือก ในที่นี้ให้เลือกทั้งหมด
5. คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป



### รูปที่ ข-3 แสดงขั้นตอนการติดตั้งซี ดี เอกซ์ ขั้นตอนที่ 3

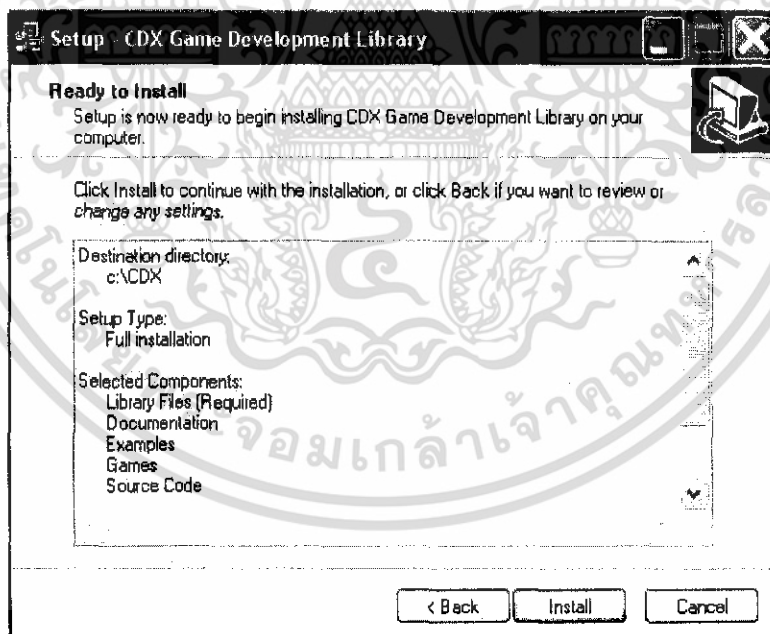
6. เลือกว่าจะเพิ่ม ซี ดี เอกซ์ ไว้ใน Start Menu ของวินโดวส์กลุ่มโปรแกรมอะไร
7. คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-4 แสดงขั้นตอนการติดตั้งซี ดี เอกซ์ ขั้นตอนที่ 4

8. คลิก Install เพื่อทำการติดตั้ง



รูปที่ ข-5 แสดงขั้นตอนการติดตั้งซี ดี เอกซ์ ขั้นตอนที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



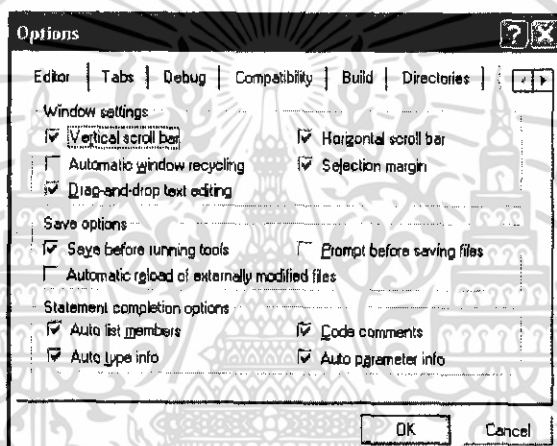
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การปรับแต่งให้โปรแกรม วิซวล ซี พลัส พลัส(Visual C++) สามารถใช้งานไคเร็กเอ็กซ์ (Direct X)

โดยปกติแล้ว โปรแกรมวิซวลซี พลัส จะสามารถใช้งานไคเร็กเอ็กซ์ เวอร์ชัน 5 ได้ แต่ในปัจจุบันนี้ ไคเร็กเอ็กซ์ได้มีการเปลี่ยนเวอร์ชันมาจนถึงเวอร์ชัน 9 แล้ว ดังนั้น ผู้ที่ต้องการเขียนเกมโดยใช้ไคเร็กเอ็กซ์ ร่วมกับโปรแกรม วิซวลซี พลัส พลัส จึงต้องทำการปรับตั้งค่าเพื่อให้โปรแกรมสามารถใช้งานไคเร็กเอ็กซ์ เวอร์ชันที่ผู้เขียนโปรแกรมใช้อยู่

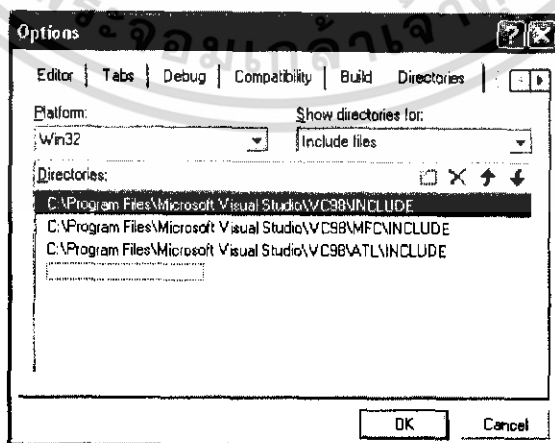
ขั้นตอนการปรับตั้งค่าให้โปรแกรม วิซวล ซี พลัส พลัส สามารถใช้งานไคเร็กเอ็กซ์ เวอร์ชันที่ผู้เขียนโปรแกรมใช้นั้น ทำได้ตามขั้นตอนต่อไปนี้

1. เปิดโปรแกรมวิซวล ซี พลัส พลัส เลือกเมนู Tools >> Option...



รูปที่ ค-1 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 1

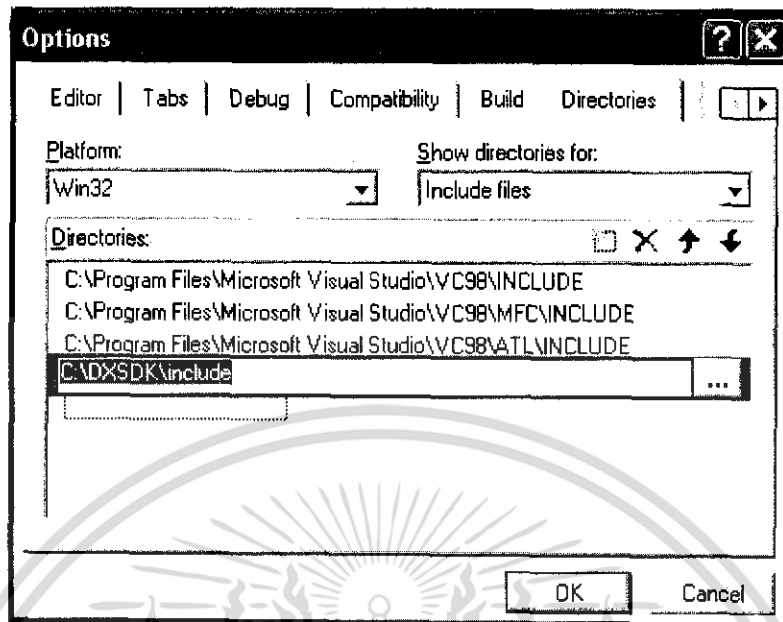
2. เลือกที่แท็บ Directories กำหนดที่ช่อง Show directories for : เป็น Include files



รูปที่ ค-2 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 2

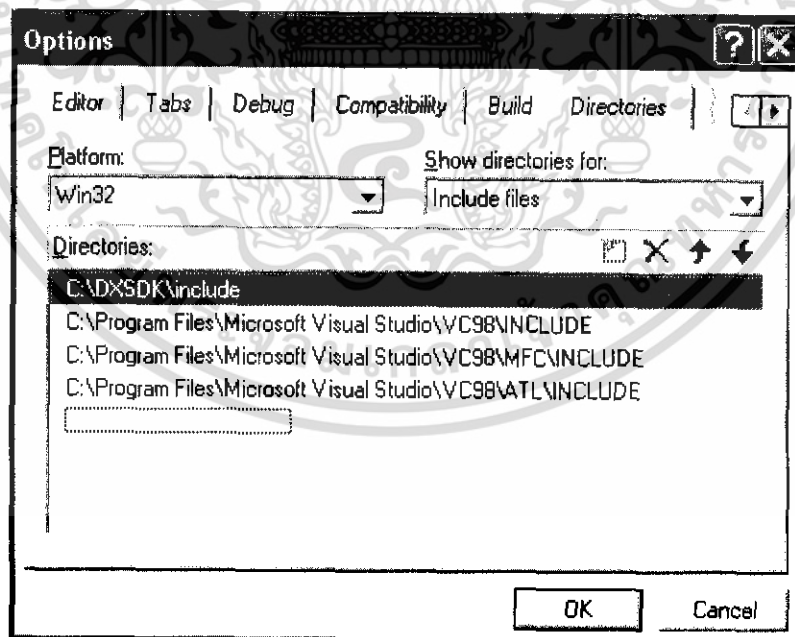
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ดับเบิลคลิกที่กรอบสี่เหลี่ยมจุดไข่ปลาแล้วเลือกพาธ include ของไคเร็กเอ็กซ์ที่เราทำการติดตั้งไว้



รูปที่ ๓-3 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 3

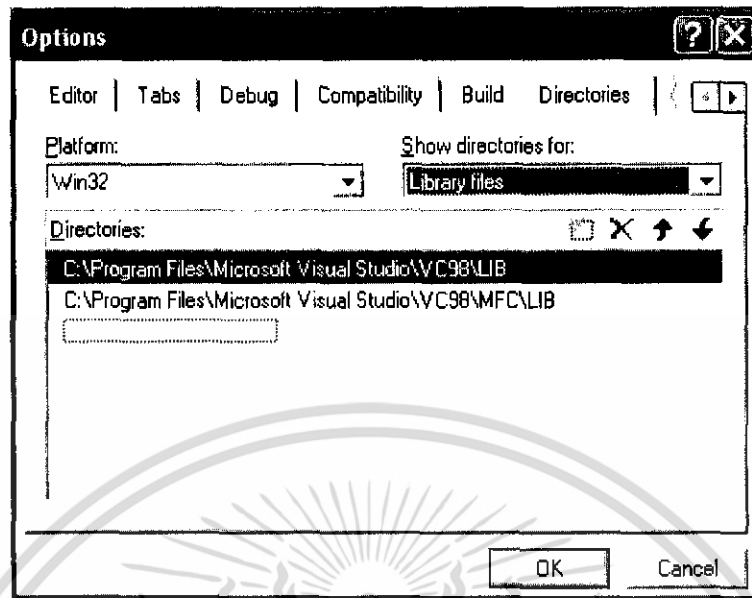
4. เลื่อนให้พาธ นี้อยู่บนสุด เพื่อให้วิซวล ซี พลัส พลัส กำหนดให้ใช้ไคเร็กเอ็กซ์ ก่อน โปรแกรมส่วนอื่น โดยคลิกที่ลูกศรชี้ขึ้นทางด้านล่างของช่อง Show directories for : จนชื่อพาธขึ้น ไปอยู่บนสุด



