สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกมกรู๊ฟ อันลิมิเต็ด

GROOVE UNLIMITED GAME

ณ พงศ์ วรัญญานนท์
มนัสนิตย์ เศรษฐาวงศ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ว.คอ.
ว26111 ว
2547

เลขหมู่.....
เลขทะเบียน 58777
วัน,เดือน,ปี 10 ก.พ. 2549

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

GROOVE UNLIMITED GAME

NA-PONGS VARUNYANONT

MANASNIT SETTHAWONG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	เกมกรู๊ฟ อันลิมิเต็ด	
	GROOVE UNLIMITED	
ชื่อนักศึกษา	นายณ พงศ์ วรัญญานนท์	44050415
	นางสาวมนัสนิตย์ เศรษฐาวงศ์	44050453
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2547

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผศ.ดร.จีรพร ศรีสวัสดิ์	
กรรมการ	ดร.กรกช ประชุมรักษ์	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	เกมกรู๊ฟ อันลิมิเต็ด	
ชื่อนักศึกษา	นายณ พงศ์ วรัญญานนท์	44050415
	นางสาวมนัสนิตย์ เศรษฐาวงศ์	44050453
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2547	
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

บทคัดย่อ

เกมกรู๊ฟ อันลิมิเต็ด (Groove Unlimited) เป็นเกมประเภทใหม่ประเภทหนึ่ง ที่สามารถทำให้ผู้เล่นเกิดความสนุกสนานเพลิดเพลินไปกับเกมได้ นอกจากนี้เกมกรู๊ฟ อันลิมิเต็ด ยังสามารถช่วยฝึกทักษะทางด้านดนตรี และประสาทสัมผัสต่าง ๆ ให้ทำงานประสานกัน โดยผู้เล่นต้องกดคีย์บอร์ดให้ตรงกับจังหวะหรือตัวโน้ตของเพลง เกมกรู๊ฟ อันลิมิเต็ดสามารถตอบสนองความต้องการของผู้เล่นได้มากกว่าเกมอื่น ๆ ที่เป็นประเภทเดียวกัน นั่นก็คือ นอกจากที่ผู้เล่นสามารถเลือกเล่นเพลงที่มีให้อยู่แล้วในเกม ผู้เล่นยังสามารถนำเพลงที่ต้องการเล่นซึ่งเป็นไฟล์เพลงต่าง ๆ เข้าสู่เกมได้

การพัฒนาเกม นอกจากผู้พัฒนาต้องมีความสามารถทางด้านการเขียนโปรแกรมแล้ว จะต้องใช้ความคิดสร้างสรรค์ในการออกแบบกติกา รูปแบบการเล่นเกม รวมไปถึงภาพกราฟิกและเสียงประกอบ ซึ่งเป็นส่วนประกอบที่สำคัญอย่างยิ่งที่จะทำให้เกมนั้นเป็นเกมที่สมบูรณ์และสามารถดึงดูดความสนใจของผู้เล่นได้เป็นอย่างดี ซึ่งเกมกรู๊ฟ อันลิมิเต็ดได้รวมส่วนประกอบของการทำเกมเข้าไว้ด้วยกัน ซึ่งทำให้เกมสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองความต้องการของผู้เล่นได้มากที่สุดเท่าที่จะเป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	GROOVE UNLIMITED GAME
Students	Mr.Na-Pongs Varunyanont 44050415
	Miss.Manasnit Setthawong 44050453
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Academic Year	2004
Special Project Advisor	Assoc.Prof.Teerawat Prakobphol

ABSTRACT

Groove Unlimited Game is a kind of modern entertainment game. The players can enjoy and learn from playing this game, furthermore, Groove Unlimited Game can help players improve their music skill and perception because players should press button on keyboards to synchronize with the rhythm or notes of the playing music. Groove Unlimited Game can be well-responsive game to the players need, not only they can select any available songs in the game, they can have their own choices by importing any music files as they want into the game.

In this game development, knowledge in game programming only, seems to be not enough. Developers should have creative thinking for game designing, including graphic and sound designing, which are very important components to develop a complete, perfect and attractive game. Groove Unlimited Game integrates all those components leads to high performance and well-responsive to the players need.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลุล่วงได้ คณะผู้จัดทำขอกราบขอบพระคุณ รศ. ธีรวัฒน์ ประกอบผล อาจารย์ที่ปรึกษาปัญหาพิเศษ ซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณ ผศ. ดร. จีรพร ศรีสวัสดิ์ ประธานกรรมการสอบปัญหาพิเศษ และ ดร. กรกช ประชุมรักษ์ กรรมการสอบปัญหาพิเศษ ที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของปัญหาพิเศษ รวมทั้งอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้ ซึ่งทำให้ทางคณะผู้จัดทำสามารถนำความรู้ที่ได้รับมาแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นจนปัญหาพิเศษนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณเจ้าหน้าที่ห้องคอมพิวเตอร์ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่านที่ให้ความช่วยเหลือด้านทรัพยากรคอมพิวเตอร์และระเบียบวินัยอันดีแก่คณะผู้จัดทำ รวมทั้งผู้มีพระคุณทุกท่านที่มิได้เอ่ยนามไว้ ณ ที่นี้ด้วย

นอกจากนี้คณะผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ที่ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนกระทั่งการทำปัญหาพิเศษครั้งนี้สำเร็จได้ด้วยดี รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคนที่ได้ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VIII
สารบัญภาพ.....IX

บทที่ 1 บทนำ
1.1 ที่มาของปัญหาพิเศษ.....1
1.2 วัตถุประสงค์ของการศึกษา.....1
1.3 ขอบเขตของปัญหา.....2
1.4 ขั้นตอนในการดำเนินงาน.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3
1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ.....3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง
2.1 Microsoft Visual C++ 6.0.....5
2.1.1 VC++ Developer Studio.....5
2.1.2 VC++ Runtime Libraries.....6
2.1.3 VC++ MFC and Template Libraries.....6
2.1.4 VC++ Build Tools.....6
2.1.5 ActiveX 7.....7
2.1.6 Data Access.....7
2.1.7 Enterprise Tools.....7
2.1.8 Graphics.....7
2.1.9 Tools.....7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2 การเขียนโปรแกรมด้วย Visual C++ บน Window 8

2.2.1 ฟังก์ชันและวิธีการสร้างฟังก์ชัน.....9

2.2.2 การเขียนโปรแกรมเชิงวัตถุ.....11

2.3 DirectX.....22

2.3.1 ประวัติความเป็นมาของ DirectX.....22

2.3.2 ความรู้เบื้องต้นเกี่ยวกับ DirectX.....23

2.3.3 DirectX คืออะไร.....24

2.3.4 API ของ DirectX.....25

2.4 CDX Game Engine 33

2.4.1 องค์ประกอบของ CDX.....34

2.4.2 ขั้นตอนการติดตั้ง CDX และการปรับแต่งVisual C++ ให้รองรับ CDX.....35

2.4.3 ขั้นตอนการสร้างเกมด้วย CDX.....46

2.4.4 ตัวอย่างโค้ดโปรแกรมที่สร้างขึ้นมาจากCDX Application Wizard.....49

2.5 FMOD Sound System.....56

2.5.1 วิธีการเรียกใช้ FMOD.....56

2.5.2 การเล่นไฟล์เพลงและไฟล์เสียงประเภทต่าง ๆ.....58

บทที่ 3 ขั้นตอนการดำเนินงานวิจัย

3.1 ขั้นตอนการวิเคราะห์และออกแบบเกม.....62

3.1.1 การออกแบบรูปแบบ กติกาการเล่น และระบบเกม.....62

3.1.2 การออกแบบภาพและกราฟิก.....63

3.1.3 การออกแบบเสียงและซาวน์เอฟเฟกต์ประกอบ.....63

3.1.4 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น.....63

3.1.5 การออกแบบการจัดเก็บข้อมูล.....65

3.1.6 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้.....65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 ขั้นตอนการลงมือปฏิบัติจริง.....65
3.2.1 สร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม.....65
3.2.2 สร้างเสียงเพลงและซาวน์เอฟเฟกต์ประกอบ.....73
3.3 ขั้นตอนการเขียนโปรแกรม.....82
3.3.1 คลาส GUBackground.....85
3.3.2 คลาส GUSprite.....86
3.3.3 หน้าจอ Main.....88
3.3.4 หน้าจอ Sound Select.....92
3.3.5 หน้าจอ Training.....98
3.3.6 หน้าจอ Import Song.....99
3.3.7 หน้าจอ Option.....101
3.3.8 หน้าจอ Key Config.....103
3.3.10 หน้าจอ Sound Test.....109
3.3.11 หน้าจอ High Score.....111
3.3.12 หน้าจอ Loading.....115
3.3.13 หน้าจอ Game Play.....116
3.3.14 หน้าจอ Result.....126
3.4 ขั้นตอนการทดสอบเกม.....128
บทที่ 4 ผลการทดลองและวิเคราะห์ปัญหา
4.1 ขั้นตอนการดำเนินการทดสอบการติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็นต้องใช้.....130
4.2 ขั้นตอนการดำเนินการทดสอบการประมวลผลภายใต้ระบบที่กำหนด.....130
4.3 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกม.....131
4.3.1 หน้าจอการเลือกเพลง.....131
4.3.2 หน้าจอการเล่นเกม.....132
4.3.3 หน้าจอสรุปผลคะแนน.....133

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.3.4 หน้าจอสอนการเล่นเกม.....133
4.3.5 หน้าจอนำเพลงเข้าสู่เกม.....134
4.3.6 หน้าจอตั้งค่าตัวเลือก.....134
4.3.7 หน้าจอการปรับตั้งค่าคีย์บอร์ด.....135
4.3.8 หน้าจอทดสอบฟังเพลง.....135
4.3.9 หน้าจอคะแนนสูงสุด 10 อันดับ.....136
4.4 ผลการทดลองและการวิเคราะห์ปัญหา.....137
4.5 ประเมินประสิทธิภาพของเกม.....137
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ
5.1 สรุปผลการดำเนินงาน.....138
5.1.2 การวิเคราะห์และการออกแบบเกม.....138
5.1.3 การสร้างตัวละคร ภาพ และเสียงต่าง ๆ.....138
5.1.4 การพัฒนาโปรแกรม.....139
5.2 ข้อจำกัดของโปรแกรม.....139
5.3 ข้อเสนอแนะ.....140
บรรณานุกรม.....141
ภาคผนวก
ภาคผนวก ก. การติดตั้งและใช้งานโปรแกรม.....142
ภาคผนวก ข. กฎและกติกาการเล่น.....150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงการวางแผนงาน 4
3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ของเกมกรู๊ฟ อันลิมิเต็ด.....64
4.1 ขั้นตอนการดำเนินการทดสอบการติดตั้งโปรแกรมและคอมโพเนนท์ที่จำเป็น.....130
4.2 ขั้นตอนการดำเนินการทดสอบการประมวลภายใต้ระบบที่กำหนด.....130
4.3 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการเลือกเพลง.....131
4.4 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการเล่นเกม.....132
4.5 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการสรุปผลคะแนน.....133
4.6 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการสอนการเล่นเกม.....133
4.7 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการนำเพลงเข้าสู่เกม.....134
4.8 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการตั้งค่าตัวเลือก.....134
4.9 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในการปรับตั้งค่าคีย์บอร์ด.....135
4.10 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการทดสอบฟังเพลง.....135
4.11 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของคะแนนสูงสุด 10 อันดับ..136

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงรูปคลาสดินสอ.....12
2.2 แสดงข้อมูลของคน 5 คน..... 13
2.3 แสดงการเก็บข้อมูลของคน 5 คนแบบอาเรย์.....14
2.4 แสดงการเก็บข้อมูลของคน 5 คนแบบโครงสร้าง.....15
2.5 แสดงการเก็บข้อมูลตัวแปร person.....16
2.6 แสดงสถาปัตยกรรมทั่วไปของ DirectDraw.....25
2.7 แสดงการทำ Flipping ระหว่าง 2 buffer.....27
2.8 แสดงการทำ Flipping ระหว่าง 3 buffer.....28
2.9 แสดง Source Surface และ Destination Surface ที่การกำหนด Color Keys ก่อนการ Blitting.....29
2.10 (ซ้าย) Surface ที่ได้หลังการ Blitting โดยใช้ Source Color Key อย่างเดียว
(ขวา) Surface ที่ได้หลังการ Blitting โดยใช้ Source Color Key และ Destination Color Key.....29
2.11 แสดงสถาปัตยกรรมของ DirectSound.....31
2.12 CDX Source Folder.....34
2.13 CDX Setup Window.....34
2.14 การปรับแต่ง C++.....35
2.15 หน้าต่าง Options แถบ Directoried.....36
2.16 Open Dictionary สำหรับ Include Files.....36
2.17 เลือกไปยัง Directory ของ Include Files ของ CDX.....37
2.18 เลือกไปยัง Directory ของ Include Files ของ DirectX SDK.....37
2.19 หลังการเพิ่ม Directory Include Files.....38
2.20 หลังการเพิ่ม Directory Include Files.....38
2.21 สลับไปเลือกที่ Library Files.....39
2.22 การ new Project ในครั้งแรกที่ไม่มี CDX Application Wizard.....40
2.23 คัดลอกไฟล์ CDXAppWiz.awx.....40
2.24 วางไว้ที่โฟลเดอร์ปลายทาง.....41
2.25 การ new Project ที่มี CDXAPplication Wizard.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

2.26 โฟลเดอร์ของ CDX ที่ได้ติดตั้งไว้ภายในเครื่อง ซึ่งไม่มีไฟล์ Library ของโหลด Debug.....42
2.27 เปิด Workspace ของ CDX (src\cdx.dsw).....43
2.28 การเลือกเมนู Batch Build.....43
2.29 เลือกเฉพาะ 2 หัวข้อแรก.....44
2.30 โฟลเดอต์ vc ที่ได้จากการ compile ใหม่.....44
2.31 เข้าไปยังซับโฟลเดอร์ vc\intel แล้วทำการคัดลอกไฟล์ cdxd.lib มาวางไว้ที่โฟลเดอร์ lib.....45
2.32 โฟลเดอร์ lib หลังจากการคัดลอกไฟล์ cdxd.lib เรียบร้อยแล้ว.....45
2.33 แสดงขั้นตอนการสร้างเกมด้วย CDX.....46
2.34 แสดงการกำหนดค่าต่าง ๆ ให้กับเกม.....46
2.35 แสดงขั้นตอนการตั้งชื่อเกม, กำหนดขนาดหน้าต่างของเกม และการแสดงเมาส์.....47
2.36 แสดงขั้นตอนการตั้งค่าอื่น ๆ ของเกม.....48
2.37 แสดงหน้าต่างข้อมูลของโปรเจกต์.....48
2.38 แสดงไดเรกทอรีของ fmod.....56
2.38 แสดงการ include file fmod.h และ fmodvc.lib.....57
2.40 แสดงการคัดลอกไฟล์ fmod.dll.....58
2.41 แสดงคำเตือนเมื่อโปรแกรมไม่สามารถเข้าถึง fmod.dll ได้.....58
3.1 แสดงภาพเมาส์.....65
3.2 รูปภาพสำหรับรอการโหลด.....66
3.3 รูปฉากหลังของหน้าจอ Main.....66
3.4 รูปปุ่มต่าง ๆ ของหน้าจอ Main.....67
3.5 รูปฉากหลังของหน้าจอ Game Play.....67
3.6 รูปปุ่มและตัวละครต่าง ๆ ของหน้าจอ Game Play.....68
3.7 รูปฉากหลังของหน้าจอ Import Song.....69
3.8 รูปปุ่มต่าง ๆ ของหน้าจอ Import Song.....69
3.9 รูปฉากหลังของหน้าจอ Option.....69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

3.10 รูปปุ่มต่าง ๆ ของหน้าจอ Option.....70
3.11 รูปฉากหลังของหน้าจอก Key Config.....70
3.12 รูปกรอบที่ใช้สำหรับแสดงจุดโฟกัสของหน้าจอ Key Config.....70
3.13 รูปฉากหลังของหน้าจอ Sound Test.....71
3.14 รูปปุ่มต่าง ๆ ของหน้าจอ Sound Test.....71
3.15 รูปฉากหลังของหน้าจอ High Score.....71
3.16 รูปฉากหลังของหน้าจอ Sound Select.....72
3.17 รูปปุ่มต่าง ๆ ของหน้าจอ Sound Select.....72
3.18 รูปฉากหลังของหน้าจอแสดงผลคะแนนเมื่อเกมจบ.....73
3.19 รูปกล่องข้อความสำหรับใส่ชื่อของผู้เล่น เมื่อได้คะแนนรวมสูงสุด 10 อันดับแรก.....73
3.20 แสดงหน้าต่างของโปรแกรม Fruity Loop Studio 5.....74
3.21 แสดงหน้าต่าง Browse ของโปรแกรม Fruity Loop.....74
3.22 แสดงหน้าต่าง Channels ของโปรแกรม Fruity Loop.....75
3.23 แสดงหน้าต่าง Channel Settings ของโปรแกรม Fruity Loop.....75
3.24 แสดงน้าต่าง Piano Roll ของโปรแกรม Fruity Loop.....76
3.25 แสดงหน้าต่าง Playlist ของโปรแกรม Fruity Loop.....76
3.26 แสดงหน้าต่าง Effect ของโปรแกรม Fruity Loop.....77
3.27 แสดงการเลือกเครื่องดนตรีที่ต้องการ.....77
3.28 แสดงการเลือก Channel เพื่อปรับแต่งเสียง.....78
3.29 แสดงการเพิ่ม แก้ไข และลบโน้ตโดยใช้หน้าต่าง Channel.....78
3.30 แสดงการเพิ่ม แก้ไข และลบโน้ตโดยใช้หน้าต่าง Piano Roll.....79
3.31 แสดงการนำแต่ละ Pattern มาต่อกันเป็นเพลง โดยใช้หน้าต่าง Playlist 80
3.32 แสดงการ Export เพลง.....80
3.33 แสดงหน้าต่างของโปรแกรม Adobe Audition 1.5.....81
3.34 แสดงการเลือกส่วนของเสียงโดยใช้เมาส์ลาก.....81
3.35 แสดงการตัดต่อไฟล์เสียง.....82
3.36 แสดงการเปลี่ยนจากหน้าจอหนึ่งไปยังหน้าจออื่น ๆ.....83
3.37 แสดงโครงสร้างของคลาส GUBackground.....84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

3.38 แสดงโครงสร้างของคลาส GUSprite.....86
3.39 แสดงหน้าจอ Main.....88
3.40 แผนผังการทำงานในหน้าจอ Main.....89
3.41 แสดงหน้าจอ Sound Select.....92
3.42 แผนผังการทำงานในหน้าจอ Sound Select.....93
3.43 แสดงโครงสร้างของตัวแปร songlist.....94
3.44 แสดงหน้าจอ Training.....98
3.45 แผนผังการทำงานในหน้าจอ Training.....98
3.46 รูปของหน้าจอ Import Song.....99
3.47 แผนผังการทำงานในหน้าจอ Import Song.....100
3.48 แสดงหน้าจอ Option.....101
3.49 แผนผังการทำงานในหน้าจอ Option.....102
3.50 แสดงหน้าจอ Key Config.....103
3.51 แผนผังการทำงานในหน้าจอ Key Config.....104
3.52 แสดงโครงสร้างของตัวแปร keys.....105
3.53 แสดงหน้าจอ Sound Test.....109
3.54 แผนผังการทำงานในหน้าจอ Sound Test.....110
3.55 แสดงหน้าจอ High Score.....111
3.56 แผนผังการทำงานในหน้าจอ High Score.....112
3.57 แสดงโครงสร้างของตัวแปร hiscore.....113
3.58 แสดงหน้าจอ Loading.....115
3.59 แสดงหน้าจอ Game Play.....116
3.60 แสดงโครงสร้างของคลาส Score.....117
3.61 แสดงโครงสร้างของคลาส SpiteBar.....119
3.62 หน้าจอของการสรุปผลคะแนน.....126
3.63 Dialog Box ให้ใส่ชื่อของผู้เล่นที่ได้คะแนนสูงสุดติดอันดับ 1 ใน 10.....127
3.64 หน้าจอสรุปผู้ทำคะแนนสูงสุด 10 อันดับ.....127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

ก-1 หน้าจอแรกของการ Setup.....142
ก-2 หน้าจอแสดงข้อตกลงใบอนุญาต.....143
ก-3 หน้าจอพร้อมที่จะติดตั้งโปรแกรม.....144
ก-4 หน้าจอขณะกำลังติดตั้งเกมกรู๊ฟ อันลิมิเต็ด.....145

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของปัญหาพิเศษ

เนื่องด้วยปัจจุบัน เกมได้เข้ามามีบทบาทต่อเด็ก ๆ หรือบุคคลที่เล่นอย่างมากโดยเฉพาะอย่างยิ่งเกมคอมพิวเตอร์ ซึ่งข้อดีของเกมคอมพิวเตอร์ต่อเด็ก หรือบุคคลที่เล่นนั้นก็มีอยู่ไม่น้อยหากจะเลือกเล่น และใช้เวลาในการเล่นพอสมควรโดยเกม Groove Unlimited เป็นเกมที่ช่วยฝึกฝนทักษะทางประสาทสัมผัสและทางดนตรีให้กับผู้เล่น เพราะผู้เล่นต้องใช้ความสามารถทั้งทางตา หู และการควบคุมนิ้วมือในการกดปุ่มคีย์บอร์ดให้สอดคล้องกับจังหวะของเพลงในเกม ผู้เล่นที่ไม่สามารถเล่นดนตรีจริง ๆ ได้ ก็สามารถสนุกสนานเพลิดเพลินและพัฒนาความคิดสร้างสรรค์ไปกับเกม นอกจากนี้ยังสามารถตอบสนองความต้องการของผู้เล่นได้ดี โดยผู้เล่นสามารถนำเพลงที่ต้องการหรือชื่นชอบซึ่งในเกมไม่ได้จัดเตรียมไว้ให้เข้าสู่โปรแกรมเกมได้

ในการพัฒนาโปรแกรมเกมต้องใช้ความรู้ความสามารถหลายด้าน ทั้งทางด้านการออกแบบรูปแบบการเล่นและส่วนติดต่อกับผู้เล่น ซึ่งต้องมีรูปแบบที่น่าสนใจ และด้านการพัฒนาโปรแกรมเกมมัลติมีเดียซึ่งต้องมีความรู้ทั้งในด้าน การติดต่อกับฮาร์ดแวร์ กราฟิกแอนิเมชัน และโดยเฉพาะอย่างยิ่งทางด้านเสียง

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อพัฒนาเกมคอมพิวเตอร์ที่สามารถสร้างสรรค์เกมได้โดยการนำเข้าเพลงของตนเองสู่เกม

1.2.2 เพื่อฝึกฝนทักษะและความสามารถในการพัฒนาโปรแกรมเกมและมัลติมีเดีย

1.2.3 เพื่อศึกษาและพัฒนาโปรแกรมเกมโดยใช้ภาษา Visual C++ 6.0

1.2.4 เพื่อศึกษาและพัฒนาโปรแกรมที่มีการติดต่อกับ DirectX

1.2.5 เพื่อศึกษาวิธีการออกแบบและตกแต่งภาพที่นำมาประกอบเกมด้วย Adobe Photoshop CS และ Adobe Illustrator CS

1.2.6 เพื่อให้ผู้เล่นได้รับความสนุกสนานพร้อมไปกับการฝึกฝนทักษะทางด้านประสาทสัมผัสและทางด้านดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปัญหา

เกมจะกำหนดปุ่มอินพุตคีย์บอร์ดว่าปุ่มไหนจะเกิดการกระทำอะไร โดยผู้เล่นเป็นผู้เลือกเพลงที่ต้องการเล่นขึ้นมา จากนั้นผู้เล่นต้องกดปุ่มบนคีย์บอร์ด เพื่อให้ตรงตามจังหวะหรือโน้ตของเพลงที่ผู้เล่นเลือกและกำลังเล่นอยู่

การพัฒนาโปรแกรมเกมและมัลติมีเดียซึ่งมีจำนวนผู้เล่น 1 - 2 คน โต้ตอบกับเกมโดยผ่านทางคีย์บอร์ดและเมาส์ หน้าจอถูกแบ่งออกเป็นสองฝั่งสำหรับผู้เล่นแต่ละคน ผู้เล่นสามารถเลือกเล่นเพลงที่มีให้แล้วในเกมหรือนำเข้าเพลงที่ต้องการเล่นซึ่งอยู่ในรูปของไฟล์เสียงแบบ .WAV .MP3 หรือ .OGG และสามารถบันทึกไฟล์เพลงนั้นเพื่อเก็บไว้เล่นครั้งต่อไปได้

1.4 ขั้นตอนในการดำเนินงาน

1.4.1 ขั้นตอนการวิเคราะห์และออกแบบเกม

1.4.1.1 การวิเคราะห์ความต้องการ เป็นการออกแบบรูปแบบของเกม ออกแบบกติกาการเล่น ระบบของเกม และลักษณะแนวทางการเล่น รูปแบบตัวละคร ฉาก เสียง และเนื้อหาของเกม

1.4.1.2 ออกแบบหน้าจอส่วนที่ใช้ติดต่อกับผู้เล่น เป็นรูปแบบที่ใช้ง่าย และดึงดูดความสนใจของผู้เล่น

1.4.1.3 ออกแบบวิธีการจัดเก็บข้อมูล โดยที่ต้องใช้พื้นที่ในการจัดเก็บน้อย เป็นระเบียบและสามารถนำมาใช้ได้ง่ายเพื่อเพิ่มประสิทธิภาพการทำงานให้กับเกม

1.4.1.4 ออกแบบวิธีการเขียนโปรแกรม และเลือกใช้ฮาร์ดแวร์, ซอฟต์แวร์ รวมถึงคอมโพเนนต์และเครื่องมือต่าง ๆ ที่นำมาใช้ในการสร้างเกม

1.4.2 ขั้นตอนการลงมือปฏิบัติการจริง

1.4.2.1 ด้านภาพและกราฟิก การทำงานในส่วนนี้จะใช้โปรแกรมทางด้านกราฟิกเข้ามาช่วยในการสร้าง ฉากที่ได้ทำการออกแบบไว้แล้ว

1.4.2.2 ด้านซาวน์เอฟเฟกต์ การทำงานในส่วนนี้จะใช้โปรแกรมการสร้างเสียงต่าง ๆ มาช่วยในการสร้างเสียงประกอบ

1.4.2.3 ด้านโปรแกรมมิ่ง การทำงานในส่วนนี้จะเกี่ยวข้องการเขียนคำสั่งต่าง ๆ โดยมีการนำ ฉาก ตัวละคร และเสียงประกอบที่สร้างไว้มาใช้ ซึ่งมีขั้นตอนการทำงานดังนี้

1. ใช้โปรแกรม Visual C++ ในส่วนของการเขียนและแก้ไขซอร์สโค้ดของเกม
2. ใช้ CDX Game Developer Kit ช่วยในการเขียนโปรแกรม
3. นำฉาก ตัวละคร และเสียง มาประกอบเข้าด้วยกัน
4. เขียนคำสั่งที่ทำให้เกิดการกระทำตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.3 ขั้นตอนการทดสอบเกม การทำงานในส่วนนี้จะเป็นขั้นตอนในการทดสอบเกมที่เราผลิตขึ้นว่ามีความสมบูรณ์มากน้อยเพียงใด เพื่อหาข้อผิดพลาดและประเมินประสิทธิภาพของเกม เมื่อพบข้อผิดพลาดก็จะทำการแก้ไขโดยทันที แล้วจัดทำเอกสารการวิจัยสรุปผลการทดลอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 มีความรู้เกี่ยวกับการพัฒนาโปรแกรมโดยใช้ Visual Basic ร่วมกับ CDX Game Developer Kit

1.5.2 สามารถตกแต่งและสร้างภาพด้วย Adobe Photoshop CS และ Adobe Illustrator CSได้

1.5.3 สามารถแต่งเพลงและตัดต่อเพลงด้วย Fruity Loop Studio 5 และ Adobe Audition ได้

1.5.3 ได้โปรแกรมเกมที่สนุกสนานและตอบสนองความต้องการของผู้เล่นได้ดี

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1.6.1 เครื่องคอมพิวเตอร์ Pentium IV 2.56 MHz

1.6.2 หน่วยความจำ 256 MB

1.6.3 Hard disk 40 GB และ Removable rack

1.6.4 Microsoft Visual C++

1.6.5 CDX Game Developer Kit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 แสดงการวางแผนงาน

ขั้นตอนการทำงาน	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1) ขั้นตอนการวิเคราะห์และออกแบบเกม										
● การวิเคราะห์ความต้องการของรูปแบบในการสร้างเกม										
● ออกแบบกติกาการเล่น										
● ออกแบบหน้าจอส่วนที่ใช้ติดต่อกับผู้เล่น										
● ออกแบบวิธีการจัดเก็บข้อมูลที่จำเป็นต้องใช้ภายในเกม										
● ออกแบบวิธีการเขียนโปรแกรม และเลือกใช้ฮาร์ดแวร์, ซอฟต์แวร์ ร่วม ถึง component และ tool ต่าง ๆ ที่นำมาใช้ในการสร้างเกม										
2) ขั้นตอนการลงมือปฏิบัติการจริง										
● ด้านภาพและกราฟิก										
● ด้านซาวน์เอฟเฟกต์										
● ด้านโปรแกรมมิ่ง										
3) ขั้นตอนการทดสอบเกม										
● ทดสอบความสมบูรณ์ของเกม										
● หาข้อผิดพลาด										
● ประเมินประสิทธิภาพของเกม										
● ปรับปรุงแก้ไขข้อผิดพลาด										
● สรุปผล										

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 2
ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

2.1 Microsoft Visual C++ 6.0

Microsoft Visual C++ 6.0 เป็นภาษาการโปรแกรมมิ่งเชิงวัตถุ (Object – Oriented Programming) แบบ GUI (Graphic User Interface) ตัวหนึ่งจากบริษัทไมโครซอฟต์ เป็นเครื่องมือพัฒนาโปรแกรมที่มีความสามารถสูงในยุคนี้ Microsoft Visual C++ ได้รับกาพัฒนาให้มีความยืดหยุ่น และมีประสิทธิภาพสูงขึ้นมาจากภาษา C++ และได้สนับสนุนการพัฒนาโปรแกรมในหลาย ๆ ด้านไม่ว่าจะเป็นการสร้างโปรแกรมทั่วไป, การสร้างโปรแกรมการจัดการฐานข้อมูล, การสร้างโปรแกรมบนระบบเครือข่าย หรือมัลติมีเดีย อย่างครบครัน

ในปัจจุบัน Microsoft Visual C++ ได้รับการพัฒนาจนถึงเวอร์ชันที่ 6 มีลักษณะเป็น IDE (Integrated Development Environment) คือเป็นโปรแกรมซึ่งมีไว้ใช้สำหรับเพิ่มความสะดวกในการสร้าง และแก้ไขโปรเจ็กต์ให้ง่ายขึ้น โดยจะมีเครื่องมืออำนวยความสะดวกในการพัฒนาโปรแกรมต่าง ๆ ให้เรียกใช้งานใน IDE เช่น โปรแกรมที่ใช้ในการดีบัก, คอมไพเลอร์ และลิงค์เกอร์ เป็นต้น นอกจากนี้ยังมีส่วนรองรับการพัฒนาโปรแกรมบนวินโดว์ โดยมี MFC (Microsoft Foundation Class) เป็นไลบรารีที่จะช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมบนวินโดว์

นอกจากนี้ Microsoft Visual C++ ยังมีส่วนที่สนับสนุนทางด้านของมัลติมีเดีย เกี่ยวกับรูปภาพ, การเล่นไฟล์ภาพเคลื่อนไหวรวมถึงทางด้านของเสียง (Sound) ทั้งยังมีส่วนของเครื่องมือที่สนับสนุนการใช้งานที่เกี่ยวข้องกับ Buffer I/O, Riff files ที่เรียกกันว่า Multimedia file I/O services อีกด้วย โดยมีความสามารถ และเครื่องมือต่าง ๆ ให้ใช้งานมากมาย (Component) ซึ่งใน Visual C++ นั้นประกอบไปด้วยคอมโพเนนต์ต่าง ๆ ดังนี้

2.1.1 VC++ Developer Studio

ทำหน้าที่เป็นศูนย์กลางในการสร้างแอพพลิเคชั่น ซึ่งจะประกอบไปด้วยองค์ประกอบย่อย ๆ ภายในดังนี้

- Project Manager ทำหน้าที่จัดการกับการสร้างแอพพลิเคชั่นในลักษณะของโปรเจ็กต์ (แอพพลิเคชั่นหนึ่ง ๆ ประกอบด้วยองค์ประกอบหลาย ๆ ส่วนซึ่งเรียกรวมว่าโปรเจ็กต์)
- Text Editor ทำหน้าที่ในการเขียนโปรแกรม โดยจะเรียกโปรแกรมที่เขียนว่า Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Resource Editor ทำหน้าที่อกแบบหน้าตาโปรแกรม เช่น เมนู ไดอะล็อกต่าง ๆ และไอคอน
- Wizard ต่างๆ เพื่อสร้างแอพพลิเคชั่นในลักษณะวิซาร์ดต่าง ๆ เช่น AppWizard และ ClassWizard จะช่วยให้สร้างโค้ดพื้นฐานแก่แอพพลิเคชั่นอย่างรวดเร็ว โดยเพียงแต่กำหนดคลาส C++ จัดการกับวินโดว์แมสเสจ และกระทำงานอื่น ๆ เพิ่มเติมอีก
- Compiler จะทำการคอมไพล์อย่างอัตโนมัติ มีการลิงค์ (Link) ไฟล์ต่าง ๆ เข้าด้วยกัน
- Debugger ดีบัก เพื่อแก้ไขความผิดพลาดของแอพพลิเคชั่นโดยผ่าน Debugger
- Online Help รายละเอียดความช่วยเหลือ (กรณีติดตั้ง MSDN แล้ว)

2.1.2 VC++ Runtime Libraries

คอมโพเนนต์นี้ทำหน้าที่เก็บฟังก์ชันมาตรฐานต่าง ๆ ของ ANSI C เช่น ฟังก์ชัน sin ซึ่งสามารถเรียกใช้ภายในโปรแกรม C หรือ C++

2.1.3 VC++ MFC and Template Libraries

MFC เป็นไลบรารี่คลาส C++ ที่ถูกสร้างมาโดยเฉพาะสำหรับการสร้างแอพพลิเคชั่นเพื่อใช้งานกับ Windows ซึ่งมีรูปแบบการติดต่อกับผู้ใช้แบบกราฟิก หรือที่เรียกว่า GUI (Graphic User Interface) ซึ่งคลาส MFC ทำให้การเขียนโปรแกรมง่ายขึ้น และยังช่วยประหยัดเวลาในการเขียนโค้ด

นอกจากนี้ยังติดตั้ง ATL (Active Template Libraries) ซึ่งเป็นชุดของคลาส C++ ที่เป็นเทมเพลตหรือต้นแบบ ที่จะช่วยอำนวยความสะดวกในการสร้าง ActiveX Control และรวมทั้งรูปแบบการสร้างแอพพลิเคชั่นอื่นๆ ที่เป็นวัตถุ COM (Component Object Model)

2.1.4 VC++ Build Tools

ประกอบด้วยคอมไพเลอร์ C/C++, Linker, คอมไพเลอร์ Resource (สำหรับเตรียมการเกี่ยวกับ Resource โปรแกรมต่าง ๆ เช่น เมนู ไดอะล็อกต่าง ๆ และไอคอน เป็นต้น) รวมทั้งเครื่องมืออื่น ๆ ที่จำเป็นต่อการสร้างแอพพลิเคชั่นสำหรับวินโดว์ (การสร้างแอพพลิเคชั่นแบบ Win32)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ActiveX

เป็นซอฟต์แวร์ย่อย ๆ (Software Component) หรือองค์ประกอบย่อย ๆ ที่เพิ่มเข้าไปในแอพพลิเคชั่นที่สร้างขึ้น ซึ่ง ActiveX นั้นทำให้ไม่จำเป็นต้องสร้างทุกส่วนของแอพพลิเคชั่นเอง เพียงแต่เลือกใช้องค์ประกอบย่อย ๆ ที่เหมาะสมกับงานเพื่อสร้างเป็นแอพพลิเคชั่นที่สมบูรณ์

2.1.6 Data Access

เป็นคอมโพเนนต์ที่รวมไดรเวอร์ฐานข้อมูลชนิดต่าง ๆ ไว้ให้สามารถสร้างแอพพลิเคชั่นที่เชื่อมต่อกับฐานข้อมูลได้อย่างสะดวก และนอกจากไดรเวอร์แล้วยังประกอบไปด้วยคอนโทรล และเครื่องมืออื่น ๆ ที่ช่วยในการสร้างแอพพลิเคชั่นกับฐานข้อมูลทำได้อย่างรวดเร็ว

2.1.7 Enterprise Tools

เป็นคอมโพเนนต์ที่ประกอบด้วยคอมโพเนนต์ย่อย ๆ อีกดังต่อไปนี้

- Microsoft Visual SourceSafe 6.0 Client
- Application Performance Explorer
- Repository
- Visual Component Manager
- Self-installing .exe redistributable files
- Visual Basic Enterprise Components
- VC++ Enterprise Tools
- Microsoft Visual Modeler
- Visual Studio Analyzer

2.1.8 Graphics

เป็นคอมโพเนนต์เกี่ยวกับรูปภาพในรูปแบบต่าง ๆ เช่น metafile, bitmap, cursor และ icon รวมทั้ง video clip ซึ่งคอมโพเนนต์เหล่านี้ใช้ในการเขียนโปรแกรมเกี่ยวกับกราฟิกต่าง ๆ

2.1.9 Tools

เป็นคอมโพเนนต์ที่เป็นเครื่องมือเสริมการทำงานชนิดต่าง ๆ ของ Visual C++ ดังนี้

- API Text Viewer
- MS Info
- MFC Trace Utility

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Spy++
- Win32 SDK Tools
- OLE/COM Object Viewer
- ActiveX Control Test Container

2.2 การเขียนโปรแกรมด้วย Visual C++ บน Window

ภาษา C++ ซึ่งใช้โครงสร้างภาษาเหมือนกับภาษา C ทุกอย่าง แต่ได้เพิ่มความรัดกุมมากขึ้นในเรื่องของการใช้คำสั่งและฟังก์ชัน พร้อมกับได้เพิ่มความสามารถในการประกาศคลาส ซึ่งเป็นตัวแปรในแบบการเขียนโปรแกรมเชิงวัตถุ และหลังจากที่ภาษา C++ และแนวคิดของการเขียนโปรแกรมเชิงวัตถุ เป็นที่ยอมรับ และได้นำไปใช้ในการพัฒนาโปรแกรมต่าง ๆ มากมาย ไม่ว่าจะเป็นตัวระบบปฏิบัติการหรือโปรแกรมประยุกต์ ทั้งบนวินโดวส์และบนระบบปฏิบัติการอื่น ๆ เช่น ลีนุกซ์ (Linux) และก็ยังมีการพัฒนาตัวแปลภาษาให้มีความสามารถสูงขึ้น ๆ ตามความสามารถของระบบปฏิบัติการ เช่น Visual C++ , Visual Studio , Visual Studio .NET เป็นต้น แต่อย่างไรก็ตาม พื้นฐานของภาษา C และการเขียนโปรแกรมแบบการเขียนโปรแกรมเชิงวัตถุ ใน C++ ก็ยังเป็นรากฐานในการพัฒนาภาษาใหม่ ๆ ขึ้นมาอีกหลายภาษา เช่น จาวา (Java) , ซีชาร์ป (C#) เป็นต้น

การที่เราจะพัฒนาโปรแกรมภาษา C++ ได้นั้น จะต้องมีโปรแกรมสำหรับแปลภาษาก่อน นั่นก็คือ Turbo C++, Borland C++ , Microsoft C/C++ , Visual C++ ฯลฯ โปรแกรมแปลภาษาเหล่านี้สามารถแปลโปรแกรมได้ทั้งภาษา C และภาษา C++ ซึ่งก็ขึ้นอยู่กับว่าเรากำหนดให้นามสกุลของไฟล์เป็นภาษาอะไร ถ้าเขียนโปรแกรมภาษา C ให้กำหนดนามสกุลของไฟล์โปรแกรมเป็น .C แต่ถ้าเขียนโปรแกรมที่มีการประกาศคลาสหรือใช้ความสามารถของการเขียนโปรแกรมเชิงวัตถุ ด้วย ให้กำหนดนามสกุลของไฟล์โปรแกรมนั้นให้เป็น .CPP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ฟังก์ชันและวิธีการสร้างฟังก์ชัน

- **การประกาศฟังก์ชัน**

ฟังก์ชันต้นแบบ หรือ Prototype Function นี้ มีไว้เพื่อบอกให้ตัวแปลภาษารู้ว่า ในโปรแกรมนี้ ยังมีฟังก์ชันชื่อว่า call( ) อยู่ แต่อยู่ด้านล่างของ main( ) ดังนั้น เราเขียนโปรแกรมให้เป็นดังนี้

```
void call( );

void main( )
{
    call( );
}

void call( )
{
    //
}
```

หรืออีกวิธีหนึ่งก็คือ ให้ประกาศฟังก์ชัน call( ) เอาไว้ด้านบนก่อน main( ) ดังนี้

```
void call( )
{
    //
}

void main( )
{
    call( );
}
```

- **การคืนค่าของฟังก์ชัน**

เราสามารถกำหนดให้ฟังก์ชันส่งค่ากลับมายังจุดที่เรียกใช้ได้ โดยใช้คำสั่ง return แต่ก่อนที่จะใช้คำสั่ง return นั้นต้องตรวจสอบว่า จะส่งค่าประเภทใดกลับไป ถ้าเป็นค่าจำนวนเต็ม เราจะต้องเปลี่ยนคำว่า void ที่อยู่หน้าฟังก์ชันให้เป็น int ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int call( );

void main( )
{
        int a;
        a = call( );
}

int call( )
{
        return 5+6;
}
```

จากตัวอย่าง จะเห็นได้ว่าฟังก์ชัน call( ) เป็นฟังก์ชันแบบ int และให้คืนค่า 5+6 (ซึ่งก็คือ 11) กลับไปยังจุดที่เรียกใช้ ดังนั้น ถ้าเราไม่เปลี่ยนจาก void ให้มาเป็น int เราก็จะไม่สามารถส่งค่ากลับไปได้

- **การส่งค่าไปให้ฟังก์ชัน**

เราสามารถส่งค่าให้กับฟังก์ชันได้ เพื่อให้ฟังก์ชันนำเอาค่าที่ส่งไปนี้ไปประมวลผล เราเรียกค่าที่ส่ง นี้ว่า "พารามิเตอร์" หรือจะเรียกว่า "อากิวเมนต์" ก็น่าจะได้เหมือนกัน ดังตัวอย่างต่อไปนี้

```
#include <stdio.h>

int call( int a , int b );
void main( )
{
        printf("answer = %d\\n", call( 10 , 20 ) );
}

int call( int a , int b )
{
        return a+b;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างจะเห็นได้ว่า ฟังก์ชัน call รับพารามิเตอร์ 2 ตัวคือ a กับ b เราได้เรียกโดยผ่านฟังก์ชัน printf ส่งค่า 10 และ 20 ไปให้กับ call( ) และมันก็จะคืนค่ากลับมาโดยเอาเลขทั้งสองบวกกัน เพราะฉะนั้น ฟังก์ชัน call( ) จึงคืนค่ากลับมาเป็น 30 แสดงออกทางจอภาพโดย printf

2.2.2 การเขียนโปรแกรมเชิงวัตถุ

- **ธรรมชาติของวัตถุ และทฤษฎีของการเขียนโปรแกรมเชิงวัตถุ**

นิยามในเรื่องของการเขียนโปรแกรมเชิงวัตถุ คือ "การมองทุกอย่างให้เป็นวัตถุ" เรามาทำความเข้าใจกับแนวคิดนี้กันดังนี้ ทุก ๆ สิ่งที่อยู่รอบ ๆ ตัวเราก็คือ "วัตถุ" ซึ่งแต่ละชิ้นแต่ละอย่างต่างก็จะมีชื่อเรียก เช่น

- โทรศัพท์ จะยี่ห้ออะไร แบบไหน เราเรียกวัตถุนี้ว่า โทรศัพท์
- แก้วน้ำ จะเป็นแก้วกระเบื้อง หรือแก้วพลาสติก วัตถุนี้ก็คือแก้ว
- กระดาษ จะเป็นเศษกระดาษ หรือ A4 วัตถุนี้ก็คือ กระดาษ
- คน ไม่ว่าจะเป็นอาชีพใด สูงต่ำดำขาว ก็คือคนด้วยกันทั้งหมด
- รถ ไม่ว่ายี่ห้อใด แบบใด ประเภทใด ก็เรียกว่ารถ เช่น รถยนต์ รถบรรทุก ฯลฯ

ดังนั้นสามารถสรุปได้ว่า ในโลกนี้ ทุก ๆ อย่างล้วนแล้วแต่มี "ชื่อที่บอกว่าสิ่งนั้นคืออะไร หรือเป็นอะไร" ชื่อที่เราสมมติขึ้นมาเพื่อใช้เรียกวัตถุต่าง ๆ นั้นก็คือ "คลาส" (Class) นั่นเอง

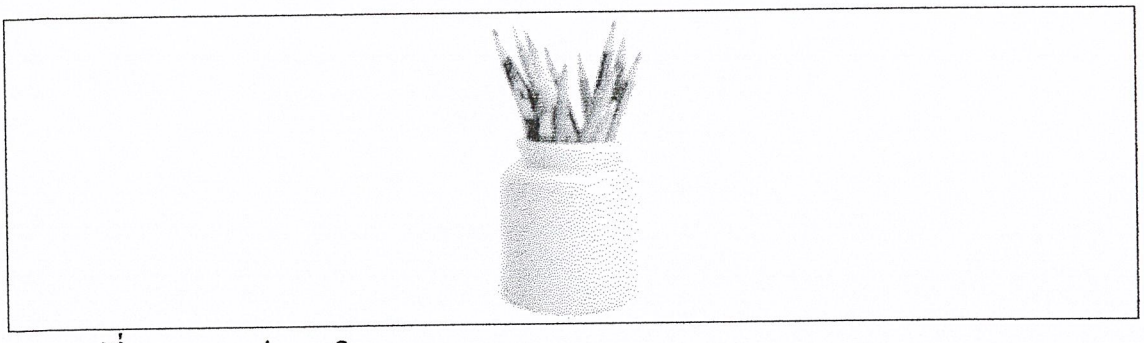
ถ้าเรามองสิ่งของรอบ ๆ ตัว ก็มีหลายคลาส เช่น

- คลาสของโทรศัพท์
- คลาสของดินสอ
- คลาสของยางลบ
- คลาสของคอมพิวเตอร์
- คลาสของเมาส์
- คลาสของพัดลม

สาเหตุที่มีคลาส หรือมีชื่อที่เรียกวัตถุตามลักษณะของวัตถุนั้นเพราะว่า ถ้าไม่มีชื่อที่ใช้เรียก ในเวลาเราจะอ้างถึงสิ่ง ๆ นั้นก็จะทำได้ยาก

พิจารณารูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงรูปคลาสดินสอ

จากรูป จะเห็นได้ว่าเป็นวัตถุที่มีรูปร่างยาว ๆ เรียว ๆ มีไส้เป็นถ่าน ปลายด้านหนึ่งจะแหลม ๆ ใช้เขียนบนวัสดุรองรับ เช่น กระดาษได้ วัตถุชิ้นนี้เรียกว่า "ดินสอ" และในคลาสของวัตถุใด ๆ จะมีอยู่ 2 องค์ประกอบที่ใช้บอกถึงความเป็นวัตถุนั้น คือ

1. คุณลักษณะ (Attribute หรือ Properties) ก็คือ ลักษณะที่บอกถึงความเป็นสิ่ง ๆ นั้น

2. การกระทำ (Method หรือ Functions) ก็คือ มันทำอะไรได้บ้าง

ยกตัวอย่างเช่น ดินสอ คุณลักษณะของมันก็คือ เรียว, ยาว, มีไส้เป็นถ่าน การกระทำของดินสอก็คือ เขียน, หัก, ปา, จิ้ม, แทง พูดง่าย ๆ ก็คือ ดินสอทำอะไรได้ และเราทำอะไรกับมันได้บ้าง เราสามารถใช้มันทำอะไรได้ตั้งแต่เขียน, ทิ่มแทง ไปจนถึงใช้ทัดหู ก็ขึ้นอยู่กับการนำไปใช้งานนั่นเอง เพราะฉะนั้น ถ้าเราไปพบวัตถุใด ที่มีคุณลักษณะตรงกันคุณลักษณะของดินสอ และยังมีการกระทำที่เหมือนกัน จะสรุปได้เลยว่า วัตถุสิ่งนั้นก็คือ "ดินสอ"

เราลองมาพิจารณารถยนต์ คุณลักษณะของรถยนต์ก็คือ

1. มีล้อ 4 ล้อ
2. มีที่นั่ง อย่างน้อย 1-2 ที่นั่ง
3. มีพวงมาลัย
4. มีเกียร์

สำหรับการกระทำ (Method) ของรถยนต์ก็คือ ขับเดินหน้าถอยหลัง เลี้ยวซ้าย เลี้ยวขวาได้ หยุดได้นั่นเอง จะเห็นได้ว่าทุก ๆ อย่างล้วนแล้วแต่ก็มีคุณลักษณะและการกระทำด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การเขียนโปรแกรมเชิงวัตถุกับการออกแบบโปรแกรม**

ในการเขียนโปรแกรมคอมพิวเตอร์ก็เช่นกัน เราจะต้องมองสิ่งที่อยู่ในโปรแกรมของเราให้เป็นเชิงวัตถุเพื่อที่จะได้ออกแบบวิธีการเก็บและจัดการข้อมูลที่เป็นอิสระต่อกัน แต่ก่อนที่จะเข้าสู่การเขียนโปรแกรมเชิงวัตถุ ในภาษา C++ นั้น คล้ายกับตัวแปรแบบโครงสร้าง (Structure) ในภาษา C การประกาศตัวแปรแบบโครงสร้างนี้ เป็นแนวคิดพื้นฐานที่จะนำมาใช้ในการสร้างคลาสต่อไป ลักษณะการทำงานของตัวแปรแบบโครงสร้างจะช่วยแก้ไขปัญหาในเรื่องของความซับซ้อนของตัวแปรได้ เพราะเป็นการเปลี่ยนวิธีการเขียนโปรแกรมโดยอาศัยชุดของข้อมูลเป็นหลัก จากรูปเป็นยกตัวอย่างปัญหาในการเก็บข้อมูลของคน 5 คน ดังรูป

รูปที่ 2.2 แสดงข้อมูลของคน 5 คน

สมมติว่าเราจะต้องเขียนโปรแกรมให้เก็บข้อมูลของพวกเขาเหล่านั้นเอาไว้ ถ้าเราใช้การเขียนโปรแกรมภาษา C ธรรมดา สามารถทำได้โดยใช้อะเรย์แบบ 1 มิติ โดยการประกาศตัวแปรตามสิ่งที่เราต้องการจะเก็บ คือ อายุ, ส่วนสูง, น้ำหนัก จำนวน 5 คนดังนี้

```
int age[5];
int height[5];
int weight[5];
```

ถ้าเราจะนำเอาข้อมูลมาใส่ลงในตัวแปร age , height และ weight นั้น เราก็จะอาศัยลำดับในอะเรย์เข้ามาช่วย นั่นคือ คนที่ 1 ก็จะเก็บลงไปในอะเรย์ช่องที่ 0 ในทุกตัวแปร ดังนี้

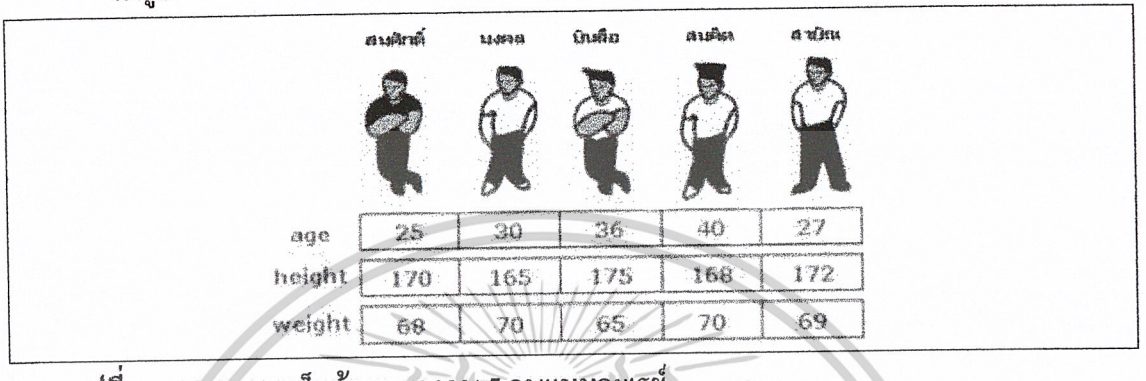
```
age[0] = 25;
height[0] = 170;
weight[0] = 68;
```

สำหรับคนที่ 2 ก็จะเก็บลงไปในอะเรย์ช่องที่ 1 ของทุกตัวแปร เช่นกัน ดังนี้

```
age[1] = 30;
height[1] = 165;
weight[1] = 70;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับคนที่ 3 , 4 และ 5 ก็จะเก็บลงไปในอะเรย์ช่องที่ 2,3 และ 4 ของทุก ๆ ตัวแปร ตามลำดับเช่นกัน จากตัวอย่างนี้ เราจะสรุปได้ว่า การใช้อะเรย์ในการเก็บข้อมูลที่มีจำนวนมาก ๆ และมีตัวแปรหลาย ๆ ตัวค่อนข้างจะยุ่งยากพอสมควร เพราะเราใช้ลำดับของอะเรย์เป็นตัวกำหนดลำดับการเก็บข้อมูล ถ้าเราต้องการเรียกดูข้อมูลของคนที่ 1 เราก็จะต้องแสดงข้อมูลที่อยู่ในช่องที่ 0 ของอะเรย์ทุกตัวออกมา รูปแบบของการเก็บข้อมูลในลักษณะนี้ เป็นดังรูป

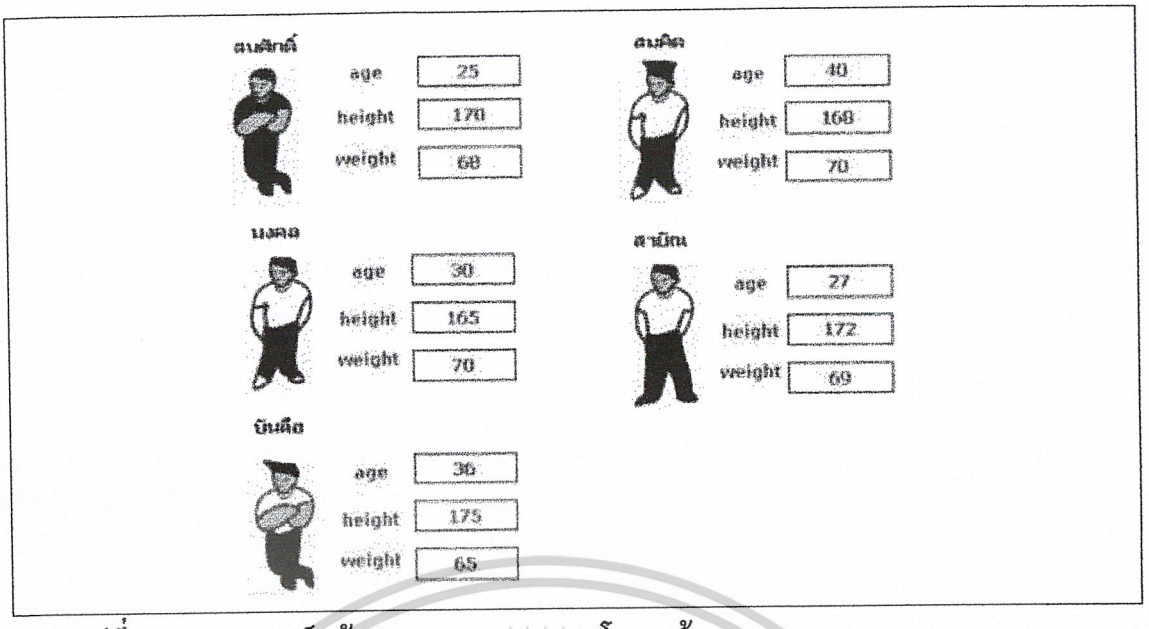


age	25	30	36	40	27
height	170	165	175	168	172
weight	68	70	65	70	69

รูปที่ 2.3 แสดงการเก็บข้อมูลของคน 5 คนแบบอะเรย์

จะเห็นว่า อะเรย์ 1 ตัวที่เก็บอายุ ก็จะเก็บอายุทุก ๆ ช่อง ตัวแปรที่เก็บส่วนสูง ก็จะเก็บแต่ส่วนสูงทั้งหมด วิธีการลักษณะนี้เสี่ยงต่อการผิดพลาดอย่างมากในเรื่องของลำดับ เพราะตัวแปรแต่ละตัวมีความอิสระต่อกัน การอ้างลำดับผิดในกรณีที่มีปริมาณข้อมูลมาก ๆ เช่น เก็บข้อมูลเป็นร้อย ๆ ข้อมูลขึ้นไป การไล่ลำดับจะต้องไล่ให้ตรงกัน ซึ่งจากปัญหาที่ได้กล่าวไปนี้ เราจะต้องใช้ตัวแปรแบบโครงสร้างมาช่วยแก้ไขปัญหา การใช้งานตัวแปรโครงสร้างนี้ จะคล้าย ๆ กับการที่เราสร้างตัวแปรชนิดใหม่ขึ้นมาเป็นของเราเอง โดยพิจารณาจากข้อมูลที่เราต้องการจัดเก็บ รูปแบบของการเก็บข้อมูลแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการเก็บข้อมูลของคน 5 คนแบบโครงสร้าง

การประกาศตัวแปรโครงสร้าง เราจะใช้คำสั่ง typedef struct ดังนี้

```
typedef struct _person
{
        int age;
        int height;
        int weight;
}person;
```

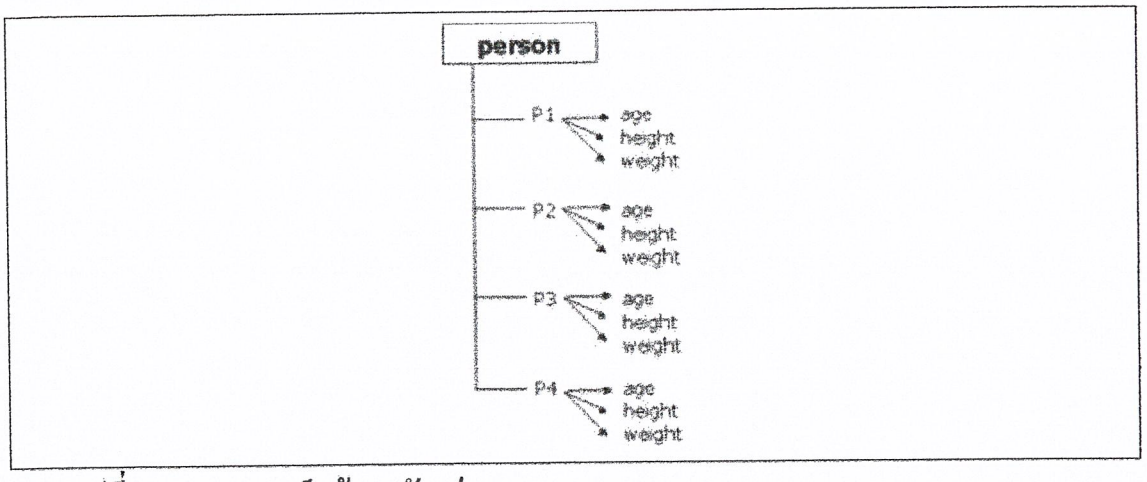
จากข้างต้น เป็นการสร้างตัวแปรแบบโครงสร้างชนิดใหม่ขึ้นมา ชื่อว่า person เราจะถือว่าเป็นการสร้างตัวแปรแบบใหม่ และตัวแปรนี้สามารถเก็บค่าได้ 3 ค่า คือ อายุ (age) , น้ำหนัก (Weight) และส่วนสูง (Height) เมื่อเราประกาศตัวแปรแบบใหม่นี้เอาไว้ในโปรแกรมของเราแล้ว เราก็สามารถสร้างตัวแปรแบบ person นี้ได้ทันที โดยเขียนโปรแกรมดังนี้

```
person p1,p2,p3,p4;
```

ตอนนี้ เราได้ตัวแปร p1,p2,p3 และ p4 เป็นตัวแปรแบบ person แล้ว

สำหรับตัวอย่างนี้ก็เช่นกัน คือ เมื่อเราได้สร้างตัวแปรโครงสร้างชนิดใหม่ขึ้นมาแล้ว โดยชื่อว่า person เราก็สามารถที่จะสร้างตัวแปรแบบ person ขึ้นมาได้ และตัวแปรที่ถูกสร้างขึ้นมานี้ จะมีคุณลักษณะความเป็นตัวแปรแบบ person ทุกอย่าง คือ ภายในตัวมันเองสามารถเก็บค่าย่อย ๆ ได้อีก 3 ค่า ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงการเก็บข้อมูลตัวแปร person

ลักษณะของตัวแปรแบบโครงสร้างนี้ จะจัดเก็บข้อมูลเป็นหมวดหมู่ และแบ่งแยกชัดเจนมากขึ้นอย่างเห็นได้ชัดว่า age , height และ weight ก็เป็นของคน ๆ นั้นเลย เพราะฉะนั้นมันจะไม่ปะปนกันคือ age ของ p1 จะเป็นของ p1 โดยเฉพาะ อาจจะไม่เท่ากับ age ของ p2 ก็ได้ เพราะฉะนั้นด้วยหลักการของโครงสร้างนี้ เมื่อเรานำมาใช้ในการเก็บข้อมูลอายุ, น้ำหนัก และส่วนสูงของคน 5 คนดังที่ได้ยกตัวอย่างไว้ในข้างต้น ถ้าเขียนด้วยภาษาซี ก็จะเขียนลักษณะนี้

```
typedef struct _person
{
        int age;
        int height;
        int weight;
}person;

person p[5];

p[0].age=25;
p[0].height=170;
p[0].weight=68;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


```
p[1].age=30;
p[1].height=165;
p[1].weight=70;
p[2].age=36;
p[2].height=175;
p[2].weight=65;

p[3].age=40;
p[3].height=168;
p[3].weight=70;

p[4].age=27;
p[4].height=172;
p[4].weight=69;
```

จากตัวอย่าง จะเหมือนกับว่าเรามีตัวแปรทั้งหมด 15 ตัว และแต่ละตัวก็ถูกจัดหมวดหมู่แบ่งออกเป็นคน ๆ เป็นข้อมูลเฉพาะส่วน ๆ ไป เราสามารถเข้าถึงข้อมูลที่เป็นอายุของคนที่ต้องการได้โดยอ้างอิงลำดับของคน ๆ นั้นและตามด้วยชื่อตัวแปรที่เราต้องการ เช่น ต้องการอายุของนายมงคล ซึ่งอยู่ลำดับที่สอง ก็คือ p[1].age เป็นต้น จะเห็นได้ว่า วิธีการนี้เป็นการเปลี่ยนแนวคิดและมุมมองจากอะเรย์ ให้มาเป็นแนวคิดที่คล้าย ๆ กับเชิงวัตถุเลย คือ เรามองแต่ละคน ๆ เป็นวัตถุมากขึ้น 1 คน ก็จะมีอายุ , น้ำหนัก และส่วนสูง หรือมีข้อมูลมากกว่านี้ใน 1 โครงสร้าง ทำให้การจัดการข้อมูลทำได้ง่ายกว่าการใช้อะเรย์แบบในวิธีแรกที่ได้นำเสนอไป

58777
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การสร้างคลาส**

การสร้างคลาสใน C++ เราจะเขียนคล้าย ๆ กับการประกาศตัวแปรแบบโครงสร้าง ดังที่ได้กล่าวไปแล้ว คลาสจะเป็นโครงสร้างที่รวมการทำงานทั้งหมดเข้าด้วยกัน ให้เป็นวัตถุก้อนเดียวกัน

ดูตัวอย่างโปรแกรม EASY.CPP ต่อไปนี้

```
#include

class Person
{
        int age;
        void setAge( int a )
        {
                age=a;
        }
        void showAge( )
        {
                printf( "age = %d\n",age );
        }
};

void main( )
{
        Person p;
}
```

โปรแกรม EASY.CPP ในข้างต้น มีการประกาศคลาส Person เอาไว้ ภายในคลาสมีสมาชิกดังนี้

1. ตัวแปร age
2. ฟังก์ชัน setage(int a)
3. ฟังก์ชัน showage( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส Person ที่ยกตัวอย่างมานี้ แสดงให้เห็นถึงการจัดเก็บข้อมูล ก็คือ อายุ และมีฟังก์ชัน setage ที่ใช้ในการกำหนดค่าอายุโดยรับพารามิเตอร์จากภายนอก และมีฟังก์ชัน showage ที่ใช้ในการแสดงค่าอายุออกมาทางจอภาพโดยฟังก์ชัน printf และในฟังก์ชัน main( ) ก็ได้มีการประกาศวัตถุของคลาส Person นี้ชื่อว่า p เอาไว้ จากนั้นก็จบโปรแกรม โปรแกรมนี้เป็นตัวอย่างที่แสดงให้เห็นถึงการประกาศคลาส ซึ่งมีรูปแบบดังนี้

```
class ชื่อคลาส
{
        //ประกาศตัวแปร
        //ประกาศฟังก์ชัน
}
```

ในคลาส ๆ หนึ่งจะมีแค่การสร้างตัวแปรหรือฟังก์ชันเพียงอย่างเดียวก็ได้ หรือเขียนแค่คลาสเพียงอย่างเดียว มีแต่ {....} ไม่ต้องมีตัวแปรหรือฟังก์ชันไว้เลยก็ได้

จากโปรแกรม Easy.cpp ในข้างต้นนั้น เราจะมาทำความเข้าใจกันดังนี้ ในคลาส Person มีตัวแปร age ตัวหนึ่ง และมีฟังก์ชัน setAge ฟังก์ชันนี้รับค่าพารามิเตอร์ 1 ตัว คือ int a และเมื่อรับเข้ามาแล้ว ก็จะทำการกำหนดให้กับตัวแปร age ที่อยู่ภายในคลาส และอีกฟังก์ชันหนึ่งก็คือ ฟังก์ชัน showAge ซึ่งเป็นฟังก์ชันที่ใช้ในการแสดงค่า age ออกทางจอภาพ จะเห็นได้ว่าการทำงานจะอยู่ภายในคลาสเท่านั้น จบแค่ภายในคลาส เมื่อเราประกาศวัตถุของคลาสดังนี้

```
Person p;
```

ก็เท่ากับว่า ตอนนี้คลาส Person มีทายาทแล้ว นั่นก็คือ วัตถุ p นั่นเอง และ p นี้ก็มีคุณลักษณะทุกอย่างเช่นเดียวกับคลาส Person นี้เลย นั่นก็คือ p สามารถเรียกใช้ตัวแปร a , ฟังก์ชัน setAge และ showAge ได้ ให้เราแก้ไขโปรแกรม Easy.cpp ให้เป็นดังนี้

```
void main( )
{
        Person p;
        p.setage( 25 );
        p.showage( );
}
```

จากข้างต้น เป็นการสั่งให้ p เรียกฟังก์ชัน setAge โดยใส่พารามิเตอร์ 25 ให้กับฟังก์ชัน เป็นการเรียกฟังก์ชันที่อยู่ในคลาส Person เอง จากนั้นก็เรียกฟังก์ชัน showAge ตามลำดับ เพื่อแสดงค่าอายุในตัวแปร age ออกมา ปัญหานั้นก็คือ เมื่อเราแปลโปรแกรมนี้โดยใช้ตัวแปรภาษา C++ ใด ๆ จะพบกับข้อความผิดพลาดลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\'setAge\' : cannot access private member declared in class \'Person\'

\'showAge\' : cannot access private member declared in class \'Person\'

จากข้อความที่ปรากฏในข้างต้น แปลแล้วได้ความว่า ฟังก์ชัน setAge และฟังก์ชัน showAge ไม่สามารถเรียกใช้ได้ ข้อความผิดพลาดจะฟ้องออกมาในบรรทัดนี้

```
p.setAge( 25 );
p.showAge( );
```

ที่เป็นเช่นนี้ก็เพราะว่ากฎของ OOP ข้อหนึ่งก็คือ คลาสจะมีการปกป้องข้อมูลไม่ให้กระบวนการภายนอกคลาสใด ๆ เข้าถึงสมาชิกในคลาสได้เลย เมื่อเราประกาศวัตถุ p ซึ่งวัตถุนี้เป็นตัวแปรที่อยู่ภายนอกคลาส ดังนั้น p จึงไม่สามารถเข้าถึงตัวแปร age และฟังก์ชัน setAge กับ showAge ได้ คุณสมบัติดังกล่าวเป็นคุณสมบัติของการเขียนโปรแกรมเชิงวัตถุที่ช่วยให้สมาชิกภายในคลาสไม่ถูกรบกวน ทำให้คลาสมีคุณสมบัติปกป้องข้อมูลหรือ Encapsulation ได้นั่นเอง ซึ่งเป็นสิ่งที่แตกต่างออกไปจากตัวแปรแบบโครงสร้างที่เห็นได้ชัด ถ้าเราจะทำให้วัตถุ p สามารถเรียกใช้สมาชิกในคลาสได้ ผู้เขียนโปรแกรมจะต้องอนุญาตให้สมาชิกในคลาสตัวที่ต้องการ มีคุณสมบัติเป็น "สาธารณะ" หรือ public ดังนี้

```
class Person
{
    int age;
public:
    void setage( int a )
    {
        age=a;
    }
    void showage( )
    {
        printf( "age = %d\\n",age );
    }
};
void main( )
{
    Person p;
    p.setage( 25 );
    p.showage( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

จากข้างต้น ในตอนนี้ฟังก์ชันทั้งสองในคลาสได้กลายเป็น public แล้ว ดังนั้น จึงสามารถเรียกใช้ p.setAge และ p.showAge ได้อย่างสมบูรณ์

จากตัวอย่าง Easy.cpp ในข้างต้น จะเห็นได้ว่า ไม่สามารถเข้าถึงวัตถุภายในคลาสได้ทันที วัตถุจะเรียกใช้ฟังก์ชันหรือตัวแปรในคลาสได้จะต้องกำหนดให้เป็นแบบ public ซึ่งตรงจุดนี้เป็นความสามารถของการเขียนโปรแกรมเชิงวัตถุที่จะช่วยให้เกิดความปลอดภัยของข้อมูลในคลาสมากขึ้น

ระดับในการเข้าถึงข้อมูลในคลาสมี 3 ระดับ คือ

1. ส่วนตัว หรือ Private
2. สาธารณะ หรือ Public
3. ปกป้อง หรือ Protect

ถ้าเราสร้างคลาสขึ้นมา โดยไม่ได้ระบุว่าเป็น private, public หรือ protect ใด ๆ สมาชิกภายในคลาสก็จะเป็นแบบ private ทั้งหมด

พิจารณาคลาสต่อไปนี้

```
class Person
{
    int age;
    void setage( int a )
    {
        age=a;
    }
    void showage( )
    {
        printf( "age = %d\\n",age );
    }
};
```

สมาชิกทุกตัวในคลาสเป็นแบบ private หมด สมาชิกแบบ private นั้น จะอนุญาตให้กระบวนการที่ดำเนินการอยู่ภายในคลาสเดียวกันเรียกใช้งานกันได้ ในคลาส Person จะเห็นว่ามีฟังก์ชัน setAge และ showAge ซึ่งต่างก็เรียกใช้ตัวแปร age ได้ พูดง่าย ๆ ก็คือ ตัวแปร และฟังก์ชันนั้นสามารถเรียกใช้กันภายในคลาสได้อย่างปกติทุกอย่าง แต่ถ้าเกิดเราสร้างวัตถุของคลาสขึ้นมา และเรียกใช้ตัวแปรที่อยู่ในคลาส

```
p.age=10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่สมารถเรียกใช้แบบนี้ได้ ทั้งนี้เป็นเพราะว่าวัตถุ p เป็นตัวแปรที่ถูกสร้างขึ้นมาใหม่อยู่ภายนอกคลาส สำหรับการประกาศให้เป็น public นั้น เราจะต้องเขียนคำว่า public: ไว้ที่บรรทัดก่อนที่จะทำการประกาศตัวแปรหรือฟังก์ชันนั้น เช่น

```
class Person
{
        public:
        int age;
        void setage( int a )
        {
                age=a;
        }
        void showage( )
        {
                printf( "age = %d\\n",age );
        }
};
```

2.3 DirectX

2.3.1 ประวัติความเป็นมาของ DirectX

ในช่วงที่ระบบปฏิบัติการไมโครซอฟต์วินโดวส์เริ่มต้นขึ้นมานั้น สิ่งที่เป็นเอกลักษณ์ของวินโดวส์ ก็คือเรื่องของ GUI (Graphical User Interface) โดยมี GDI (Graphical Device Interface) เป็นเครื่องมือสำหรับจัดการทางด้านภาพ และเกมทั่วไปก็ยังคงพัฒนาภายใต้ระบบปฏิบัติการเดิมอยู่ ซึ่งก็คือระบบปฏิบัติการ “DOS” นั่นเอง

การเริ่มต้นครั้งแรกสำหรับไมโครซอฟต์ คือ “เครื่องมือ Win-G” สำหรับ Window95 และดูเหมือนว่าเครื่องมือตัวนี้จะไม่ได้รับการตอบรับที่ดีนักจากผู้พัฒนาเกมทั่วโลก ยังผลให้ไมโครซอฟต์เองต้องปรับปรุงรูปแบบใหม่ให้กับเครื่องมือตัวนี้อีกครั้ง

ไมโครซอฟต์ได้ทำการจัดตั้งทีมงานขึ้นมาใหม่เพื่องานกราฟิก งานมัลติมีเดีย งานเนตเวิร์ก งานด้านการรับข้อมูล Input และ งานกราฟิก 3 มิติ (ชื่อเทคโนโลยี “Render Graphic) โดยใช้ชื่อว่า “DirectX”

DirectX 2.0 สามารถเข้ามาจัดการงานด้านเกมได้เป็นอย่างดี การทำงานเร็วขึ้นกว่าระบบปฏิบัติการ DOS ก็ เลยทำให้โปรแกรมเมอร์ทั่วโลกได้เริ่มหันมามองเทคโนโลยี ตัวนี้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกครั้ง และจากการพัฒนาอย่างต่อเนื่องของไมโครซอฟต์เพื่อเพิ่มประสิทธิภาพของ DirectX ให้มีความสามารถมากขึ้น ทำให้เกิด DirectX 2.0, DirectX 3.0, DirectX 5.0, DirectX 6.0, DirectX 7.0, DirectX 8.0 และ DirectX 9.0 ซึ่งเป็นเวอร์ชันใหม่ล่าสุดที่ไมโครซอฟต์ผลิตขึ้น และยังมีการพัฒนาเวอร์ชันใหม่ ๆ ต่อไป จากเทคโนโลยีนี้เอง เป็นผลให้เกมบนระบบปฏิบัติการ DOS ค่อย ๆ เลือนหายไปในที่สุด

ทุกวันนี้ DirectX 8.0 ได้นำเทคโนโลยี COM (Component object Model) มาใช้งาน ทำให้ได้รับความสะดวกสบายในการใช้งานเป็นอย่างยิ่ง ผู้พัฒนาเกมสามารถควบคุมและสร้างการทำงานได้อย่างอิสระ

2.3.2 ความรู้เบื้องต้นเกี่ยวกับ DirectX

DirectX เป็น API ของไมโครซอฟท์พัฒนาเพื่อใช้จัดเตรียมอินเตอร์เฟส สำหรับควบคุมฮาร์ดแวร์ มัลติมีเดียบนระบบ Microsoft Windows ได้อย่างมีประสิทธิภาพ และเป็นเครื่องมือให้โปรแกรมเมอร์ทำงานกับคำสั่ง และโครงสร้างข้อมูลในระดับใกล้ชิดฮาร์ดแวร์ โดยไม่ต้องสร้างโค้ดติดต่อในระดับล่างซึ่งวิธี ติดต่อจะแตกต่างกันไปตามประเภทของอุปกรณ์ การเขียนโค้ดที่เป็นอิสระจากอุปกรณ์ในลักษณะนี้ ช่วยให้โปรแกรมเมอร์สามารถสร้างซอฟต์แวร์เพื่อทำงานดังกล่าวได้อย่างดี แม้ผู้ใช้จะปรับเปลี่ยนอุปกรณ์ตัวใหม่ และเพิ่มการ์ดเร่งความเร็วแบบสามมิติ เสียง อุปกรณ์อินพุต และอื่นๆก็ ตาม

DirectX ได้รับการออกแบบให้นักพัฒนามีสภาพแวดล้อมคล้ายคลึงกับสภาพแวดล้อมที่มีประสิทธิภาพ ของ MS-DOS ซึ่งทำงานได้เร็วกว่าโค้ดที่ทำงานบนวินโดวส์ เนื่องจากไม่ต้องสูญเสียประสิทธิภาพจาก API สำหรับจัดการงานมัลติมีเดียของ วินโดวส์รุ่นก่อน แต่อย่างไรก็ตามการสนับสนุนความสามารถในการทำงานของฮาร์ดแวร์ที่มีอยู่บนระบบ โค้ดที่เขียนขึ้นด้วย DirectX สามารถรันได้เร็วกว่าแอพพลิเคชั่นของ MS-DOS

เมื่อใดก็ตามที่สร้างวัตถุ DirectX ให้กับดีไวซ์นั้น DirectX จะเข้าไปซักถามฮาร์ดแวร์ผ่าน HAI เพื่อดึงเอาข้อมูลเกี่ยวกับดีไวซ์ออกมานี้มีอยู่ในตาราง Cap Bits (Capability Bits) ข้อมูลที่มี อยู่ใน Cap Bits เป็นข้อมูลที่ ใช้บอกความสามารถที่ฮาร์ดแวร์สามารถทำได้ หรือความสามารถใดที่ HEL ต้องจำลอง

ไมโครซอฟท์จัดเตรียม Cap Bits ให้รับทราบฟีเจอร์ของฮาร์ดแวร์ที่มีอยู่ใน HAL และฟีเจอร์ที่ต้องจำลองขึ้นโดยซอฟต์แวร์ HEL ดังนั้นวิธีที่ดีที่สุดก็คือเขียนแอพพลิเคชั่นโดยใช้ค่าระบบ หรือฟีเจอร์ต่ำสุดที่ยอมรับได้ และ Optimize โค้ดที่เขียนให้รันได้เร็วและมีประสิทธิภาพมากที่สุด อีกทางหนึ่งควรเขียนโค้ดเผื่อการทำงานของ HEL ในอนาคตไว้ด้วย การสนับสนุนการใช้ฟีเจอร์ขั้นสูงบนระบบที่มีความสามารถของฮาร์ดแวร์ มากกว่าค่าระบบที่ตั้งไว้นั้น สามารถจัดเตรียมฟีเจอร์เหล่านี้เป็นฟีเจอร์พิเศษที่จะมีให้เลือกบนเกมได้ เช่น การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน Texture ขั้นสูง ,การทำโพลีกอนที่มีความซับซ้อนสูง หรือแม้แต่การสร้างแสงแบบไดนามิก เผื่อว่าใครก็ตามทีมีฮาร์ดแวร์ที่มีความสามารถสูงจะสามารถใช้งานฟีเจอร์ดังกล่าวได้ ดังนั้นควรออกแบบให้รองรับฟีเจอร์ ต่าง ๆ ที่มีได้ทั้งหมด

2.3.3 DirectX คืออะไร

ก่อนที่จะมี DirectX เกิดขึ้นนั้น นักพัฒนาเกมคอมพิวเตอร์สำหรับ ดอสหรือวินโดวส์ จะต้องเขียนเกม คอมพิวเตอร์ให้รู้จักกับฮาร์ดแวร์ ซึ่งมีอยู่มากมายในท้องตลาด ซึ่งในกรณีที่มีฮาร์ดแวร์ตัวใหม่เกิดขึ้น อาจเกิดปัญหาความไม่สนับสนุนกันระหว่างเกมคอมพิวเตอร์กับฮาร์ดแวร์ตัวใหม่นั้น นักเล่นเกมจะต้องรอจนกว่านักพัฒนาเกมจะทำการอัพเดตเกมนั้น ๆ ให้ใช้ความสามารถของฮาร์ดแวร์ตัวใหม่ได้ เมื่อเป็นเช่นนี้ไมโครซอฟต์จึงได้ทำการพัฒนาเทคโนโลยีที่มีความสามารถในการเป็นสื่อกลางติดต่อระหว่างเกมคอมพิวเตอร์ หรือโปรแกรมมัลติมีเดียต่าง ๆ กับฮาร์ดแวร์ขึ้นมาโดยใช้ชื่อว่า “DirectX”

DirectX ตามความหมายจะหมายถึง ไลบรารีคำสั่ง (Run Time Library) ที่ช่วยทำงานด้านมัลติมีเดีย Graphic โดยตัว DirectX Foundation จะมีส่วนประกอบที่เรียกว่า HAL (Hardware Abstraction Layer) จะใช้ซอฟต์แวร์ในการตรวจสอบความสามารถของฮาร์ดแวร์ที่อยู่ในเครื่องคอมพิวเตอร์นั้นอย่างอัตโนมัติ แล้วนำมากำหนดพารามิเตอร์ของแอพพลิเคชั่นให้ตรงตามความเหมาะสมระหว่างเกมคอมพิวเตอร์ หรือโปรแกรมมัลติมีเดียกับไดร์เวอร์ของฮาร์ดแวร์ที่มีส่วนเกี่ยวข้องกับเกมหรือโปรแกรมนั้น ทำให้การประมวลผลโปรแกรมทำได้เร็วขึ้น เนื่องจากขั้นตอนต่าง ๆ จะถูกนำไปประมวลผลโดยตรง ไม่ต้องอาศัยตัวกลางอย่างเช่น GDI (Graphic Device Interface) ก่อน ทำให้นักพัฒนาเกมคอมพิวเตอร์สามารถเขียนโปรแกรมให้สื่อสารกับ DirectX เท่านั้นก็เพียงพอ นอกจากนี้ DirectX Foundation ยังมีส่วนประกอบที่เรียกว่า HEL (Hardware Emulation Layer) ทำให้สามารถใช้โปรแกรมมัลติมีเดีย หรือโปรแกรมที่เกี่ยวข้องกับ 3D บน Hardware ที่ไม่สนับสนุนการใช้งานทางด้าน 3 มิติ โดยจะทำการจำลองความสามารถบางอย่างที่ฮาร์ดแวร์ตัวนั้นไม่มี ให้สามารถใช้งานได้กับโปรแกรมที่ต้องการ แม้จะมีข้อเสียอยู่บ้างตรงที่อาจทำให้การช้าลงบ้างก็ตาม แต่ก็คุ้มค่ากับความสามารถของ DirectX ที่มีอยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

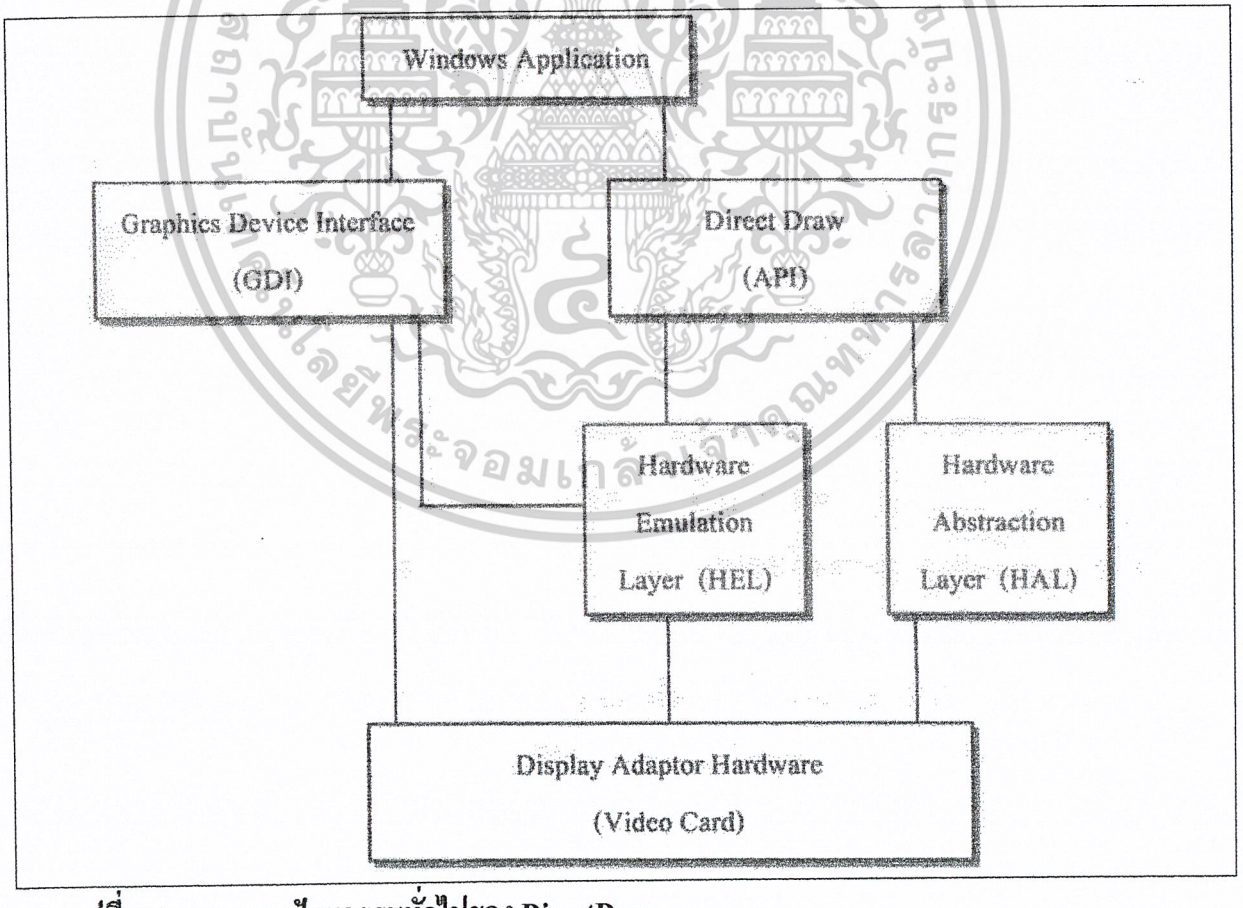
2.3.4 API ของ DirectX

DirectX ประกอบไปด้วย API มากมายที่ได้รับการออกแบบเพื่อใช้พัฒนาเกม และการจำลองแบบแบบสามมิติ (3D Simulation) (ส่วนใหญ่ยังไม่ได้พัฒนาให้เป็นแบบสามมิติ) โดย DirectX มีไลบรารีที่เก็บฟังก์ชันที่ใช้ในการเรนเดอร์แบบสองมิติและสามมิติ ,สร้างเสียงแบบปกติ และแบบสามมิติ ,ดนตรี ,ติดต่อแป้นพิมพ์ ,จอยสติ๊ก และอุปกรณ์อินพุตชนิดต่าง ๆ รวมทั้งฮาร์ดแวร์ที่มีความสามารถสร้างปฏิกิริยาสะท้อนกลับ (จอยสติ๊กแบบสั่น) และการเล่นเกมผ่านเครือข่าย สามารถใช้ไลบรารีของคำสั่งต่าง ๆ ที่รวมเข้ามาเพื่อสร้างเกม และทำซิมูเลชั่นที่งดงาม

ชุด API ที่มีอยู่ใน DirectX คือ DirectDraw, Direct3D, DirectMusic, DirectSound, DirectPlay, DirectInput และ DirectSetup

2.3.4.1 DirectDraw

DirectDraw เป็นชุด API สำหรับใช้จัดการอุปกรณ์แสดงผล, ควบคุมข้อมูลบิตแมปหน่วยความจำที่ ออกนอกพื้นที่สกรีน และสร้างการติดต่อที่รวดเร็วให้กับฟีเจอร์ของฮาร์ดแวร์ เช่น Blitting และ Page Flipping ซึ่งเป็นฟีเจอร์พื้นฐานที่ Direct3D สามารถทำได้



รูปที่ 2.6 แสดงสถาปัตยกรรมทั่วไปของ DirectDraw

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Cooperative level และ Display mode

ก่อนที่จะคำสั่งต่าง ๆ ใน DirectX ให้แสดงผลภาพได้นั้น เราจะต้องเซต Cooperative level และ Display mode ก่อน โดยการเซต Cooperative level ก็คือการบอก DirectDraw ว่าเราจะทำงานกับ DirectDraw ในลักษณะใด เช่น window mode หรือ full screen mode สามารถ interrupt ด้วยการกดปุ่ม Ctrl-Alt-Del ได้หรือไม่ และสามารถที่จะเปลี่ยนขนาดของ window ได้หรือไม่ (ใน window mode) เป็นต้น หลังจากเซต Cooperative level แล้วจะต้องเซต Display mode เป็นลำดับถัดมา เพื่อระบุ screen Resolution, color depths และ refresh rate ที่ต้องการ โดยจะต้องระบุความกว้าง ความยาวของ resolution เป็นหน่วย pixel และจะใช้สีแบบใด 6, 8, 24 หรือ 32 บิต หรือจะใช้ตารางสี (palette) ทั้งนี้ video hardware ที่ใช้จะต้องสนับสนุนกับ mode ที่เลือกด้วย มิฉะนั้นจะเกิด error Display mode

- Surface

Surface เป็นพื้นที่บน memory ที่ใช้สำหรับเก็บข้อมูลภาพ bitmap เพื่อเตรียมสำหรับการนำมาแสดงผลบนจอภาพต่อไป โดยพื้นที่นี้อาจอยู่บน video memory หรือ system memory ก็ได้ขึ้นอยู่กับค่า flags ที่ส่งให้ตอนสร้าง โดย surface แบ่งออกเป็น 2 ชนิด คือ Primary Surface และ Off-screen surface