รูปที่ ๓-4 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 4

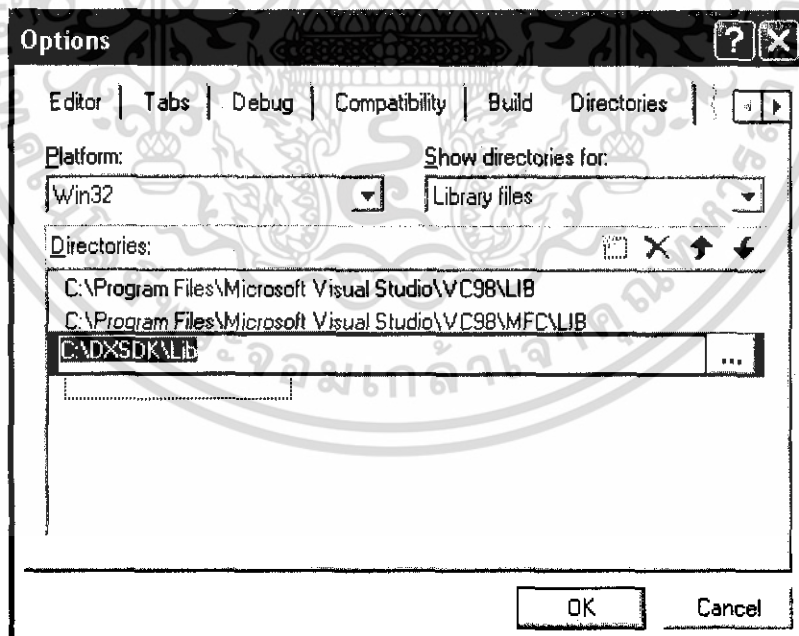
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เปลี่ยนค่าในช่อง Show directories for : ให้เป็น Library Files



รูปที่ ก-5 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 5

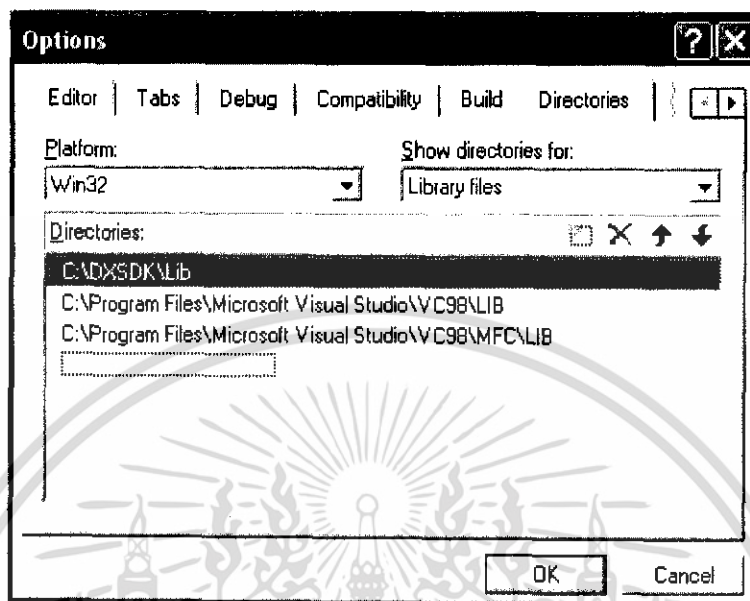
6. ดับเบิลคลิกที่กรอบสี่เหลี่ยมจุดไข่ปลาแล้วเลือกพาท (path) Library ของไคเร็กเอ็กซ์ ที่เราทำการติดตั้งไว้ ดังรูป



รูปที่ ก-6 แสดงการปรับตั้งค่าการใช้งานไคเร็กเอ็กซ์ ขั้นตอนที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลื่อนให้พารามิเตอร์อยู่บนสุด เพื่อให้วิซวก ซี พลัส พลัส กำหนดให้ใช้ไดเรกทอรีก่อนโปรแกรมส่วนอื่น โดยคลิกที่ลูกศรชี้ขึ้นทางด้านล่างของช่อง Show directories for : จนชื่อพารามิเตอร์ขึ้นไปอยู่บนสุด จากนั้นคลิกปุ่ม OK เป็นการเสร็จสิ้นการปรับแต่ง



รูปที่ ค-7 แสดงการปรับตั้งค่าการใช้งานไดเรกทอรี ขั้นตอนที่ 7

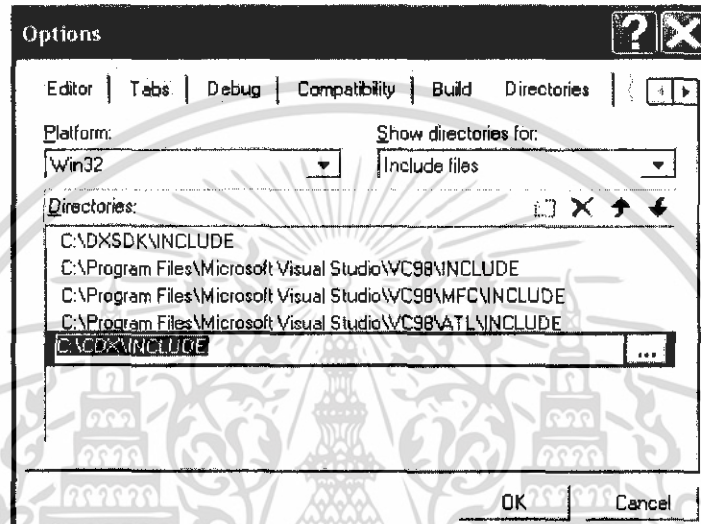
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

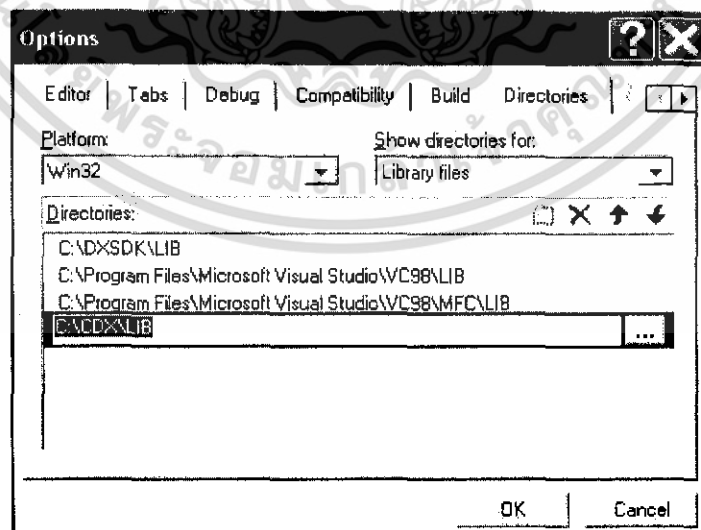
## การทำให้โปรแกรมวิชวล ซี พลัส พลัส(Visual C++) สามารถใช้งาน ซี ดี เอกซ์ (CDX)

- 1 เปิด วิชวลซีพลัสพลัส เลือกเมนู Tools > Options
- 2 คลิกที่แท็บ Directories
- 3 ใน Include files ให้เพิ่ม C:\CDX\INCLUDE เข้าไป



รูปที่ ง-1 แสดงขั้นตอนการทำให้ วิชวลซี พลัส พลัส รู้จัก ซี ดี เอกซ์ ขั้นตอนที่ 1

- 4 ใน Library files ให้เพิ่ม C:\CDX\LIB เข้าไป



รูปที่ ง-2 แสดงขั้นตอนการทำให้ วิชวลซี พลัส พลัส รู้จัก ซี ดี เอกซ์ ขั้นตอนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