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

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

- Blitting และ Flipping

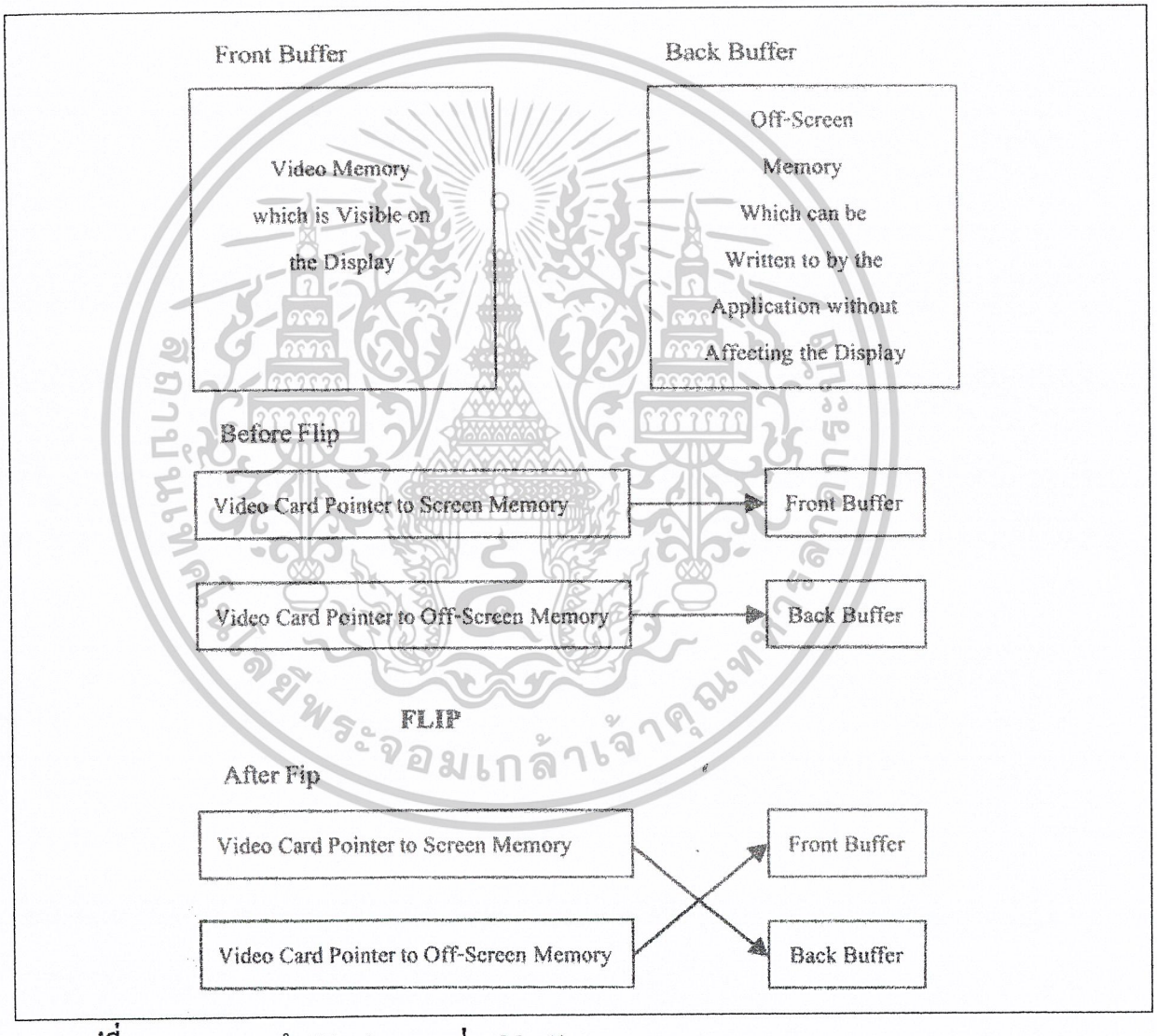
Blitting คือ การนำภาพต่าง ๆ จาก source ไปวาดบนบริเวณ destination ที่กำหนด โดยอาจมีการย่อ-ขยายภาพให้พอดีกับขนาด destination ที่กำหนด และสามารถทำการ transparent สีบางสีได้ด้วย การทำ Blitting สามารถทำได้หลายครั้ง โดยภาพที่ทำ Blitting จะเรียงซ้อนทับกันเป็นชั้น ๆ

Flipping เป็นการสลับที่กันของ buffer โดยก่อนการทำ Flipping จะต้องมีการสร้าง buffer ขึ้นมาอย่างน้อย 2 buffer คือ Front buffer กับ Back buffer และอาจมี Third buffer เพิ่มขึ้นมาก็ได้ โดย Front buffer คือ buffer ที่แสดงภาพออกที่หน้าจอ หรือ Primary surface นั่นเอง ส่วน Back buffer กับ Third buffer เป็นที่พักของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่จะรอแสดงผลบนหน้าจอ buffer ทั้ง 3 ชนิดนี้จะมีขนาดเท่ากับ Display mode ที่เลือกไว้

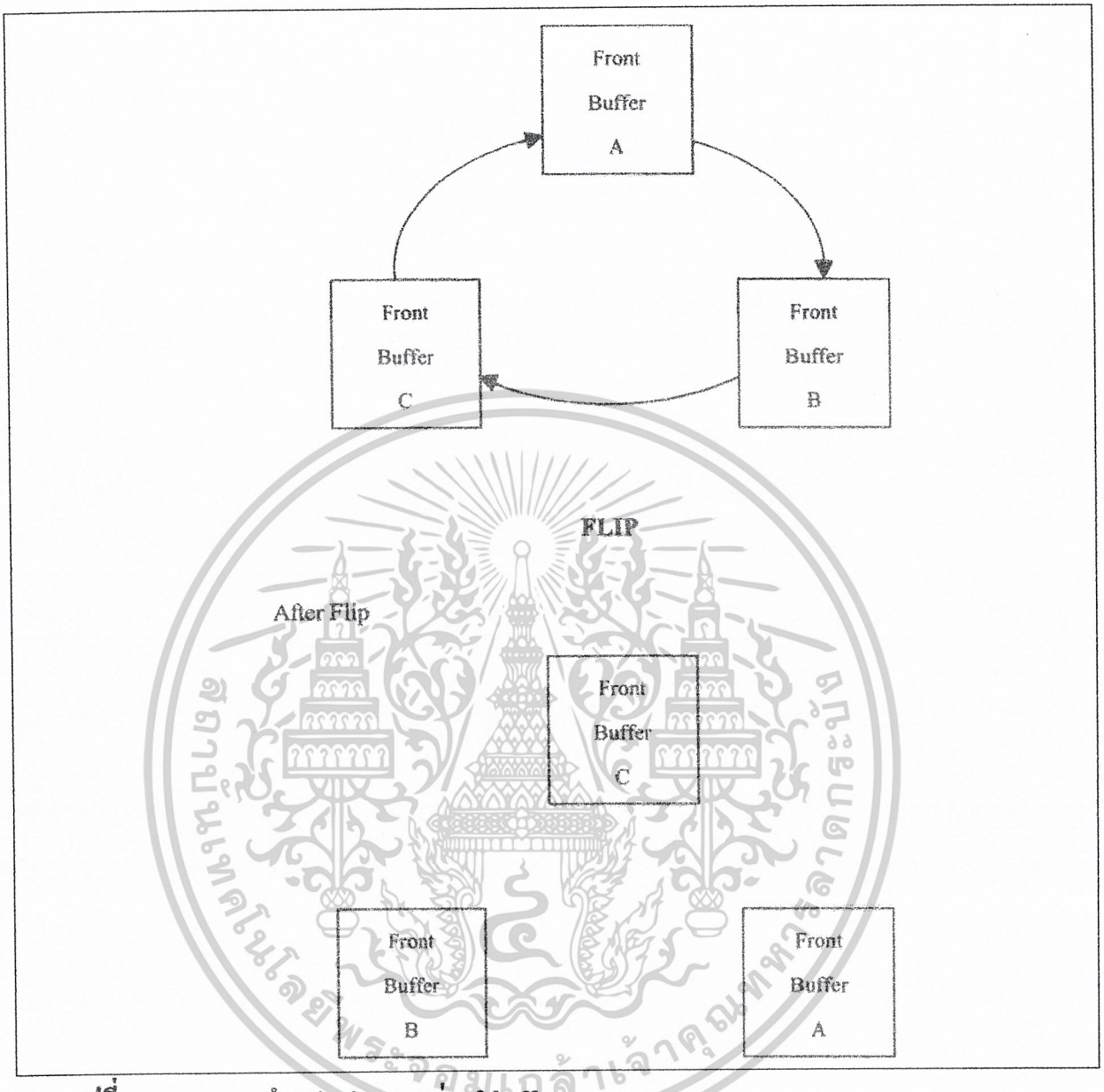
การวาดหรือ Blitting ลงบน Front buffer โดยตรงจะให้ภาพกระพริบ ควรเตรียมภาพที่จะแสดงผลไว้ให้พร้อมบน Back buffer แล้วทำการ Flipping โดย DirectDraw จะสลับภาพระหว่าง Front buffer กับ Back buffer เพื่อนำภาพบน Back buffer มาแสดงบนจอภาพ โดยการสลับในที่นี้ไม่ได้หมายถึง การสลับที่ข้อมูลภาพแต่เป็นการสลับเฉพาะ pointer ที่ชี้พื้นที่ที่เก็บข้อมูลอยู่เท่านั้น ซึ่งทำได้อย่างรวดเร็ว และไม่ทำให้เกิดการกระพริบ หรือการกระตุกของภาพ



รูปที่ 2.7 แสดงการทำ Flipping ระหว่าง 2 buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามี Third buffer การสลับภาพจะทำเป็นวงจรดังรูป



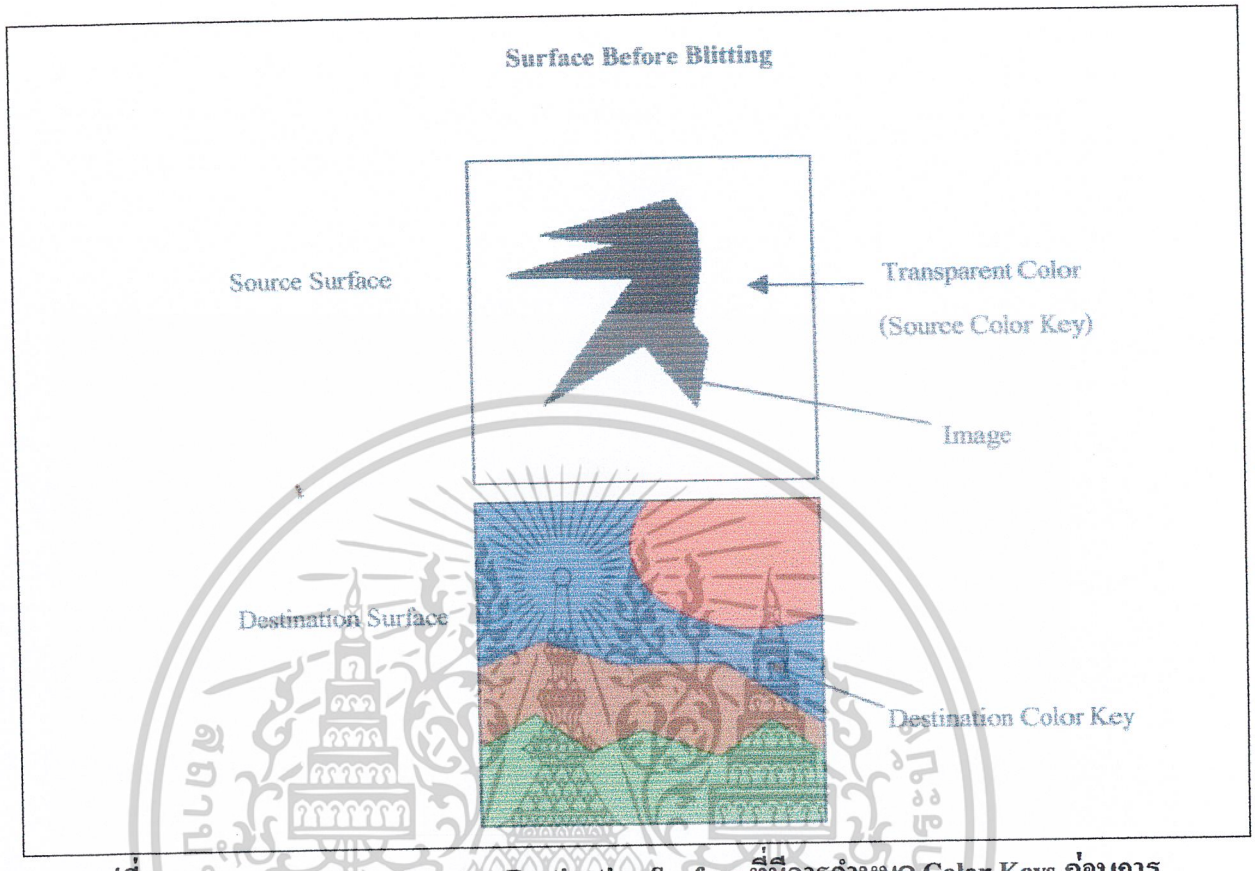
รูปที่ 2.8 แสดงการทำ Flipping ระหว่าง 3 buffer

- Transparent Blitting และ Color Keys

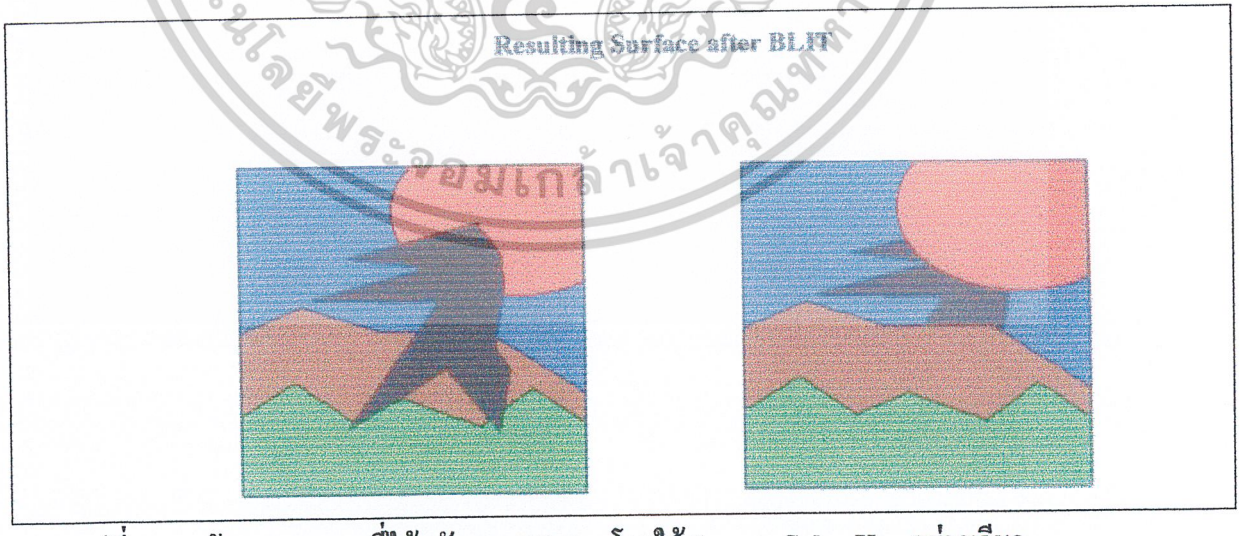
DirectDraw สามารถทำ Transparent Blitting ได้โดยอาศัย Color Keys Surface ต่าง ๆ โดยสามารถเซต Color Keys เป็นสีใดสีหนึ่ง หรือช่วงของค่าสีก็ได้ Color Keys ที่ถูกเซตไว้จะแสดงผลตอน Blitting โดยถ้า source มีการเซต Color Keys ไว้ ค่าสีตาม Color Keys จะไม่ถูก Copy ไปบน destination ทำให้บริเวณสีนั้นโปร่งแสงไป ถ้า destination มีการทำ Color Keys ไว้ ค่าสีตาม Color Keys เท่านั้นที่จะถูกทับโดยภาพที่ Copy มาจาก source การทำ Color Keys กับ source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มักจะใช้กับตัว sprite ที่มีการเคลื่อนที่ไปบนฉากหลัง ส่วนการทำ Color Keys กับ destination จะใช้กับกรณีที่ต้องการปิดบังบางส่วนของตัว sprites



รูปที่ 2.9 แสดง Source Surface และ Destination Surface ที่มีการกำหนด Color Keys ก่อนการ Blitting



รูปที่ 2.10 (ซ้าย) Surface ที่ได้หลังการ Blitting โดยใช้ Source Color Key อย่างเดียว

(ขวา) Surface ที่ได้หลังการ Blitting โดยใช้ Source Color Key และ Destination Color Key

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Palettes

Palettes คือ ตารางสีที่ใช้ในระบบ Index Color ซึ่งสามารถมีจำนวนสีได้ตั้งแต่ 2, 4, 16 และ 256 สี ระบบ Index Color นั้นจะมี Array สำหรับเก็บค่าสี RGB โดยแต่ละค่าอาจจะมีความละเอียดที่ 16 หรือ 24 บิต แล้วในข้อมูลภาพจะเก็บเป็นค่า Index จากตาราง Palettes แทนการเก็บค่าสีจริง ซึ่งจะใช้พื้นที่ 8 บิตต่อ 1 จุด DirectDraw สามารถสร้าง Palettes เพื่อเชื่อมต่อกับแต่ละ Surface

- Clippers

Clippers เป็น Object ที่ช่วยในการตัดภาพส่วนที่เกินออกจากพื้นที่ที่กำหนดไว้ รวมถึงตัดภาพที่นำมา Blitting บนพื้นที่ที่มี Clippers ด้วย พื้นที่ที่จะทำ Clippers นี้จะเป็นรูปสี่เหลี่ยมใน window mode Clippers จะช่วยในการจำกัดขอบเขตของการวาดภาพไม่ให้เกินออกไปจาก window ที่กำหนด ส่วนใน full screen Clippers จะช่วยในการแบ่งหน้าจอออกเป็นส่วนๆ ตามที่ต้องการ

2.3.4.2 Direct3D

Direct3D คือ API ที่สามารถใช้เขียนโปรแกรมกราฟิกสามมิติ และติดต่อใช้งานฮาร์ดแวร์เร่งความเร็วสามมิติ การ์ดแสดงผลส่วนใหญ่ที่มีขายในท้องตลาดสนับสนุนความสามารถในการเร่งการแสดงผลสามมิติ และเกมสามมิติส่วนใหญ่ที่มีอยู่ในปัจจุบันก็มักจะรันบนวินโดว์ซึ่งใช้ Direct3D

Direct3D ในยุคแรกมี API อยู่สองโหมดคือ โหมด Immediate (IM) และโหมด Relation (RM) ในโหมด IM (การเรนเดอร์วัตถุทำได้โดยฉับพลันได้ตามความต้องการของโปรแกรมเมอร์) เป็นโหมดที่ใช้งานยากแต่มีความยืดหยุ่นสูง เป็น API ในระดับล่างสำหรับใช้เขียนเกมที่ทำงานได้เร็ว และมีประสิทธิภาพเท่าที่จะเป็นไปได้บนระบบ ในโหมด RM (API จะเก็บซีนลงในฐานข้อมูล แล้วนำมาเรนเดอร์ทั้งหมดในคราวเดียวกัน) เป็นโหมดที่สร้างขึ้นมาเป็นเลเยอร์ที่อยู่บนสุดของโหมด IM โดยโหมดนี้จะจัดเตรียมบริการต่าง ๆ เช่น การจัดการ Texture, การโหลด Object File, การจัดลำดับเฟรม และการทำวัตถุเคลื่อนไหว การศึกษา และใช้งานโหมด RM นั้นง่ายกว่าเมื่อเทียบกับโหมด IM แต่ในโหมด IM เป็นโหมดที่มีประสิทธิภาพ และความยืดหยุ่นสูงกว่า ดังนั้นการพัฒนาการทำงานในโหมด RM จึงได้หยุดลงใน DirectX เวอร์ชัน 6 และมุ่งพัฒนาการทำงานในโหมด IM ให้มีความสามารถ และใช้งานง่ายในเวอร์ชันต่อมา ด้วยเหตุนี้การทำงานในโหมด RM จึงไม่สนับสนุนเทคโนโลยีใหม่ เช่น Multitexturing Bump Mapping Hardware Transformation และ Lighting ดังนั้นความสามารถทั้งหมดของโปรแกรมสามมิติควรเขียนขึ้นมาด้วยการใช้โหมด IM

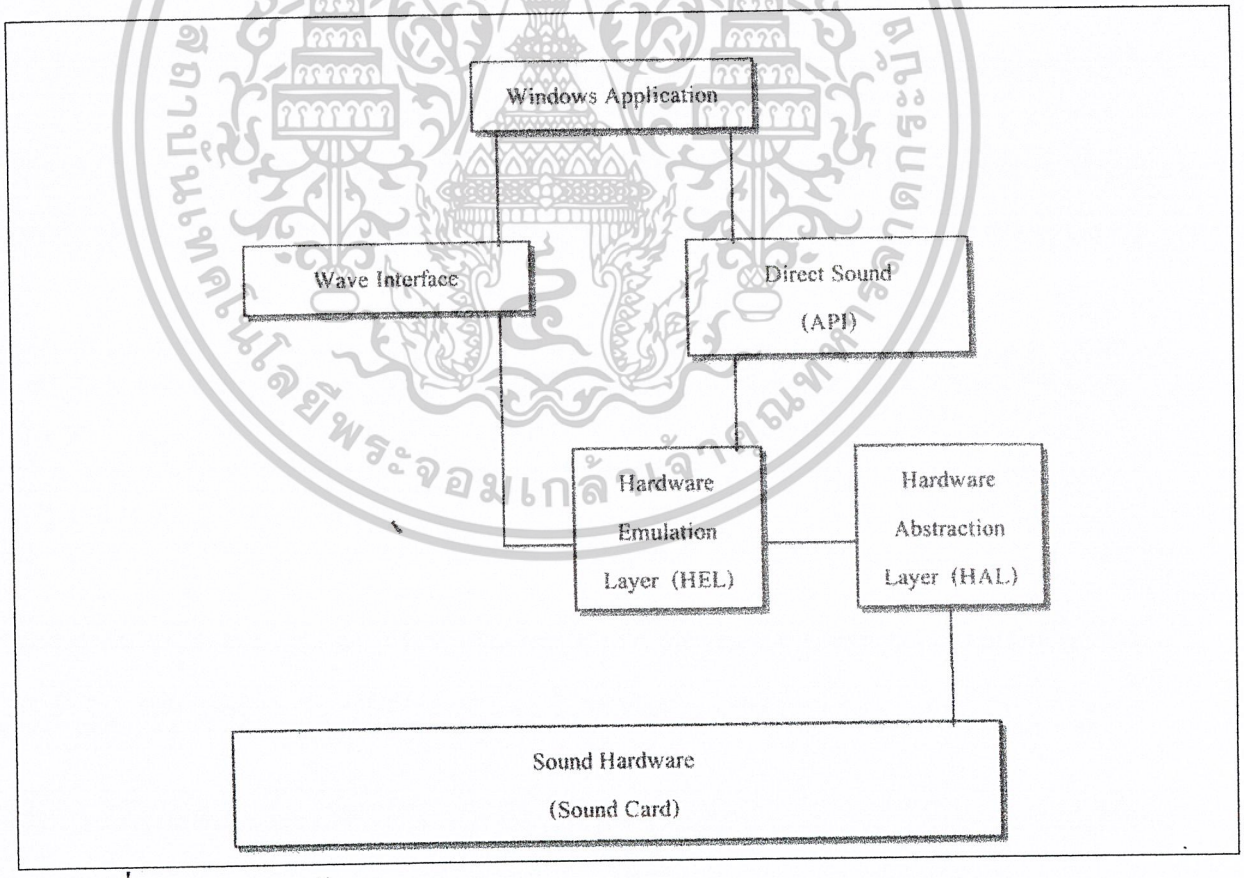
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.3 DirectMusic

เป็นชุดของ API ที่ทำงานร่วมกับข้อมูลประเภท Message-based Musical Data ซึ่งเป็นข้อมูลที่แปลงมาจาก Wave Sample ด้วยซินธีไซเซอร์ (Synthesizer) ทั้งแบบฮาร์ดแวร์ หรือซอฟต์แวร์ โดยปกติถ้าเป็นแบบซอฟต์แวร์จะใช้โปรแกรม Microsoft Software Synthesizer เพื่อสร้าง Wave Sample ให้กับ DirectSound การใช้เสียงดนตรีที่ได้จากซินธีไซเซอร์จะเป็นไปตามมาตรฐาน DLS (Downloadable Sound) นอกจากนี้ DirectMusic ยังเป็นกลไกที่ใช้สร้างเพลงตามที่ กำหนด

2.3.4.4 DirectSound

API ชุดนี้เป็นเครื่องมือทีใช้จัดการเสียงแบบเสียงสเตริโอ และแบบสามมิติอย่างมีประสิทธิภาพ ประกอบด้วยความสามารถในการจัดการหน่วยความจำ และการผสมเสียงฮาร์ดแวร์ DirectSound ได้รับการออกแบบมาเพื่อดึงความสามารถของฮาร์ดแวร์บนระบบ การรวมเสียงสามมิติเข้าไปในเกม หรือการซิมูเลชัน ทำให้แอพพลิเคชั่นให้เสียงได้สมจริงสมจัง เช่น การได้ยินเสียงทางซ้าย ทางขวา ด้านบน หรือเสียงที่ มารอบ ๆ ตัว



รูปที่ 2.11 แสดงสถาปัตยกรรมของ DirectSound

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.4 DirectPlay

DirectPlay ทำให้เกมสามารถเล่นได้หลายคน และทำให้สามารถเชื่อมต่อแบบ Transport-independent รวมทั้งยังให้บริการMessaging Service ด้วย ซึ่งเกมที่มีการพัฒนาออกมากันอยู่ในทุกวันนี้ล้วนสนับสนุนความสามารถดังกล่าว

2.3.4.5 DirectInput

เป็น API ของ DirectX ที่สนับสนุนอินพุตความเร็วต่ำ (Low-latency Input) ซึ่งเป็นอุปกรณ์อินพุตส่วนใหญ่ที่มีในปัจจุบัน ใน DirectX7 เพิ่มการสนับสนุนอุปกรณ์อินพุตทุกชนิดที่มีความสามารถ แสดงปฏิกิริยาสะท้อนกลับ เช่น จอยสติ๊ก หรือพวงมาลัยแบบสั่นได้ อุปกรณ์อินพุตแบบนี้สามารถจำลองสถานการณ์บางส่วนให้สมจริง เช่น จำลองสถานการณ์เมื่อขับรถชนบนถนน อุปกรณ์อินพุต (พวงมาลัย) จะส่งแรงสั่นสะเทือนให้รู้สึกตามความรุนแรง และความเสียหายของรถ, แรงถีบของปืนเมื่อยิง, แรงลมปะทะเมื่อบิน และคลื่นที่กระทบเรือ เป็นต้น

2.3.4.5 DirectSetup

API ชุดนี้ของ DirectX มีหน้าที่ติดตั้งคอมโพเนนต์ DirectX Runtime ให้กับระบบโดยอัตโนมัติถ้าหากระบบดังกล่าวยังไม่ได้รับการติดตั้ง แม้ในปัจจุบันมักจะมีการติดตั้งคอมโพเนนต์มากับระบบปฏิบัติการแล้วก็ตาม แต่สำหรับ DirectX แล้วไมโครซอฟท์อนุญาตให้ใช้โดยไม่ต้องเสียค่าใช้จ่าย ดังนั้นจึงไม่ต้องกังวลเรื่องของลิขสิทธิ์ โดยสามารถติดตั้งกี่ครั้งก็ได้ ไมโครซอฟท์เตรียมคอมโพเนนต์ DirectX Runtime ไว้ในไดเร็กทอรี DFX\redist ใน DirectX7 SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 CDX Game Engine

CDX เป็นฟรี GDK (Game Development Kit) พัฒนาขึ้นโดย Dan Farley Bil Simser ประกอบด้วยคลาสที่พัฒนาด้วยภาษา C++ ใช้สำหรับเขียนเกมบนวินโดวส์ CDX ถูกสร้างขึ้นมาบนเทคโนโลยี Microsoft Windows และ DirectX ซึ่งช่วยให้การพัฒนาเกมทุกรูปแบบด้วยภาษา C++ ทำได้ง่ายขึ้น

CDX ได้รวบรวมสิ่งต่าง ๆ ที่มีบน DirectX เช่น การสร้างภาพ, ภาพเคลื่อนไหว , เสียง, เพลง ประกอบเกม, วีดีโอ, การสร้างเมนูเกม, รูปร่างสามมิติพื้นฐาน เป็นต้น โดย CDX จะเป็นตัวกลางติดต่อกับ DirectX อีกทีหนึ่ง ซึ่งช่วยให้ลดขั้นตอนในการสร้างเกมลงได้ CDX เป็น GDK ที่รวบรวมคลาสสำหรับพัฒนาเกมไว้มากมาย เป็นคลาสที่ยืดหยุ่น ง่ายต่อการเรียนรู้และนำไปใช้

ในปัจจุบัน CDX สามารถคอมไพล์และรันบนระบบปฏิบัติการ Windows 32bit โดยรองรับ DirectX เวอร์ชัน 3 เป็นต้นไป และสามารถทำการคอมไพล์บนคอมไพเลอร์ ทั้ง Microsoft Visual C++ และ Borland C++ ได้

2.4.1 องค์ประกอบของ CDX

ใน CDX จะประกอบไปด้วยคลาสหลายๆคลาส ที่ทำหน้าที่ต่างๆกัน ดังนี้

- CDXScreen ใช้สร้างหน้าจอเกม
- CDXSurface ควบคุม Surface
- CDXLayer ใช้ควบคุม Surface สำหรับทำภาพพื้นหลัง
- CDXSprite ควบคุมตัวละครแบบสไปรท์ในเกม
- CDXInput รับค่าจากผู้ใช้ผ่านทางคีย์บอร์ด เมาส์ และจอยสติ๊ก
- CDXSound เก็บไฟล์เสียง .wav
- CDXSoundBuffer เล่นไฟล์เสียง .wav
- CDXMidi เล่นไฟล์เสียง MIDI (.mid)

หากเปรียบเทียบกับส่วนต่าง ๆ ใน DirectX ก็จะได้ดังนี้

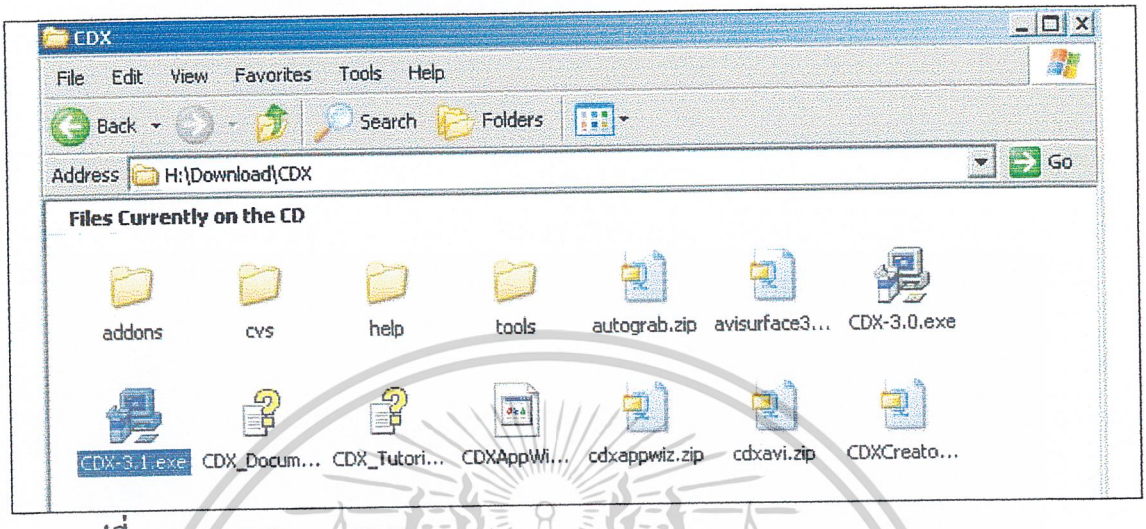
- DirectDraw คือ CDXScreen , CDXSurface , CDXSprite
- DirectInput คือ CDXInput
- DirectSound คือ CDXSound และ CDXSoundBuffer
- DirectMusic คือ CDXMidi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ขั้นตอนการติดตั้ง CDX และการปรับแต่ง Visual C++ ให้รองรับ CDX

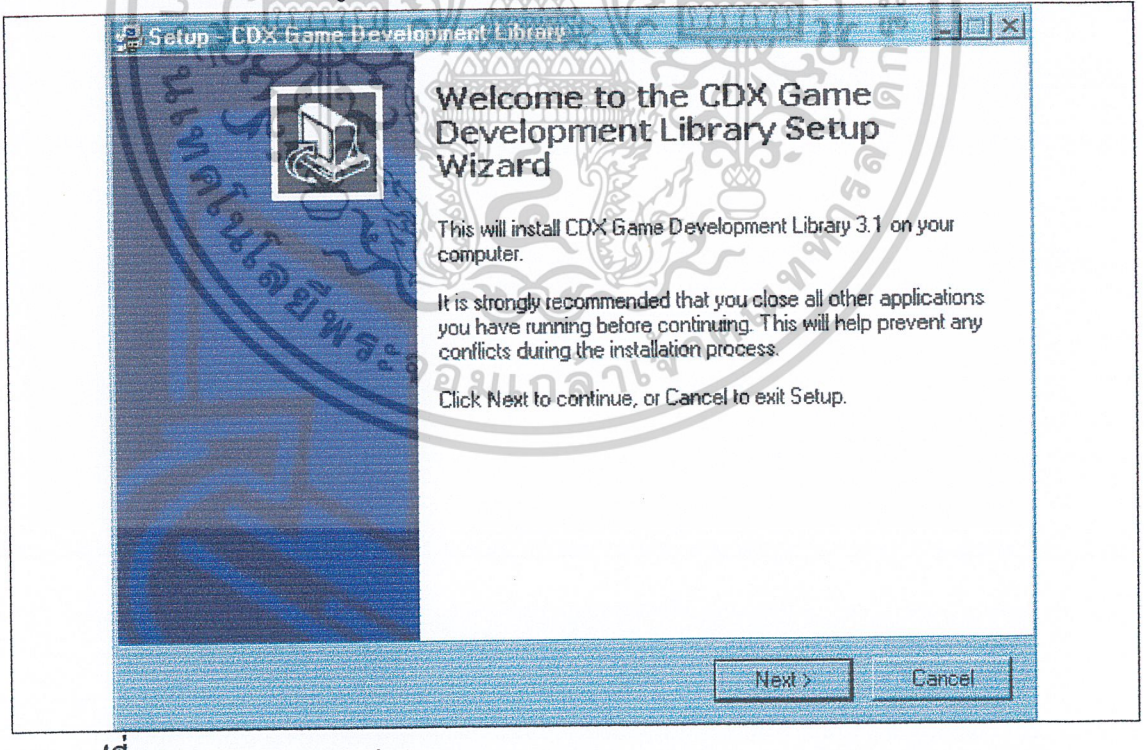
2.4.2.1 ขั้นตอนการติดตั้ง CDX การติดตั้ง CDX มีขั้นตอนดังต่อไปนี้

1. ในการติดตั้ง CDX ให้เลือกติดตั้งจากแผ่นซีดี โดยเข้าไปที่โฟลเดอร์ CDX



รูปที่ 2.12 CDX Source Folder

2. ดับเบิ้ลคลิกเลือกไฟล์ CDX-3.1.exe เพื่อเริ่มต้นทำการติดตั้ง CDX จะได้หน้าต่างดังรูปให้ทำการติดตั้งตามขั้นตอนไปเรื่อย ๆ จนเสร็จสิ้น



รูปที่ 2.13 CDX Setup Window

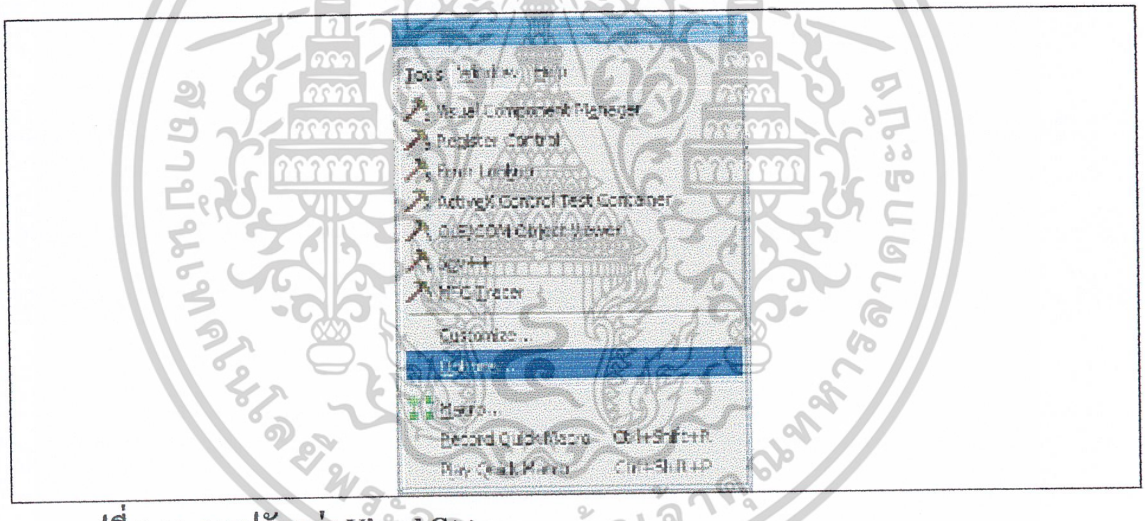
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นจะได้โฟลเดอร์ของ CDX อยู่ในตำแหน่งบน Storage ตามที่ได้ติดตั้งไป เมื่อเข้ามาในโฟลเดอร์ที่ได้ทำการติดตั้ง CDX ไว้ จะพบกับโฟลเดอร์ include ซึ่งจะมี Header File ที่จำเป็นของ CDX มากมาย และเราต้องทำให้ Visual C++ รู้จักกับ Header File เหล่านี้ ส่วนอีกโฟลเดอร์หนึ่งที่สำคัญและจำเป็นได้แก่โฟลเดอร์ lib ซึ่งจะมี Library ซึ่งเป็นตัวเชื่อมโยงชุดคำสั่งของ CDX และตัวเกมที่ได้พัฒนาไว้

2.4.2.2 ขั้นตอนการปรับแต่ง Visual C++ และติดตั้ง CDXAppWizard

หลังจากที่ได้ติดตั้ง CDX แล้ว สำหรับขั้นตอนของการพัฒนาเกม จำเป็นต้องปรับแต่ง Visual C++ และติดตั้ง CDXAppWizard ซึ่ง AppWizard นั้นก็คือการสร้าง Wizard หรือตัวช่วยในการสร้างโปรเจ็กต์ของ Visual C++ ซึ่งมีขั้นตอนต่อไปนี้

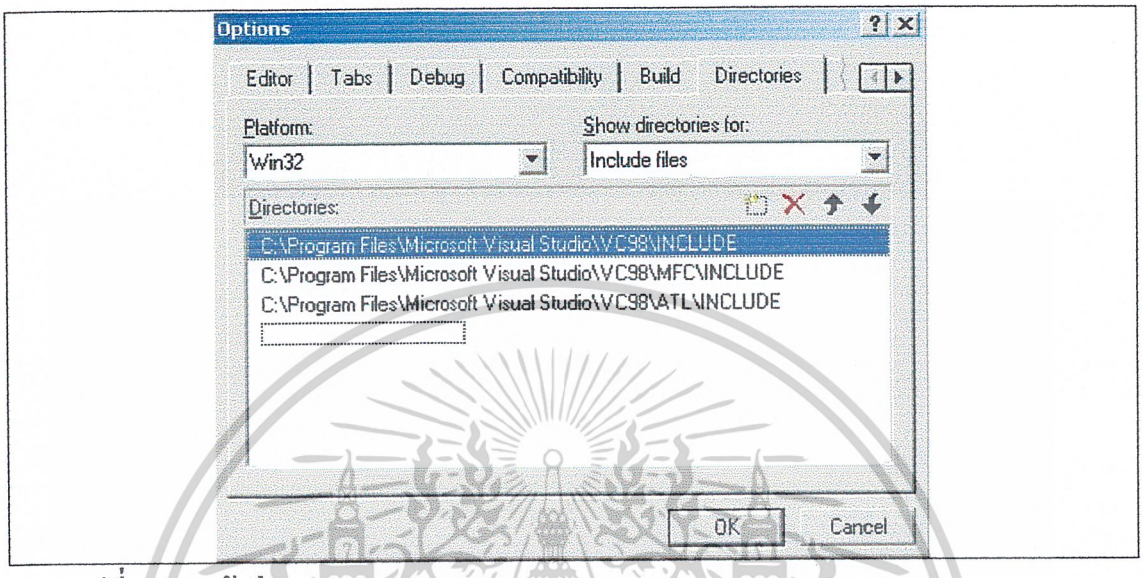
1. ในการปรับแต่ง Visual C++ นั้นให้ทำการเลือกไปที่เมนู Tools/Options บนเมนูบาร์ของ Visual C++



รูปที่ 2.14 การปรับแต่ง Visual C++

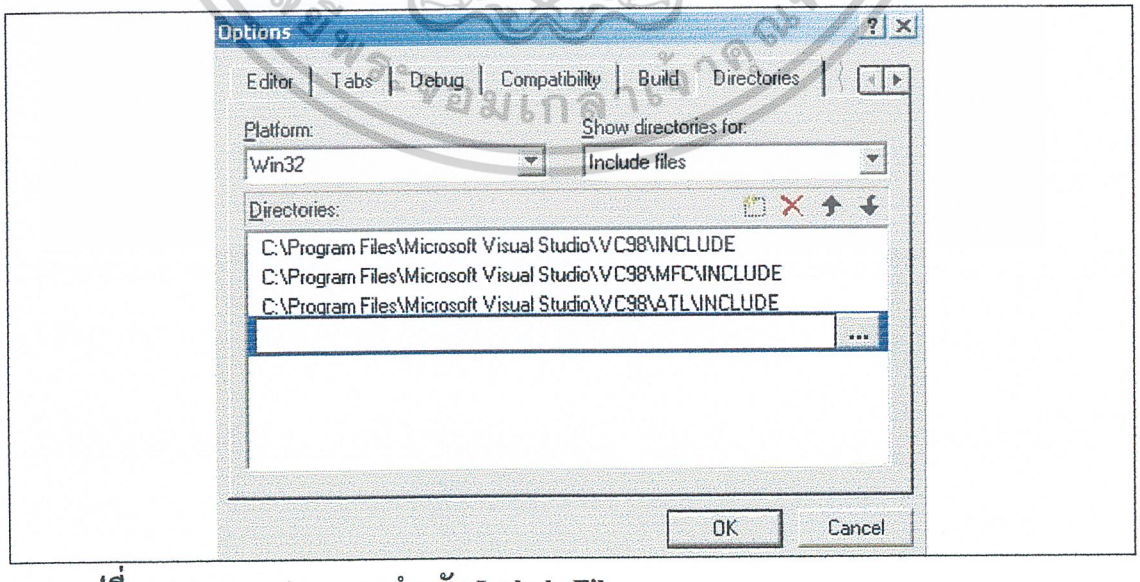
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าต่างของ Options จะปรากฏออกมา หลังจากนั้นให้เลือกที่แถบ Directories แล้วจะได้หน้าต่างดังรูป



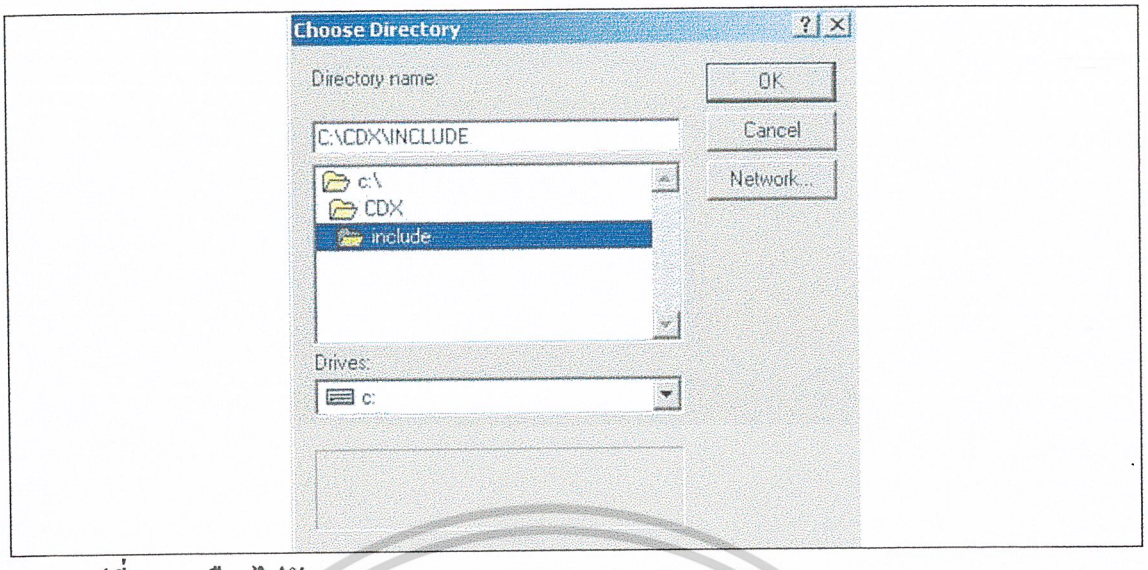
รูปที่ 2.15 หน้าต่าง Options แถบ Directories

3. ขั้นตอนนี้เราจะทำการให้ Visual C++ รู้จักกับ Directory ของ Include Files ของ CDX เราจึงจะเพิ่มที่อยู่ของ Directory นั้นเข้ามาใน Options ของ Visual C++ โดยทำการ Open Directory แล้วเลือกไปยัง Directory ของ Include Files ของ CDX ดังรูป



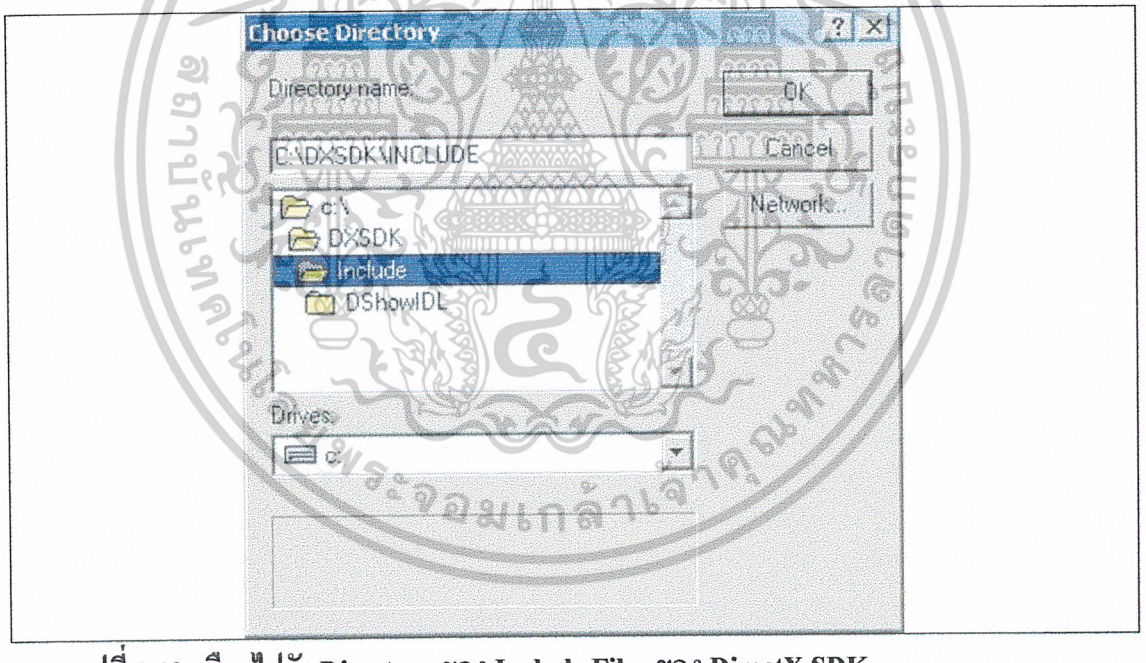
รูปที่ 2.16 Open Directory สำหรับ Include Files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 เลือกไปยัง Directory ของ Include Files ของ CDX

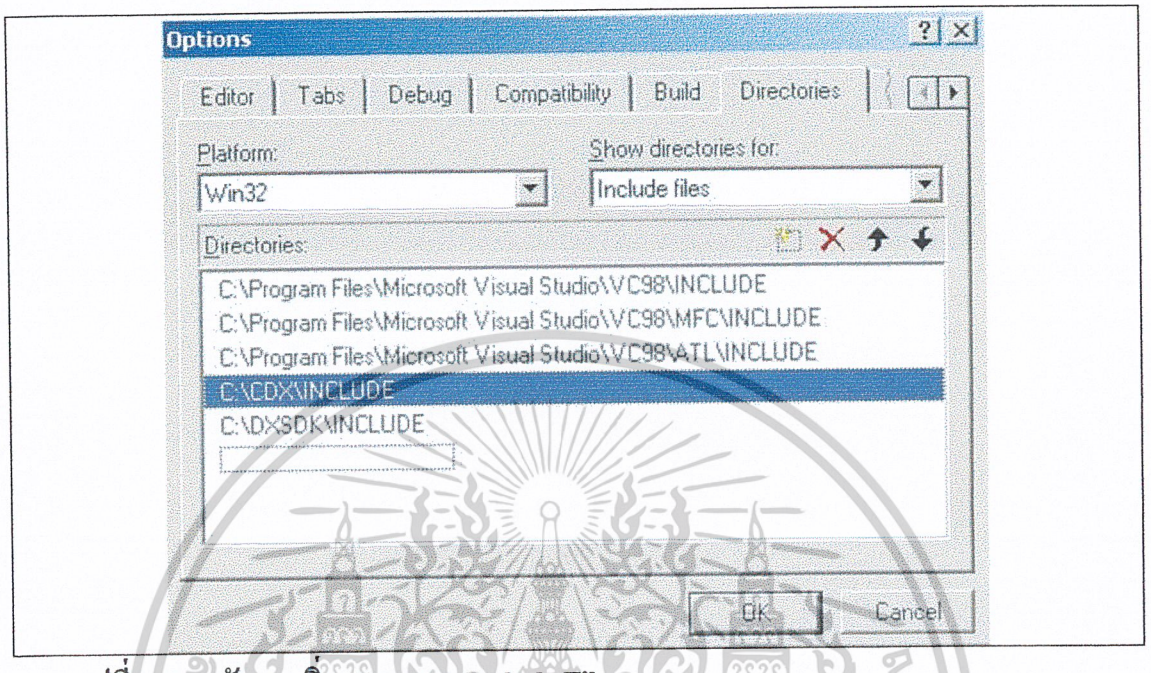
4. ทำการ Open Directory -> Directory Include Files ของ DirectX SDK ดังรูป



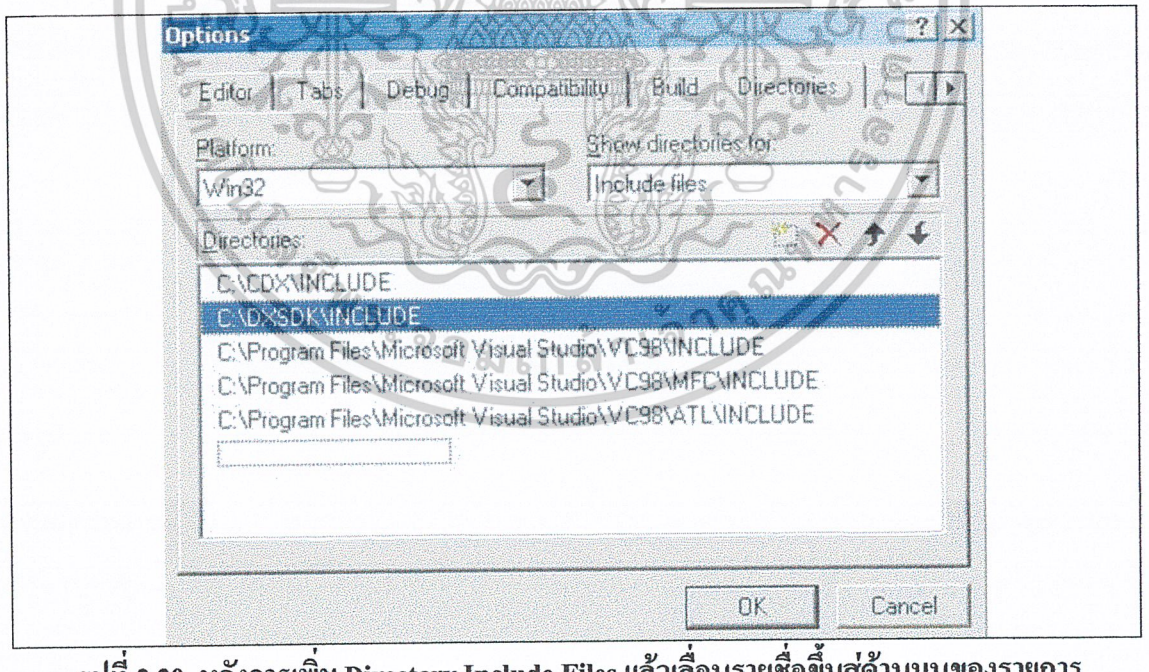
รูปที่ 2.18 เลือกไปยัง Directory ของ Include Files ของ DirectX SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จะได้รายชื่อ Directory ของ Include Files ของทั้ง DirectX SDK และ CDX จากนั้นให้ทำการเลื่อน รายชื่อ Directory ทั้งสองชื่อที่เพิ่มเข้ามาใหม่ขึ้นไปไว้ด้านบนสุดของรายการ ดังรูป



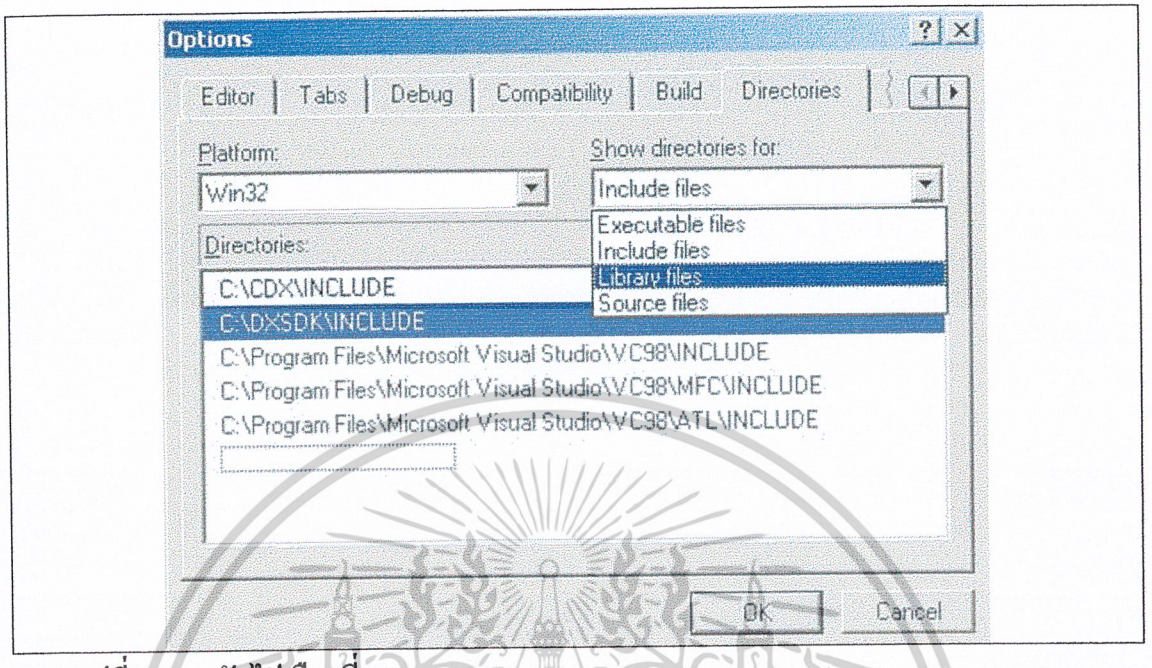
รูปที่ 2.19 หลังการเพิ่ม Directory Include Files



รูปที่ 2.20 หลังการเพิ่ม Directory Include Files แล้วเลื่อนรายชื่อขึ้นสู่ด้านบนของรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จากนั้นในหน้าต่างเดิมให้ทำการเปลี่ยนไปเลือกที่ Library Files เพื่อที่จะทำการเพิ่มรายชื่อของ Directory Library Files ของ CDX และ DirectX SDK

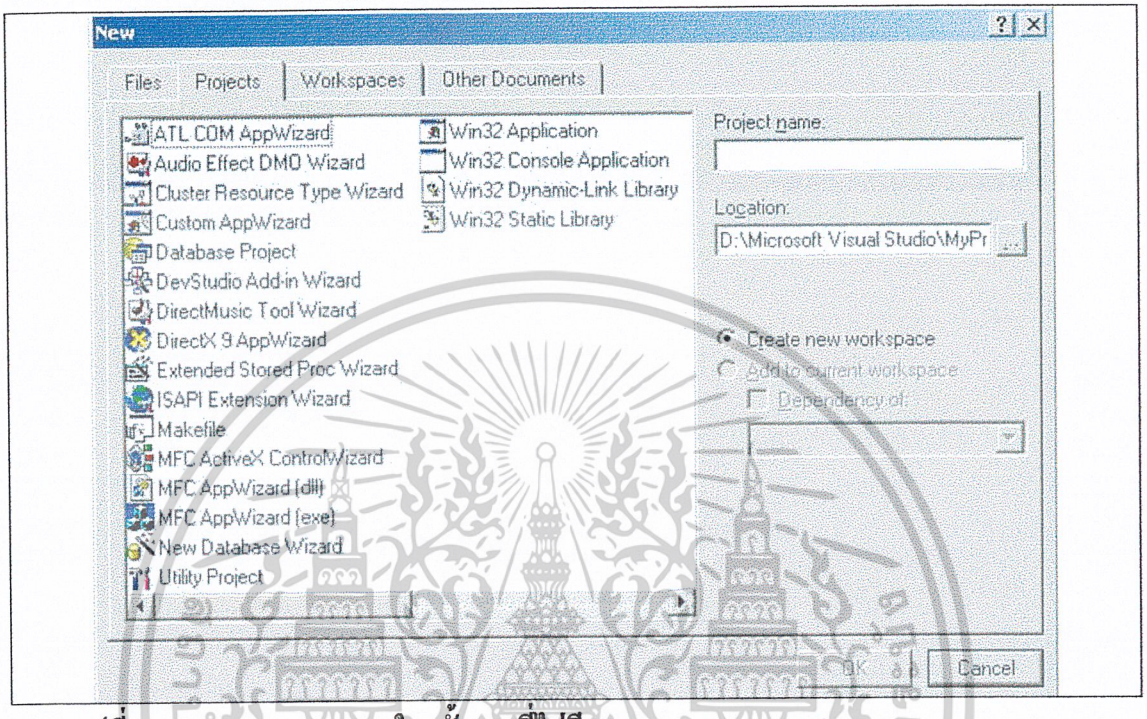


รูปที่ 2.21 สลับไปเลือกที่ Library Files

7. จากนั้นให้เลือกเพิ่ม Directory ของ Library Files ของทั้ง CDX และ DirectX SDK โดยใช้ขั้นตอนเดียวกันกับวิธีการเพิ่มรายชื่อ Directory ของทางด้าน Include Files โดยเปลี่ยน Directory จาก include เป็นการเลือก lib แทนในโฟลเดอร์ของ CDX และ DirectX SDK ตามลำดับ หลังจากการเพิ่มเสร็จแล้วก็ให้ทำการเลื่อนรายชื่อของทั้งสองรายชื่อไปอยู่บนสุดของรายการ Library Files นั้น

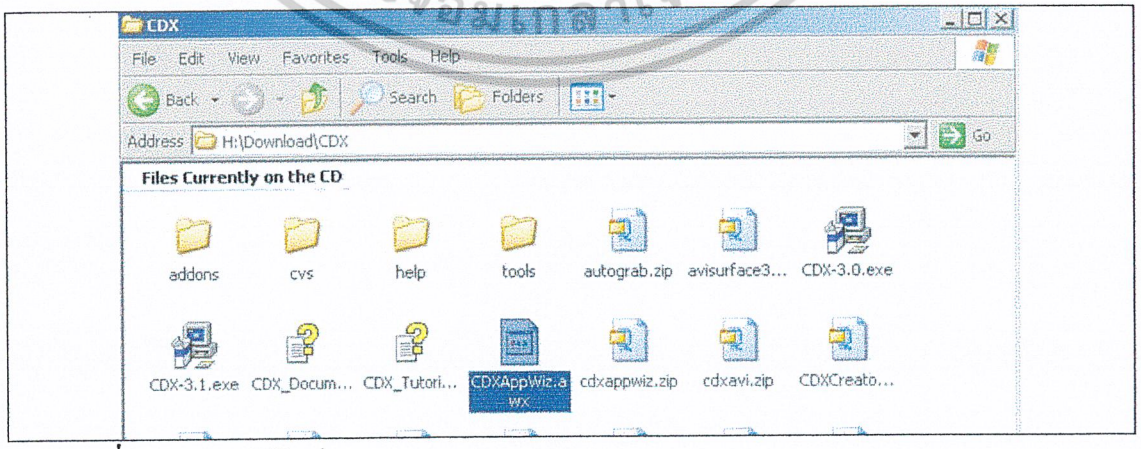
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ขั้นตอนถัดมาคือการสร้าง Application Wizard ไว้สำหรับเป็นตัวช่วยในการสร้างโปรเจ็กต์การพัฒนาเกม โดยใช้ CDX Game Engine ซึ่งจำเป็นต้องใช้ไฟล์ที่เกี่ยวข้องชื่อ CDXAppWiz.awx ในโฟลเดอร์ CDX ซึ่งถูกบรรจุมาในแผ่นซีดีแผ่นนี้แล้วด้วย ในครั้งแรก จะไม่มี CDXApplicationWizard เตรียมไว้ให้ เมื่อเราทำการ new Project ใน Visual C++



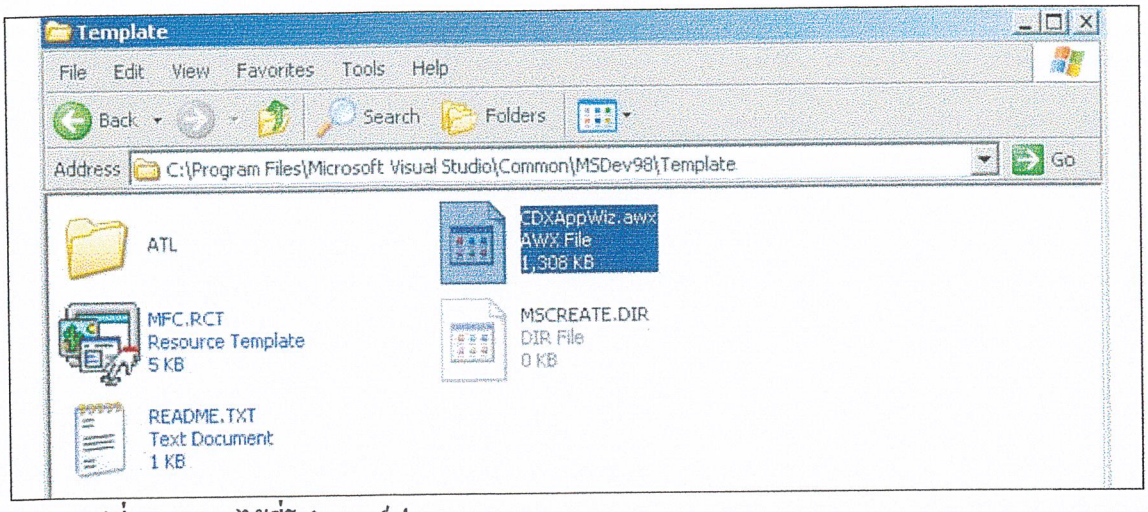
รูปที่ 2.22 การ new Project ในครั้งแรกที่ไม่มี CDXApplicationWizard

9. ทำการคัดลอกไฟล์ CDXAppWiz.awx จากโฟลเดอร์ต้นทางในแผ่นซีดี ไปวางไว้ในโฟลเดอร์ของ Microsoft Visual Studio\Common\MSDev98\Template ดังรูป



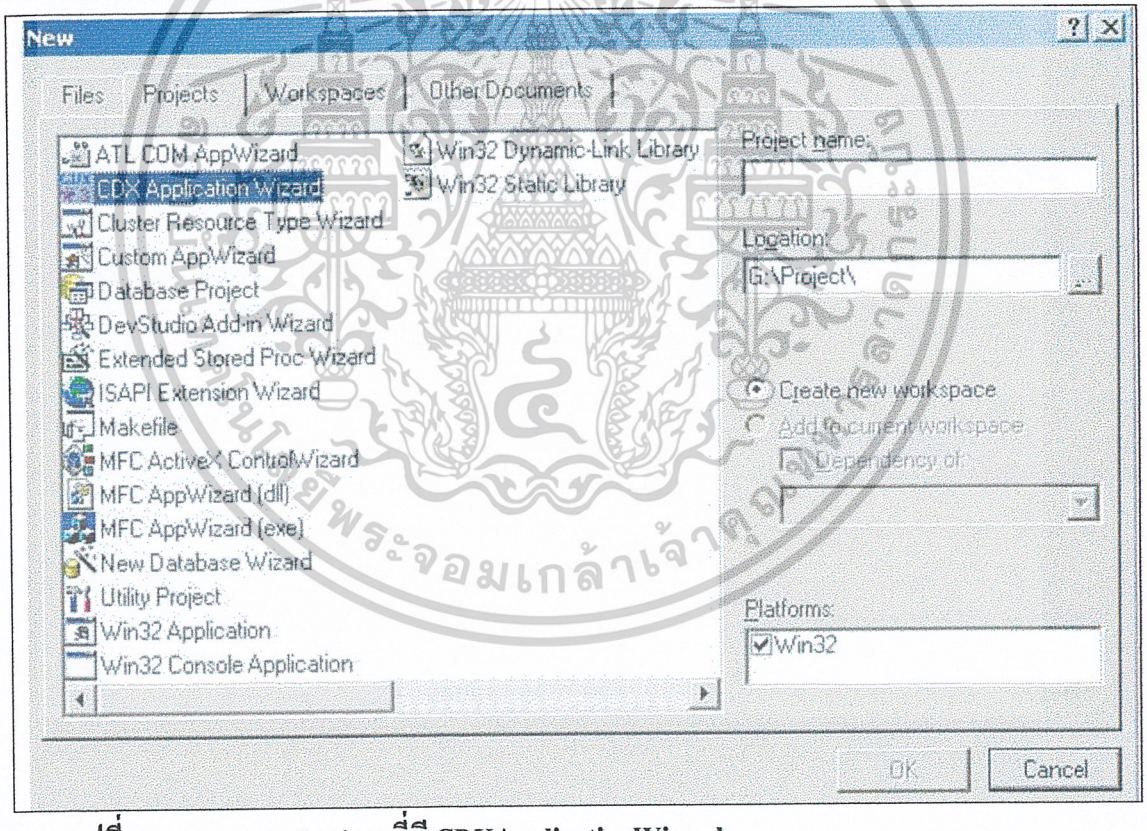
รูปที่ 2.23 คัดลอกไฟล์ CDXAppWiz.awx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 วางไว้ที่โฟลเดอร์ปลายทาง

10. หลังจากนั้นเมื่อเราทำการ new Project ใน Visual C++ อีกครั้ง เราก็จะพบ CDXApplicationWizard เพื่อช่วยในการเริ่มต้นสร้างโปรเจ็กต์การพัฒนาเกม



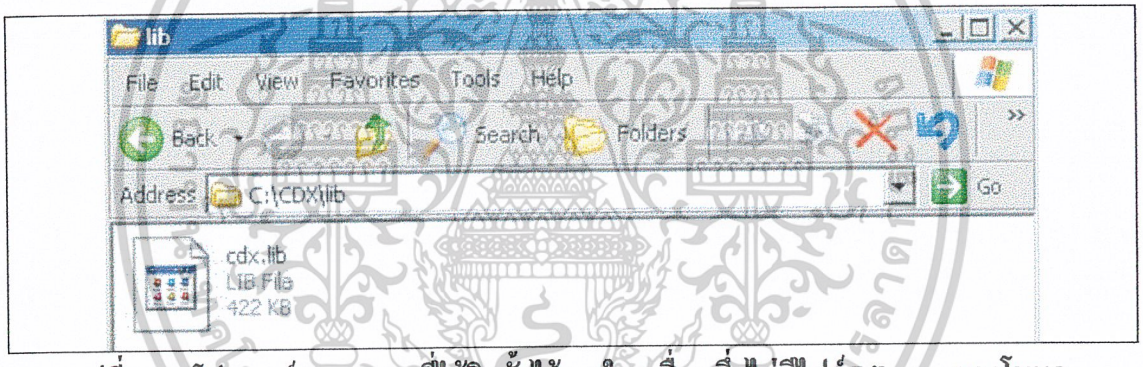
รูปที่ 2.25 การ new Project ที่มี CDXApplicationWizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.3 ข้อกำหนดในการใช้ CDX ร่วมกับ Visual C++

เนื่องจากในชุดคำสั่ง CDX มีเพียงโหมด Release และไม่มีโหมด Debug มาให้ ทำให้เมื่อทำการสร้างโปรเจ็กต์เกมด้วย CDX แล้วนั้นเมื่อเราทำการ Build Project นั้น ๆ แล้วอาจจะเกิดปัญหาได้ เราจึงต้องทำการ Compile ตัว CDX ใหม่อีกครั้งหนึ่งหลังจากการติดตั้ง CDX Game Engine ลงใน Hard Disk (โดยทำการ Compile เพียงครั้งเดียวหลังจากการติดตั้ง เมื่อ Compile แล้ว หากสร้างโปรเจ็กต์เกมด้วย CDX ไม่ว่ากี่โปรเจ็กต์ก็ตาม ก็สามารถ Build Project ได้โดยไม่ต้องกลับมา Compile ตัว CDX อีกครั้ง) โดยขั้นตอนของการ Compile CDX ใหม่ เพื่อสร้าง Debug Library มีขั้นตอนดังต่อไปนี้

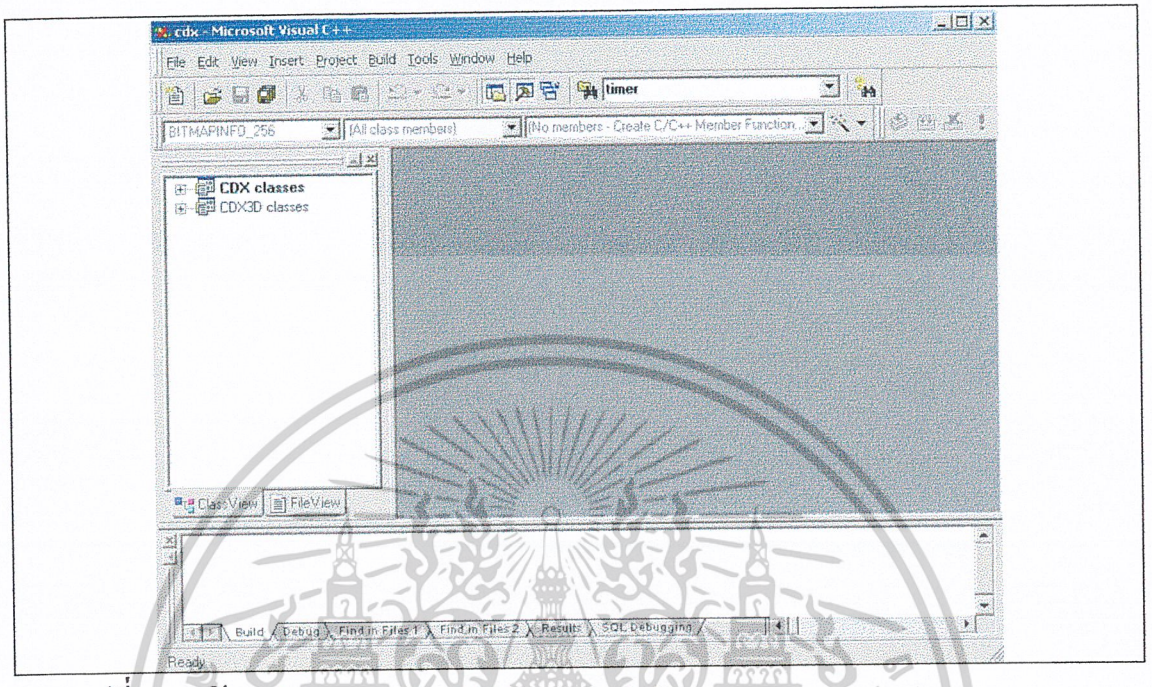
1. จากโฟลเดอร์ของ CDX ที่ได้ทำการติดตั้งไว้ จะเห็นได้ว่าในโฟลเดอร์ lib จะมีเพียง cdx.lib ซึ่งเป็น Library ของ Release แต่เมื่อเราสร้างโค๊ดโปรแกรมโดยใช้ CDXApplicationWizard โปรแกรมจะมีการวิ่งหา cdxd.lib ด้วย เพราะโปรแกรมต้องการหาข้อมูล Information ของการ Debug



รูปที่ 2.26 โฟลเดอร์ของ CDX ที่ได้ติดตั้งไว้ภายในเครื่อง ซึ่งไม่มีไฟล์ Library ของโหมด Debug

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการ Compile ชุดคำสั่งใหม่ทั้งหมดเพื่อจะได้มีโหมด Debug โดยการเปิด Workspace ของ CDX ใน Visual C++ (ไฟล์ cdx.dsw ภายในโฟลเดอร์ src ที่อยู่ภายในโฟลเดอร์ของ CDX ที่ได้ทำการติดตั้งไว้)



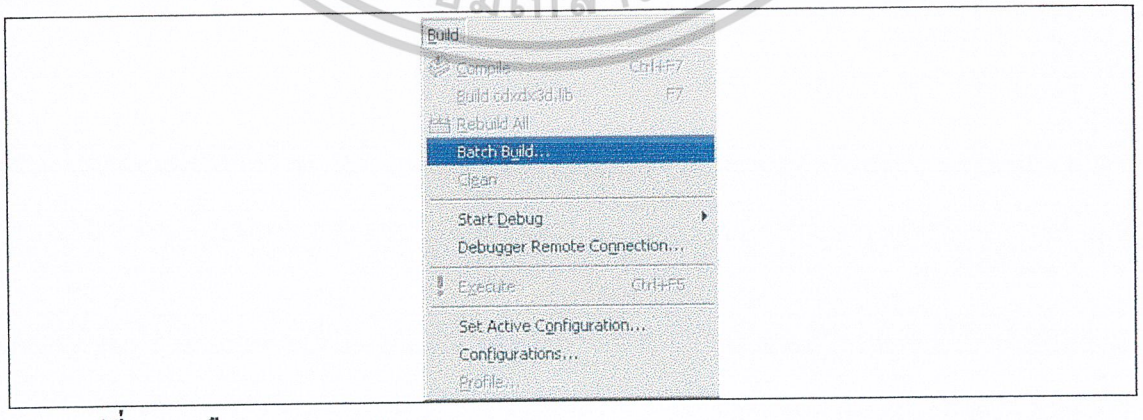
รูปที่ 2.27 เปิด Workspace ของ CDX (src\cdx.dsw)

3. เลือกเมนู build บนแถบเมนูบาร์แล้วเลือกที่ Batch Build จะมีหน้าต่างใหม่เปิดขึ้นมา ให้ทำการเลือกตัวเลือกเพียงแค่ 2 หัวข้อเท่านั้นได้แก่

1) CDX – Win32 Release

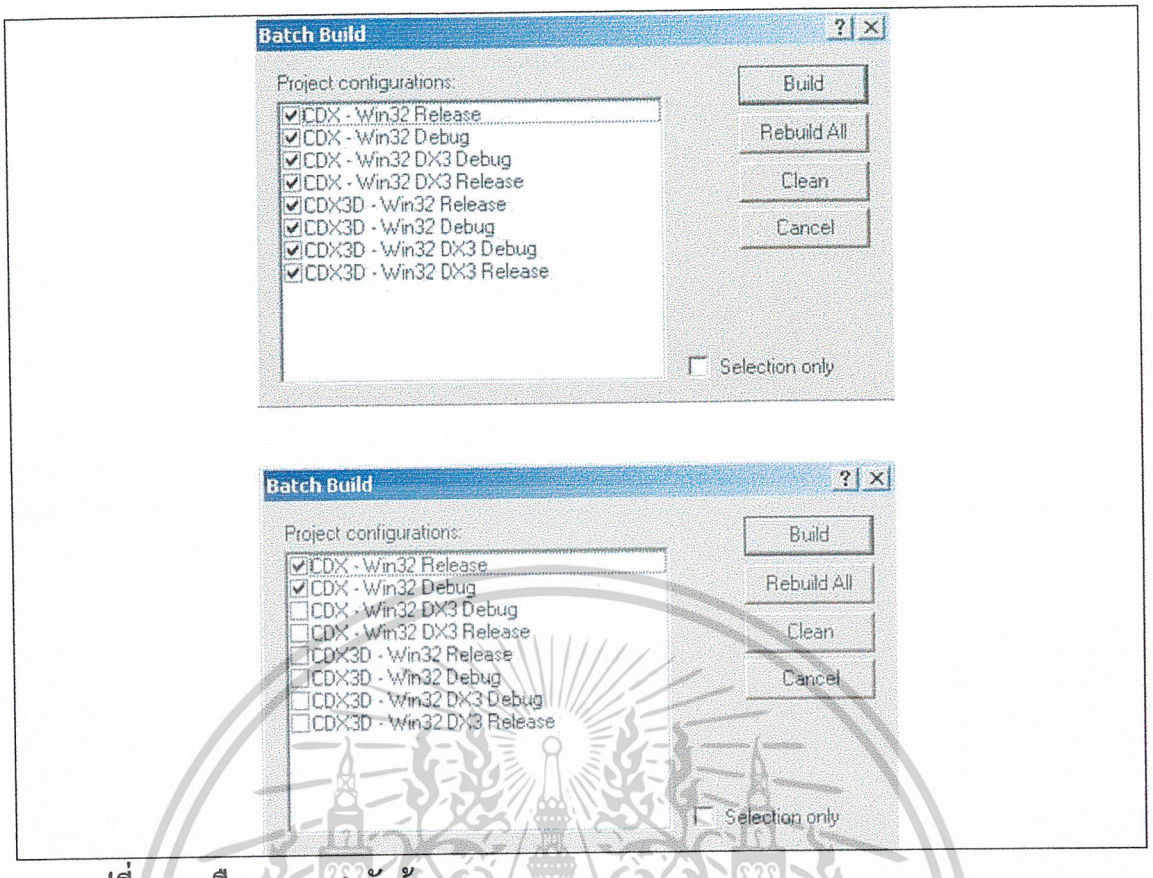
2) CDX – Win32 Debug

แล้วทำการกดปุ่ม Build ดังรูป



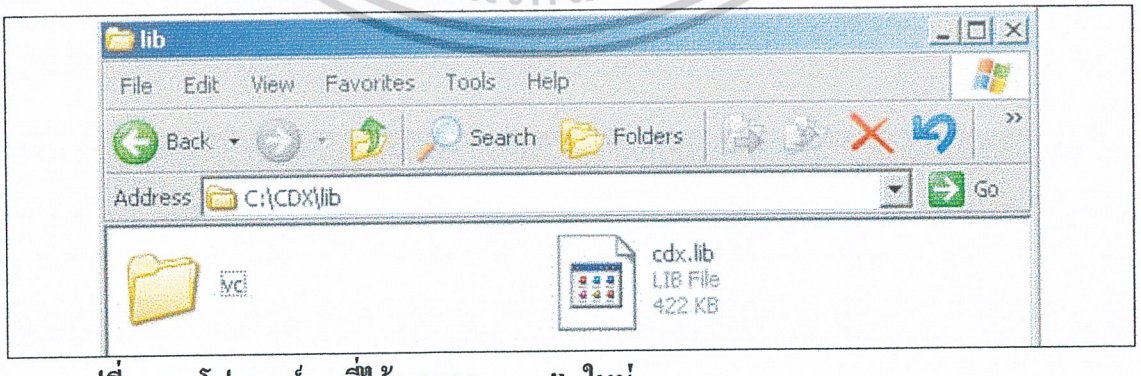
รูปที่ 2.28 เลือกเมนู Batch Build

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



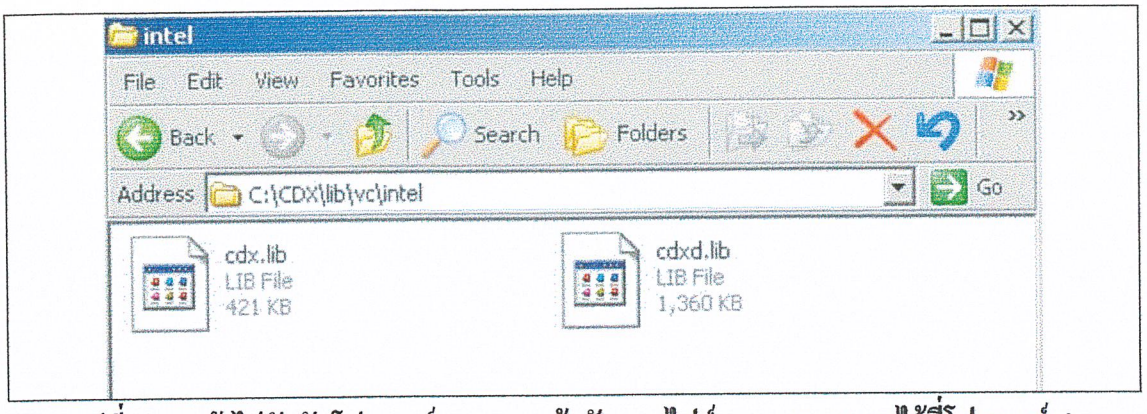
รูปที่ 2.29 เลือกเฉพาะ 2 หัวข้อแรก

4. เมื่อทำการ compile เสร็จเรียบร้อยแล้วให้กลับไปดูที่โฟลเดอร์ที่ติดตั้ง CDX ไว้ ในโฟลเดอร์ lib จะพบโฟลเดอร์ใหม่ขึ้นมาหนึ่งโฟลเดอร์ชื่อ vc เมื่อเข้าไปยังโฟลเดอร์ vc\intel จะพบไฟล์ Library 2 ไฟล์ที่ได้มาจากการ compile นั่นก็คือ Library ที่จะใช้ในการลิงค์แบบ Debug (cdxd.lib) และแบบ Release (cdx.lib) โดยเราจะทำการคัดลอกไฟล์ cdxd.lib ที่เราต้องการ มาวางไว้นอกซับโฟลเดอร์ vc\intel ที่โฟลเดอร์ lib

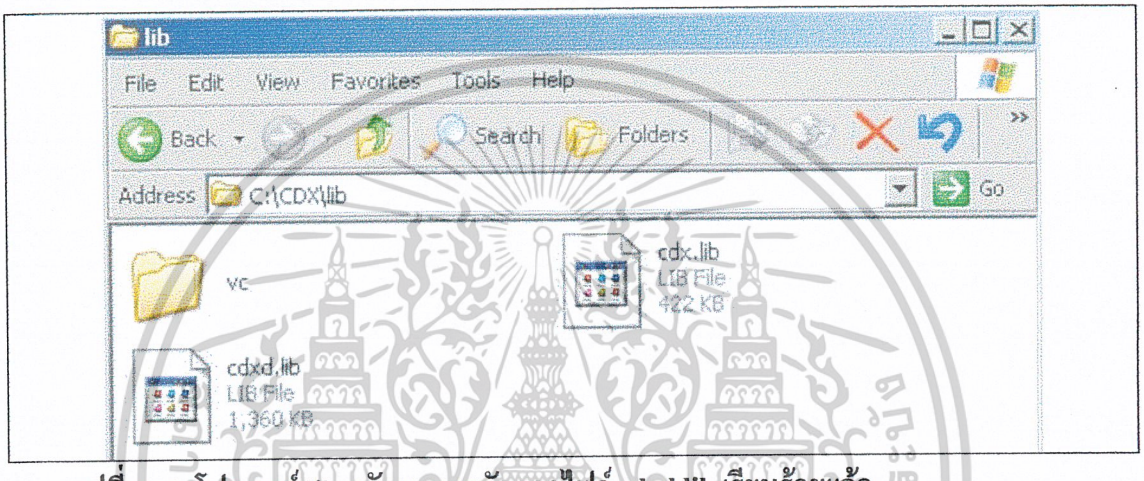


รูปที่ 2.30 โฟลเดอร์ vc ที่ได้จากการ compile ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 เข้าไปยังซับโฟลเดอร์ vc\intel แล้วคัดลอกไฟล์ cdxd.lib มาวางไว้ที่โฟลเดอร์ lib



รูปที่ 2.32 โฟลเดอร์ lib หลังจากการคัดลอกไฟล์ cdxd.lib เรียบร้อยแล้ว

5. หลังจากนั้นก็สามารถสร้างโปรเจ็กต์เกม ด้วย CDXApplicationWizard ใน Visual C++ ได้ตามปรกติ

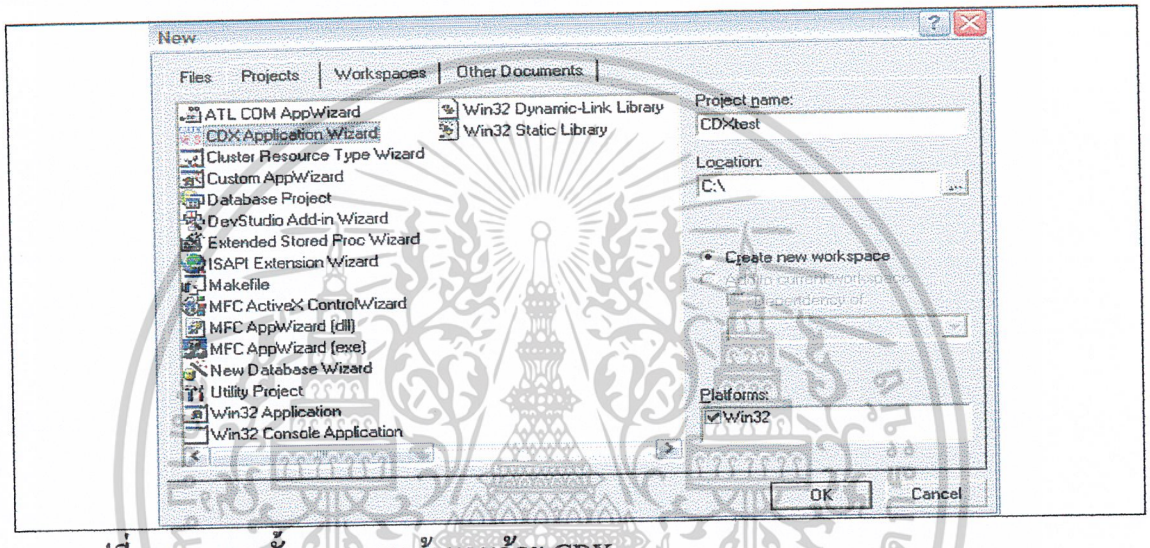
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ขั้นตอนการสร้างเกมด้วย CDX

ทดลองสร้างเกมด้วย CDX AppWizard ตามขั้นตอนต่อไปนี้

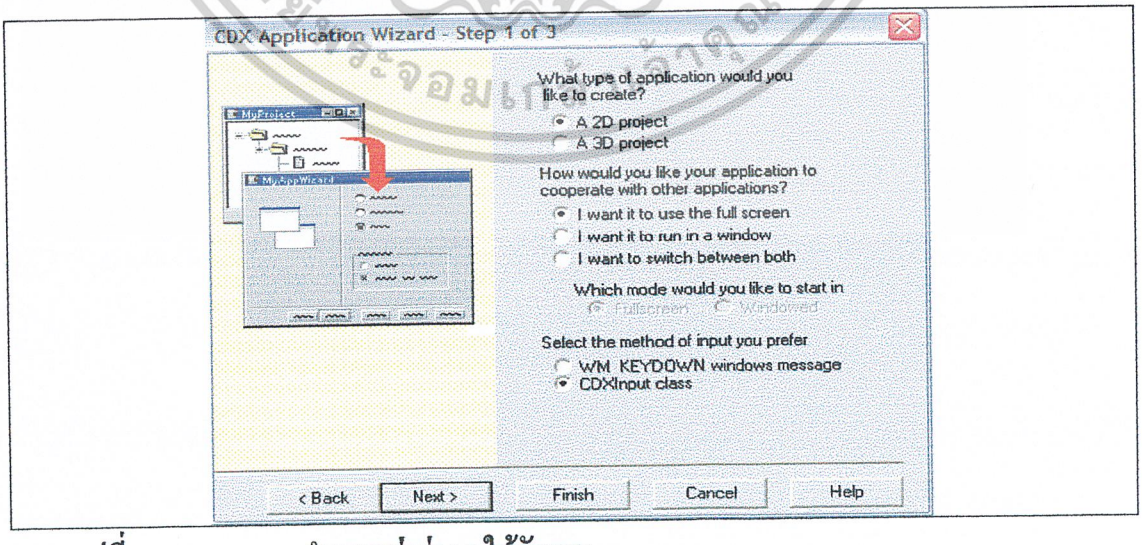
ขั้นตอนที่ 1 เริ่มใช้งาน CDX AppWizard โดยเรียกใช้งานผ่าน Visual C++ ดังนี้

1. คลิกเมนู File > New
2. จะปรากฏไดอะล็อก New ให้เลือกรูปแบบของโปรเจ็กต์โดยคลิกที่ CDX Application Wizard
3. กำหนดชื่อ Project Name ว่า CDXtest
4. คลิกปุ่ม OK



รูปที่ 2.33 แสดงขั้นตอนการสร้างเกมด้วย CDX

5. ในไดอะล็อก CDX Application Wizard ให้กำหนดค่าดังรูป



รูปที่ 2.34 แสดงการกำหนดค่าต่าง ๆ ให้กับเกม

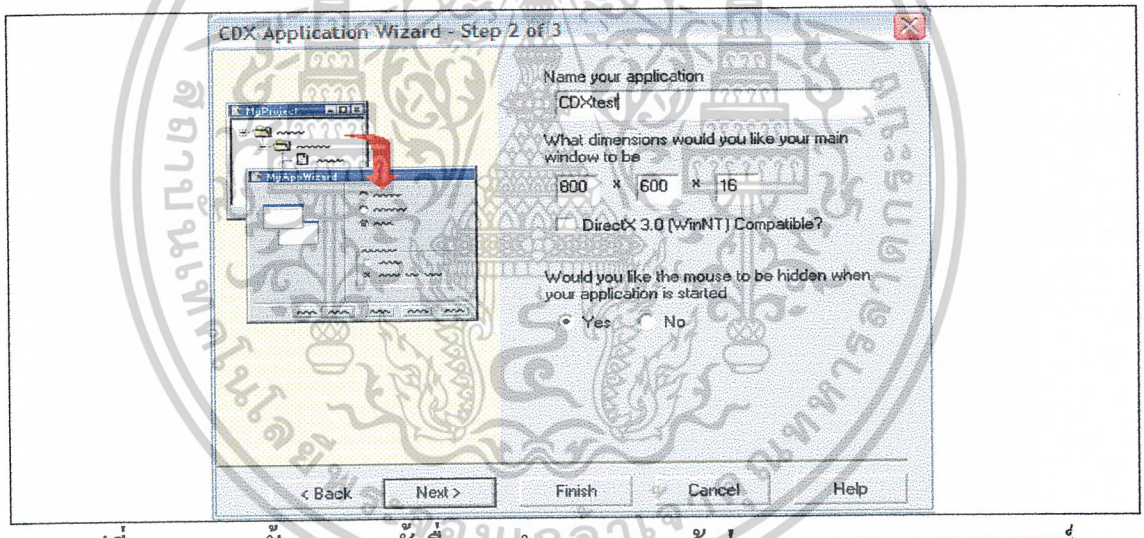
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเกมที่จะสร้างนี้ จะสร้างเป็นแบบ 2 มิติ และให้เกมแสดงผลแบบเต็มหน้าจอ และใช้ CDX Input รับค่าจากคีย์บอร์ด เนื่องจาก WM KEYDOWN windows message จะช้า

6. จากนั้นคลิกปุ่ม Next > เพื่อไปสู่ขั้นตอนต่อไป

ขั้นตอนที่ 2 ในไดอะล็อกขั้นตอนที่ 2 ของ CDX Application Wizard จะเป็นการกำหนดขนาดของหน้าต่างเกม ซึ่งมีขั้นตอนดังต่อไปนี้