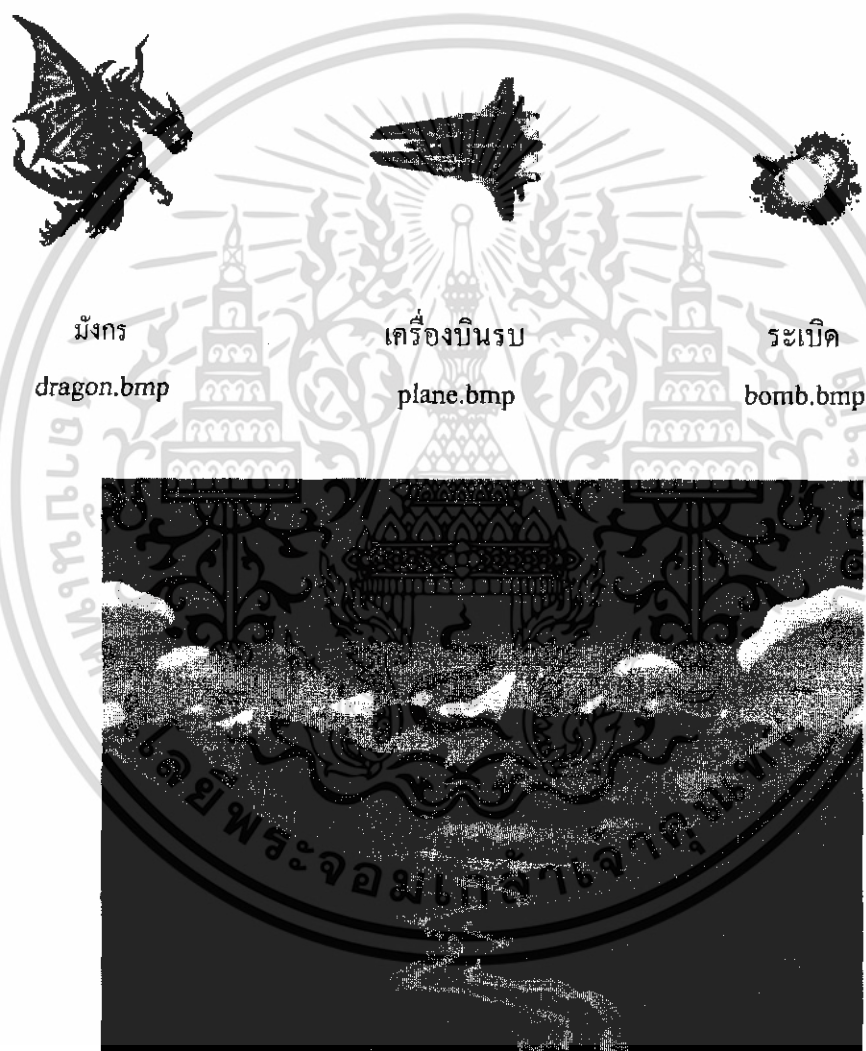


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างการสร้างเกมขั้นพื้นฐาน

### ตัวอย่างการใช้งานโปรแกรมช่วยสร้างเกม 2 มิติ ขั้นพื้นฐาน

ในตัวอย่างนี้ เป็นการสอนวิธีการใช้โปรแกรมสร้างเกมแบบง่ายๆ โดยให้ผู้เล่นเกม บังคับมังกรซึ่งลอยอยู่บนท้องฟ้า เมื่อมังกรบินไปชนเครื่องบิน จะทำให้เครื่องบินระเบิด และตกลงสู่พื้นดิน ดังรูป



รูปที่ จ-1 แสดงภาพที่ใช้ในการทำตัวอย่างเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อันดับแรกเราจะต้องทำการคัดลอกโฟลเดอร์ ResourceGame ไปเก็บไว้ที่ไดรฟ์ ซี (C:/) ก่อน จากนั้น ดับเบิลคลิกที่ gameCreator.exe เราก็สามารถใช้งานโปรแกรมช่วยสร้างเกม 2 มิติได้ ดังนี้

1. ให้เลือก สร้างเพิ่ม ( Add Folder ) ดังรูป



รูปที่ จ-2 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 1

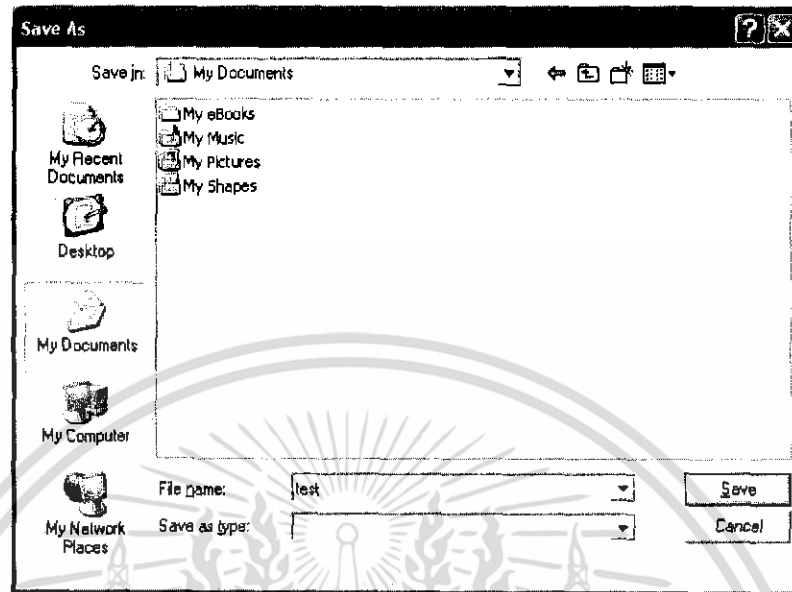
2. จะปรากฏหน้าต่างให้เลือกตำแหน่งที่จะเก็บไฟล์เกม คลิกที่ปุ่ม เลือก ดังรูป



รูปที่ จ-3 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 2

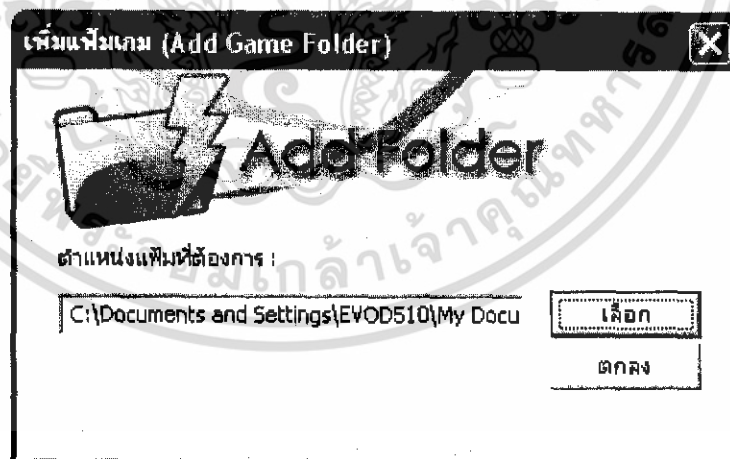
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ให้เลือกตำแหน่งที่จะเก็บไฟล์เกมที่จะสร้าง ให้ตั้งชื่อ ในที่นี้ตั้งเป็น test จากนั้นคลิกที่ปุ่ม Save ดังรูป



รูปที่ จ-4 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 3

4. หลังจาก Save แล้ว โปรแกรมจะแสดงตำแหน่งที่จะสร้าง ถ้าจะเลือกตำแหน่งใหม่ให้คลิกปุ่ม เลือก ถ้าตำแหน่งถูกต้องแล้วให้คลิกปุ่ม ตกลง ดังรูป



รูปที่ จ-5 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 4

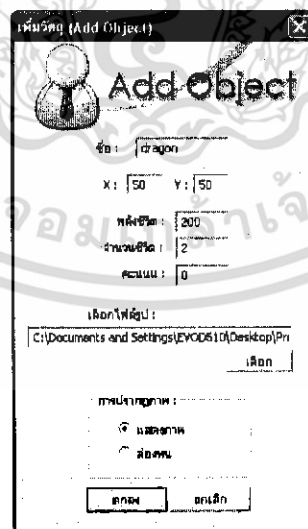
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการเพิ่ม ส่วนประกอบต่างๆ ภายในเกม ในที่นี้จะทำการ เพิ่ม ตัวละคร (Object) เข้าไปเป็นอันดับแรก ให้คลิกที่ปุ่ม เพิ่มตัวละคร ดังรูป



รูปที่ จ-6 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 5

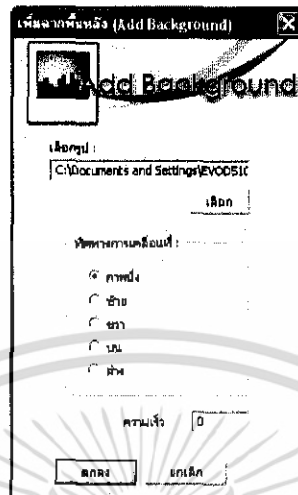
6. เมื่อหน้าต่างเพิ่มตัวละครปรากฏขึ้นมาให้ใส่ ชื่อว่า dragon ตำแหน่งที่จะให้ ตัวละครนั้นปรากฏที่พิกัด  $(x,y) = (50,50)$  ค่าพลังเป็น 200 ค่าชีวิตเป็น 2 คะแนนเป็น 0 เลือกตำแหน่งของรูปตัวละครนั้นในที่นี้ให้เป็นรูปมังกร และ กำหนดการปรากฏเป็น แสดงภาพ แล้วคลิกตกลง ดังรูป



รูปที่ จ-7 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 6

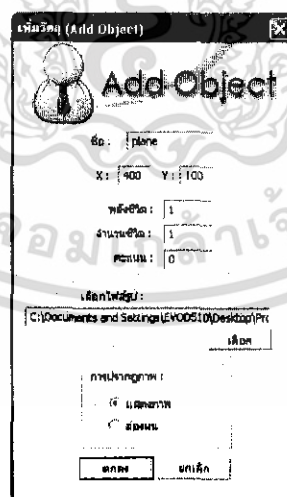
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการ เพิ่มฉากพื้นหลัง ให้เลือกตำแหน่งของภาพพื้นหลัง กำหนดทิศทาง การเลื่อนภาพเป็น ฉากนิ่ง ความเร็วเป็น 0 แล้วคลิก ตกลง ดังรูป



รูปที่ จ-8 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 7

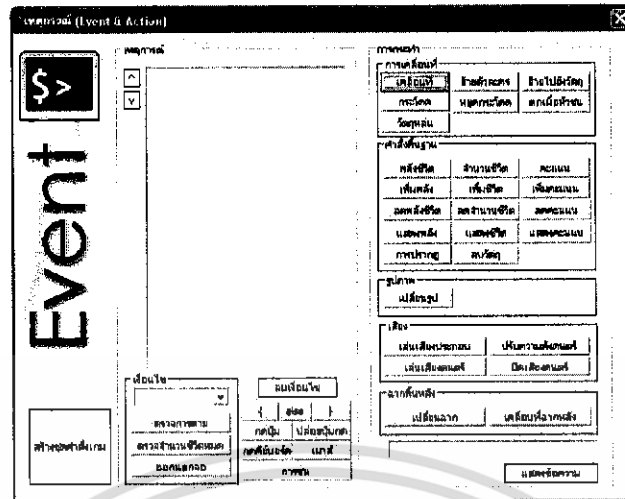
8. ทำการ เพิ่ม ตัวละครรูปเครื่องบินรบ เข้าไปอีกรูปใส่ ชื่อว่า plane ตำแหน่งที่จะให้ตัวละครนั้นปรากฏที่พิกัด  $(x,y) = (400,100)$  ค่าพลังเป็น 1 ค่าชีวิตเป็น 1 คะแนนเป็น 0 เลือกตำแหน่งของรูปตัวละครนั้นในที่นี่ให้เป็นรูปเครื่องบิน และ กำหนดการปรากฏเป็น แสดงภาพ แล้วคลิก ตกลง ดังรูป