1. ตั้งชื่อเกม
2. กำหนดขนาดหน้าต่างเกม ในที่นี้ให้อยู่ที่ 800x600 ในโหมด 16 บิต (65,536 สี)
3. ไม่ต้องเลือกที่ช่อง DirectX 3.0 เพราะจะมี error เกิดขึ้นถ้าในคอมพิวเตอร์ของเราไม่ได้ติดตั้ง DirectX 3.0 หรือมีการติดตั้งเวอร์ชันอื่นไว้
4. ถ้าไม่ต้องการให้มีการแสดงเมาส์ในเกมให้เลือก Yes แต่ถ้าต้องการให้เลือก No
5. คลิก Next> เพื่อเข้าสู่ขั้นตอนต่อไป

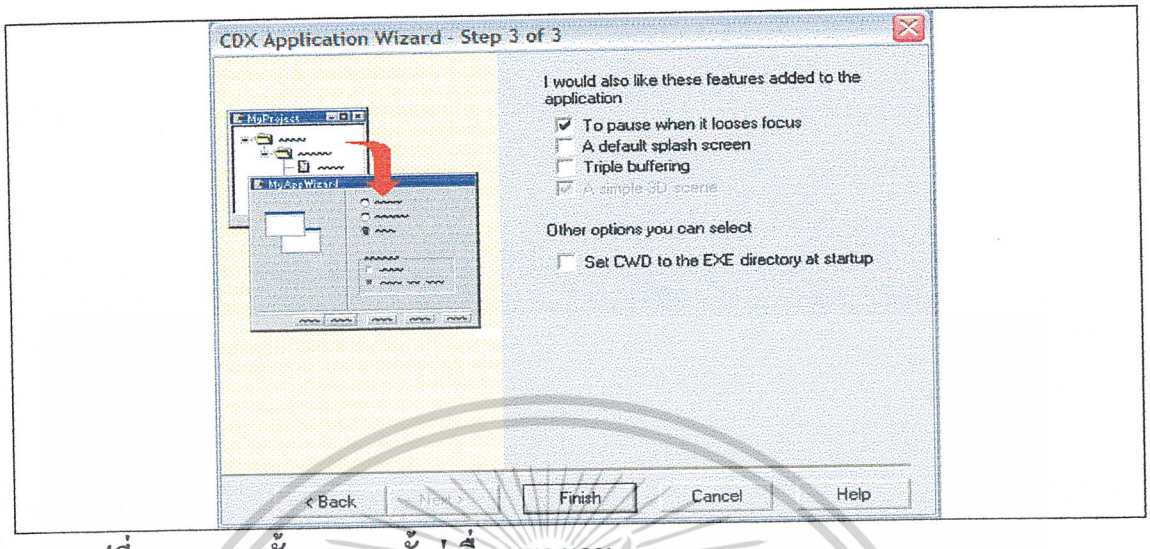


รูปที่ 2.35 แสดงขั้นตอนการตั้งชื่อเกม, กำหนดขนาดหน้าต่างของเกม และ การแสดงเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

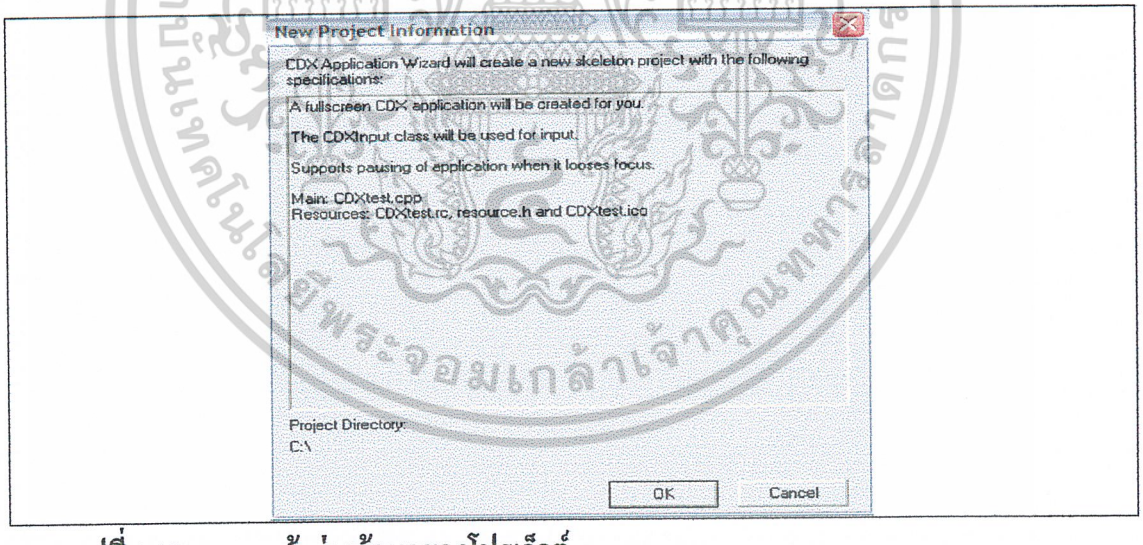
ขั้นตอนที่ 3 เป็นการกำหนดค่าอื่นๆให้กับโปรเจ็กต์ มีขั้น ตอนดังนี้

1. คลิกที่ตัวเลือกที่ต้องการ



รูปที่ 2.36 แสดงขั้นตอนการตั้งค่าอื่น ๆ ของเกม

2. จากนั้นคลิกปุ่ม Finish
3. หน้าต่างข้อมูลต่าง ๆ ของ โปรเจ็กต์จะปรากฏขึ้นมา ให้คลิก OK



รูปที่ 2.37 แสดงหน้าต่างข้อมูลของโปรเจ็กต์

4. เมื่อถึงขั้นตอนนี้แล้ว โปรเจ็กต์ CDXtest จะถูกสร้างขึ้นมา เราสามารถ Build โปรเจ็กต์นี้ได้เลยโดยการกด F5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ตัวอย่างโค้ดโปรแกรมที่สร้างขึ้นมาจาก CDX Application Wizard

พิจารณาดูรายละเอียด และหน้าที่การทำงานของโปรแกรมนี้ โดยจะดูทีละส่วน ๆ ดังนี้

```
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#define CDXINCLUDEALL
#include <cdx.h>
#include "resource.h"
char      szAppName[]    = "CDXtest";
char      szClassName[]  = "CDXtestWndClass";

HINSTANCE   g_hInst;                          // instance handle
HWND        g_hWnd;                                   // window handle
BOOL        g_IsAppActive;                    // is the app active

CDXScreen   * Screen    = 0;    // The screen object, every program must have one
CDXInput    * Input     = 0;    // The input object
```

ในบรรทัดที่เขียนว่า CDXINCLUDEALL ก็คือ include ไฟล์ที่เกี่ยวข้องกับ CDX ทั้งหมดจากนั้นเป็นการประกาศตัวแปรที่ใช้ในการสร้างเกม ซึ่งโปรแกรมนี้ จะใช้คลาสของ CDX 3 คลาส คือ

- CDXScreen ใช้ในการสร้างหน้าจอเกม กำหนดความละเอียดของจอภาพ และจำนวนบิตสี
- CDXSurface ใช้โหลดรูปภาพ SPLASH จาก Resource
- CDXInput สร้าง DirectInput เพื่อรับค่าจากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโปรแกรมถูกสั่งให้ทำงาน ฟังก์ชัน WinMain จะถูกเรียก และภายใน WinMain จะมีการเรียกคำสั่งต่าง ๆ ต่อไปนี้

```
int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst, LPSTR lpCmdLine,
int nCmdShow)
{
        MSG msg;

        // save the app instance
        g_hInst = hInstance;


        if(!InitApp(nCmdShow))
    CDXError( NULL , "could not initialize application" );

        if(!cdx_Init())
        {
                PostQuitMessage(0);
                return FALSE;
        }

        while(1)
        {
                if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE))
                {
                        if(!GetMessage(&msg, NULL, 0, 0 )) return msg.wParam;
                        TranslateMessage(&msg);
                        DispatchMessage(&msg);
                }
                else if(g_IsAppActive)
                {
                        cdx_DoFrame();
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else
            {
                WaitMessage();
            }
        }
}
```

ฟังก์ชัน WinMain เป็นฟังก์ชันเริ่มต้นที่ใช้ในการสร้างหน้าต่าง และกำหนดค่าให้กับโปรแกรม โดยมีการเรียกฟังก์ชัน InitApp ซึ่งใช้ในการสร้างหน้าต่าง และจากภายในฟังก์ชัน InitApp ก็จะเรียกฟังก์ชันAdJustWinStyle เพื่อกำหนดรูปแบบของหน้าต่าง จากนั้นก็จะกลับมาเรียกฟังก์ชัน cdx_Init เพื่อทำการสร้างสภาวะแวดล้อมของเกม แบบ DirectX ก็คือปรับความละเอียดของจอภาพนั่นเอง ซึ่ง cdx_init นั้น เป็นฟังก์ชันที่ถูกสร้างขึ้นมาใหม่โดย CDX Application Wizard ภายในฟังก์ชันนี้มีคำสั่ง CDX ที่ใช้ในการปรับความละเอียดการแสดงผลดังนี้

```
BOOL cdx_Init()
{
    // Create the CDXSreen object
    Screen = new CDXScreen();
    if (Screen==NULL)
  CDXError( NULL , "Could not create CDXScreen object" );
    // start app fullscreen
    if( Screen->CheckIfVideoModeExists( 800 , 600 , 16 ) == TRUE )
    {
    if(FAILED(Screen->CreateFullScreen(g_hWnd, 800, 600, 16)))
        CDXError( Screen , "Could not set 800x600x16 video mode" );
    }
    else
    {
    if(FAILED(Screen->CreateFullScreen(g_hWnd, 800, 600, 15)))
        CDXError( Screen , "Could not set 800x600x15 video mode" );
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // create our input object
    Input = new CDXInput( );
    if( Input == NULL )
CDXError( Screen , "Could not create CDXInput object" );

    // Create input devices
    if(FAILED(Input->Create( g_hInst , g_hWnd )))
CDXError( Screen , "Could not create direct input object" );

    // TODO: Initialize your own CDX objects here
    return TRUE;
}
```

จากโปรแกรม จะเห็นได้ว่า มีการสร้างหน้าต่างเกมขนาด 800x600 โดยใช้คำสั่ง CreateWindowed ซึ่งมีการตรวจสอบโดยใช้คำสั่ง if เพื่อดูว่าการสร้างหน้าต่างเกมสำเร็จหรือไม่ และต่อมาก็มีการสร้าง DirectInput โดยใช้ CDXInput ตัวแปรที่ใช้ในฟังก์ชันนี้ ก็คือตัวแปรที่ได้มีการประกาศเอาไว้ตอนต้นของโปรแกรมนั่นเอง จะเห็นได้ว่า ในฟังก์ชัน cdx_Init นี้ จะเป็นคำสั่งที่ใช้เข้าสู่โหมด DirectX ของเกม

เมื่อ cdx_Init ทำงานเสร็จแล้ว ก็จะส่งการทำงานกลับไปสู่ WinMain ซึ่งจะเห็นว่า ใน WinMain ก็ได้มีการตรวจสอบการทำงานของ cdx_Init เหมือนกัน ดังนี้

```
if(!cdx_Init())
{
    PostQuitMessage(0);
    return FALSE;
}
```

จากตัวอย่างในข้างต้นนี้ ถ้า cdx_Init ทำงานไม่สำเร็จ เช่น สร้าง DirectInput ไม่ได้ หรือปรับหน้าจอให้มีขนาด 800x600 ไม่ได้ ก็จะคืนค่า False กลับมา และจบโปรแกรม แต่ถ้าทำงานสำเร็จ ก็จะเข้าสู่ส่วนต่อไป นั่นก็คือ วนรอบเพื่อตรวจจับเมสเสจในวินโดวส์

```
while(1)
{
    if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE))
    {
        if(!GetMessage(&msg, NULL, 0, 0 )) return msg.wParam;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            TranslateMessage(&msg);
            DispatchMessage(&msg);
    }
    else
    {
            cdx_DoFrame();
    }
}
```

ในการวนรอบนี้ จะใช้คำสั่ง while ในการทำซ้ำ โดยจะตรวจสอบว่ามีเมสเสจใดในระบบส่งมา ซึ่งในการตรวจจับเมสเสจนี้ จะทำงานร่วมกับฟังก์ชัน WinProc คือ ถ้ามีเมสเสจใด ๆ ส่งมา เราจะเขียนโค้ดเพื่อตรวจจับ และสั่งให้ทำงานในส่วนนี้

```
static long PASCAL WinProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    switch(message)
    {
        case WM_SETCURSOR:      SetCursor(NULL);
            return 1;
            case WM_CLOSE:    cdx_DeInit();
            case WM_DESTROY:      PostQuitMessage(0);
            return 0;
            default: return DefWindowProc(hWnd, message, wParam, lParam);
    }
}
```

ตัวอย่างเช่น ถ้ามีการปิดโปรแกรมเกม ก็จะมีเมสเสจ WM_CLOSE ส่งมา ก็จะส่งการทำงานต่อไปให้กับฟังก์ชัน cdx_DeInit เพื่อคืนค่าตัวแปรให้กับหน่วยความจำและจบโปรแกรม

```
void cdx_DeInit( void )
{
    SAFEDELETE( Splash );
    SAFEDELETE( Input );
    SAFEDELETE( Screen );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าในระหว่างที่โปรแกรมทำงาน ไม่มีเมสเสจส่งมา ก็จะเข้าสู่โหมดการทำงานของเกม นั่นก็คือในส่วนของคำสั่ง else ที่อยู่ภายใน while เราจึงเรียกฟังก์ชัน cdx_DoFrame ในส่วนนี้

```
while(1)
{
    if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE))
    {
            if(!GetMessage(&msg, NULL, 0, 0 )) return msg.wParam;
            TranslateMessage(&msg);
            DispatchMessage(&msg);
    }
    else
    {
            cdx_DoFrame();
    }
}
```

ฟังก์ชัน cdx_DoFrame เป็นฟังก์ชันที่มีความสำคัญมากตัวหนึ่ง เพราะว่าในระหว่างที่เกมกำลังดำเนินการนั้น สิ่งต่าง ๆ ภายในฟังก์ชัน cdx_DoFrame นี้จะถูกกระทำ เพราะฉะนั้นถ้าต้องการสร้างตัวละครให้เคลื่อนไหว สร้างฉากเลื่อน หรือเพิ่มค่าตำแหน่งของตัวละครใด ๆ ตรวจสอบเงื่อนไขใด ๆ ก็ตาม ที่เกี่ยวกับเกม จะใส่ลงไปในฟังก์ชัน cdx_DoFrame นี้ทั้งหมด

```
void cdx_DoFrame()
{
    Input->Update();
    if (Input->GetKeyState(CDXKEY_ESCAPE)==CDXKEY_PRESS)
    {
            SendMessage(g_hWnd, WM_CLOSE, 0, 0);
            return;
    }
        // put code here
        Screen->GetBack()->Fill(0);
    Splash->DrawBlk(Screen->GetBack(), 0, 0);
    Screen->Flip();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดโปรแกรมที่ CDX Application Wizard สร้างขึ้นมาให้ ภายในฟังก์ชัน cdx_DoFrame จะมีการตรวจจับการกดปุ่มคีย์บอร์ด ซึ่งจะเห็นว่า ได้เรียกฟังก์ชัน Input->Update() เพื่อดูว่ามีการกดปุ่มหรือไม่ ถ้ามีก็จะตรวจสอบว่า มีการกดปุ่มอะไร โดยใช้ฟังก์ชัน GetKeyState ตามด้วยค่าของปุ่ม ขึ้นต้นด้วย CDXKEY_XXXX ดังโค้ดต่อไปนี้

```
Input->Update();
if (Input->GetKeyState(CDXKEY_ESCAPE) = CDXKEY_PRESS)
{
    SendMessage(g_hWnd, WM_CLOSE, 0, 0);
    return;
}
```

ให้ if ตรวจสอบว่ามีการกดปุ่ม ESC หรือไม่ โดย CDXKEY_PRESS นั้นมีความหมายว่า ถ้ามีการกดปุ่มเพียงครั้งเดียว เพราะฉะนั้น ถ้ากดปุ่ม ESC ค้างเอาไว้มันก็จะถือว่ากดครั้งเดียวเท่านั้น ต่อจากนั้น ก็จะเป็นส่วนที่จะ ต้องเขียนโปรแกรมลงไป ในตอนกลางของฟังก์ชัน cdx_DoFrame เช่น การปรับตำแหน่งของตัวละคร , การตรวจ สอบเงื่อนไขของโปรแกรม และอื่น ๆ อีก และสุดท้ายก็จะทำการฟลิป (Flip) เพื่อสลับพื้นผิว ทำให้ตัวละครในเกมเกิดภาพเคลื่อนไหวนั่นเอง

และสุดท้ายเมื่อมีการกด ESC โปรแกรมก็จะSendMessage ของ WM_CLOSE เพื่อทำการออกจากโปรแกรม และถอนตัวออกจากหน่วยความจำของเครื่อง เพราะฉะนั้น ฟังก์ชัน cdx_DeInit จะถูกเรียกให้ทำงานเพื่อคืนตัวแปรทุกอย่างที่ใช้ในเกม ดังนี้

```
void cdx_DeInit( void )
{
    // TODO: Destroy your CDX objects here

    SAFEDELETE( Input );
    SAFEDELETE( Screen );
}
```

เพราะฉะนั้น เมื่อประกาศตัวแปรใด ๆ เอาไว้ จะต้องคืนหน่วยความจำด้วยทุกตัว โดยจะเขียนว่า SAFEDELETE(….) ในวงเล็บก็คือตัวแปรที่ประกาศเอาไว้นั่นเอง เช่นถ้าสร้างตัวละครด้วยตัวแปร x ก่อนที่จะรันโปรแกรม ก็จะต้องเขียน SAFEDELETE(x) ไว้ในฟังก์ชัน cdx_DeInit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

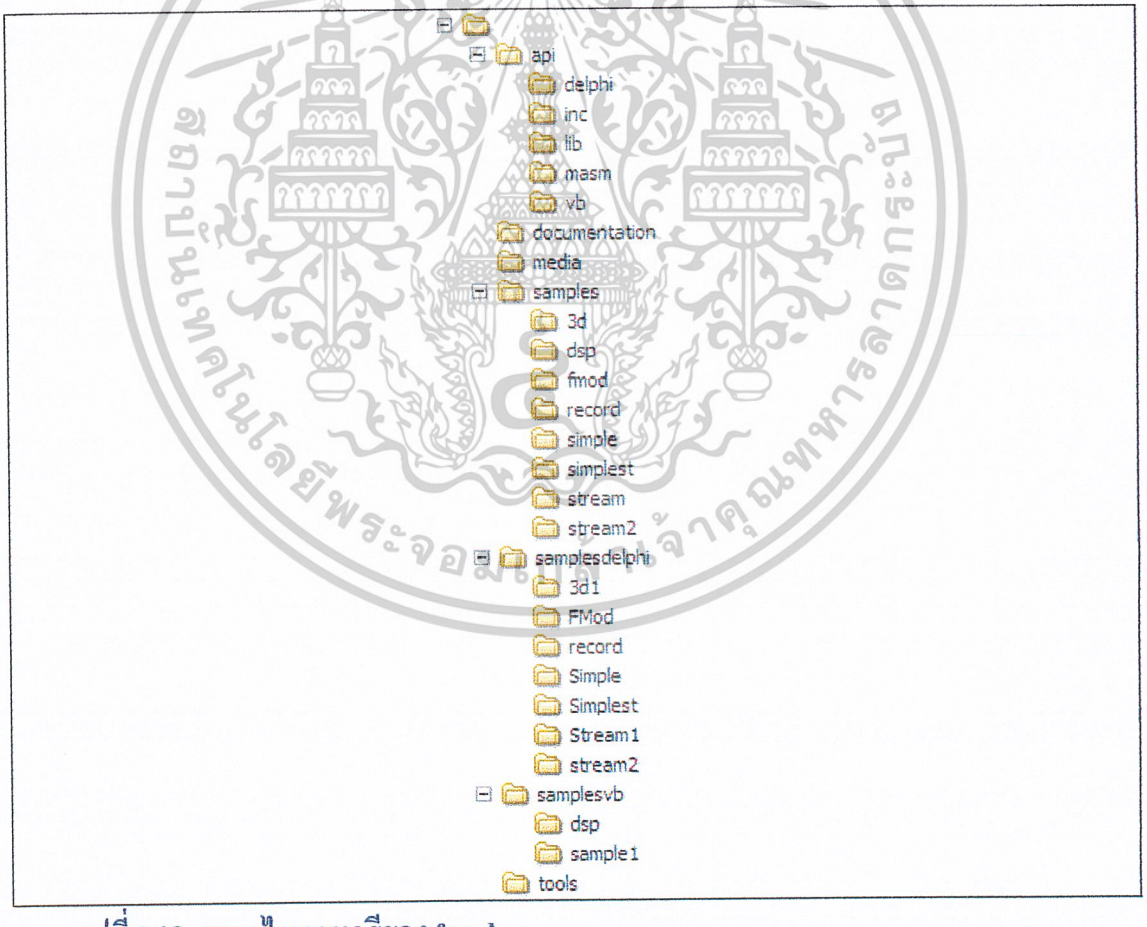
2.5 FMOD Sound System

เนื่องจากเกมกรู๊ฟอันลิมิเต็ด เป็นเกมที่การเรียกใช้ไฟล์เพลงหลายไฟล์ และหลายชนิด ทั้ง midi, wave และ mp3 ซึ่งตัว CDX Game Engine ที่ใช้ไม่สามารถรองรับการเล่นไฟล์ mp3 ได้ ผู้จัดทำจึงได้ใช้ FMOD Sound System แทน เพราะ FMOD สามารถรองรับไฟล์หลายประเภท เช่น 3d sound, midi, mods, mp3, ogg vorbis, wma, aiff, recording, obstruction/occlusion, cd playback (ทั้ง analog และ digital), cd ripping, mmx, internet streaming ฯลฯ นอกจากนี้ FMOD ยังสามารถใช้ได้บนระบบปฏิบัติการทุกระบบทั้ง Windows, Linux, Windows CE, Macintosh, GameCube, Play Station 2 และ Xbox การเล่นไฟล์เพลงหรือไฟล์เสียงต่าง ๆ FMOD ไม่ต้องเรียกผ่าน DirectSound ของ DirectX เลย FMOD จึงสามารถทำงานได้บนเครื่องเล่นใดก็ได้ที่มีการ์ดเสียงแบบ 16 Bit

2.5.1 วิธีการเรียกใช้ FMOD

เมื่อต้องการนำ FMOD เข้าไปใช้ในโปรเจ็กต์มีวิธีการดังนี้

1. ทำการเปิด fmodapi374win.zip และแตกไฟล์ออกมาจะได้ไดเรคทอรีดังรูป

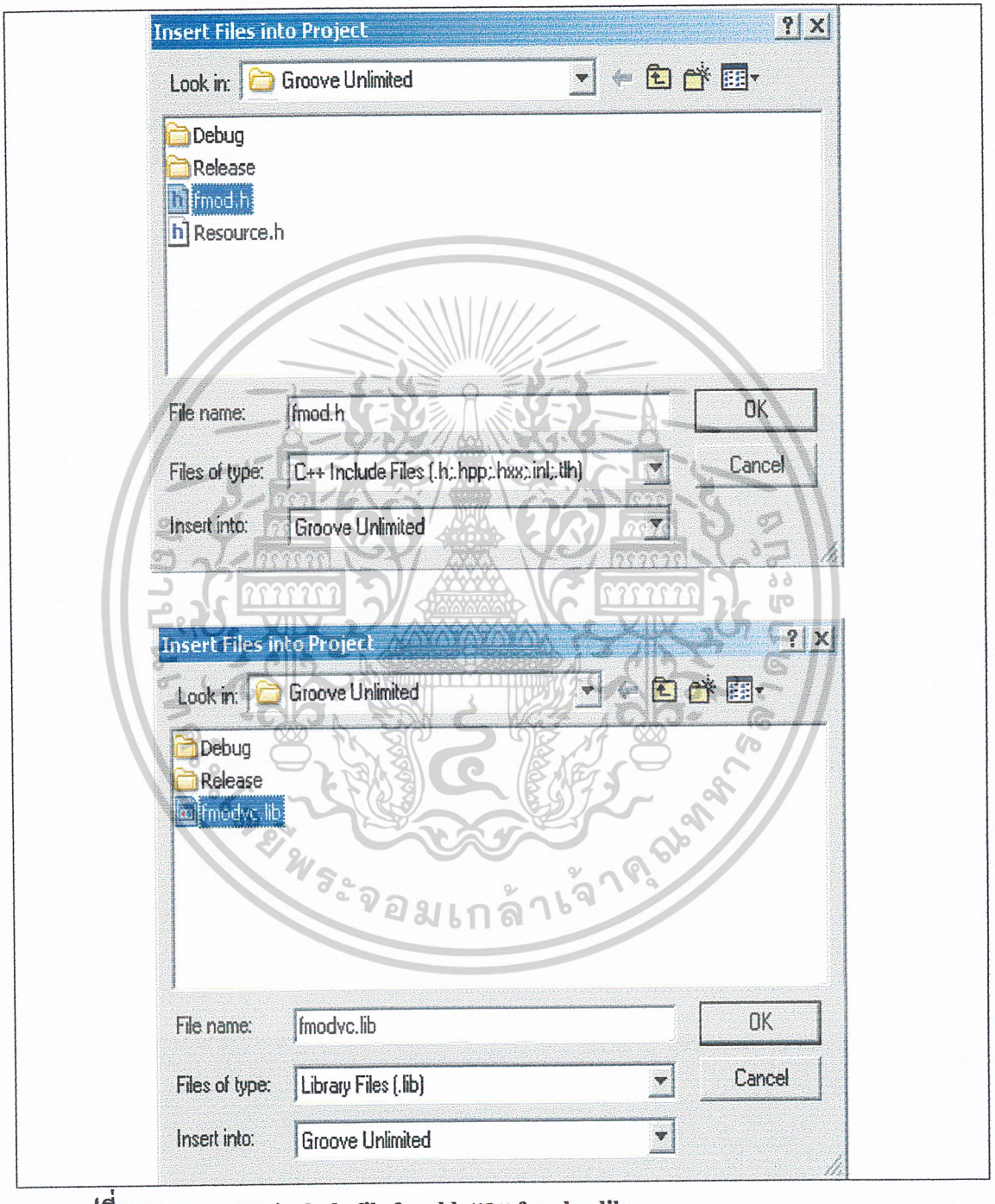


รูปที่ 2.38 แสดงไดเรคทอรีของ fmod

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการคัดลอกไฟล์ fmodvc.lib ในโฟลเดอร์ /api/lib และ fmod.h ในโฟลเดอร์ /api/inc โดยคัดลอกไว้ในโฟลเดอร์เดียวกับโปรเจ็กต์

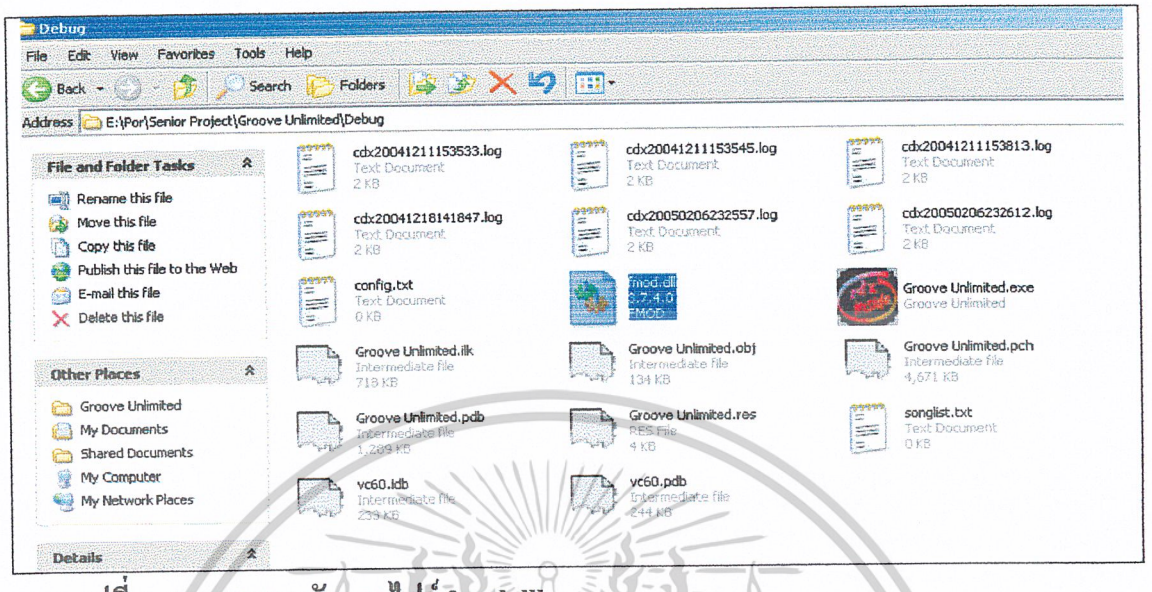
3. ทำการ include ไฟล์ fmod.h และ importไฟล์ fmodvc.lib โดยไปที่เมนูโปรเจ็กต์ → Add To Project → คลิกเลือกที่ Files จะมีหน้าต่างปรากฏขึ้นดังรูป



รูปที่ 2.39 แสดงการ include file fmod.h และ fmodvc.lib

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คัดลอกไฟล์ fmod.dll ในโฟลเดอร์ /api ไว้โฟลเดอร์เดียวกับไฟล์ที่เป็น .exe ของโปรเจ็กต์

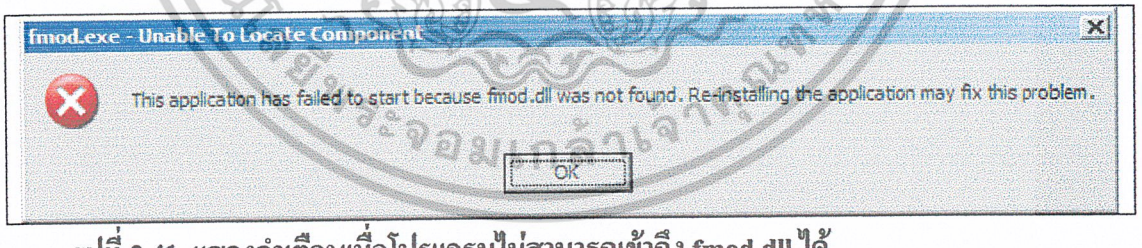


รูปที่ 2.40 แสดงการคัดลอกไฟล์ fmod.dll

2.5.2 การเล่นไฟล์เพลงและไฟล์เสียงประเภทต่าง ๆ

การเล่นไฟล์เพลงและไฟล์เสียงทุกประเภทมีขั้นตอนเหมือนกัน จะแตกต่างกันที่ชนิดตัวแปร และชื่อฟังก์ชันที่เรียกใช้ ขั้นตอนการเล่นไฟล์เพลงและไฟล์เสียงมีดังนี้

1. ก่อนที่จะเขียนโปรแกรม ต้องแน่ใจก่อนว่าโปรแกรมของเราสามารถเข้าถึง fmod.dll ได้ มิฉะนั้นเมื่อรันโปรแกรมจะมีคำเตือนขึ้นมาดังรูป



รูปที่ 2.41 แสดงคำเตือนเมื่อโปรแกรมไม่สามารถเข้าถึง fmod.dll ได้

2. ถ้าไม่มีคำเตือนดังกล่าวก็แสดงว่าสามารถเขียน โปรแกรมของเราสามารถเข้าถึง fmod.dll ได้แล้ว จากนั้นให้ทำการ initialize FMOD โดยใช้คำสั่ง FSOUND_Init ซึ่งจะทำครั้งเดียวภายในโปรแกรม ในโปรเจ็กต์นี้กำหนดให้ FMOD ทำการมิกซ์เพลงที่ความถี่ 44100 Hz และใช้ได้ 16 software channels เขียนได้เป็น

FSOUND_Init(44100,16,0);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ประกาศตัวแปร pointer เพื่อใช้ชี้ไปยังตำแหน่งไฟล์เพลงหรือไฟล์เสียงที่เราต้องการเล่น โดยชนิดของตัวแปร pointer ขึ้นอยู่กับไฟล์ที่จะนำมาเล่น

- ไฟล์ประเภท **.MOD, .S3M, .XM, .IT** หรือ **.MID** ต้องใช้ handle ชนิด FMUSIC_MODULE เช่น

  FMUSIC_MODULE *mySong;

- ไฟล์ประเภท **.WAV, .MP2, .MP3, .OGG** หรือ **.RAW** ต้องใช้ handle ชนิด FSOUND_SAMPLE เช่น

  FSOUND_SAMPLE *mySound;

- เล่นไฟล์แบบสตรีม ใช้ handle ชนิด FSOUND_STREAM เช่น

  FSOUND_STREAM *myStream;

4. กำหนดชื่อไฟล์ที่ต้องการเล่นให้กับตัวแปรที่ประกาศไว้

- ไฟล์ประเภท **.MOD, .S3M, .XM, .IT** หรือ **.MID** กำหนดดังนี้

  mySong = FMUSIC_LoadSong (“ ชื่อไฟล์ ”);

- ไฟล์ประเภท **.WAV, .MP2, .MP3, .OGG** หรือ **.RAW** กำหนดดังนี้

  mySound = FMUSIC_Sample_Load (0,“ ชื่อไฟล์ ”,FSOUND_16BITS,0,0);

- เล่นไฟล์แบบสตรีม กำหนดดังนี้

  myStream = FSOUND_Stream_Open (“ ชื่อไฟล์ ”,FSOUND_16BITS,0,0);

5. เล่นไฟล์นั้น ซึ่งต้องกำหนดให้เล่นไฟล์นั้นในลูปของเกม

- ไฟล์ประเภท **.MOD, .S3M, .XM, .IT** หรือ **.MID** ใช้คำสั่ง FMUSIC_PlaySong เช่น

  FMUSIC_PlaySong(mySong);

- ไฟล์ประเภท **.WAV, .MP2, .MP3, .OGG** หรือ **.RAW** ใช้คำสั่ง FSOUND_PlaySound เช่น

  FSOUND_PlaySound(0,mySound);

- เล่นไฟล์แบบสตรีม ใช้คำสั่ง FSOUND_Stream_Play เช่น

  FSOUND_Stream_Play(0,myStream);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คืนหน่วยความจำให้กับระบบโดยใช้คำสั่ง **FSOUND_Close();**

ตัวอย่างการเล่นไฟล์แบบต่าง ๆ

- **การเล่นไฟล์ประเภท .MOD, .S3M, .XM, .IT, .MID**

```
FMUSIC_MODULE      *mySong;
mySong = FMUSIC_LoadSong (“ ชื่อไฟล์ ”);
while(1)  //Game loop
{
        FMUSIC_PlaySong(mySong);
}
FSOUND_Close();
```

- **การเล่นไฟล์ประเภท .WAV, .MP3, .MP2, .OGG, .RAW**

```
FSOUND_SAMPLE      *mySound;
mySound = FMUSIC_Sample_Load (0,“ ชื่อไฟล์
”,FSOUND_16BITS,0,0);
while(1)  //Game loop
{
        FSOUND_PlaySound(0,mySound);
}
FSOUND_Close();
```

- **การเล่นไฟล์แบบ Stream**

```
FMUSIC_STREAM      *myStream;
myStream = FSOUND_Stream_Open  (“ ชื่อไฟล์
”,FSOUND_16BITS,0,0);
while(1)  //Game loop
{
        FSOUND_Stream_Play(0,myStream);
}
FSOUND_Close();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการดำเนินงานวิจัย

ลำดับขั้นตอนในการดำเนินการวิจัยทั้งหมดแบ่งออกได้เป็น 3 ส่วนใหญ่ ได้แก่ ส่วนของการวิเคราะห์และออกแบบเกม ส่วนของการลงมือปฏิบัติการจริง และส่วนของการทดสอบเกม ซึ่งแต่ละส่วนล้วนมีความสำคัญกับการพัฒนาเกมกรู๊ฟ อันลิมิเต็ดเป็นอย่างมาก และแต่ละขั้นตอนประกอบด้วยขั้นตอนย่อย ๆ ดังนี้

1. **ขั้นตอนการวิเคราะห์และออกแบบเกม**
   - การวิเคราะห์ความต้องการว่าต้องการจะสร้างเกมในรูปแบบใด เนื้อหาทั้งหมดมีลักษณะอย่างไร ออกแบบกติกาการเล่น ระบบของเกม และลักษณะแนวทางการเล่นรูปแบบตัวละคร ฉาก เสียง และเนื้อหาของเกม ว่าควรมีรูปแบบใด
   - ออกแบบหน้าจอส่วนที่ใช้ติดต่อกับผู้เล่น ซึ่งต้องเป็นรูปแบบที่ใช้ง่าย และดึงดูดความสนใจของผู้เล่น
   - ออกแบบวิธีการจัดเก็บข้อมูลที่จำเป็นต้องใช้ภายในเกม โดยที่ต้องใช้พื้นที่ในการจัดเก็บน้อย เป็นระเบียบ และสามารถนำมาใช้ได้ง่ายเพื่อเพิ่มประสิทธิภาพการทำงานให้กับเกม
   - ออกแบบวิธีการเขียนโปรแกรม และเลือกใช้ฮาร์ดแวร์, ซอฟต์แวร์ ร่วมถึง component และเครื่องมือ ต่าง ๆ ที่นำมาใช้ในการสร้างเกม
2. **ขั้นตอนการลงมือปฏิบัติการจริง**
   - ด้านภาพและกราฟิก การทำงานในส่วนนี้จะใช้โปรแกรมทางด้านกราฟิกเข้ามาช่วยในการสร้าง ฉากที่ได้ทำการออกแบบไว้แล้ว
   - ด้านซาวน์เอฟเฟกต์ การทำงานในส่วนนี้จะใช้โปรแกรมการสร้างเสียงต่าง ๆ มาช่วยในการสร้าง เสียงประกอบสำหรับเนื้อหาภายในเกม
   - ด้านโปรแกรมมิ่ง การทำงานในส่วนนี้จะเกี่ยวข้องการเขียนคำสั่งต่าง ๆ โดยมีการนำฉาก ตัวละคร และเสียงประกอบ ที่ได้สร้างไว้มาใช้ในส่วนนี้ ซึ่งมีการจัดลำดับขั้นตอนการทำงานดังนี้
     - ใช้โปรแกรม Visual C++ ในส่วนของการเขียนและแก้ไขซอร์สโค้ดของเกม
     - ใช้ CDX Game Developer Kit ช่วยในการพัฒนาโปรแกรมเกม
     - นำฉาก ตัวละคร และเสียง มาประกอบเข้าด้วยกัน
     - เขียนคำสั่งที่ทำให้เกิดการกระทำตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขั้นตอนการทดสอบเกม การทำงานในส่วนนี้จะเป็นขั้นตอนในการทดสอบเกมที่เราผลิตขึ้นว่ามีความสมบูรณ์มากน้อยเพียงใด เพื่อหาข้อผิดพลาดและประเมินประสิทธิภาพของเกม เมื่อพบข้อผิดพลาดก็จะทำการแก้ไขโดยทันที

3.1 ขั้นตอนการวิเคราะห์และออกแบบเกม

3.1.1 การออกแบบรูปแบบ กติกาการเล่น และระบบเกม

กรู๊ฟ อันลิมิเต็ด เป็นเกมที่ส่งเสริมทักษะทางด้านดนตรีและประสาทสัมผัสต่าง ๆ รูปแบบของเกมคือ สามารถเล่นได้ 2 ผู้เล่น โดยผู้เล่นเลือกเพลงที่ต้องการเล่น โดยเลือกจากเพลงที่มีให้อยู่แล้วในเกม หรือผู้เล่นสามารถเลือกเพลงที่ต้องการเล่น โดยรูปแบบของไฟล์เพลงที่เลือกเป็นแบบ .WAV .MP3 หรือ .OGG การที่ผู้เล่นสามารถเลือกเล่นเพลงเองได้ ทำให้เกมกรู๊ฟ อันลิมิเต็ด สามารถตอบสนองความต้องการของผู้เล่นได้มากขึ้น ซึ่งต่างจากเกมรูปแบบเดียวกัน ที่เป็นที่รู้จักกันดี เช่น Beat Mania, DJ Simulation เป็นต้น ที่ผู้เล่นสามารถเลือกเล่นได้เฉพาะเพลงที่มีให้ไว้แล้วในเกม เมื่อผู้เล่นเลือกเพลงเรียบร้อยแล้ว ก็จะเข้าสู่หน้าจอของเพลงนั้น โดยหน้าจอจะเป็นตัวบอกว่าเมื่อถึงจังหวะนี้ หรือตัวโน้ตตัวนี้ต้องกดคีย์บอร์ดคีย์ใด ซึ่งคีย์ที่เป็นตัวแทนตัวโน้ตมีทั้งหมด 6 คีย์ ซึ่งผู้เล่นสามารถปรับแต่งให้ 6 คีย์นั้นเป็นคีย์ใดก็ได้บนคีย์บอร์ด ผู้เล่นจะได้คะแนนจากการกดคีย์ให้ถูกคีย์และตรงจังหวะ โดยแบ่งระดับของคะแนนในการเล่นดังนี้

- กดถูกคีย์ และไม่ช้ากว่า 0.2 วินาทีของโน้ตตัวนั้น : Cool !! ได้ 200 คะแนน
- กดถูกคีย์ และไม่ช้ากว่า 0.3 วินาทีของโน้ตตัวนั้น: Good !! ได้ 100 คะแนน
- กดถูกคีย์ และไม่ช้ากว่า 0.5 วินาทีของโน้ตตัวนั้น: Bad !! ได้ 50 คะแนน
- กดถูกคีย์ แต่กดเร็วหรือช้าไปเกิน 0.5 หรือกดไม่ถูกคีย์เลย : Missed !! ได้ 0 คะแนน

เมื่อจบเกม จะทำการสรุปคะแนนที่ได้ว่าได้ Cool, Good, Bad และ Missed เป็นจำนวนกี่ครั้ง และ คำนวณว่าได้คะแนนรวมเท่าไร ถ้าผู้เล่นได้คะแนนรวมสูงสุด 10 อันดับแรก โดยเปรียบเทียบคะแนนของผู้เล่นแต่ละคนจากสัดส่วนของคะแนนที่ได้กับคะแนนเต็ม จะได้บันทึกใน High Score

รูปแบบการเล่นแบบ 2 ผู้เล่นคล้ายกับการเล่นคนเดียว ผู้เล่นแต่ละผู้เล่นจะมีหน้าต่างของตนเองโดยเฉพาะ แบบสุดท้ายคือแบบผู้เล่น 2 คนจะแข่งกันเล่น และมีการปล่อยพลังใส่กัน โดยถ้ากดคีย์ได้ตรงตามจังหวะมากเท่าใดก็สามารถปล่อยพลังได้มากขึ้นเท่านั้น เมื่อผู้เล่นคนที่ถูกปล่อยพลังใส่ พลังชีวิตของตนก็จะลดลง เมื่อจบเพลงผู้เล่นที่มีพลังชีวิตเหลือมากกว่าเป็นผู้ชนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบภาพและกราฟิก

รูปแบบของภาพกราฟิกในเกมกรู๊ฟ อันลิมิเต็ด มีลักษณะของรูปภาพเป็นแบบ 2 มิติ ใช้สีสันสดใสเพื่อดึงดูดความสนใจผู้เล่น ซึ่งกลุ่มเป้าหมายเป็นกลุ่มเด็กวัยรุ่น ในเกมประกอบด้วยหน้าจอต่าง ๆ ดังนี้ หน้าจอหลักแสดงเมนูต่าง ๆ เพื่อเข้าสู่เกม หน้าจอขณะเล่นเกม หน้าจอ Option เพื่อเข้าสู่การปรับแต่งค่าของเกม หน้าจอ Import Song สำหรับนำเพลงที่ผู้เล่นต้องการเข้าสู่เกม หน้าจอ Sound Test สำหรับให้ผู้เล่นทดลองฟังเพลง และหน้าจอ Training สำหรับสอนวิธีการเล่นเกม ซึ่งโปรแกรมที่นำมาใช้สำหรับสร้างและตกแต่งภาพที่เลือกใช้สำหรับเกมนี้ได้แก่ โปรแกรม Adobe Photoshop CS และ Adobe Illustrator CS

3.1.3 การออกแบบเสียงและซาวน์เอฟเฟกต์ประกอบ

ไฟล์เสียงที่ใช้เป็นเอฟเฟกต์ในเกมมีสองประเภทคือ WAVE และ MIDI ส่วนไฟล์เพลงเป็นแบบ MP3 ซึ่งไฟล์เสียงและซาวน์เอฟเฟกต์ที่ใช้จะเป็นแบบที่มีความสนุกสนาน เร้าใจ ซึ่งช่วยให้ผู้เล่นมีความเพลิดเพลินกับเกมได้มากขึ้น ส่วนไฟล์เพลงที่ใช้ในตัวเกมจะมีหลากหลายแนว ทั้งป๊อป ร๊อค อิเล็คทรอนิก แดนซ์ ฯลฯ ซึ่งทำให้สามารถตอบสนองความพึงพอใจของผู้เล่นได้หลายกลุ่ม โปรแกรมที่นำมาใช้สำหรับสร้างและตัดต่อไฟล์เสียงและซาวน์เอฟเฟกต์ได้แก่ Fruity Loop Studio 5 และ Adobe Auditon 1.5

3.1.4 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น

ในเกมกรู๊ฟ อันลิมิเต็ด ใช้อุปกรณ์ติดต่อกับผู้เล่น 2 อย่าง คือ เมาส์ และ คีย์บอร์ด โดยใช้เมาส์สำหรับเลือกกดปุ่มเมนูต่าง ๆ ใช้คีย์บอร์ดสำหรับกดปุ่มเลือกเมนูต่าง ๆ และใช้สำหรับกดปุ่มตามจังหวะดนตรีเพื่อเล่นเกม เกมกรู๊ฟ อันลิมิเต็ด ประกอบด้วย Interface หลัก ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงหน้าที่ของปุ่มที่มีในแต่ละ interface ของเกมกรู๊ฟ อันลิมิเต็ด

Interface	ปุ่ม	หน้าที่
Main		
Main	Game Start	เข้าสู่ส่วนการเล่นเกม
	Training	เข้าสู่ส่วนการสอนวิธีการเล่นเกม
	Import Song	เข้าสู่ส่วนการนำเพลงเข้าสู่เกม
	Option	เข้าสู่ส่วนการปรับแต่งค่าของเกม
	Exit	ออกจากเกม
Game Start		
Sound Select	1 player	เลือกเล่นคนเดียว
	2 player	เลือกเล่น 2 คน
	VS battle	เลือกเล่น 2 คนแบบแข่งกัน
	Easy	เลือกเกมระดับง่าย
	Medium	เลือกเกมระดับปานกลาง
	Hard	เลือกเกมระดับยาก
	play	เข้าสู่การเล่นเกม
Import song		
Import Song	Import	วิเคราะห์ไฟล์เพลง เพื่อนำไปใช้ในเกม
Option		
Option	Key Config	เข้าสู่ส่วนการปรับตั้งค่าการใช้งานคีย์บอร์ด
	Sound Test	เข้าสู่ส่วนฟังตัวอย่างเพลงที่มีอยู่ในเกม
	High Score	เข้าสู่ส่วนแสดงผู้ทำคะแนนสูงสุด 10 อันดับ
Sound Test	<< (ซ้าย)	เล่นเพลงในลำดับก่อนหน้า
	>> (ขวา)	เล่นเพลงในลำดับถัดไป
	▶ (Play)	เล่นเพลง
	■ (Stop)	หยุดเพลงที่กำลังเล่นอยู่
	■■ (Pause)	หยุดเพลงที่กำลังเล่นอยู่ชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การออกแบบการจัดเก็บข้อมูล

ข้อมูลที่จำเป็นที่ใช้ในเกมนี้ที่จะต้องจัดเก็บ จะประกอบไปด้วยพารามิเตอร์เริ่มต้นต่าง ๆ ของเกม ไฟล์ที่ได้จากการวิเคราะห์ไฟล์จากการนำเพลงเข้าสู่เกม และไฟล์ในการจัดเก็บคะแนน High Score สำหรับไฟล์ทั้งหมดนั้นต้องการให้เข้าถึงได้อย่างรวดเร็ว มีการเปลี่ยนแปลงข้อมูลในตัวไฟล์ไม่บ่อยนัก และมีความซับซ้อนน้อย จึงเลือกใช้วิธีเก็บเป็นแบบ Text File เพื่อให้มีการเข้าถึงข้อมูลได้ง่าย และยังใช้เนื้อที่น้อยกว่าการเก็บแบบ Database

3.1.6 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้

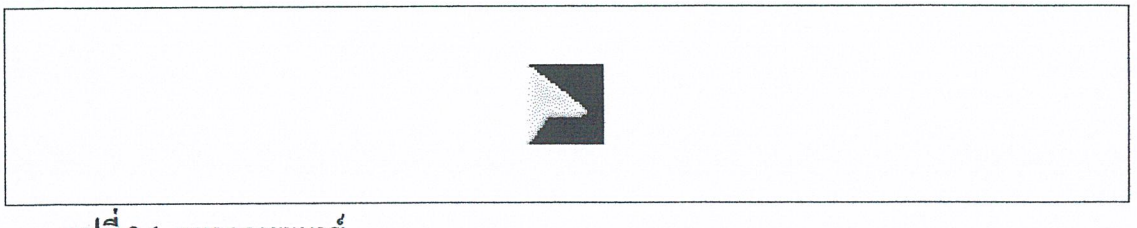
การเขียนโปรแกรมเกมกรู๊ฟ อันลิมิเต็ดนี้ ได้เลือกใช้คอมไพเลอร์ภาษา Visual C++ 6.0 มาใช้ร่วมกับ CDX Game Engine ซึ่ง CDX จะทำหน้าที่ติดต่อกับ DirectX อีกที ซึ่ง DirectX จะเป็นผู้ติดต่อกับฮาร์ดแวร์ต่าง ๆ โดยตรงโดยเฉพาะอุปกรณ์แสดงผลทั้งภาพ และเสียง จุดเด่นของ DirectX นั้นคือ สามารถทำงานติดต่อกับฮาร์ดแวร์โดยไม่ต้องผ่านระบบปฏิบัติการ ทำให้ประสิทธิภาพในการทำงานสูงขึ้น สามารถทำงานได้เร็วขึ้น ส่วนซอร์ฟแวร์ที่ใช้ในการสร้างฉากและภาพ ได้แก่ Adobe Photoshop CS และ Adobe Illustrator CS ส่วนซอร์ฟแวร์ที่ใช้ในการสร้างเสียงและซาวน์เอฟเฟกต์ ได้แก่ Fruity Loop Studio 5 และ Adobe Audition 1.5

3.2 ขั้นตอนการลงมือปฏิบัติจริง

3.2.1 สร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม

ภาพกราฟิกที่ใช้ในเกมกรู๊ฟ อันลิมิเต็ดเป็นแบบ 2 มิติ ซึ่งใช้ Adobe Photoshop CS และ Adobe Illustrator CSในการสร้างภาพกราฟิก โดยภาพที่ได้จะเก็บในรูปแบบบิตแมป (.bmp) ซึ่งการสร้างภาพกราฟิก ต้องสร้างรูปที่เป็นฉากหลัง และรูปที่เป็นตัวละครแยกกัน ซึ่งตัวละครจะต้องทำเป็นเฟรมให้ภาพค่อย ๆ เคลื่อนไหวทีละนิด และใช้สีดำเป็นสีพื้นหลัง ภายในหน้าจอหนึ่ง ๆ จะมีภาพฉากหลังและตัวละคร ดังนี้

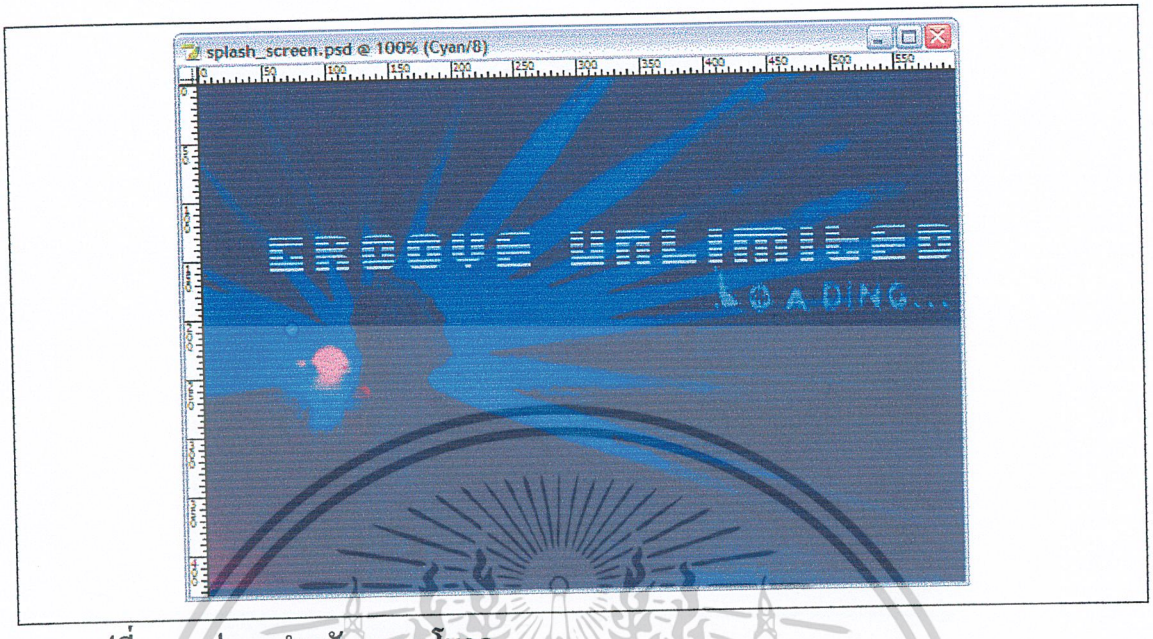
3.2.1.1 ภาพที่ใช้สำหรับเมาส์



รูปที่ 3.1 แสดงภาพเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.2 หน้าจอ Splash Screen ใช้สำหรับโชว์ขณะรอเมื่อมีการเปิดเกมและโหลดวัตถุต่าง ๆ

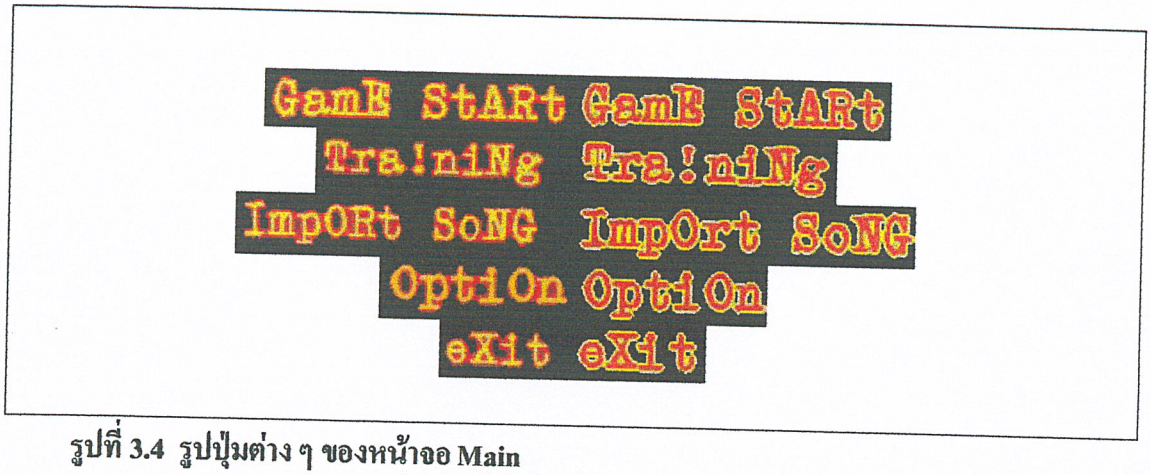


รูปที่ 3.2 รูปภาพสำหรับรอการโหลด

3.2.1.3 หน้าจอ Main

รูปที่ 3.3 รูปฉากหลังของหน้าจอ Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

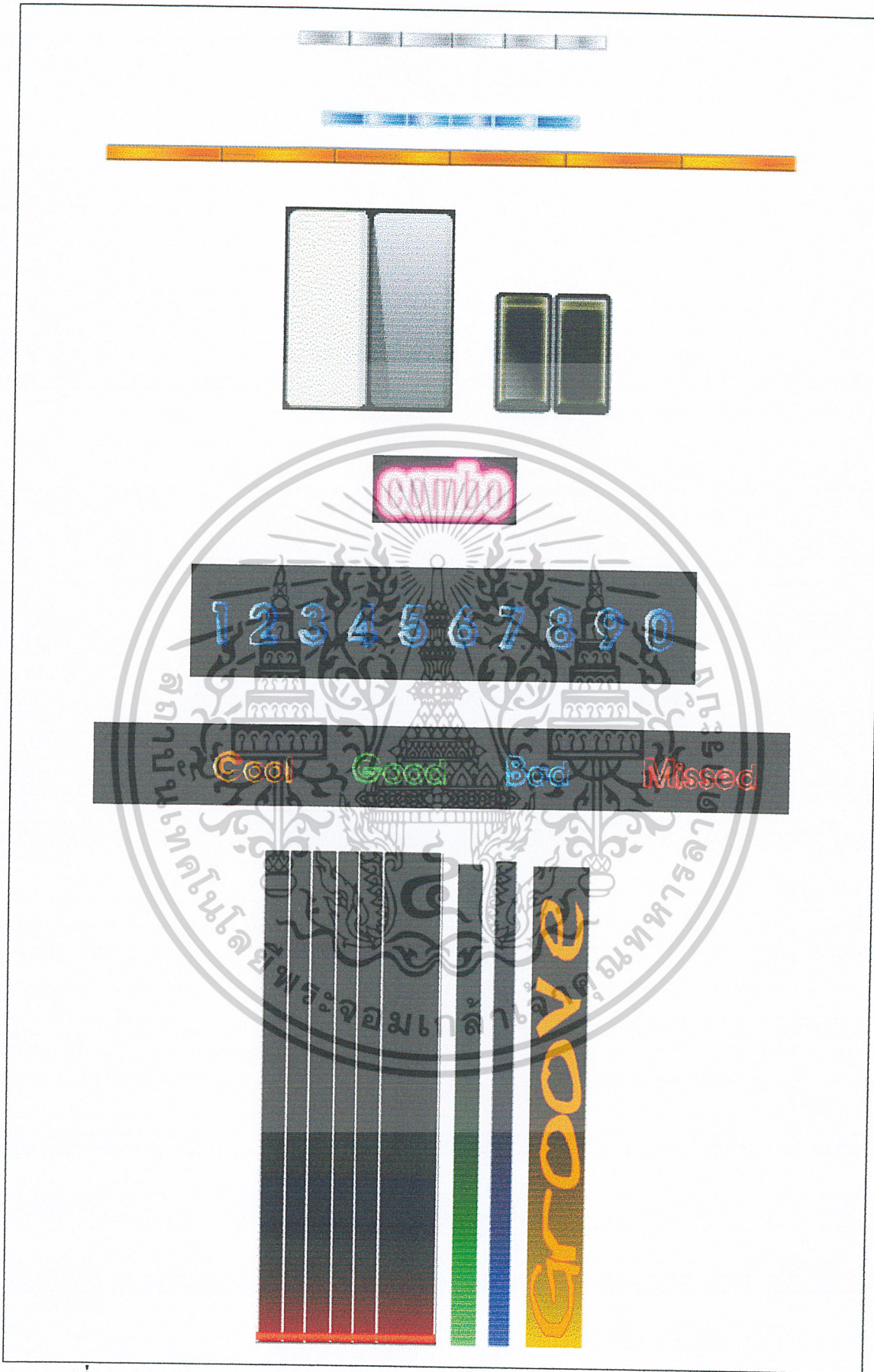


รูปที่ 3.4 รูปปุ่มต่าง ๆ ของหน้าจอ Main

3.2.1.4 หน้าจอ Game Play

รูปที่ 3.5 รูปฉากหลังของหน้าจอ Game Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

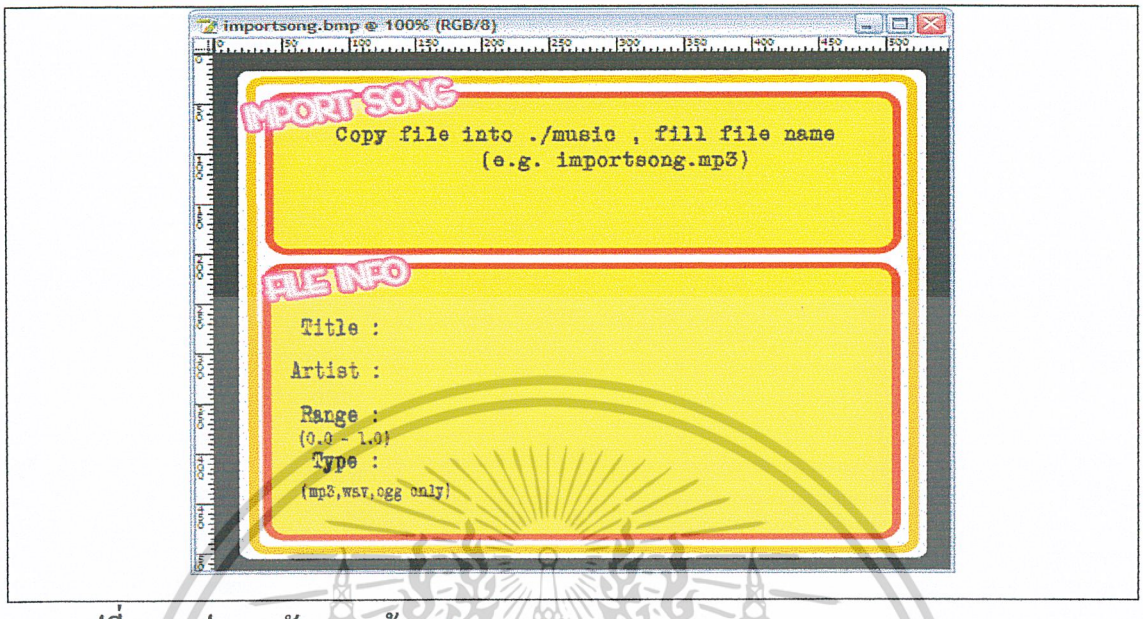


รูปที่ 3.6 รูปปุ่มและตัวละครต่าง ๆ ของหน้าจอ Game Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.5 หน้าจอ Training ใช้ภาพฉากหลังและตัวละครเดียวกับหน้าจอ Game Play

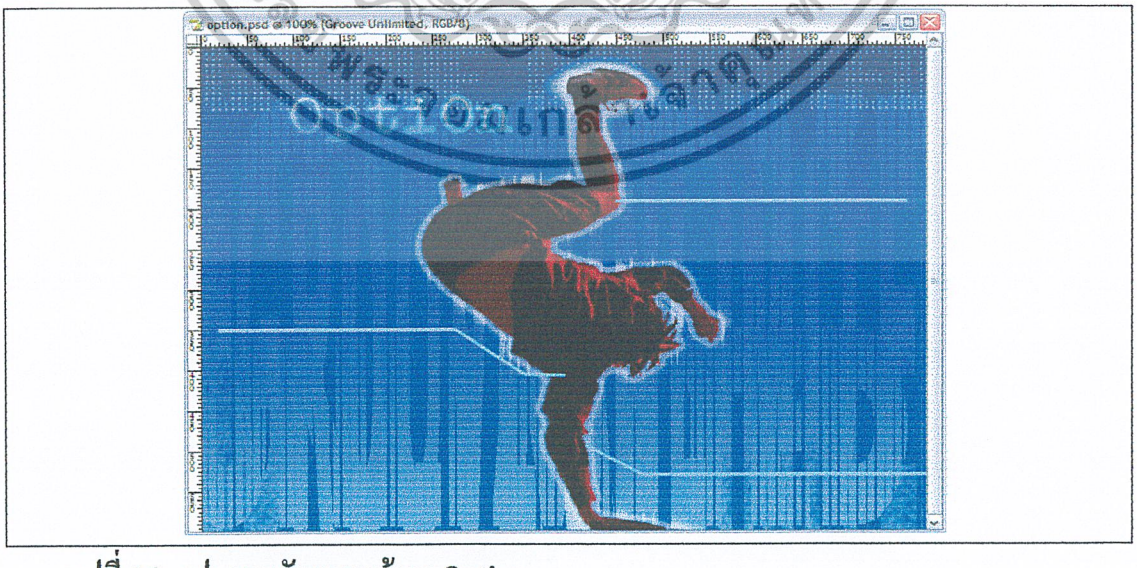
3.2.1.6 หน้าจอ Import Song



รูปที่ 3.7 รูปฉากหลังของหน้าจอ Import Song

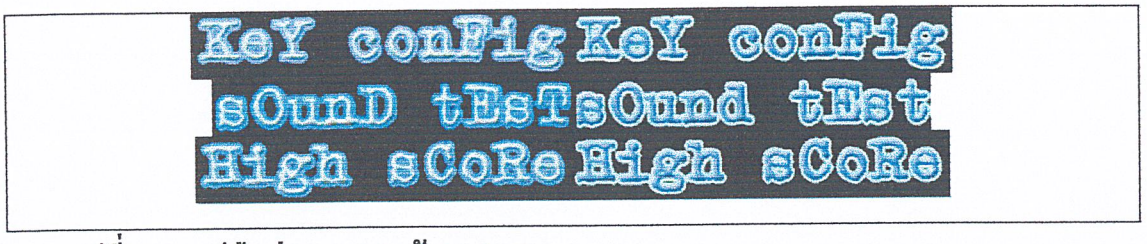
รูปที่ 3.8 รูปปุ่มต่าง ๆ ของหน้าจอ Import Song

3.2.1.7 หน้าจอ Option



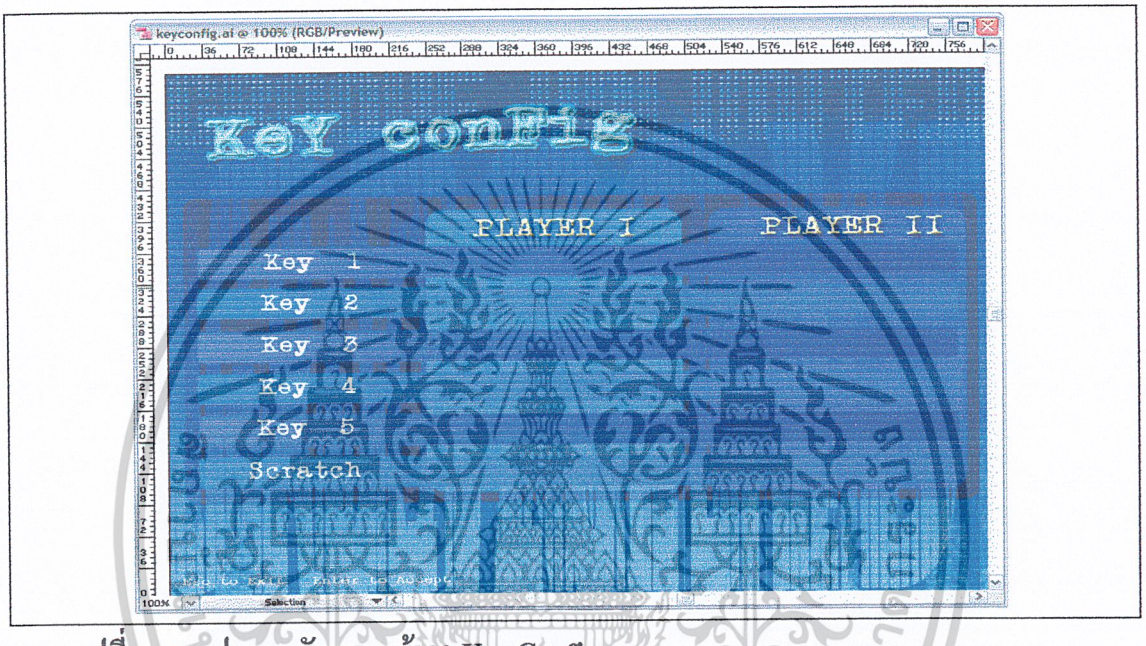
รูปที่ 3.9 รูปฉากหลังของหน้าจอ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 รูปปุ่มต่าง ๆ ของหน้าจอ Option

3.2.1.8 หน้าจอ Key Config

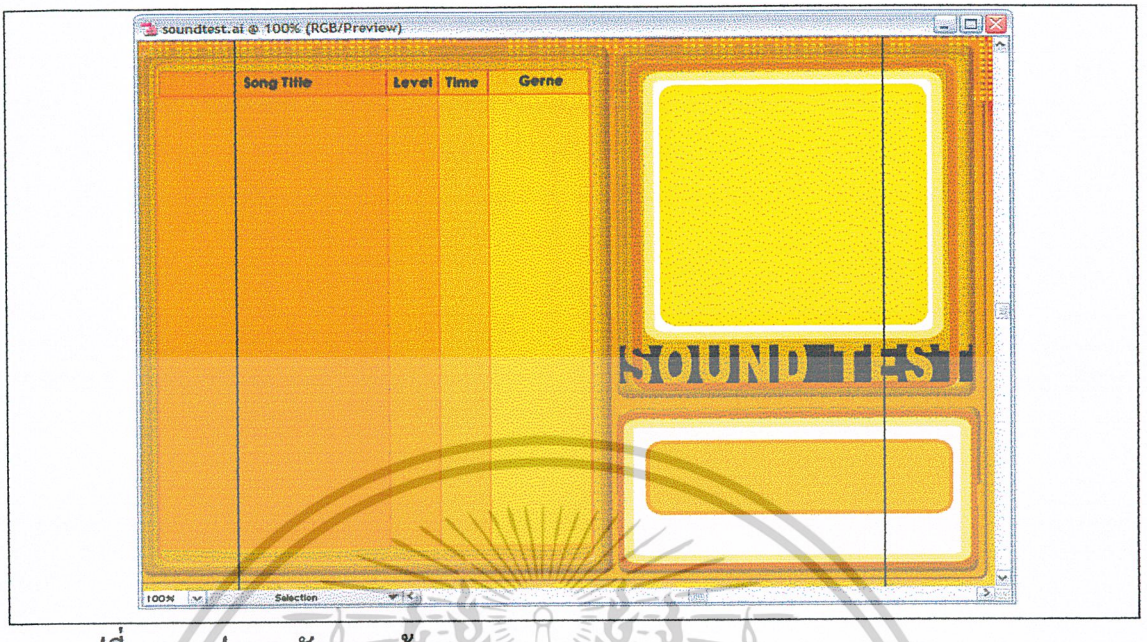


รูปที่ 3.11 รูปฉากหลังของหน้าจอ Key Config

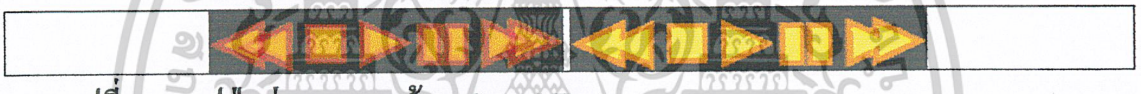
รูปที่ 3.12 รูปกรอบที่ใช้สำหรับแสดงจุดโฟกัสของหน้าจอ Key Config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.9 หน้าจอ Sound Test

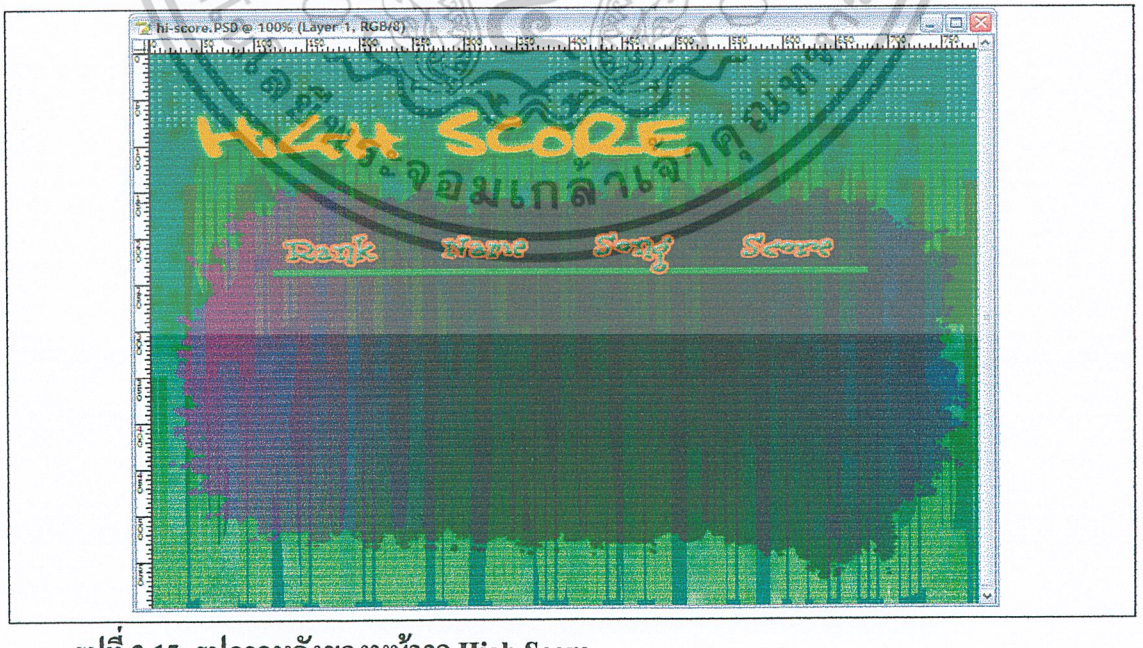


รูปที่ 3.13 รูปฉากหลังของหน้าจอ Sound Test



รูปที่ 3.14 รูปปุ่มต่าง ๆ ของหน้าจอ Sound Test

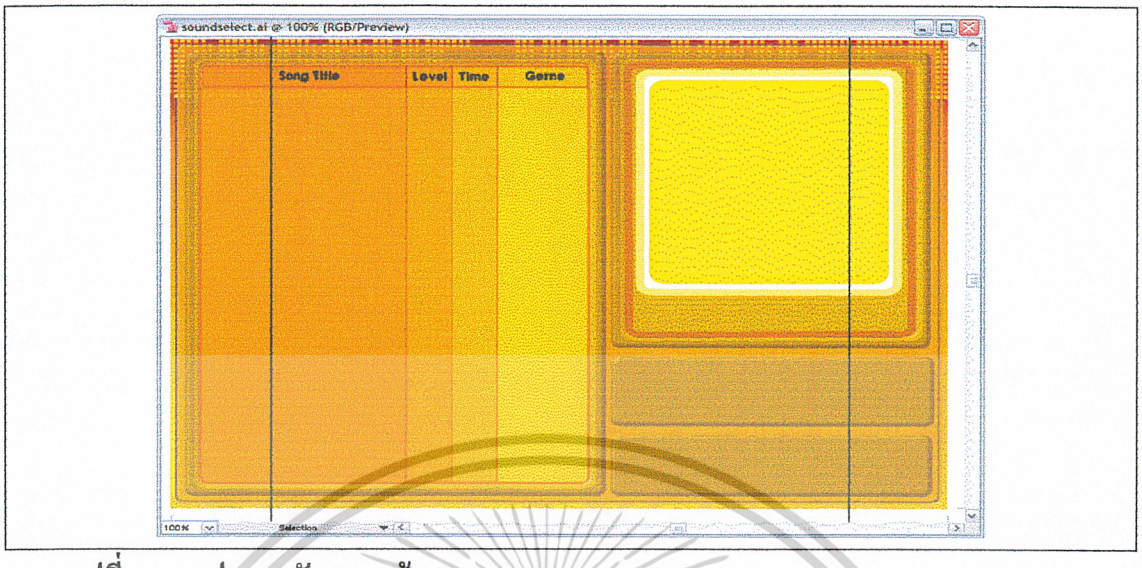
3.2.1.10 หน้าจอ High Score



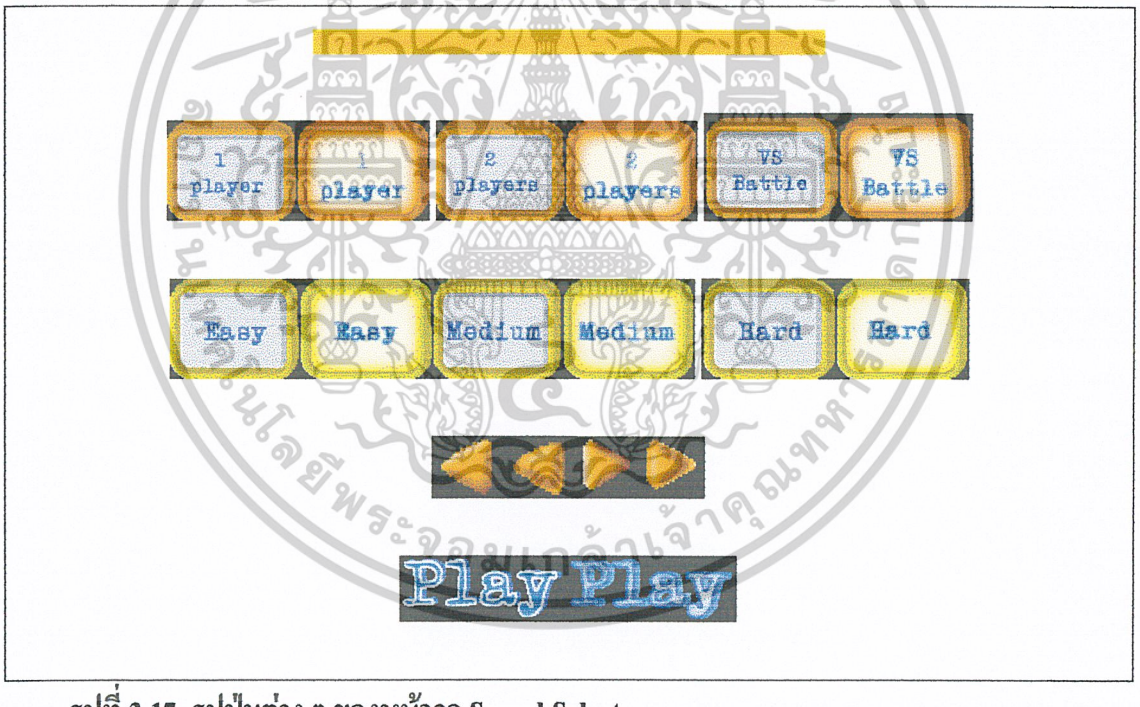
รูปที่ 3.15 รูปฉากหลังของหน้าจอ High Score

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.11 หน้าจอ Sound Select

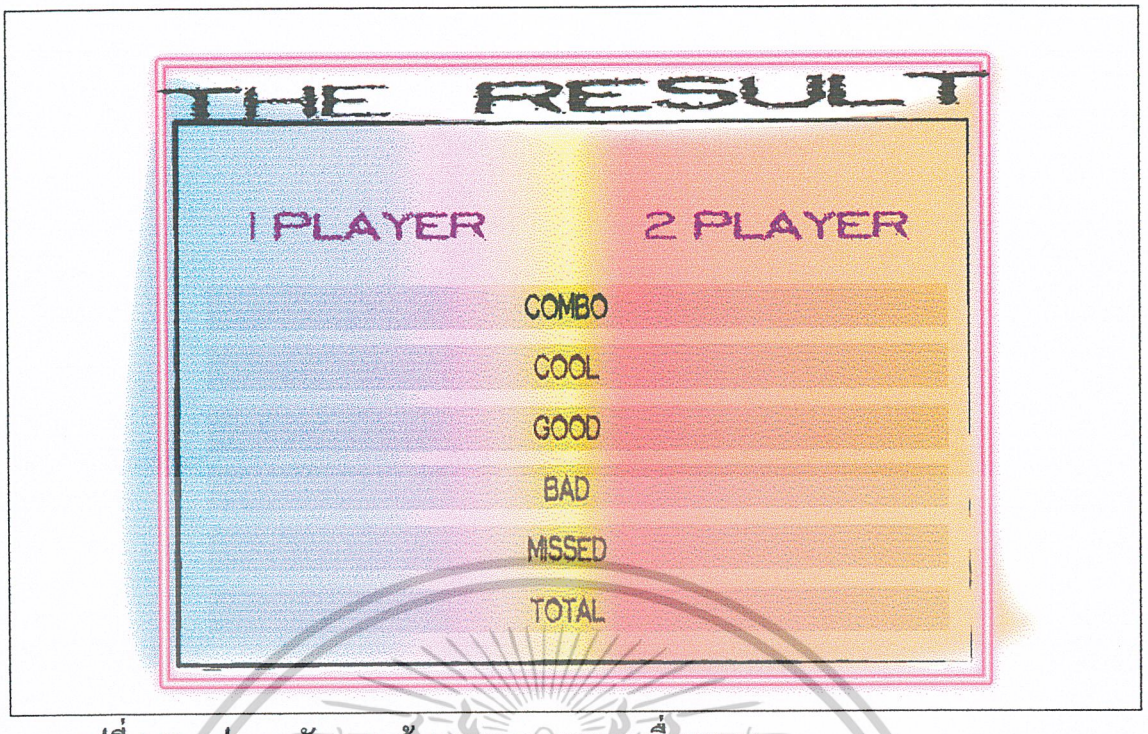


รูปที่ 3.16 รูปฉากหลังของหน้าจอ Sound Select

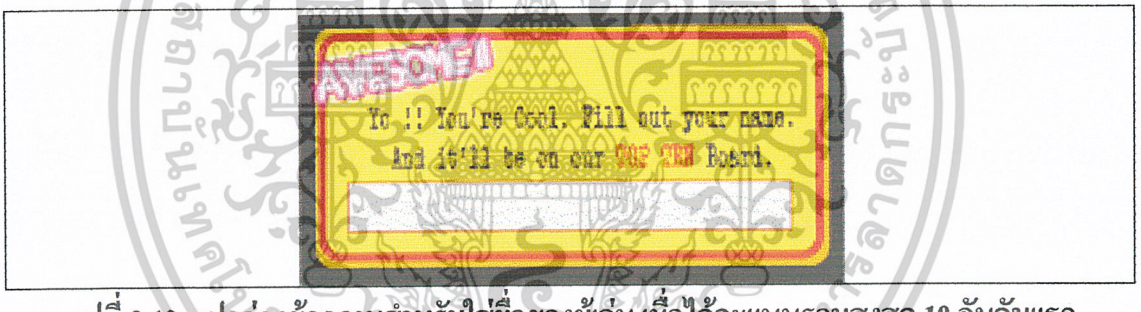


รูปที่ 3.17 รูปปุ่มต่าง ๆ ของหน้าจอ Sound Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 รูปฉากหลังของหน้าจอแสดงผลคะแนนเมื่อจบเกม



รูปที่ 3.19 รูปกล่องข้อความสำหรับใส่ชื่อของผู้เล่น เมื่อได้คะแนนรวมสูงสุด 10 อันดับแรก

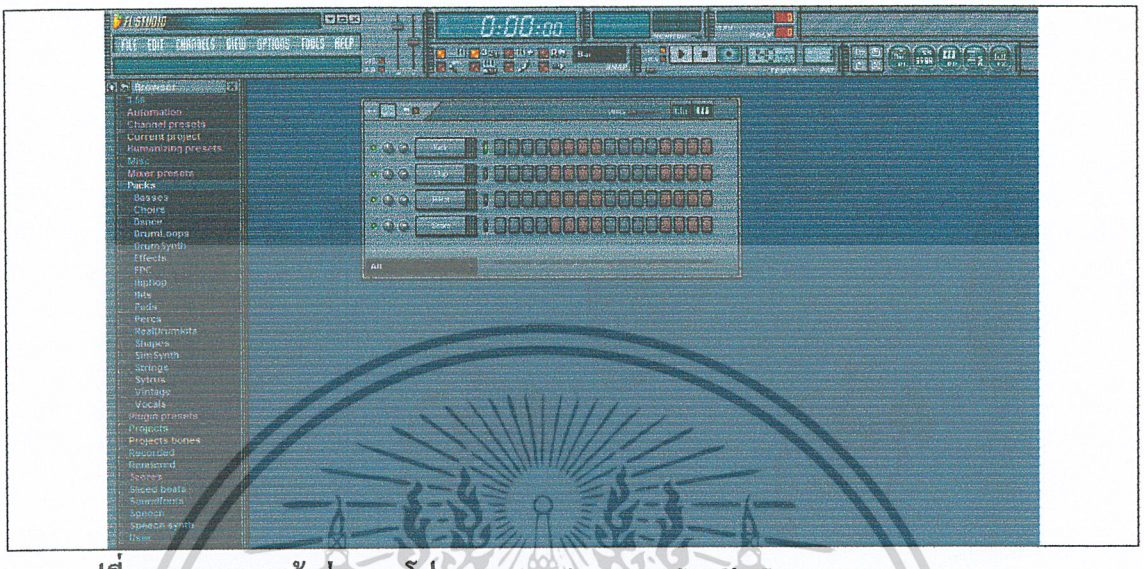
3.2.2 สร้างเสียงเพลงและซาวน์เอฟเฟกต์ประกอบ

ภายในเกม กรู๊ฟ อันลิมิเต็ด มีดนตรีประกอบเพื่อเพิ่มความสนุกสนานเพลิดเพลินให้แก่ผู้เล่น ในขณะที่ผู้เล่นกำลังอยู่ที่หน้าจอต่าง ๆ และเมื่อผู้เล่นใช้เมาส์คลิกปุ่ม หรือกดคีย์บอร์ด จะมีเสียงซาวน์เอฟเฟกต์ดังขึ้นด้วย การสร้างเสียงเพลงและซาวน์เอฟเฟกต์ต่าง ๆ จะอยู่ในรูปแบบ .wav, .mid หรือ .mp3 ซึ่งไฟล์เหล่านั้นสร้างขึ้นมาจาก Fruity Loop Studio 5 และนำไฟล์เพลงที่ได้สร้างไว้หรือไฟล์ซาวน์เอฟเฟกต์อื่น ๆ มาทำการตัดต่อโดยใช้ Adobe Audition 1.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 การสร้างเพลงด้วยโปรแกรม Fruity Loop Studio 5 มีวิธีการดังนี้

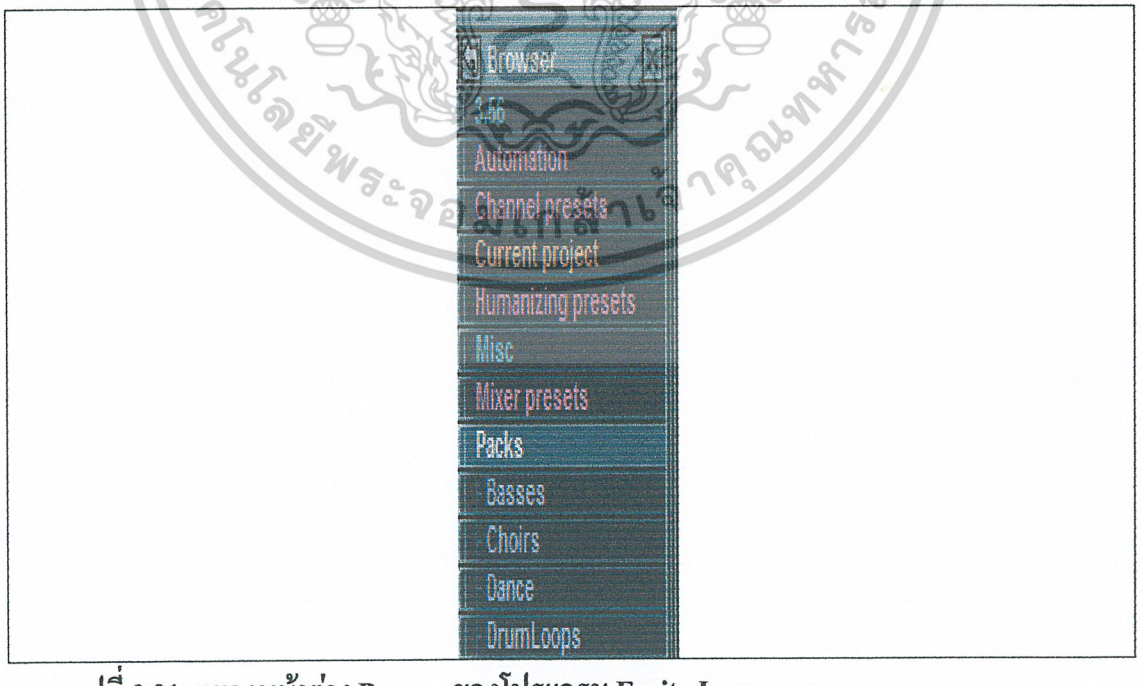
สร้างเพลงที่ใช้ในเกม โดยเปิดโปรแกรม Fruity Loop Studio 5 ขึ้นมาแล้วไปที่ FILE -> New จะได้หน้าต่างดังรูป



รูปที่ 3.20 แสดงหน้าต่างของโปรแกรม Fruity Loop Studio 5

ก่อนที่เราจะสร้างเพลงใหม่ควรทำความรู้จักหน้าต่างสำคัญ ๆ ในโปรแกรม Fruity Loop Studio 5 ซึ่งมี ดังนี้

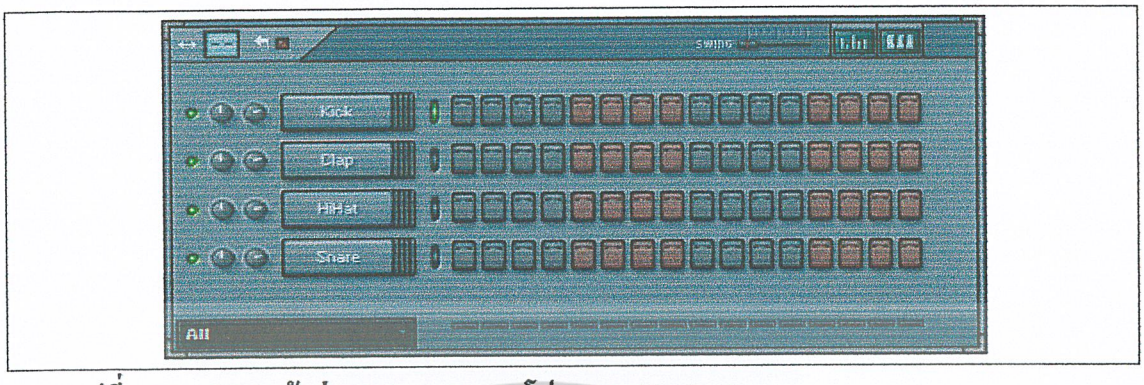
1. หน้าต่าง Browse มีไว้สำหรับเลือกเสียงดนตรีที่ต้องการ



รูปที่ 3.21 แสดงหน้าต่าง Browse ของโปรแกรม Fruity Loop

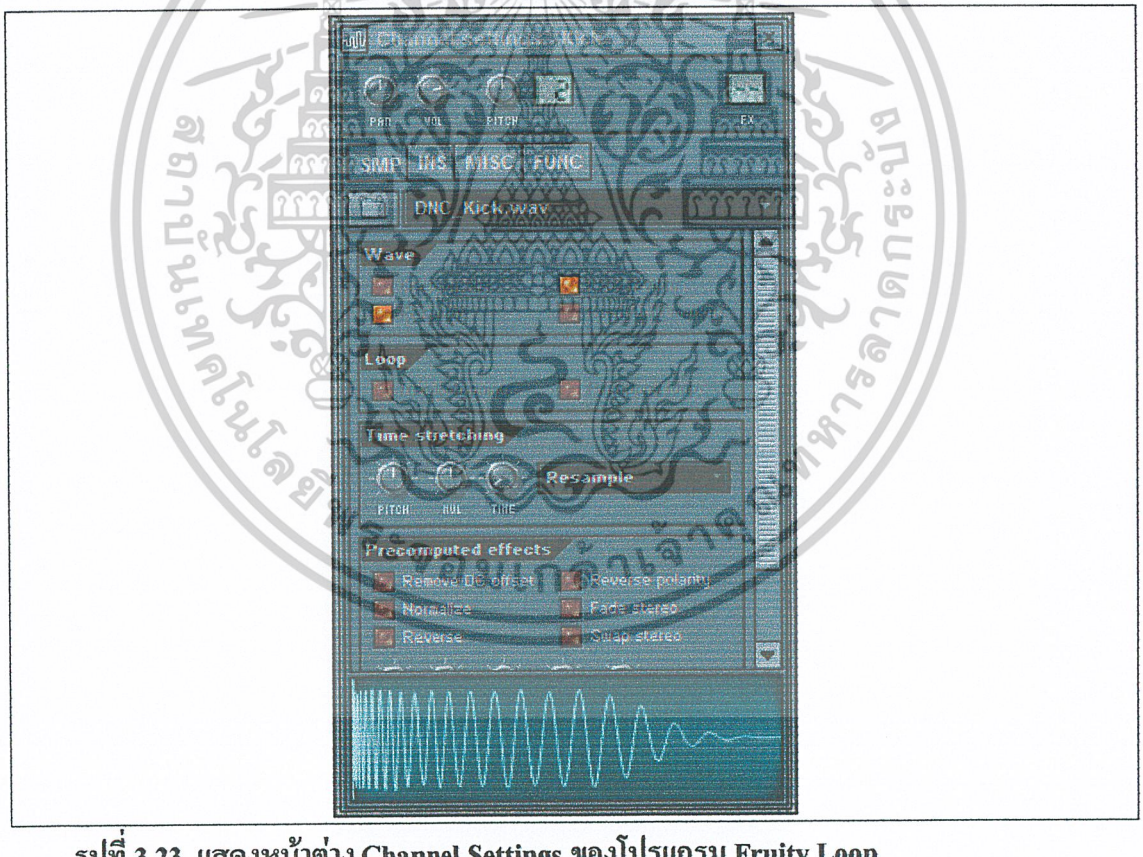
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าต่างที่แสดงเครื่องดนตรีที่เราเลือกไว้สำหรับทำเพลง ซึ่งแต่ละเครื่องดนตรีจะเรียกว่า channel



รูปที่ 3.22 แสดงหน้าต่าง Channels ของโปรแกรม Fruity Loop

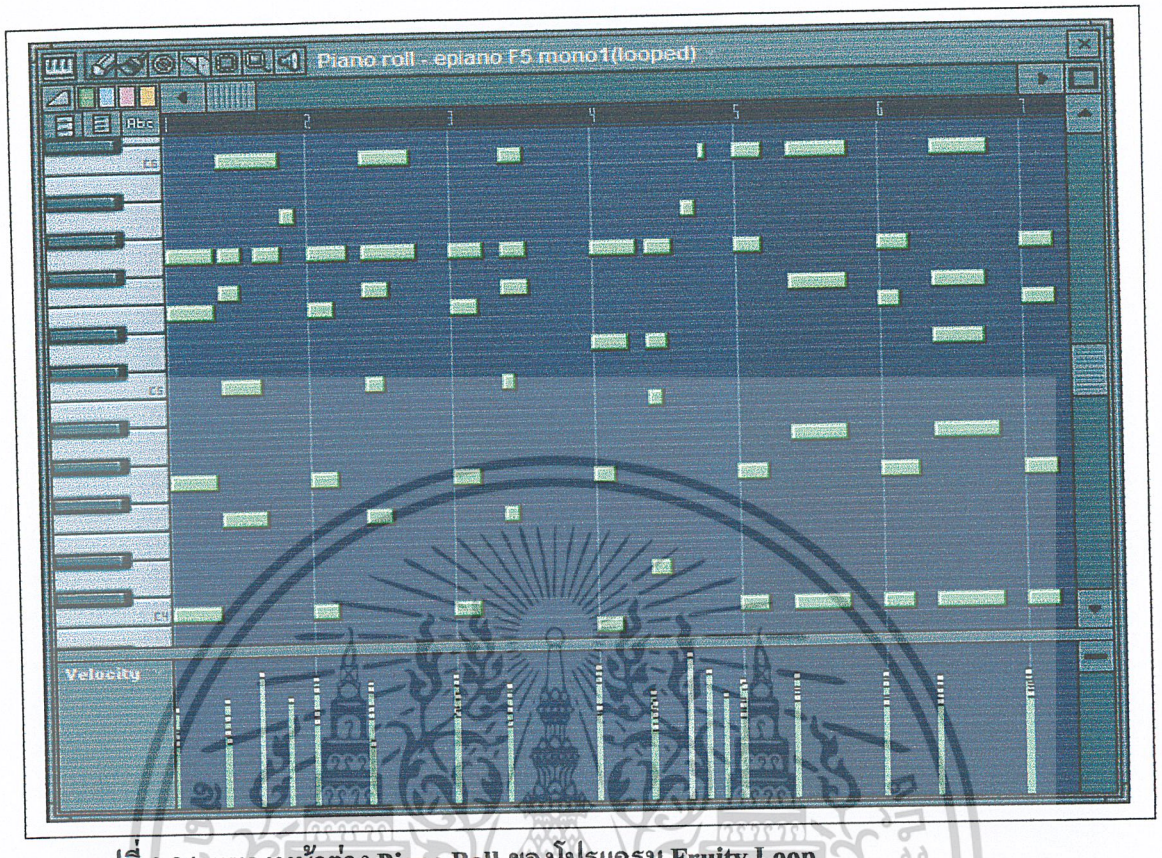
3. หน้าต่างสำหรับปรับแต่งเสียงเครื่องดนตรี



รูปที่ 3.23 แสดงหน้าต่าง Channel Settings ของโปรแกรม Fruity Loop

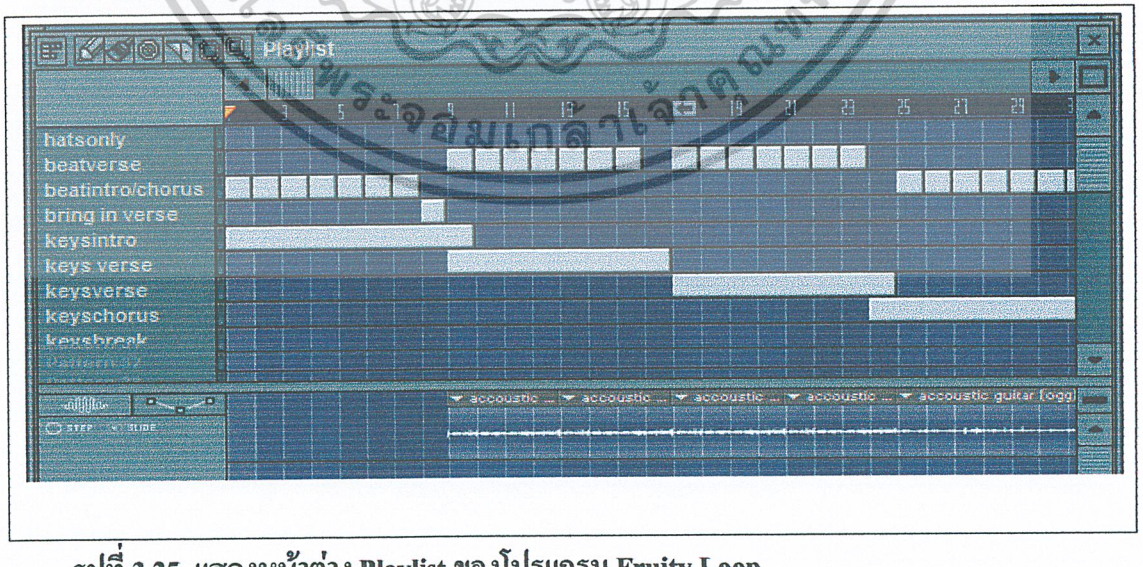
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน้าต่าง Piano Roll ใช้สำหรับพิมพ์โน้ตของแต่ละ Channel



รูปที่ 3.24 แสดงหน้าต่าง Piano Roll ของโปรแกรม Fruity Loop

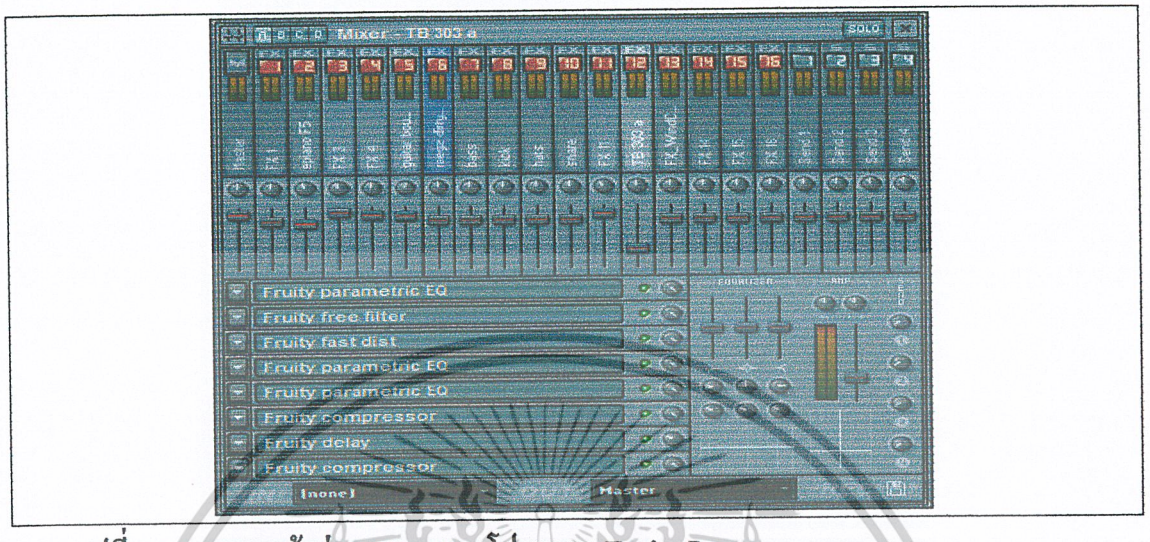
5. หน้าต่าง Playlist สำหรับนำ Pattern ย่อย ๆ ของเพลงที่สร้างไว้มาต่อให้เป็นเพลงเดียวกัน



รูปที่ 3.25 แสดงหน้าต่าง Playlist ของโปรแกรม Fruity Loop

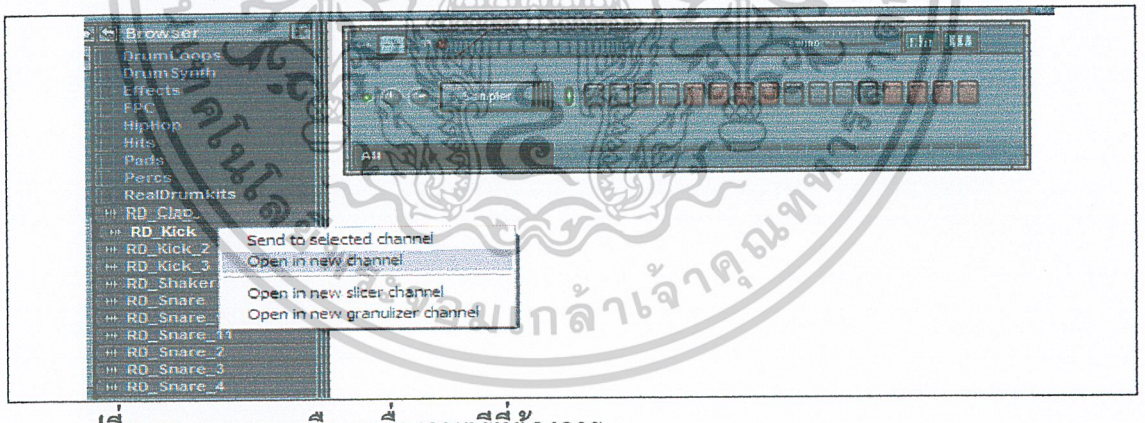
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หน้าต่าง Effect ใช้สำหรับเพิ่มซาวน์เอฟเฟกต์ให้กับ Channel ที่เราต้องการ



รูปที่ 3.26 แสดงหน้าต่าง Effect ของโปรแกรม Fruity Loop

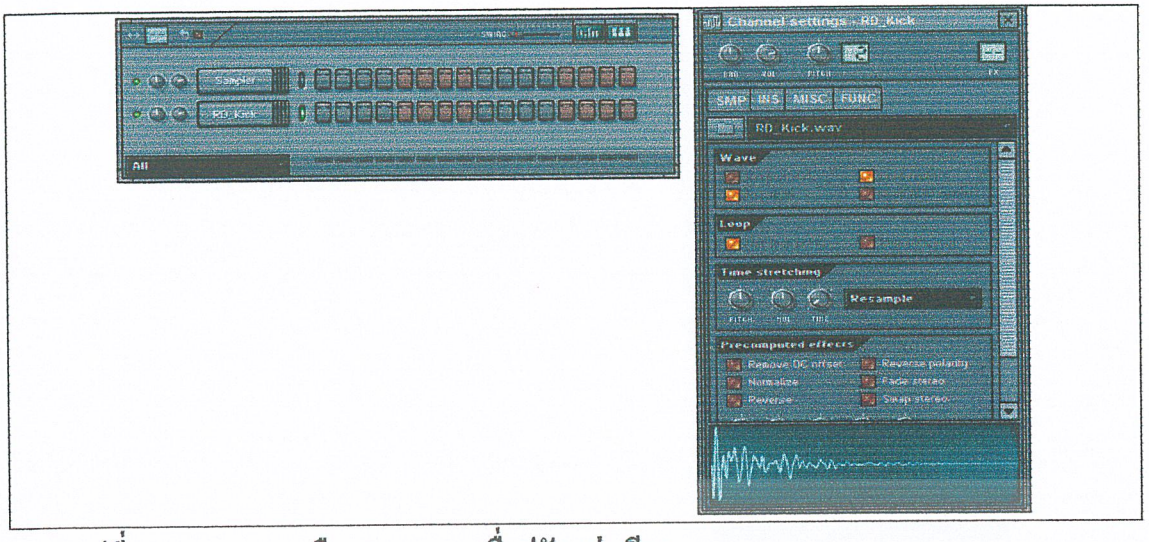
7. เลือกเสียงเครื่องดนตรีที่ต้องการจากหน้าต่าง Browse เมื่อเลือกเสียงที่ต้องการได้แล้วให้คลิกขวา -> Send To New Channel หรือ Send To Select Channel ก็ได้



รูปที่ 3.27 แสดงการเลือกเครื่องดนตรีที่ต้องการ

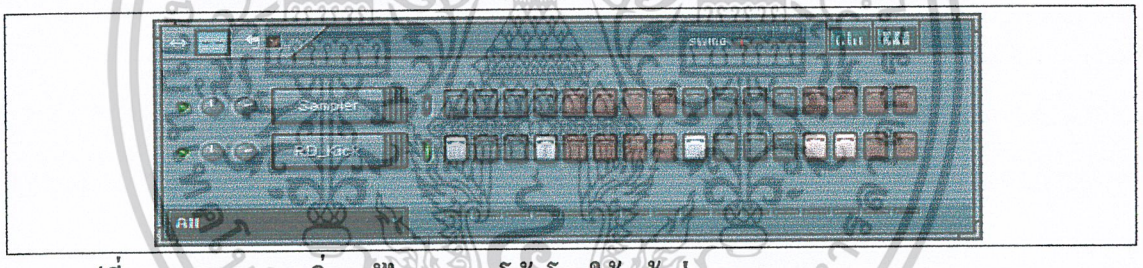
ซึ่งจะทำให้ได้ Channel ใหม่ นั่นคือ RD_Kick ดังรูป ถ้าต้องการปรับแต่งเสียงของ RD_Kick ก็คลิกที่ปุ่มของ RD_Kick หน้าต่าง Channel settings จะปรากฏขึ้นมาดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 แสดงการเลือก Channel เพื่อปรับแต่งเสียง

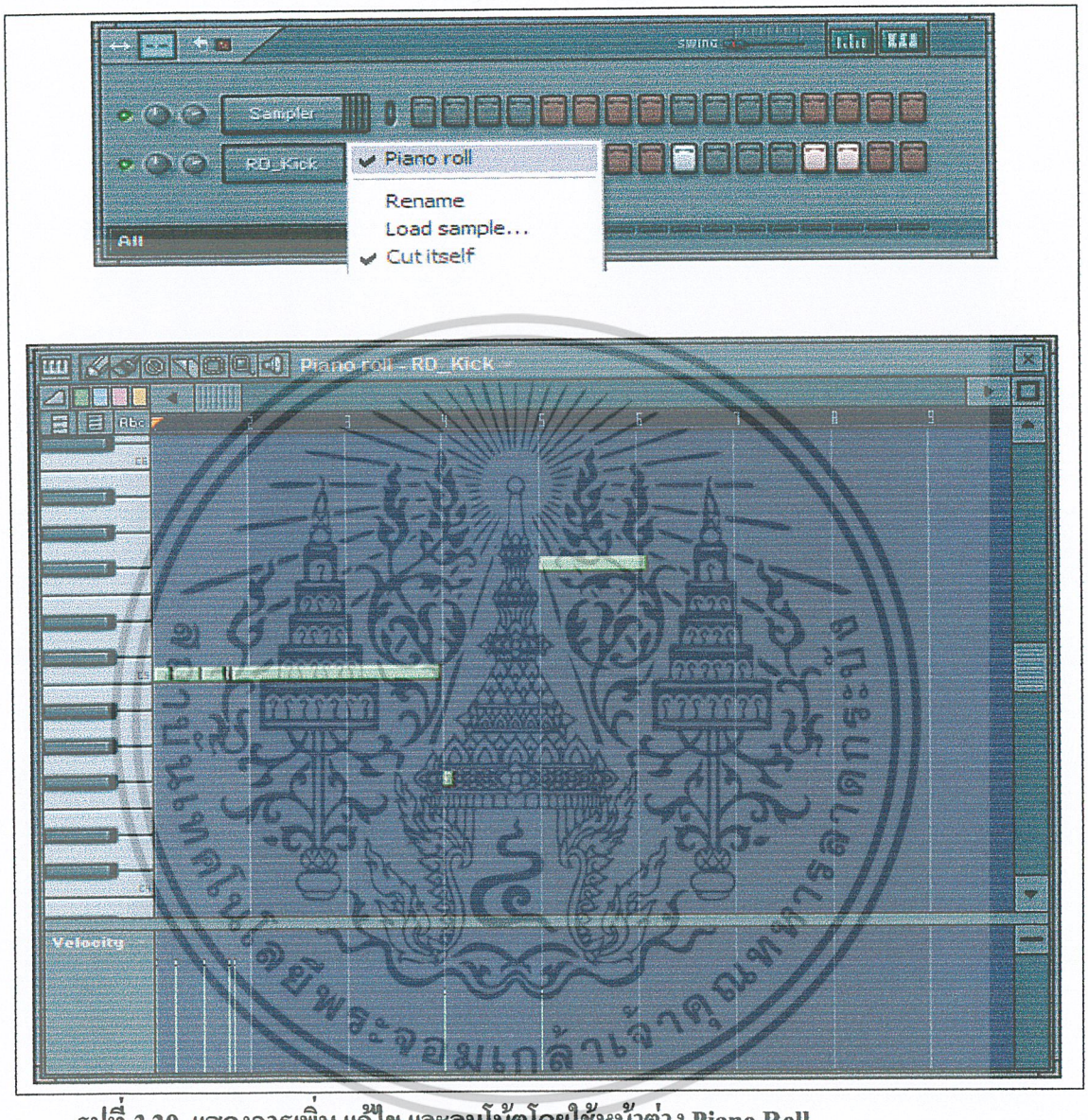
8. เพิ่ม แก้ไข และลบโน้ตของ Channel นั้น ๆ ได้ด้วยหน้าต่าง Channels หรือ Piano Roll ก็ได้ ส่วนใหญ่แต่งโน้ตจากหน้าต่าง Channels จะใช้สำหรับเครื่องดนตรีประเภทกลอง ทำได้โดยการคลิกแต่ละบาร์ของ channel นั้น ๆ เมื่อต้องการลบให้คลิกขวาที่บาร์นั้น



รูปที่ 3.29 แสดงการเพิ่ม แก้ไข และลบโน้ตโดยใช้หน้าต่าง Channels

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

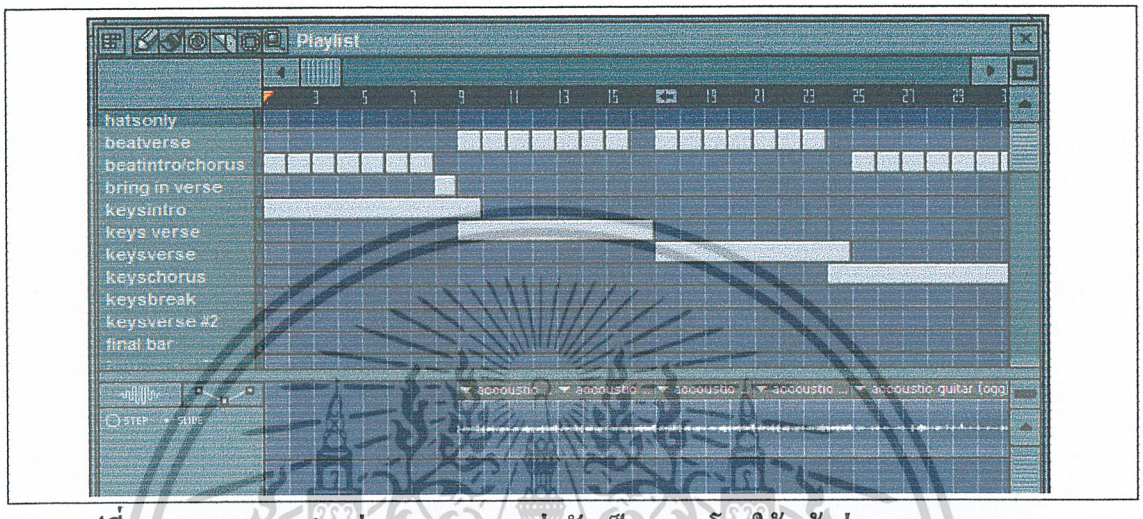
ส่วนการแต่งโน้ตจากหน้าต่าง Piano Roll ทำได้โดยคลิกขวาเลือก channel ที่ต้องการ -> Piano Roll แล้วคลิกโน้ตที่ต้องการลงในช่อง ถ้าต้องการลบให้คลิกขวา ในหน้าต่าง Piano Roll นี้สามารถใช้เมาส์ลากเพื่อเพิ่มความยาวให้โน้ตได้



รูปที่ 3.30 แสดงการเพิ่ม แก้ไข และลบโน้ตโดยใช้หน้าต่าง Piano Roll

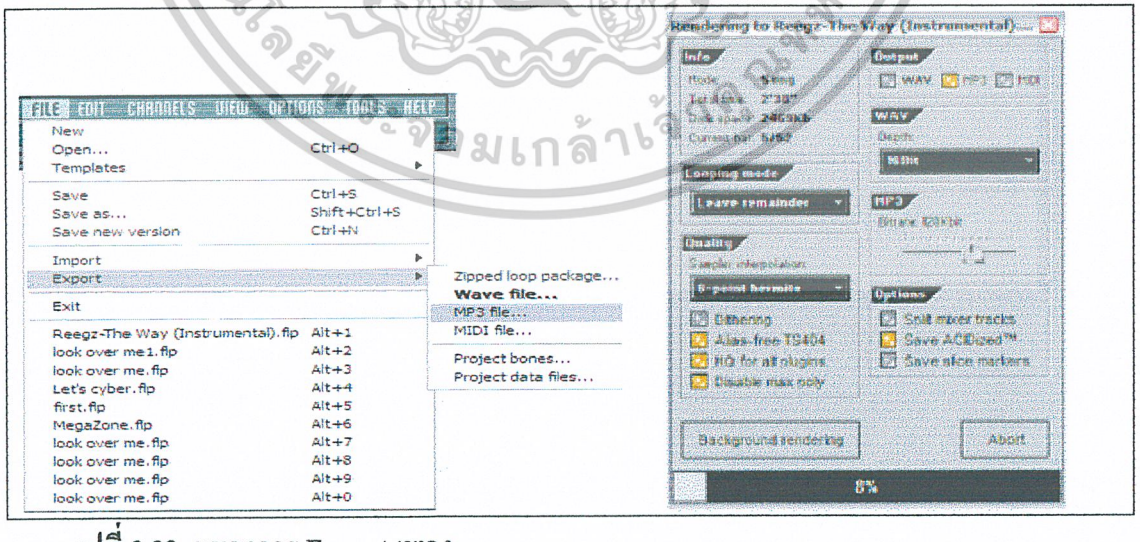
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ในการแต่งเพลงเพลงหนึ่ง เราสามารถแต่งเพลงทีละส่วนเล็ก ๆ ที่เรียกว่า Pattern ได้ เมื่อแต่งแต่ละ Pattern เสร็จแล้วก็นำแต่ละ Pattern มารวมกันโดยใช้หน้าต่าง Songlist ซึ่งสามารถเรียงลำดับก่อนหลังของการเล่นแต่ละ Pattern ตามที่เราต้องการได้ นอกจากนั้นยังสามารถนำไฟล์เสียงอื่น ๆ มาใส่ในเพลงของเราได้ที่ด้านล่างของหน้าต่าง ดังรูป



รูปที่ 3.31 แสดงการนำแต่ละ Pattern มาต่อกันเป็นเพลง โดยใช้หน้าต่าง Playlist

10. เมื่อได้เพลงตามที่ต้องการแล้วก็สามารถ Export เพลงให้เป็นไฟล์เพื่อนำไปฟังในเครื่องเล่นต่าง ๆ โดยไปที่เมนู FILE -> Export แล้วเลือกว่าต้องการบันทึกเป็นไฟล์ประเภทใด ชื่อไฟล์ว่าอะไร ดังรูป

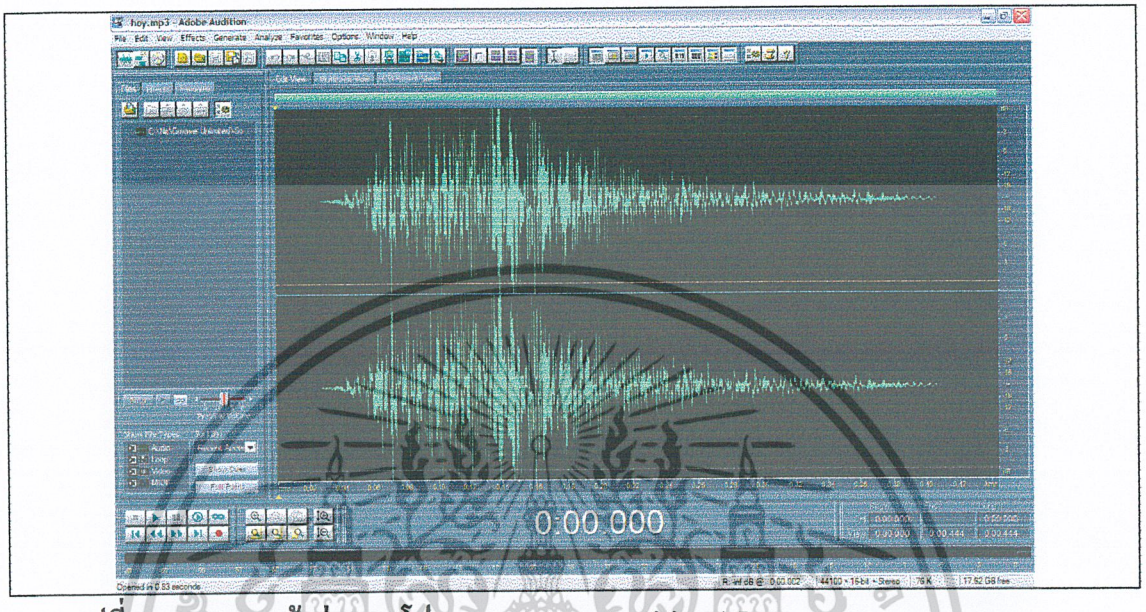


รูปที่ 3.32 แสดงการ Export เพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

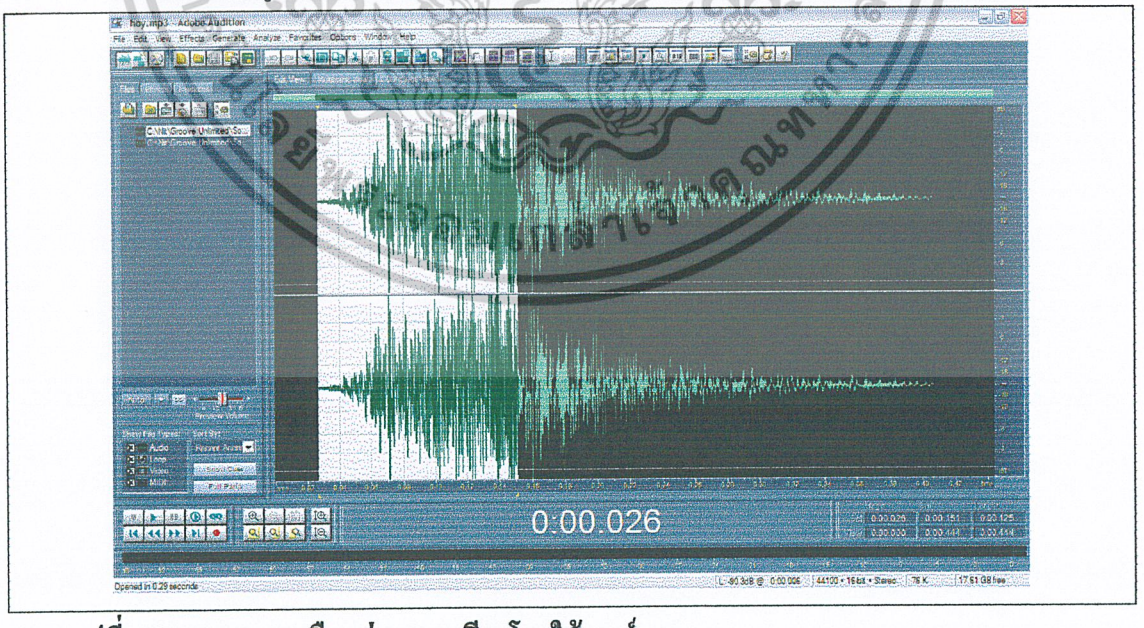
3.2.2.2 การตัดต่อเพลงด้วยโปรแกรม Adobe Audition 1.5 มีวิธีการดังนี้

1. เปิดโปรแกรม Adobe Audition 1.5 ขึ้นมาแล้วเปิดไฟล์ที่ต้องการทำการตัดต่อ โดยไปที่เมนู File -> Open และเลือกไฟล์ที่ต้องการ จะได้หน้าต่างดังรูป หน้าต่างย่อยทางซ้ายมือแสดงถึงไฟล์ที่กำลังถูกเปิดอยู่ในขณะนั้น



รูปที่ 3.33 แสดงหน้าต่างของโปรแกรม Adobe Audition 1.5

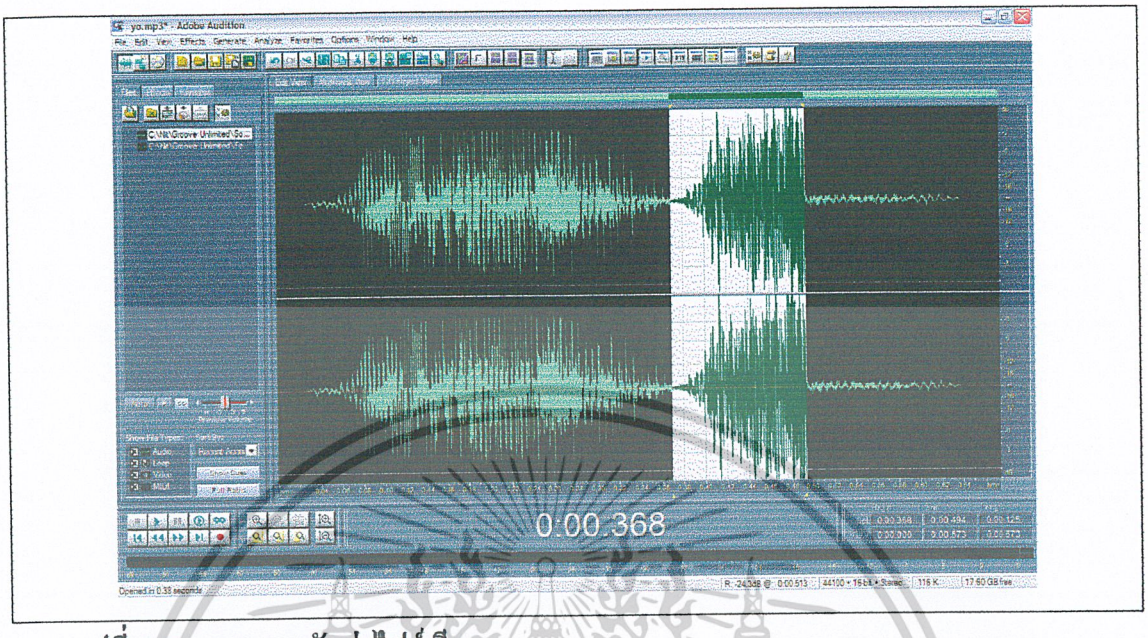
2. ใช้เมาส์ลากส่วนของเสียงที่ต้องการแล้วทำการคัดลอกโดยกด Ctrl + C หรือไปที่เมนู Edit -> Copy



รูปที่ 3.34 แสดงการเลือกส่วนของเสียงโดยใช้เมาส์ลาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. นำไปวางรวมกับไฟล์เสียงอีกไฟล์ในตำแหน่งที่ต้องการ ดังรูป



รูปที่ 3.35 แสดงการตัดต่อไฟล์เสียง

3.3 ขั้นตอนการเขียนโปรแกรม

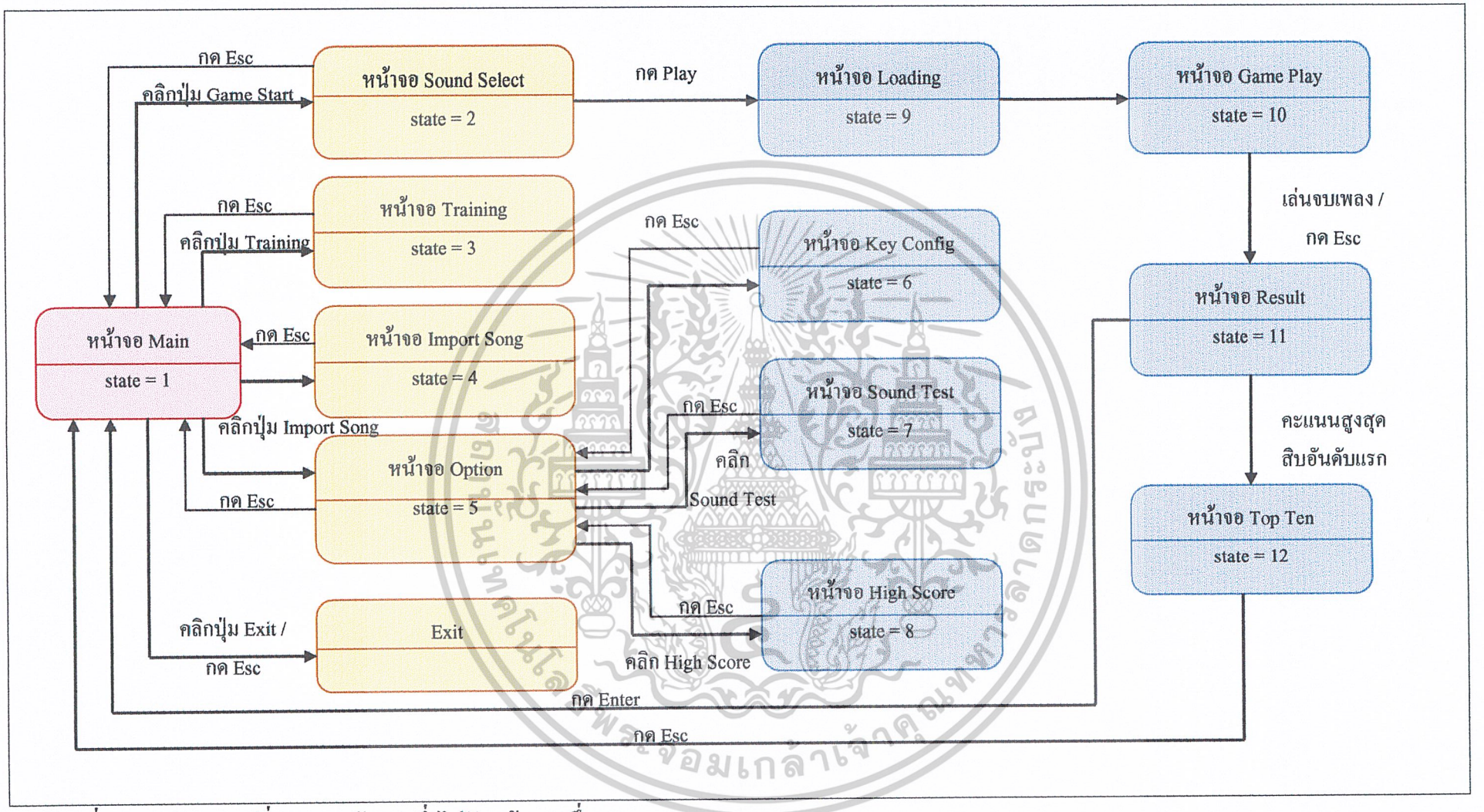
ในการพัฒนาโปรแกรมที่เป็นเกม คำสั่งควบคุมการทำงานทุกอย่างจะเขียนอยู่ภายในลูป เพื่อเป็นการง่ายในการเขียนโปรแกรม และง่ายในการตรวจสอบ แก้ไข หรือเพิ่มเติม จึงได้แบ่งการเขียนโปรแกรมให้ขึ้นอยู่กับแต่ละหน้าจอ โดยการตรวจสอบว่าขณะนี้หน้าจอของผู้เล่นเป็นหน้าจอใด แล้วเขียนโปรแกรมควบคุมการทำงานให้กับหน้าจอนั้น ๆ โดยภายในหนึ่งหน้าจอจะมีการเขียนโปรแกรมควบคุมการทำงานหลัก ๆ ดังนี้

- แสดงฉากหลัง รูปภาพและปุ่มต่าง ๆ ของหน้าจอนั้น
- รอรับอินพุตจากผู้เล่นผ่านทางการคลิกเมาส์หรือการพิมพ์จากคีย์บอร์ด
- เขียนโปรแกรมรองรับเหตุการณ์ต่าง ๆ ตามที่ผู้เล่นป้อนอินพุตเข้ามา

ในการตรวจสอบว่าขณะนี้หน้าจอใดแสดงอยู่ และเมื่อมีการคลิกปุ่มเพื่อเปลี่ยนหน้าจะไปแสดงหน้าจอใด จะใช้ค่าแต่ละค่าของตัวแปรที่ชื่อว่า state เป็นตัวแทนของแต่ละหน้าจอ แล้วใช้คำสั่งควบคุม Switch-Case ตรวจสอบเงื่อนไขว่าขณะนี้ state มีค่าเป็นเท่าใด ก็เข้าไปทำงานตามคำสั่งที่อยู่ใน case นั้น

ซึ่งในการเขียนโปรแกรมจะตั้งค่าให้ state เท่ากับ 1 เป็นค่าเริ่มต้น นั่นคือ เมื่อเปิดโปรแกรมให้เข้ามาที่หน้าจอ Main รูปภาพต่อไปนี้เป็นรูปภาพที่แสดงการเปลี่ยนจากหน้าจอหนึ่งไปยังหน้าจออื่น ๆ โดยเริ่มจากหน้าจอ Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 แสดงการเปลี่ยนจากหน้าจอหนึ่งไปยังหน้าจอหนึ่ง

จากแผนภาพดังกล่าวสามารถเขียนอัลกอริธึมควบคุมการทำงานโดยใช้ Switch-Case ได้ดังนี้

1. ประกาศตัวแปร state แบบ Global และตั้งค่าเริ่มต้นให้เท่ากับ 1