รูปที่ จ-9 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 8

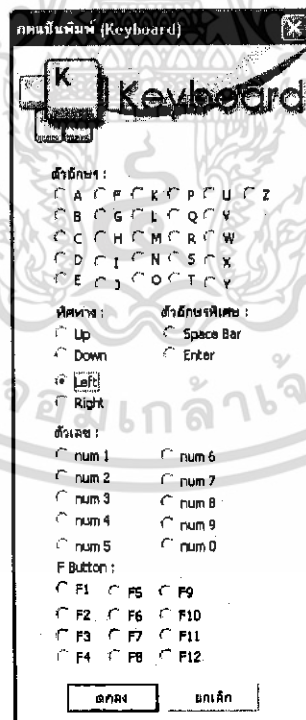
9. ทำการ เพิ่ม เหตุการณ์ (Event) เลือกที่ปุ่มเหตุการณ์ จะแสดงหน้าไฟล์ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-10 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 9

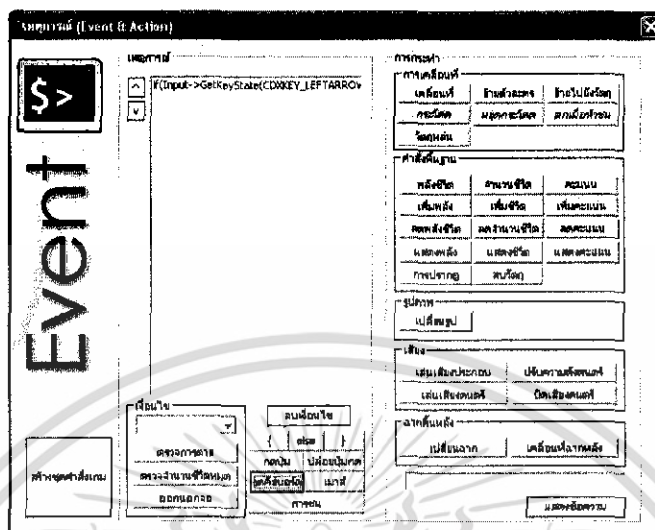
10. ต่อไปจะทำการใส่คำสั่งควบคุมมังก์ร คลิกที่ปุ่ม “กดปุ่มคีย์บอร์ด” ที่อยู่ด้านล่างของหน้าต่าง จะมีหน้าต่างปรากฏขึ้นมา เลือกทิศทางเป็น LEFT จากนั้น คลิก OK ดังรูป



รูปที่ จ-11 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 10

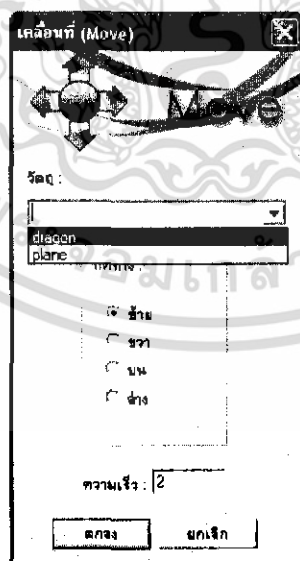
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ที่รายการว่างจะมีรายการที่เราทำปรากฏออกมาหลังจากที่เรากำหนดเหตุการณ์ ดังรูป



รูปที่ จ-12 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 11

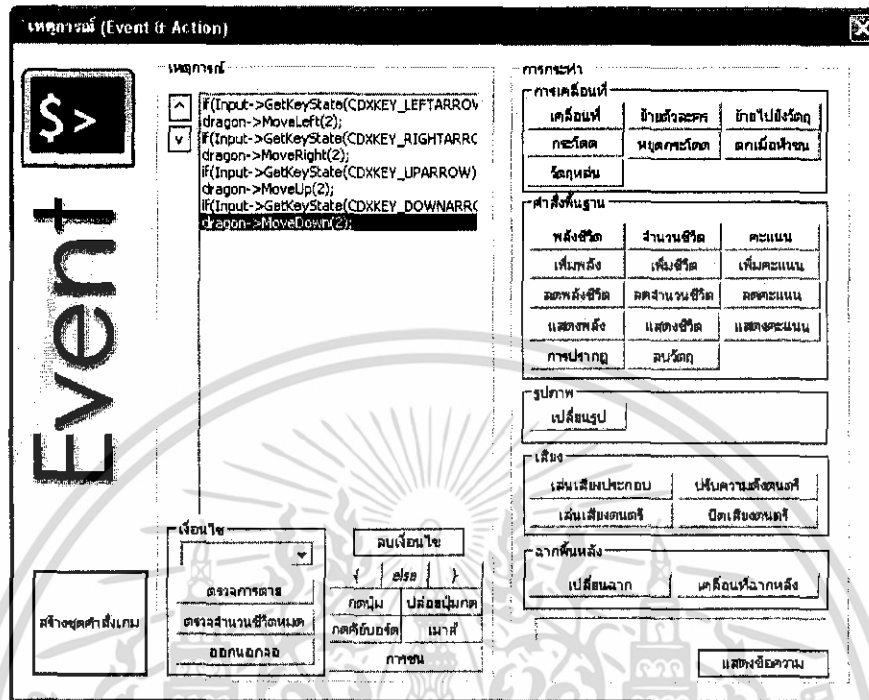
12. คลิกที่ปุ่ม “เคลื่อนที่” จะมีหน้าต่างปรากฏขึ้นมา ให้เลือก ชื่อตัวละครเป็น dragon ที่จะทำเหตุการณ์ แล้วเลือก ทิศทาง เป็น ซ้าย และ กำหนดความเร็ว เป็น 2 แล้วคลิก ตกลง ดังรูป



รูปที่ จ-13 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 12

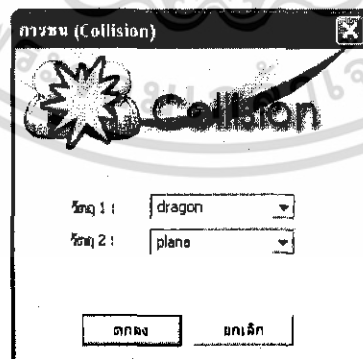
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. เนื่องจากต้องการให้บังคับตัวมังกรได้ในหลายๆ ทิศทาง ดังนั้น ให้ทำซ้ำข้อ 10 ถึง 12 แต่ให้เปลี่ยนจากทิศทาง ซ้าย เป็น ขวา ,บน ,ล่าง ตามลำดับ



รูปที่ จ-14 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 13

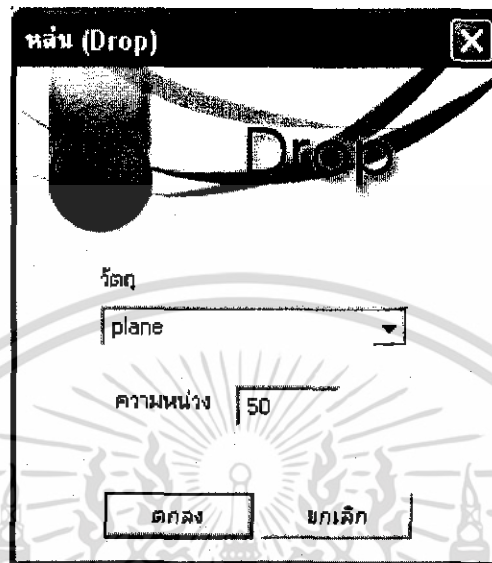
14. ในตัวอย่างนี้ ต้องการให้มังกรชนเครื่องบิน แล้วจะให้เครื่องบินตก ในขั้นตอนนี้ ให้คลิกที่ปุ่ม การชน ที่อยู่ด้านล่าง จะมีหน้าต่างโผล่มา ให้เลือกวัตถุ 1 เป็น dragon และ วัตถุ 2 เป็น plane แล้วคลิก ตกลง ดังรูป



รูปที่ จ-15 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 14

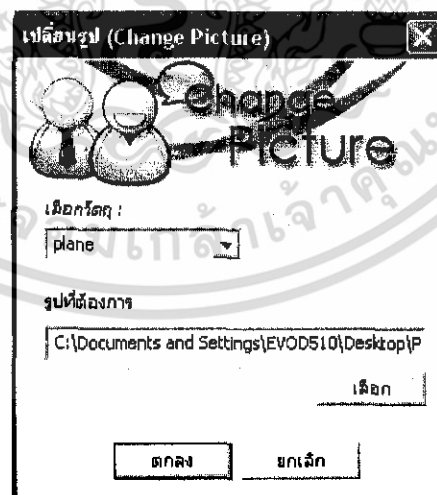
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. ให้คลิกที่ { เพื่อทำการบอกว่า ต่อไปนี้จะเป็นการใส่ชุดคำสั่งมากกว่า 1 คำสั่ง จากนั้น คลิกที่วัตถุหลัก เพื่อสั่งให้เครื่องบนตก โดยเลือกวัตถุ plane และใส่ ความหน่วงเป็น 50