```
int     state =1;
```

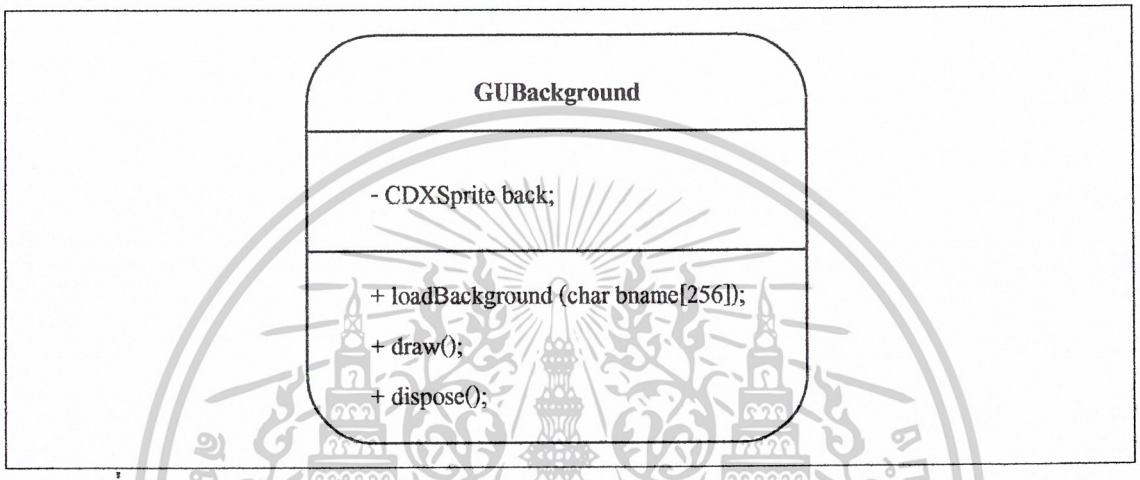
2. การเขียนโปรแกรมสำหรับตรวจสอบว่าขณะนี้ค่าของตัวแปร state เป็นเท่าไร ซึ่งเขียนในฟังก์ชัน cdx_DoFrame()

```
switch (state)
{
        /*========== Main ==========*/
        case 1:
        /*========== Game Start -> Sound Select ==========*/
        case 2:
        /*========== Game Start -> Training ==========*/
        case 3:
        /*========== Game Start -> Import Song ==========*/
        case 4:
        /*========== Game Start -> Option ==========*/
        case 5:
        /*========== Option -> Key Config ==========*/
        case 6:
        /*========== Option -> Sound Test ==========*/
        case 7:
        /*========== Option -> High Score ==========*/
        case 8:
        /*==========ช= Sound Select -> Loading ==========*/
        case 9:
        /*========== Sound Select -> Game Play ==========*/
        case 10:
        /*========== Game Play -> Result ==========*/
        case 11:
        /*========== Resutl -> Top Ten ==========*/
        case 12:
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะกล่าวถึงการเขียนโปรแกรมภายในแต่ละ case เราจะต้องสร้างคลาสที่ต้องใช้สำหรับการเตรียมและโหลดภาพและตัวละครเพื่อแสดงในแต่ละหน้าจอ ดังนี้

3.3.1 คลาส GUBackground เป็นคลาสสำหรับการโหลดและแสดงรูปภาพ background มีแอททริบิว และฟังก์ชัน ดังนี้



รูปที่ 3.37 แสดงโครงสร้างของคลาส GUBackground

3.3.1 แอททริบิว

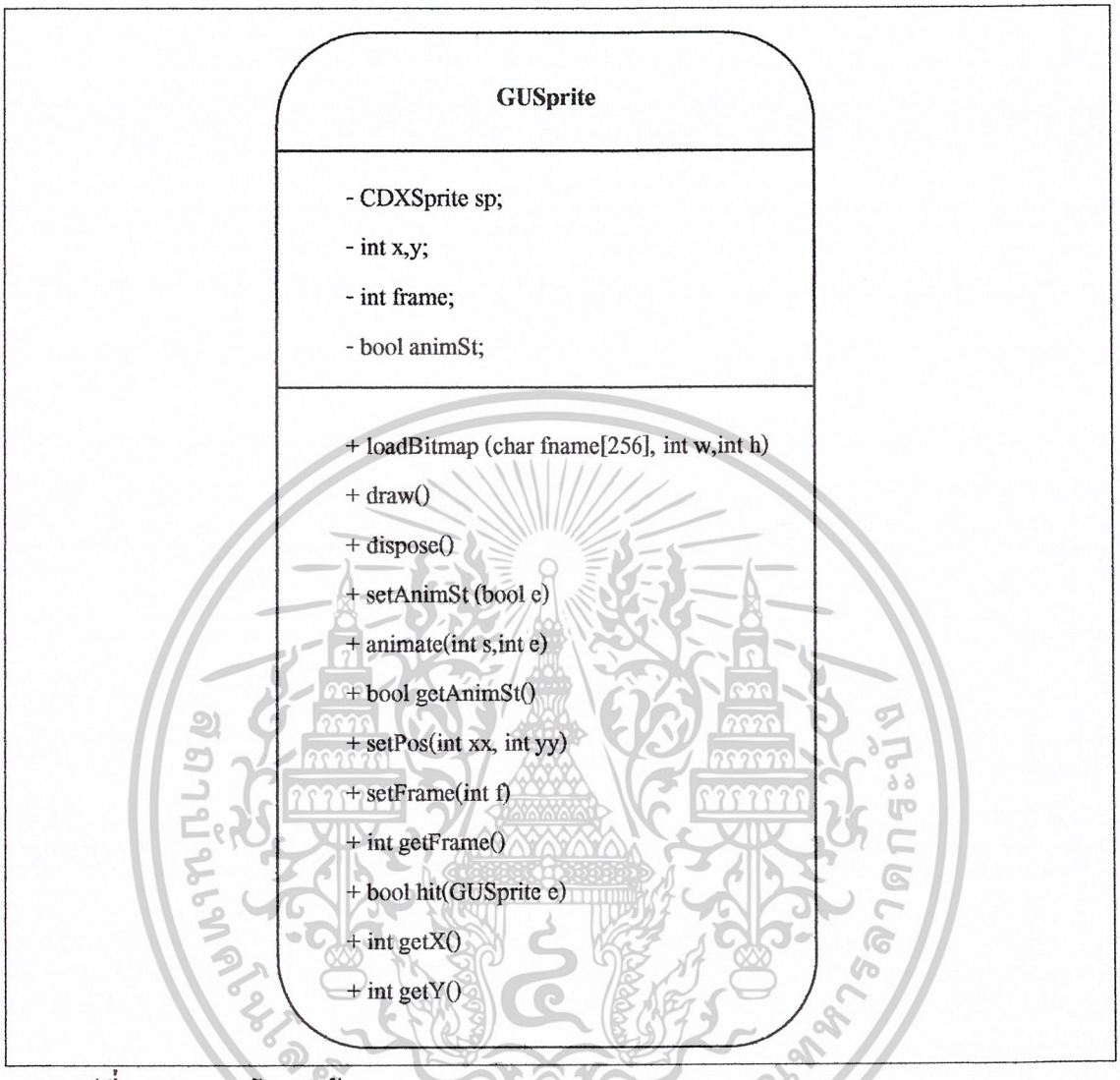
- back เป็นออบเจกต์ของคลาส CDXSprite ซึ่งเป็นคลาสพื้นฐานของ CDX