รูปที่ จ-16 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 15

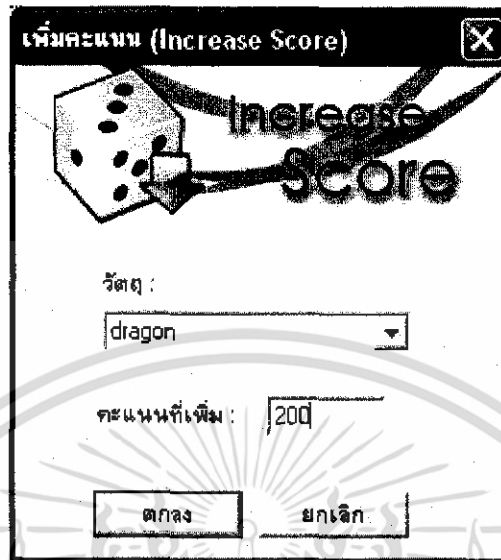
16. เปลี่ยนรูปเครื่องบินให้เป็นรูปเครื่องบินระเบิด โดยการกดปุ่ม เปลี่ยนรูป เลือก วัตถุเป็น plane และเลือกรูปภาพเป็นรูประเบิด



รูปที่ จ-17 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 16

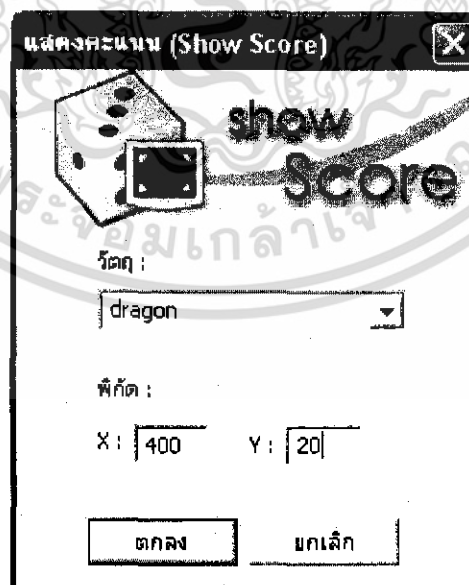
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. ทำการเพิ่มคะแนนให้กับมังกร โดยกดปุ่ม “เพิ่มคะแนน” เลือกตัวละครที่ต้องการเพิ่มเป็น dragon และใส่คะแนนที่ต้องการเพิ่มเป็น 200



รูปที่ จ-18 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 17

18. ต่อไปให้คลิกที่ } เพื่อจบการใส่ชุดคำสั่ง จากนั้นจะทำการแสดงค่าคะแนน โดยเลือกกดที่ปุ่ม “แสดงคะแนน” เลือกคะแนนของตัวละคร dragon ที่พิกัด 400,20



รูปที่ จ-19 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 19. จะได้เหตุการณ์ทั้งหมด ดังรูป

**เหตุการณ์ (Event & Action)**

**เหตุการณ์**

```

^
v
if(Input->GetKeyState(CDXKEY_LEFTARROV
dragon->MoveLeft(2);
if(Input->GetKeyState(CDXKEY_RIGHTARRC
dragon->MoveRight(2);
if(Input->GetKeyState(CDXKEY_UPARROW)
dragon->MoveUp(2);
if(Input->GetKeyState(CDXKEY_DOWNARRC
dragon->MoveDown(2);
if(dragon->Collision(plane))
{
plane->Drop(50);
dragon->IncScore(200);
plane->ChangeSprite(Screen,"Resource/Spr
dragon->ShowScore(hdc,400,20);

```

**การเคลื่อนที่**

เคลื่อนที่	ย้ายตัวละคร	ย้ายไปยังวัตถุ
กระโดด	หยุดกระโดด	ตกเมื่อทำชน
วัตถุหมุน		

**คำสั่งพื้นฐาน**

พลังชีวิต	จำนวนชีวิต	คะแนน
เพิ่มพลัง	เพิ่มชีวิต	เพิ่มคะแนน
ลดพลังชีวิต	ลดจำนวนชีวิต	ลดคะแนน
แสดงพลัง	แสดงชีวิต	แสดงคะแนน
การปรากฏ	ลบวัตถุ	

**รูปภาพ**

เปลี่ยนรูป

**เสียง**

เล่นเสียงประกอบ	ปรับคานาตั้งต้น
เล่นเสียงดนตรี	ปิดเสียงดนตรี

**ฉากหลัง**

เปลี่ยนฉาก | เคลื่อนที่ฉากหลัง

**เงื่อนไข**

สร้างชุดคำสั่งเกม

ตรวจสอบการตาย

ตรวจสอบจำนวนชีวิตหมด

ออกนอกจอ

ลบเงื่อนไข

{ else }

กดปุ่ม | ปลดปล่อยปุ่มกด

กดคีย์บอร์ด | เมาส์

การชน

แสดงข้อความ

## รูปที่ จ-20 แสดงตัวอย่างการสร้างเกม ขั้นตอนที่ 19

20. ให้คลิกที่ปุ่ม “สร้างชุดคำสั่ง” เพื่อสร้างชุดคำสั่งลงในเกมเอ็นจินที่เลือกสร้างไว้ในขั้นตอนที่ 1-4 จากนั้น กดประมวลผลในโปรแกรมวิซวล ซี พลัส พลัส ด้วยโหมดรีลีส (Release Mode) หากทำถูกต้องทุกขั้นตอน จะได้เกมที่เราเป็น มังกร บินไปชนเครื่องบินรบ แล้วเครื่องบินรบจะหล่นลงมา พร้อมทั้งเพิ่มคะแนน 200 และแสดงคะแนนที่ด้านบนกลางหน้าจอ



รูปที่ จ-21 ผลลัพธ์จากการเล่นเกม



รูปที่ จ-22 ผลลัพธ์จากการเล่นเกม

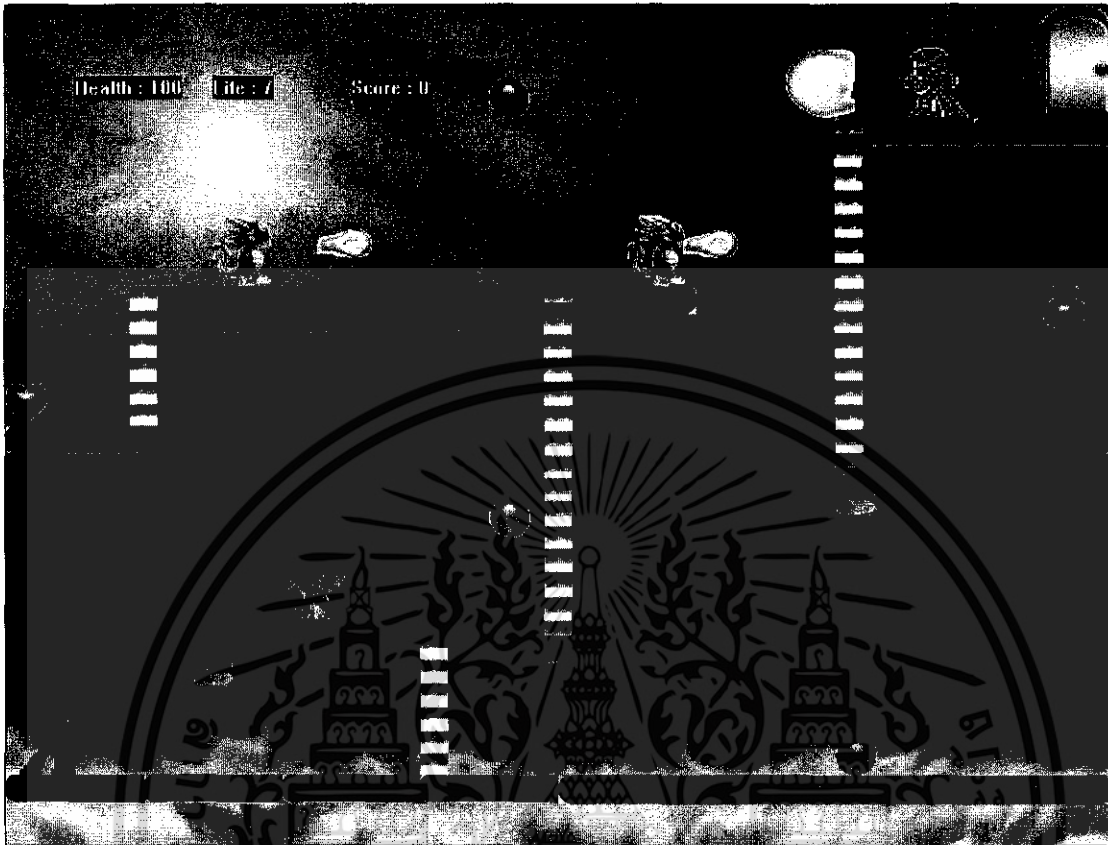
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ฉ  
ตัวอย่างการสร้างเกมแบบซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างการสร้างเกมแบบซบซ้อน

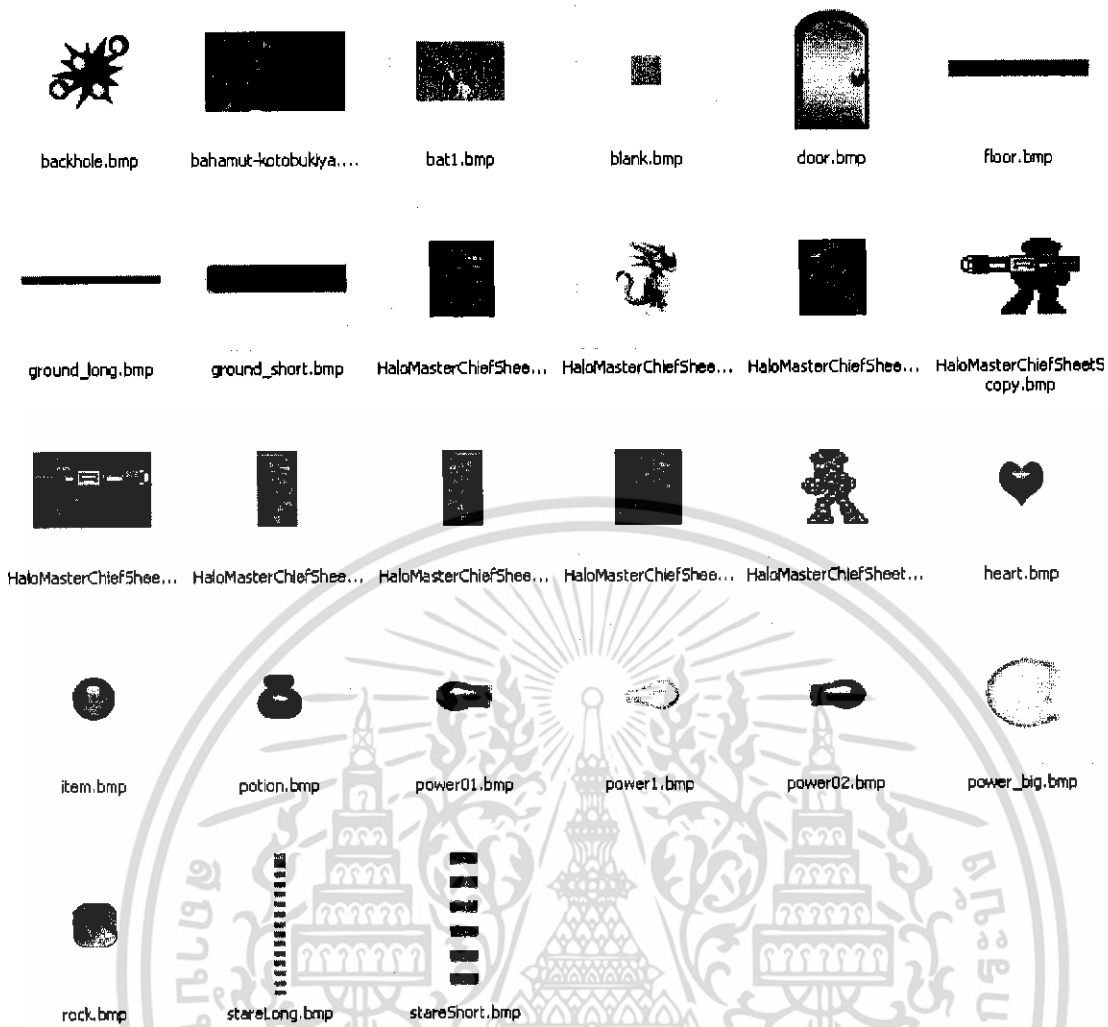


รูปที่ ๑-1 แสดงเกมตัวอย่างแบบซบซ้อน

การสร้างเกมนั้น ไม่ว่าจะสร้างด้วยการเขียนชุดคำสั่งเองหรือจะใช้เครื่องมือหรือโปรแกรมช่วยในการสร้างย่อมต้องมีการออกแบบเกมเอาไว้ก่อน ซึ่งอาจจะวาดออกมาเป็นภาพร่างของกระดาษหรือใช้โปรแกรมตกแต่งรูปภาพในคอมพิวเตอร์ช่วย การออกแบบเกมที่ดีนั้นต้องคิดให้รอบคอบก่อนลงมือสร้างเกม วางเงื่อนไขและกำหนดบทบาทของตัวละครหรือวัตถุต่าง ๆ ให้เรียบร้อยและมองเห็นเป็นภาพในจินตนาการของเราได้

ในที่นี้เราจะสร้างเกม 2 มิติแบบหน้าจอเดี่ยว ตัวละครสามารถเดิน กระโดด ปีน ได้ ต้องหลบเลี่ยงก้อนหินที่ตกลงมาและกระสุนที่ปล่อยออกมาจากศัตรู ในขณะที่เดียวกันก็ต้องเก็บของเพื่อรักษาพลังชีวิตและคะแนนที่อาจสูญเสียไป จนไปถึงประตูด้านบนสุดเพื่อไปยังด่านของหัวหน้าใหญ่

เมื่อเราคิดเกมของเราได้แล้ว ต่อไปเราก็ต้องสร้างตัวละครและวัตถุต่าง ๆ



รูปที่ ๑-2 แสดงรูปตัวละครที่ใช้ในเกม

การสร้างรูปเพื่อนำมาทำเป็นวัตถุในเกมมีความสำคัญอย่างมาก เพราะถือเป็นส่วนประกอบหลักและความสวยงามของเกม เราสามารถใช้โปรแกรมตกแต่งรูปภาพมาช่วยในการสร้างตัวละครต่างๆ ได้ เช่น โปรแกรมโฟโตช้อป เป็นต้น

ขั้นตอนต่อมาเป็นการคิดเงื่อนไขในเกมให้เป็นตรรกะเพื่อใช้ในการเขียนชุดคำสั่ง แต่สำหรับโปรแกรมช่วยสร้างเกม 2 มิติ ผู้สร้างเกมไม่จำเป็นต้องเขียนชุดคำสั่ง เพียงแค่สร้างตรรกะลำดับก่อนหลังในการเพิ่มคำสั่งด้วยการคลิกที่ปุ่มต่างๆ ในโปรแกรม ซึ่งมีขั้นตอนดังนี้

1. เปิดโปรแกรมช่วยสร้างเกม 2 มิติ
2. คลิกที่ สร้าง โฟลเตอร์ ระบุชื่อโฟลเตอร์ตามต้องการ
3. คลิกที่ปุ่ม เพิ่มวัตถุ เพื่อเพิ่มตัวละครหรือวัตถุ ดังนี้

### รูปที่ ๓-3 แสดงการเพิ่มวัตถุหรือตัวละครลงในเกม

เราจะต้องทำการเพิ่มวัตถุหรือตัวละครที่ต้องใช้ในเกมนของเรา ดังตัวอย่างรูปที่ ๓-3 ให้ใส่ชื่อของตัวละคร และคุณสมบัติต่าง ๆ ของตัวละคร

ชื่อ : ชื่อของวัตถุหรือตัวละคร

X,Y : พิกัดแกน X แกน Y ที่ต้องการให้วัตถุปรากฏในเกม

พลังชีวิต : ค่าพลังของวัตถุ ใส่ได้ไม่เกิน 999

จำนวนชีวิต : จำนวนชีวิตของตัวละคร

คะแนน : คะแนนที่เพิ่มหรือลดเมื่อกระทำเหตุการณ์ต่างๆ

จากนั้นให้เราเลือกรูปภาพที่จะนำมาเป็นภาพให้กับวัตถุ ซึ่งสามารถเลือกได้โดยที่ขนาดต้องไม่เกินขนาดของหน้าจอ คือ 800 x 600 สุดท้ายจึงเลือกการปรากฏของวัตถุ

แสดงภาพ : วัตถุปรากฏเมื่อตอนเริ่มเกม

ล่องหน : วัตถุซ่อนหรือไม่ปรากฏเมื่อเริ่มต้นโปรแกรม

ให้เรากำหนดแบบนี้ซ้ำกับวัตถุอื่น ๆ ที่เราต้องการนำมาใส่ในเกมของเรา ดังตาราง

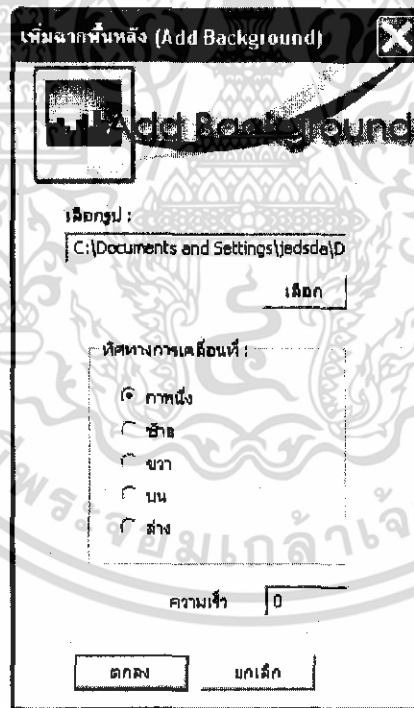
ชื่อ	แกน X	แกน Y	พลังชีวิต	จำนวนชีวิต	คะแนน	การปรากฏ
Halo	10	500	100	20	0	แสดงภาพ
bat	0	400	0	1	100	แสดงภาพ
ghost	700	500	0	2	200	แสดงภาพ
e1	150	150	0	2	200	แสดงภาพ
E2	450	150	0	2	200	แสดงภาพ
b1	650	30	0	5	500	แสดงภาพ
door	700	0	0	0	0	แสดงภาพ
groundS1	0	550	0	0	0	แสดงภาพ
groundS2	0	300	0	0	0	แสดงภาพ
groundS 3	200	550	0	0	0	แสดงภาพ
groundS4	300	450	0	0	0	แสดงภาพ
groundS5	600	80	0	0	0	แสดงภาพ
groundS6	700	80	0	0	0	แสดงภาพ
groundS7	100	550	0	0	0	แสดงภาพ
groundS8	300	550	0	0	0	แสดงภาพ
groundL1	400	550	0	0	0	แสดงภาพ
groundL2	400	300	0	0	0	แสดงภาพ
groundL3	100	200	0	0	0	แสดงภาพ
heart	0	270	0	0	0	แสดงภาพ
potion	750	200	0	0	0	แสดงภาพ
I1	350	350	0	0	0	แสดงภาพ
I2	600	520	0	0	0	แสดงภาพ
I3	250	50	0	0	0	แสดงภาพ
I4	350	50	0	0	0	แสดงภาพ
hapowL	0	0	0	0	0	กล่อง
hapowR	0	0	0	0	0	กล่อง
epowR1	190	160	0	0	0	แสดงภาพ
epowR2	490	160	0	0	0	แสดงภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bpowL1	600	30	0	0	0	แสดงภาพ
rock1	135	0	0	0	0	แสดงภาพ
rock2	600	0	0	0	0	แสดงภาพ
stareS1	300	450	0	0	0	แสดงภาพ
stareS2	90	200	0	0	0	แสดงภาพ
stareL1	390	200	0	0	0	แสดงภาพ
stareL2	600	80	0	0	0	แสดงภาพ
bahamut	999	999	0	10	0	ล่องหน
darkball	999	999	0	0	0	ล่องหน

ตารางที่ ฉ-1 ตารางแสดงรายละเอียดของตัวละครในเกม

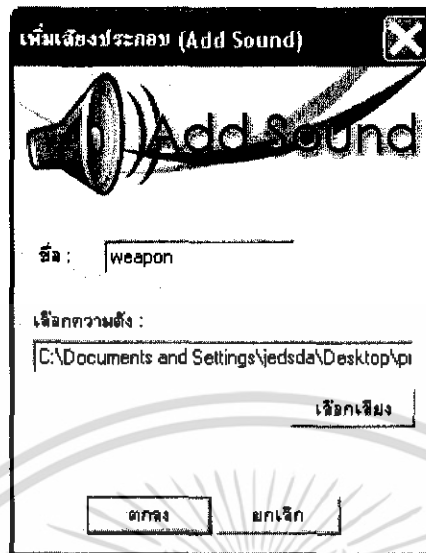
4. เพิ่มภาพพื้นหลังให้คลิกที่ปุ่ม เพิ่มฉากหลัง ระบุค่าต่าง ๆ ดังรูป



รูปที่ ฉ-4 แสดงการเพิ่มฉากหลัง

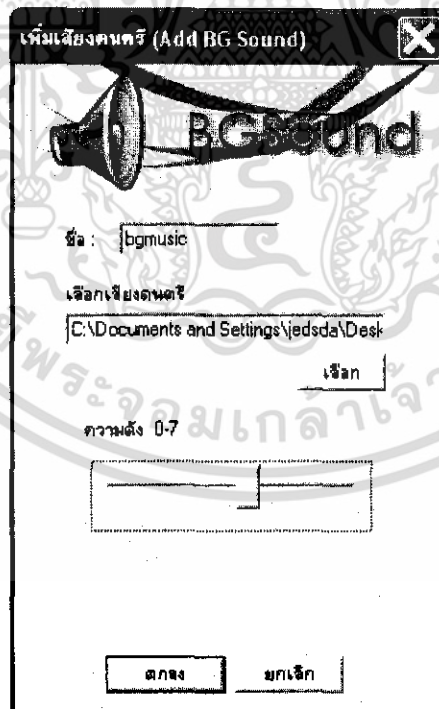
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เพิ่มเสียงประกอบให้คลิกที่ปุ่ม เพิ่มเสียงประกอบ ดังรูป



รูปที่ ฉ-5 แสดงการเพิ่มเสียงประกอบ

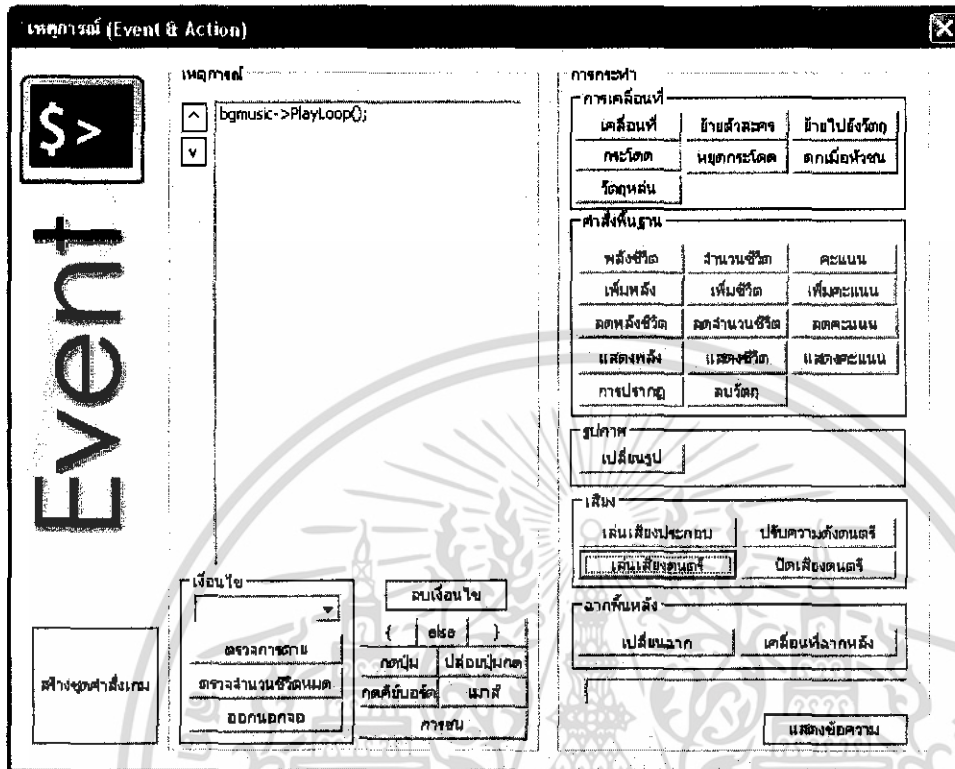
6. เพิ่มเสียงดนตรีให้คลิกที่ปุ่ม เพิ่มเสียงดนตรี ดังรูป



รูปที่ ฉ-6 แสดงการเพิ่มเสียงดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เพิ่มเหตุการณ์และคำสั่ง (Event & Action)  
เราสามารถเพิ่มคำสั่งได้จากหน้าต่างเหตุการณ์ ดังรูป



รูปที่ ๗-7 แสดงหน้าต่างเพิ่มเหตุการณ์และคำสั่ง

จากรูปเราเพิ่มคำสั่งแรกเป็น bgmusic -> PlayLoop(); ซึ่งสามารถเพิ่มคำสั่งได้โดยคลิกที่ปุ่ม เล่นเสียงดนตรี ในกลุ่ม เสียง จะปรากฏ ไดอะล็อก ดังรูป



รูปที่ ๗-8 แสดงการใส่คำสั่งเล่นเสียงดนตรี

เลือกเสียงดนตรีที่เราได้เพิ่มลงไปในช่วงตอนแรก ในที่นี้ คือ bgmusic แล้วคลิก ตกลง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้ทำการเพิ่มคำสั่งต่าง ๆ โดยใช้ปุ่มในหน้าต่างเหตุการณ์ ดังนี้

ปุ่ม	คำสั่ง
เล่นเสียงดนตรี	bgmusic->PlayLoop();
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_LEFTARROW)==CDXKEY_REPEAT)
{	{
เคลื่อนที่	halo->MoveLeft(2);
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet4.bmp");
}	}
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_RIGHTARROW)==CDXKEY_REPEAT)
{	{
เคลื่อนที่	halo->MoveRight(2);
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet9.bmp");
}	}
กดปุ่ม	if(Input->GetKeyState(CDXKEY_SPACE)==CDXKEY_PRESS)
	{
กระโดด	halo->Jump(0,-100*1.1,0,40);
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
}	}
กดปุ่ม	if(Input->GetKeyState(CDXKEY_Z)==CDXKEY_PRESS)
{	{
ย้ายไปยังตัวละคร	hapowL->WarpToObj(halo);
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet6.bmp");
การปรากฏ	hapowL->Visible(Screen);
เล่นเสียงประกอบ	weapon->Play();
}	}
else	else
เคลื่อนที่	hapowL->MoveLeft(4);
กดปุ่ม	if(Input->GetKeyState(CDXKEY_X)==CDXKEY_PRESS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

{	{
ย้ายไปยังตัว ละคร	hapowR->WarpToObj(halo);
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet5 copy.bmp");
การปรากฏ	hapowR->Visible(Screen);
เล่นเสียง ประกอบ	weapon->Play();
}	}
else	else
เคลื่อนที่	hapowR->MoveRight(4);
การชน	if(halo->Collision(stareL1))
{	{
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveUp(1);
}	}
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_DOWNARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveDown(1);
}	}
}	}
การชน	if(halo->Collision(stareL2))
{	{
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveUp(1);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}	}
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_DOWNARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveDown(1);
}	}
}	}
การชน	if(halo->Collision(stareS1))
{	{
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveUp(1);
}	}
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_DOWNARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveDown(1);
}	}
}	}
การชน	if(halo->Collision(stareS2))
{	{
กดคีย์บอร์ด	if(Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveUp(1);
}	}
คีย์บอร์ด	if(Input->GetKeyState(CDXKEY_DOWNARROW)==CDXKEY_REPEAT)
{	{
เปลี่ยนรูป	halo->ChangeSprite(Screen,"Resource/Sprite/HaloMasterChiefSheet7.bmp");
เคลื่อนที่	halo->MoveDown(1);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}	}
}	}
การชน	if(halo->Collision(groundL1))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundL2))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundL3))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS1))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS2))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS3))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS4))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS5))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS6))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS7))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(groundS8))
หยุดกระโดด	halo->StopJump();
การชน	if(halo->Collision(potion))
{	{
ลบวัตถุ	potion->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(800);
}	}
การชน	if(halo->Collision(heart))
{	{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลบวัตถุ	heart->Remove(Screen);
เพิ่มชีวิต	halo->IncLife(1);
}	}
การชน	if(halo->Collision(I1))
{	{
ลบวัตถุ	I1->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(500);
}	}
การชน	if(halo->Collision(I2))
{	{
ลบวัตถุ	I2->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(500);
}	}
การชน	if(halo->Collision(I3))
{	{
ลบวัตถุ	I3->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(500);
}	}
การชน	if(halo->Collision(I4))
{	{
ลบวัตถุ	I4->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(500);
}	}
วัตถุหล่น	rock1->Drop(40);
วัตถุหล่น	rock2->Drop(90);
ออกนอกจอ	if(rock1->CheckOutScreen())
{	{
ย้ายตัวละคร	rock1->Warp(135,0);
หยุดกระโดด	rock1->StopJump();
}	}
ออกนอกจอ	if(rock2->CheckOutScreen())

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

{	{
ย้ายตัวละคร	rock2->Warp(600,0);
หยุดกระโดด	rock2->StopJump();
}	}
เคลื่อนที่	ghost->MoveLeft(2);
ออกนอกจอ	if(ghost->CheckOutScreen())
ย้ายตัวละคร	ghost->Warp(700,500);
การชน	if(hapowL->Collision(ghost))
{	{
ลบวัตถุ	hapowL->Remove(Screen);
ลดจำนวนชีวิต	ghost->DecLife(1);
}	}
การชน	if(hapowR->Collision(ghost))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
ลดจำนวนชีวิต	ghost->DecLife(1);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(ghost->CheckGameOver())
{	{
ลบวัตถุ	ghost->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(200);
เพิ่มชีวิต	ghost->IncLife(1);
}	}
เคลื่อนที่	bat->MoveRight(3);
ออกนอกจอ	if(bat->CheckOutScreen())
ย้ายตัวละคร	bat->Warp(0,400);
การชน	if(hapowL->Collision(bat))
{	{
ลบวัตถุ	hapowL->Remove(Screen);
ลบวัตถุ	bat->Remove(Screen);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มคะแนน	halo->IncScore(100);
}	}
การชน	if(hapowR->Collision(bat))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
ลบวัตถุ	bat->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(100);
}	}
เคลื่อนที่	epowR1->MoveRight(2);
ออกนอกจอ	if(epowR1->CheckOutScreen())
ย้ายตัวละคร	epowR1->Warp(190,160);
เคลื่อนที่	epowR2->MoveRight(2);
ออกนอกจอ	if(epowR2->CheckOutScreen())
ย้ายตัวละคร	epowR2->Warp(490,160);
การชน	if(hapowL->Collision(e1))
{	{
ลบวัตถุ	hapowL->Remove(Screen);
ลดจำนวนชีวิต	e1->DecLife(1);
}	}
การชน	if(hapowR->Collision(e1))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
ลดจำนวนชีวิต	e1->DecLife(1);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(e1->CheckGameOver())
{	{
ลบวัตถุ	e1->Remove(Screen);
ลบวัตถุ	epowR1->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(200);
เพิ่มชีวิต	e1->IncLife(1);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}	}
การชน	if(hapowL->Collision(E2))
{	{
ลบวัตถุ	hapowL->Remove(Screen);
ลดจำนวนชีวิต	E2->DecLife(1);
}	}
การชน	if(hapowR->Collision(E2))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
ลดจำนวนชีวิต	E2->DecLife(1);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(E2->CheckGameOver())
{	{
ลบวัตถุ	E2->Remove(Screen);
ลบวัตถุ	epowR2->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(200);
เพิ่มชีวิต	E2->IncLife(1);
}	}
เคลื่อนที่	bpowL1->MoveLeft(2);
ออกนอกจอ	if(bpowL1->CheckOutScreen())
ย้ายตัวละคร	bpowL1->Warp(600,30);
การชน	if(hapowL->Collision(bpowL1))
ลบวัตถุ	hapowL->Remove(Screen);
การชน	if(hapowR->Collision(bpowL1))
ลบวัตถุ	hapowR->Remove(Screen);
การชน	if(hapowR->Collision(b1))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
ลดจำนวนชีวิต	b1->DecLife(1);
}	}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การชน	if(hapowL->Collision(b1))
{	{
ลบวัตถุ	hapowL->Remove(Screen);
ลดจำนวนชีวิต	b1->DecLife(1);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(b1->CheckGameOver())
{	{
ลบวัตถุ	b1->Remove(Screen);
ลบวัตถุ	bpowL1->Remove(Screen);
เพิ่มคะแนน	halo->IncScore(500);
เพิ่มชีวิต	b1->IncLife(1);
}	}
การชน	if(halo->Collision(ghost))
{	{
ลบวัตถุ	ghost->Remove(Screen);
ลดจำนวนชีวิต	halo->DecLife(1);
เพิ่มชีวิต	ghost->IncLife(1);
}	}
การชน	if(halo->Collision(bat))
{	{
ลบวัตถุ	bat->Remove(Screen);
ลดจำนวนชีวิต	halo->DecLife(1);
}	}
การชน	if(halo->Collision(epowR1))
{	{
ย้ายตัวละคร	epowR1->Warp(190,160);
ลดจำนวนชีวิต	halo->DecLife(1);
}	}
การชน	if(halo->Collision(epowR2))
{	{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย้ายตัวละคร	epowR2->Warp(490,160);
ลดจำนวนชีวิต	halo->DecLife(1);
}	}
การชน	if(halo->Collision(bpowL1))
{	{
ย้ายตัวละคร	bpowL1->Warp(600,30);
ลดจำนวนชีวิต	halo->DecLife(2);
}	}
การชน	if(halo->Collision(rock1))
{	{
ย้ายตัวละคร	rock1->Warp(135,0);
หยุดกระโดด	rock1->StopJump();
ลดพลังชีวิต	halo->DecHealth(50);
}	}
การชน	if(halo->Collision(rock2))
{	{
ย้ายตัวละคร	rock2->Warp(600,0);
หยุดกระโดด	rock2->StopJump();
ลดพลังชีวิต	halo->DecHealth(50);
}	}
ตรวจการตาย	if(halo->CheckDead())
{	{
ลดจำนวนชีวิต	halo->DecLife(1);
เพิ่มพลังชีวิต	halo->IncHealth(100);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(halo->CheckGameOver())
{	{
ลบวัตถุ	halo->Remove(Screen);
แสดงข้อความ	ShowMessage(hdc,"YOU LOSE");
}	}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การชน	if(halo->Collision(door))
{	{
ลบวัตถุ	b1->Remove(Screen); bat->Remove(Screen); bpowL1->Remove(Screen); door->Remove(Screen); epowR1->Remove(Screen); epowR2->Remove(Screen); groundL2->Remove(Screen); groundL3->Remove(Screen); groundS2->Remove(Screen); groundS4->Remove(Screen); groundS5->Remove(Screen); groundS6->Remove(Screen); heart->Remove(Screen); I1->Remove(Screen); I2->Remove(Screen); I3->Remove(Screen); I4->Remove(Screen); potion->Remove(Screen); rock1->Remove(Screen); rock2->Remove(Screen); stareL1->Remove(Screen); stareL2->Remove(Screen); stareS1->Remove(Screen); stareS2->Remove(Screen);
ย้ายตัวละคร	e1->Warp(999,999); E2->Warp(999,999); ghost->Warp(999,999);
เปลี่ยนฉาก	bg->ChangeBackground(Screen,"Resource/Background/cave.bmp");
ย้ายตัวละคร	halo->Warp(10,500);
การปรากฏ	bahamut->Visible(Screen);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย้ายตัวละคร	bahamut->Warp(500,390);
}	}
การชน	if(bahamut->Collision(hapowL))
{	{
ลดจำนวนชีวิต	bahamut->DecLife(1);
ลบวัตถุ	hapowL->Remove(Screen);
}	}
การชน	if(bahamut->Collision(hapowR))
{	{
ลดจำนวนชีวิต	bahamut->DecLife(1);
ลบวัตถุ	hapowR->Remove(Screen);
}	}
ตรวจจำนวน ชีวิตทั้งหมด	if(bahamut->CheckGameOver())
{	{
ลบวัตถุ	bahamut->Remove(Screen);
แสดงข้อความ	ShowMessage(hdc,"ENDING");
}	}
การชน	if(halo->Collision(darkball))
{	{
ย้ายตัวละคร	darkball->Warp(500,470);
ลดจำนวนชีวิต	halo->DecLife(1);
}	}
การชน	if(darkball->Collision(hapowR))
{	{
ลบวัตถุ	hapowR->Remove(Screen);
}	}
ออกนอกจอ	if(darkball->CheckOutScreen())
ย้ายตัวละคร	darkball->Warp(500,470);
else	else
เคลื่อนที่ซ้าย	darkball->MoveLeft(3);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงพลัง	halo->ShowHealth(hdc,50,50)
แสดงชีวิต	halo->ShowLife(hdc,150,50);
แสดงคะแนน	halo->ShowScore(hdc,250,50);

### ตารางที่ ๓-2 ตารางแสดงการเพิ่มคำสั่งต่าง ๆ ในหน้าต่างเหตุการณ์

8. เมื่อใส่คำสั่งครบแล้วให้คลิกปุ่ม สร้างชุดคำสั่งเกม เราก็จะได้ชุดคำสั่งเกมที่โปรแกรมได้สร้างไว้ในโฟลเดอร์ที่เรากำหนดไว้ในตอนแรก
9. เปิดโปรแกรมวิซวล ซี พลัส พลัส 6.0 ไปที่ไฟล์ เลือก Open Workspace จะปรากฏไดอะล็อกขึ้นมา ให้เราไปที่โฟลเดอร์ที่เรากำหนดไว้ คลิก project.dsw กดตกลง โปรแกรมวิซวล ซี พลัส พลัส 6.0 จะนำชุดคำสั่งทั้งหมดมาแสดงผลในโปรแกรม
10. เลือก Build > Set active configuration จะปรากฏไดอะล็อกขึ้นมา เลือก win32 Release
11. คลิกเลือกแท็บ file view ด้านซ้ายของโปรแกรม คลิกที่ไฟล์ Test.cpp
12. ไปที่ Build > Compile Test.cpp แล้วรอให้โปรแกรมประมวลผลชุดคำสั่งจนเสร็จ
13. ไปที่ Build อีกครั้ง แล้วเลือก Execute project.exe โปรแกรมวิซวล ซี พลัส พลัส 6.0 จะทำการแปลงชุดคำสั่งของเราให้เป็นไฟล์ที่สามารถประมวลผลได้ เป็นอันจบขั้นตอนการสร้างเกม