3.3.2 เมทธอด

- loadBackground มีการรับพารามิเตอร์แบบอักขระขนาด 255 ตัว ชื่อว่า bname ฟังก์ชันนี้ใช้สำหรับโหลดไฟล์รูปภาพ background สำหรับหน้าจอนั้น ๆ มาเก็บไว้ในแอททริบิว back
- draw เป็นฟังก์ชันสำหรับการวาดภาพในแอททริบิว back ลงบนหน้าจอ
- dispose เป็นฟังก์ชันสำหรับคืนหน่วยความจำให้กับระบบปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 คลาส GUSprite เป็นคลาสสำหรับโหลด, แสดงรูปภาพหรือตัวละคร และทำภาพเคลื่อนไหว มีแอททริบิวและฟังก์ชันต่าง ๆ ดังนี้



รูปที่ 3.38 แสดงโครงสร้างของคลาส GUSprite

3.3.2.1 แอททริบิว

- sp เป็นออบเจกต์ของคลาส CDXSprite ซึ่งเป็นคลาสพื้นฐานของ CDX
- x,y ใช้สำหรับกำหนดตำแหน่งของ sp
- frame ใช้สำหรับกำหนดหมายเลขเฟรมของ sp เพื่อทำให้ภาพเคลื่อนไหว
- animSt ใช้สำหรับบอกว่าขณะนั้น sp มีการเคลื่อนไหวหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 เมทธอด

- loadBitmap มีการรับพารามิเตอร์แบบอักขระขนาด 255 ตัว ชื่อว่า fname ใช้สำหรับรับชื่อไฟล์ และ พารามิเตอร์แบบ integer 2 ตัวได้แก่ w และ h ใช้สำหรับรับขนาดความกว้างและความยาวของภาพ ฟังก์ชันนี้ใช้สำหรับโหลดไฟล์รูปภาพที่เป็นตัวละครของหน้าจอนั้น ๆ มาเก็บไว้ในแอททริบิว sp
- draw เป็นฟังก์ชันสำหรับการวาดภาพในแอททริบิว sp ลงบนหน้าจอ
- dispose เป็นฟังก์ชันสำหรับคืนหน่วยความจำให้กับระบบปฏิบัติการ
- animate ใช้สำหรับทำให้ sp เป็นภาพเคลื่อนไหว
- setAnimSt ใช้สำหรับเปลี่ยนแปลงค่า animSt
- getAnimSt ใช้สำหรับเรียกดูค่าของ animSt
- setPos ฟังก์ชันนี้มีการรับพารามิเตอร์ 2 ตัว ได้แก่ xx และ yy ใช้สำหรับตั้งค่าให้กับแอททริบิว x และ y ตามลำดับ
- setFrame มีการรับพารามิเตอร์ f เพื่อใช้สำหรับการตั้งค่าให้แอททริบิว frame
- getFrame ใช้สำหรับเรียกดูค่าของ frame
- getX ใช้สำหรับเรียกดูค่าของ x
- getY ใช้สำหรับเรียกดูค่าของ y
- hit มีการรับพารามิเตอร์ GUSprite e เพื่อตรวจสอบว่าแอททริบิว sp มีการชนกับตัวละครที่ส่งเข้ามาหรือไม่

เมื่อสร้างคลาสดังกล่าวเรียบร้อยแล้ว ให้ประกาศออบเจกต์ของคลาสและตัวแปรเป็นแบบ global ดังนี้

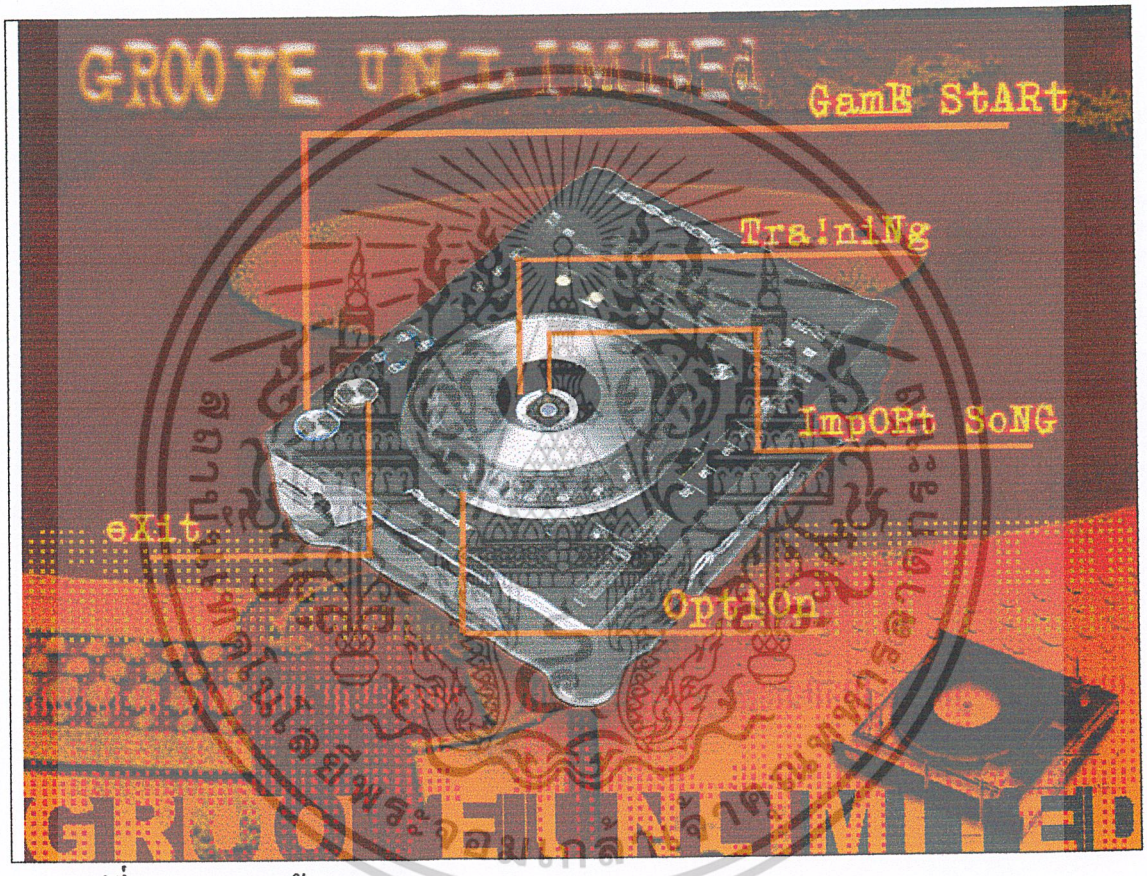
```
GUBackground bg;        //สำหรับเก็บภาพ background
GUSprite tsp[0],mouse;  //tsp สำหรับเก็บตัวละคร,mouse สำหรับภาพของเมาส์
bool showed;            // สำหรับดูว่ามีการวาดภาพลงบนหน้าจอหรือไม่
```

ขั้นตอนต่อไปคือเขียนโปรแกรมในแต่ละเงื่อนไขของ Switch และได้แสดงแผนผังการทำงานในแต่ละหน้าจอ ซึ่งส่วนของการไหลไปของกิจกรรมในระบบเกมถือว่าเป็นส่วนที่สำคัญของการเขียนและเข้าใจภาพรวมของการโปรแกรม และมีความสำคัญต่อการเล่น เพราะจะทำให้เห็นถึงกิจกรรมที่เกิดขึ้นในเหตุการณ์หนึ่ง ๆ นั้นเป็นอย่างไร และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการเปลี่ยนแปลงเหตุการณ์ กิจกรรมไหนของเกมที่จะเกิดขึ้นต่อมา โดยเมื่อเราทราบการไหลไปของระบบเกมโดยละเอียดแล้ว ก็จะเป็นการง่ายต่อการเขียนโปรแกรม เพื่อควบคุมกิจกรรมและเหตุการณ์ทั้งหมดของเกม

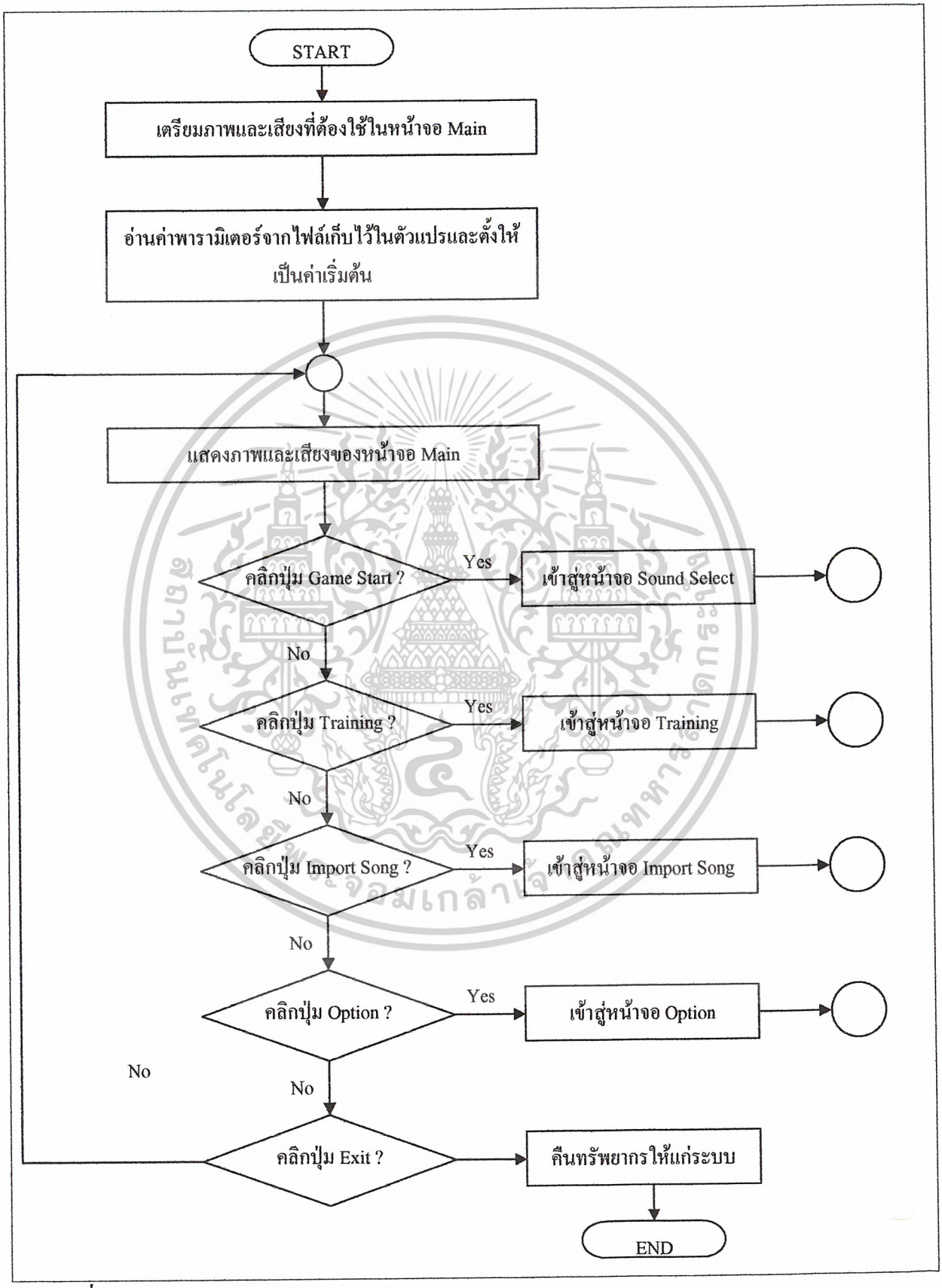
3.3.3 หน้าจอ Main (case 1) ในหน้าจอ Main มีการแสดงภาพฉากหลัง ที่ชื่อว่า main.bmp และมีการแสดงตัวละครเพื่อใช้เป็นปุ่ม 5 ปุ่ม ได้แก่ ปุ่ม Game Start, ปุ่ม Training, ปุ่ม Import Song, ปุ่ม Option และ ปุ่ม Exit ดังรูป



รูปที่ 3.39 แสดงหน้าจอ Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนผังการทำงานดังนี้



รูปที่ 3.40 แผนผังการทำงานในหน้าจอ Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีขั้นตอนในการเขียนโปรแกรมดังนี้

1. โหลดภาพฉากหลัง และตัวละคร โดยใช้คำสั่งต่อไปนี้

```
if(showed==false)
{
        bg.loadBackground("main.bmp");

        tsp[0].loadBitmap("gamestart_frame.bmp",199,36);
        tsp[0].setPos(560,45);
        tsp[0].setFrame(0);

        tsp[1].loadBitmap("training_frame.bmp",163,41);
        tsp[1].setPos(510,138);
        tsp[1].setFrame(0);
        ...
        showed = true;
}
```

//โหลดฉากหลัง

//โหลดตัวละครใส่ออบเจกต์ tsp แล้วตั้งค่าตำแหน่งและเฟรมให้กับตัวละครตามลำดับ

เนื่องจากการเขียนโปรแกรมจะทำอยู่ภายในลูป จึงต้องมีการตรวจสอบว่าภาพและตัวละครถูกวาดหรือไม่ ถ้าไม่จะทำการโหลดภาพและตัวละครใส่ตัวแปรต่าง ๆ เมื่อโหลดเสร็จแล้วจะมีการตั้งค่าว่าโหลดแล้ว นั่นคือ showed = true เมื่อวนลูปมาใหม่แล้วพบว่าภาพมีการโหลดและวาดแล้วจะได้ไม่ต้องทำซ้ำ ถ้าไม่มีการตรวจสอบเงื่อนไขนี้ เมื่อโปรแกรมวนลูปมาทำ ก็จะทำซ้ำเดิมไปเรื่อย ๆ ทำให้ดูเหมือนว่าภาพตัวละครไม่มีการเคลื่อนไหว

2. แสดงภาพ background และตัวละคร ด้วยการใช้คำสั่ง draw() เช่น

```
tsp[0].draw();
```

3. เขียนโปรแกรมรองรับอินพุตทางเมาส์และคีย์บอร์ด ซึ่งมีวิธีการตรวจสอบอินพุตทางเมาส์ และคีย์บอร์ดดังอัลกอลิธึม ต่อไปนี้

1. ตรวจว่าตำแหน่งของเมาส์ตรงกับตำแหน่งของปุ่มหรือไม่โดยใช้เมธอด hit ของคลาส GUSprite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้าตำแหน่งตรงกัน นั่นคือ mouse.hit (tsp) มีค่าเป็นจริง ให้ทำการกระพริบปุ่มนั้น และถ้ามีการคลิกเมาส์เกิดขึ้นก็ให้เปลี่ยนหน้าจอเป็นหน้าจออื่นตามที่คลิก หรือถ้าคลิกที่ปุ่ม exit ก็จะออกจากเกม

ตัวอย่างการเขียนโปรแกรม เป็นดังนี้

```
if (mouse.hit(tsp[0]))
{
        tsp[0].setAnimSt(true);

        if (Input->GetKeyState (CDXKEY_MOUSELEFT)==CDXKEY_RELEASE)
        {
                state=2;
                showed = false;
                break;
        }
}
else if (!mouse.hit(tsp[0]))
{
        tsp[0].setAnimSt(false);
        tsp[0].setFrame(0);
}
```

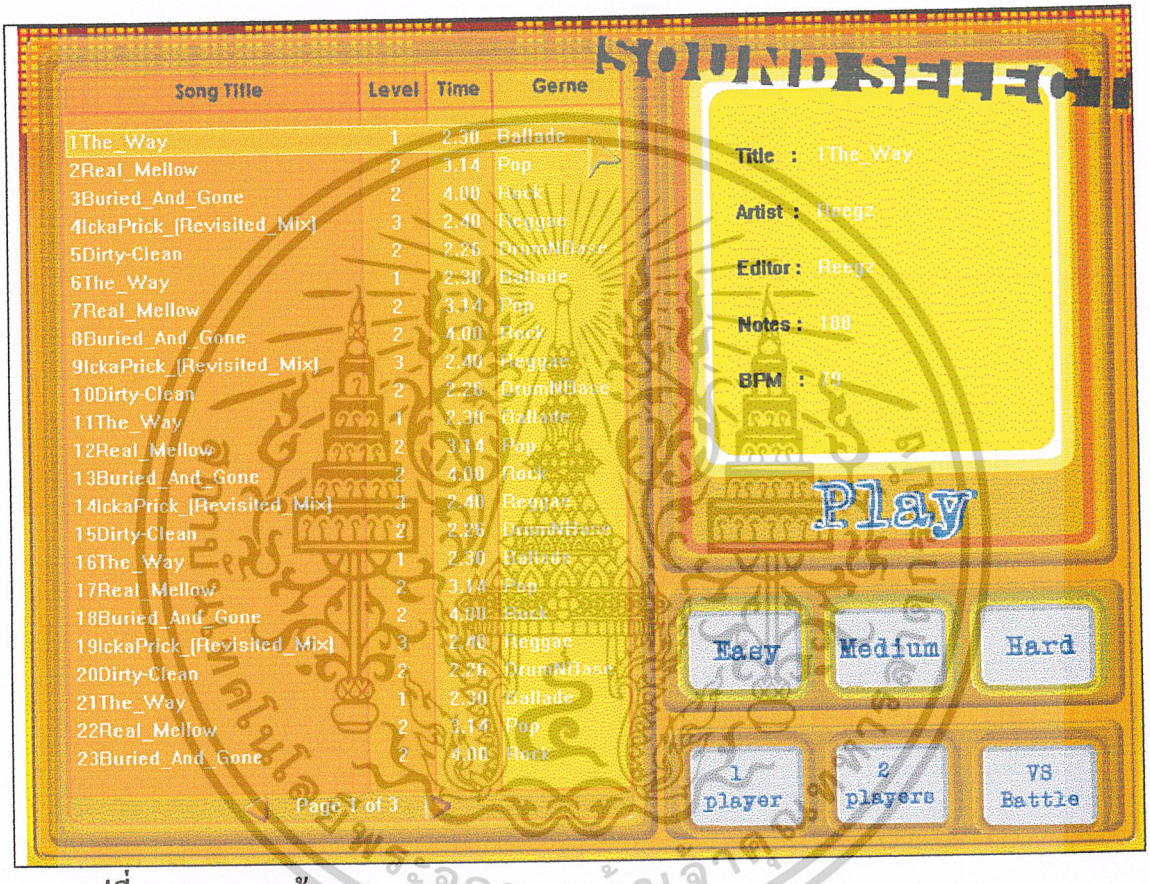
//ถ้านำตำแหน่งเมาส์ไปใกล้ ๆ กับปุ่ม Game Start ให้ทำการกระพริบปุ่ม Game Start

// ถ้ามีการคลิกปุ่ม Game Start ให้เปลี่ยน state = 2 นั่นคือให้แสดงหน้าจอ Sound Select ต่อไป

//ถ้าเมาส์ไม่ได้อยู่ใกล้ ๆ กับปุ่ม Game Start ให้เลิกกระพริบปุ่ม Game Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

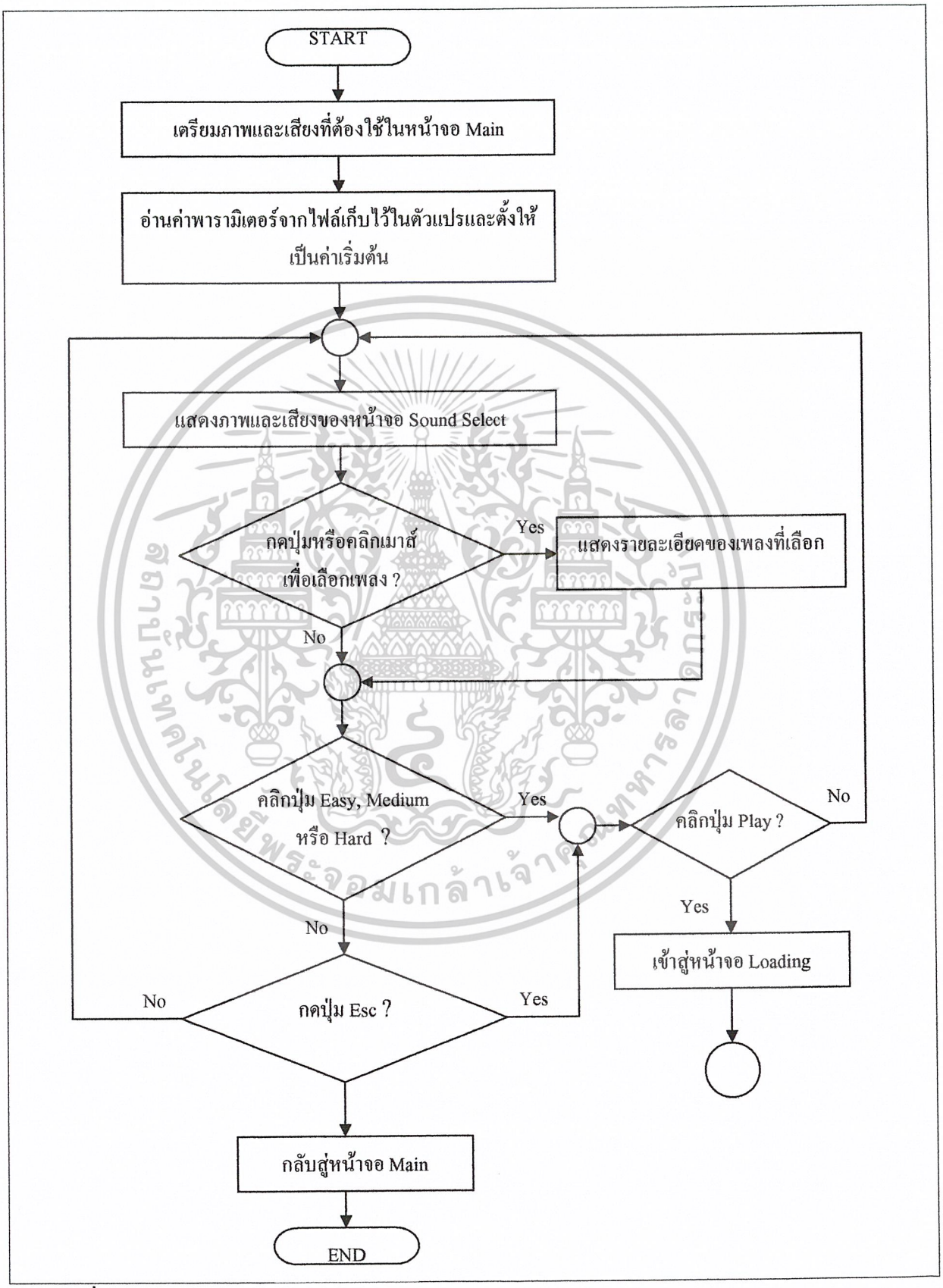
3.3.4 หน้าจอ Sound Select (case 2) ในหน้าจอนี้มีวิธีการในการโหลดและแสดงฉากหลังและตัวละครเหมือนกับหน้าจอ Main แต่มีหน้าที่ในการแสดงรายชื่อเพลงและรายละเอียดของเพลงเพิ่มขึ้นมา คือ จะแสดงรายชื่อเพลงทั้งหมดที่มีในเกมทางด้านซ้ายมือของจอ และแสดงรายละเอียดของเพลงที่เลือกทางด้านมุนขวาบน ผู้เล่นสามารถคลิกปุ่มซ้าย ขวา เพื่อเปลี่ยนหน้าของรายชื่อเพลงได้ ในหนึ่งหน้าจอแสดงได้ 23 เพลง เมื่อผู้เล่นเลือกเพลงเรียบร้อยแล้วคลิกปุ่ม play เพื่อเข้าสู่หน้าจอของ Game Play และเริ่มเล่นเกม หน้าจอ Sound Select เป็นดังรูป



รูปที่ 3.41 แสดงหน้าจอ Sound Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนผังการทำงานดังต่อไปนี้

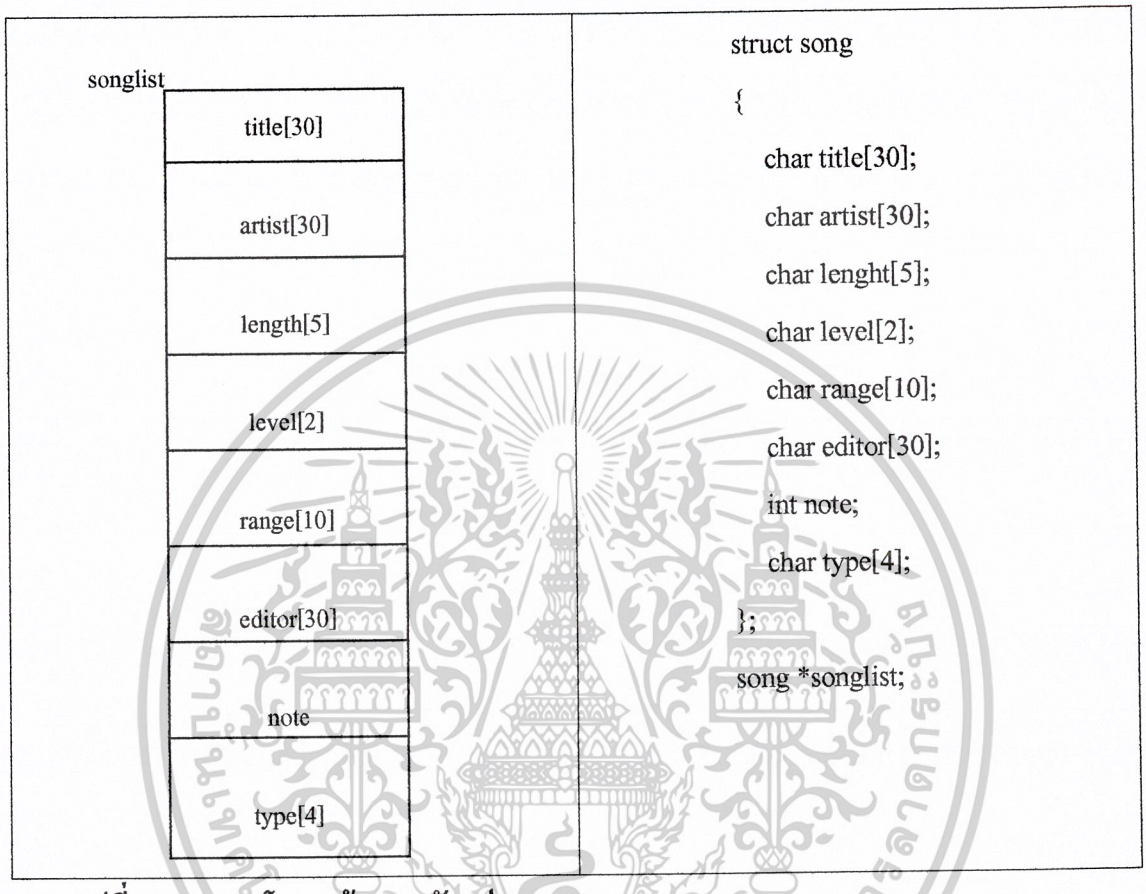


รูปที่ 3.42 แผนผังการทำงานในหน้าจอ Sound Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงรายชื่อเพลงและรายละเอียดของเพลง มีวิธีการดังนี้

1. อ่านข้อมูลเพลงและรายละเอียดของเพลงจากไฟล์ที่ชื่อว่า songlist.txt แล้วมาเก็บไว้ในตัวแปร struct ที่ชื่อ songlist ซึ่งมีโครงสร้างต่อไปนี้



รูปที่ 3.43 แสดงโครงสร้างของตัวแปร songlist

ตัวแปร title สำหรับเก็บชื่อเพลง

ตัวแปร artist สำหรับเก็บชื่อศิลปิน

ตัวแปร length สำหรับเก็บความยาวของเพลง

ตัวแปร level สำหรับเก็บระดับความยากง่ายของเพลง

ตัวแปร range สำหรับเก็บระดับของ spectrum ที่ใช้สำหรับนำเข้าเพลง

ตัวแปร note สำหรับเก็บจำนวนตัวโน้ต

ตัวแปร type สำหรับเก็บชนิดของไฟล์เพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการอ่านไฟล์เป็นดังนี้

```
int readSonglist()
{
        ifstream songList ("songlist.txt");
        if (! songList.is_open())
        {
                SendMessage(g_hWnd, WM_CLOSE, 0, 0);
                return 1;
        }
        songList >> nsongs;
        songlist = new song[nsongs];
        for(int i=0;i<nsongs;i++)
        {
                songList >> (songlist[i].title);
                songList >> (songlist[i].artist);
                songList >> (songlist[i].level);
                songList >> (songlist[i].lenght);
                songList >> (songlist[i].gerne);
                songList >> (songlist[i].editor);
                songList >> (songlist[i].note);
                songList >> (songlist[i].bpm);
        }
        songList.close();
        return 0;
}
```

// nsongs คือ จำนวนเพลงที่มีทั้งหมดใน songlist

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แสดงรายชื่อเพลงที่เก็บอยู่ในตัวแปร songlist ออกทางหน้าจอ โดยต้องดูว่าขณะนี้แสดงอยู่ที่หน้าใด และหน้านั้นมีเพลงแรกและเพลงสุดท้ายเป็นเพลงใด

```
int xx = 35;
int yy = 75;
int cpage = 1;
int begin = 23*(cpage-1);
int end = 0;
int hpos = 0;
```

// xx,yy เป็นตำแหน่งของรายชื่อเพลงแต่ละชื่อ
// cpage เป็นตัวแปรแบบ global ซึ่งตัวเลขแสดงหน้าปัจจุบัน
//begin เป็นตำแหน่งอินเด็กซ์ของเพลงในอาเรย์ songlist ที่เป็นเพลงแรกในแต่ละหน้า
//end เป็นตำแหน่งอินเด็กซ์ของเพลงในอาเรย์ songlist ที่เป็นเพลงสุดท้ายในแต่ละหน้า
//hpos เป็นตำแหน่งล่างสุดที่แถบสีสามารถจะโฟกัสได้ในแต่ละหน้า

```
if (cpage == npages) //Last Page
{
        end = nsongs;
        hpos = nsongs - ((npages-1)*23);
}
else
{
        end = begin+23;
        hpos = 23;
}
```

//ถ้าหน้าที่แสดงอยู่เป็นหน้าสุดท้าย กำหนด end เป็นอินเด็กซ์ของเพลงสุดท้าย และ กำหนด hpos ให้เป็นตำแหน่งของเพลงสุดท้ายนั้น

//ถ้าหน้าที่แสดงอยู่เป็นไม่ใช่หน้าสุดท้าย กำหนด end เป็นอินเด็กซ์ของเพลงแรกบวก 23 และกำหนด hpos ให้เป็น 23 23 (เพราะ 1 หน้าแสดงได้ 23 เพลง)

```
//Display Song List
for(int i=begin;i<end;i++)
{
        Screen->GetBack()->TextXY(xx,yy,RGB(255,255,255),songlist[i].title);
        Screen->GetBack()->TextXY(xx+228,yy,RGB(255,255,255),songlist[i].level);
        Screen->GetBack()->TextXY(xx+265,yy,RGB(255,255,255),songlist[i].lenght);
        Screen->GetBack()->TextXY(xx+305,yy,RGB(255,255,255),songlist[i].gerne);
        yy = yy+20;
        xx = 35;
}
```

// แสดงรายชื่อเพลงแต่ละเพลงโดยแต่ละบรรทัดห่างกัน 20 pixels

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้างแถบสีขึ้นมาเพื่อแสดงให้รู้ว่าเพลงใดถูกเลือกอยู่ และแสดงรายละเอียดของเพลงนั้น การเขียนโปรแกรมเป็นดังนี้

```
int pos=0;
int x=0;
int x=74;
```

// pos เป็นตัวแปรที่บอกว่าขณะนี้แถบสีเลือกอยู่ที่เพลงที่เท่าไรใน songlist (อินเด็กซ์ที่เท่าใด)
// x และ y เป็นตำแหน่งเริ่มต้นของแถบสี

```
if(Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_PRESS)
{
        x=33;
        if(y<=74)
                y=54+20*hpos;
        else
                y=y-20;
        tsp[9].setPos(x,y);
}
if(x==33&&y>=74)
        tsp[9].draw();
```

// ถ้ามีการกดขึ้นไปเรื่อย ๆ จะให้แถบสีเลื่อนขึ้นตามไป โดยเปลี่ยนตำแหน่ง y ให้ลบทีละ 20 เมื่อกดขึ้น และถ้า y น้อยกว่าตำแหน่งของเพลงแรก จะให้ y เป็น ตำแหน่งสุดท้ายของเพลง

// ทำการวาดแถบสี

4. แสดงรายละเอียดของเพลงที่แถบสีกำลังโฟกัสอยู่

```
if(x<=0)
        pos=-1;
else
        pos = (((y-54)/20)-1)+((cpage-1)*23);
if(x>0)
{
        Screen->GetBack()->TextXY(512,92,RGB(0,0,0), "Title  :");
        Screen->GetBack()->TextXY(512,132,RGB(0,0,0),"Artist :");
        Screen->GetBack()->TextXY(512,172,RGB(0,0,0),"Editor :");
                ……
}
```

// x=0 หมายถึง แถบสีไม่ได้โฟกัสที่เพลงใดเลย แต่ถ้า x > 0 นั่นคือ แถบสีไปโฟกัสที่เพลงใดเพลงหนึ่งแล้ว และตำแหน่งของเพลงใน songlist จะมีค่าเท่ากับ pos จึงแสดงรายละเอียดของเพลงในตำแหน่ง songlish[pos] ออกมา

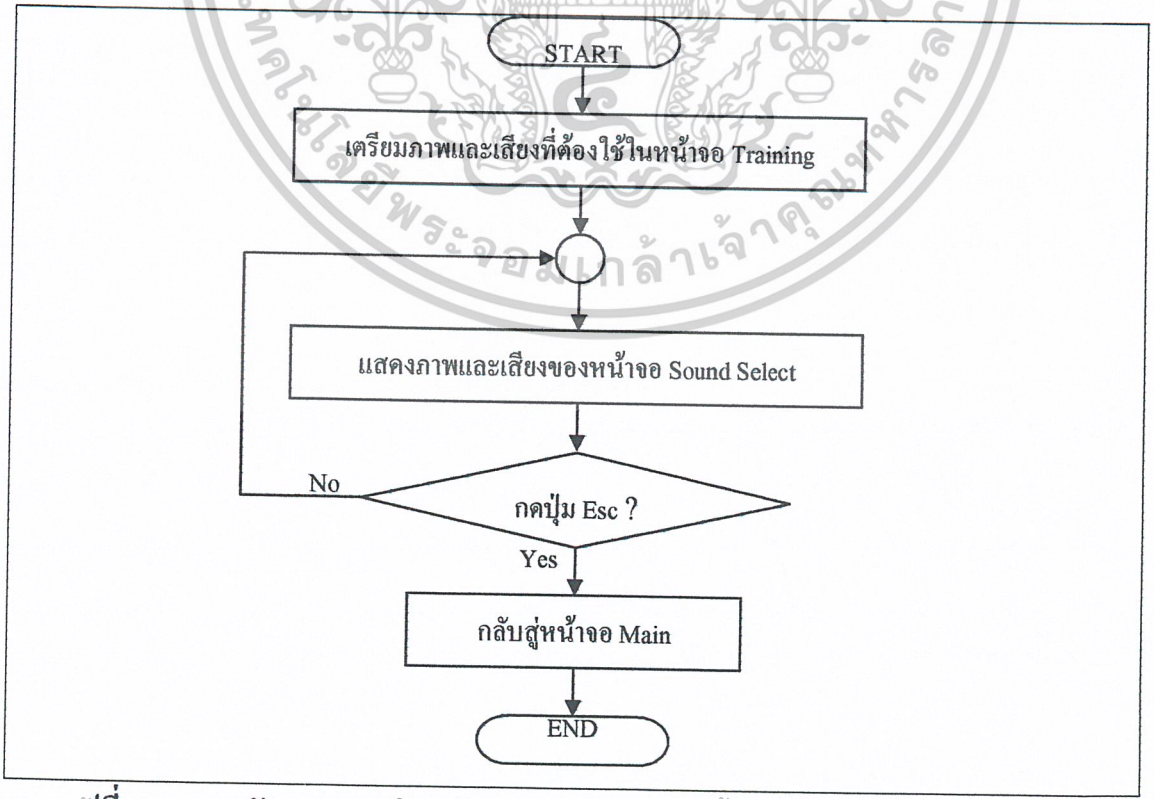
5. เมื่อเลือกเพลงได้แล้ว แล้วกด play เพื่อเข้าสู่หน้าโหลดเพื่อเตรียมเข้าสู่การเล่นเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 หน้าจอ Training (case 3) หน้าที่ของหน้าจอ Training คือ สอนวิธีการเล่นเกมให้กับผู้เล่นมือใหม่ หน้าจอ Training เป็นดังรูป

รูปที่ 3.44 แสดงหน้าจอ Training

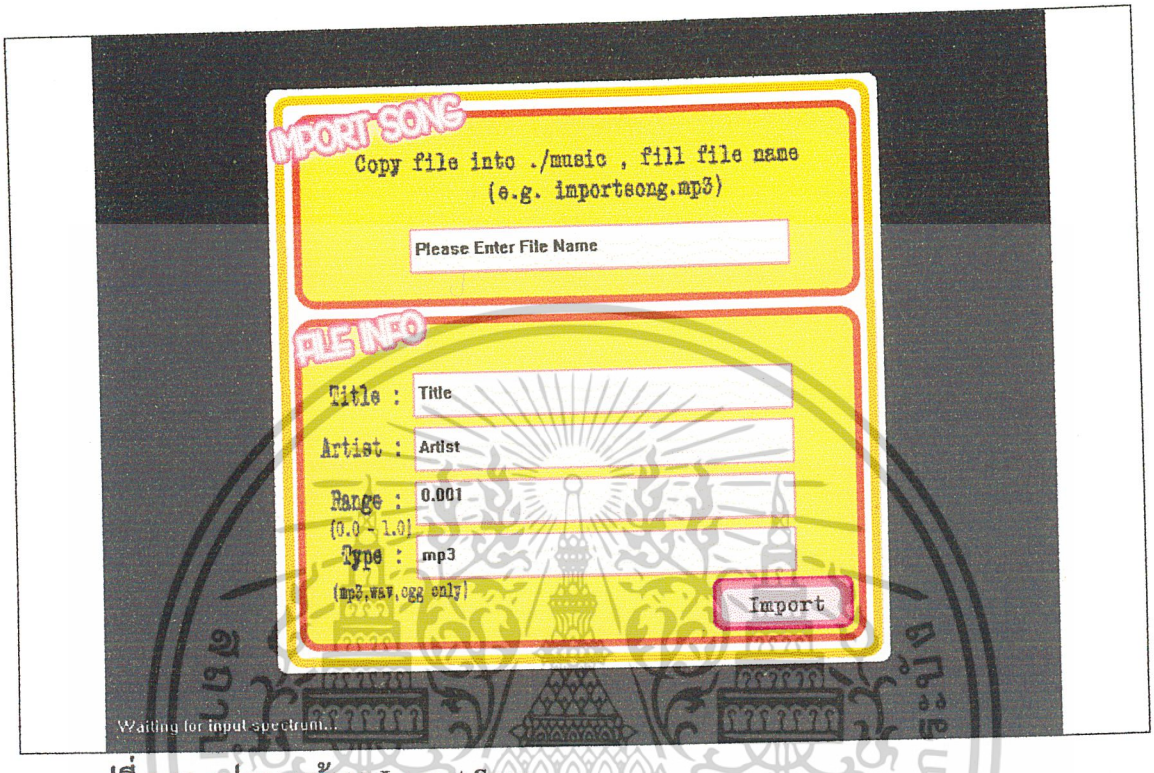
มีแผนผังการทำงานดังรูป



รูปที่ 3.45 แผนผังการทำงานในหน้าจอ Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

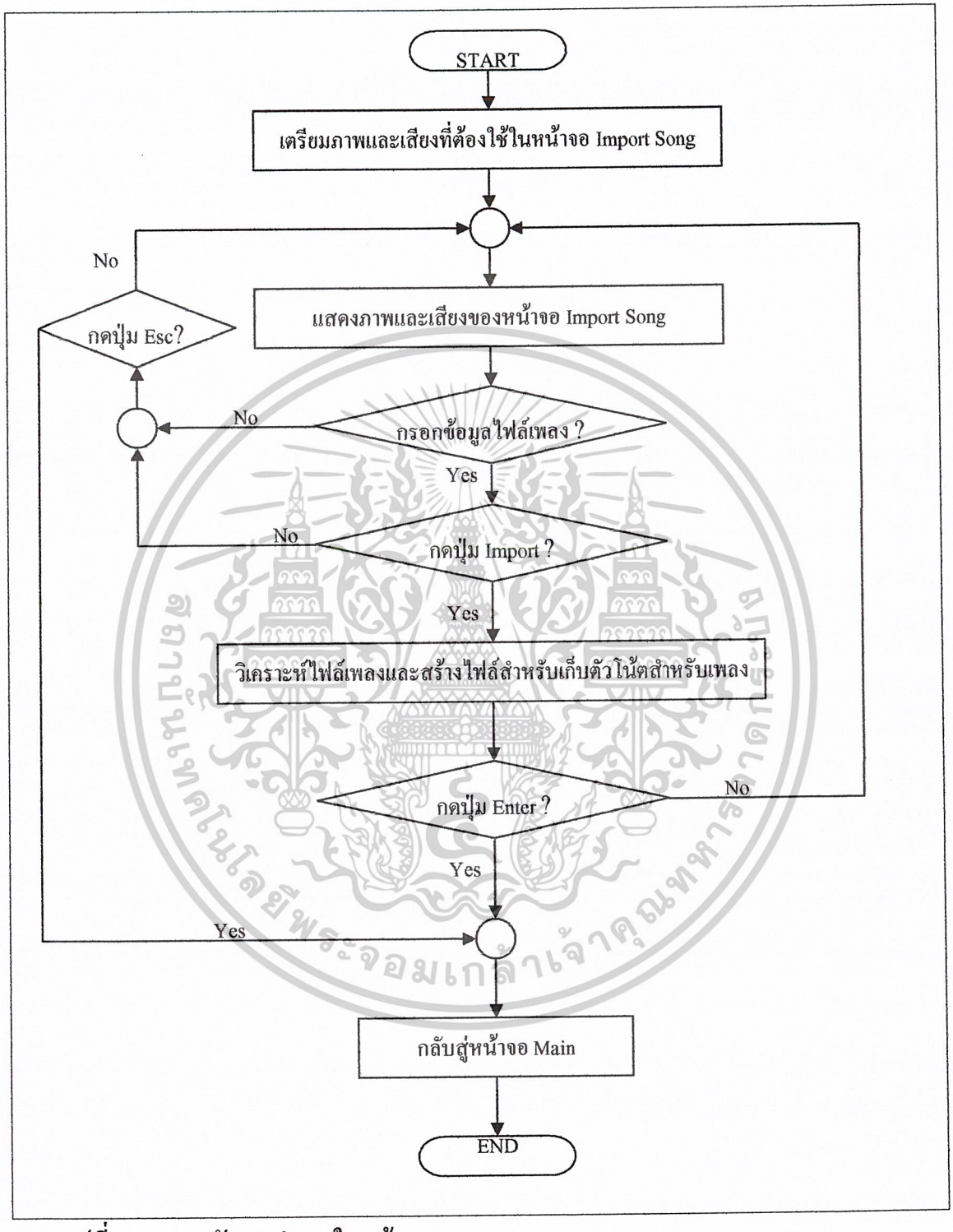
3.3.6 หน้าจอ Import Song (case 4) หน้าจอนี้ใช้สำหรับนำเพลงประเภท .WAV .MP3 หรือ .OGG เข้าสู่เกม ซึ่งเป็นเพลงที่ผู้เล่นต้องการเล่นและไม่ได้มีให้อยู่แล้วในเกม หน้าจอ Import Song เป็นดังรูป



รูปที่ 3.46 รูปของหน้าจอ Import Song

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนผังการทำงานดังนี้



รูปที่ 3.47 แผนผังการทำงานในหน้าจอ Import Song

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีหลักการทำงานดังนี้

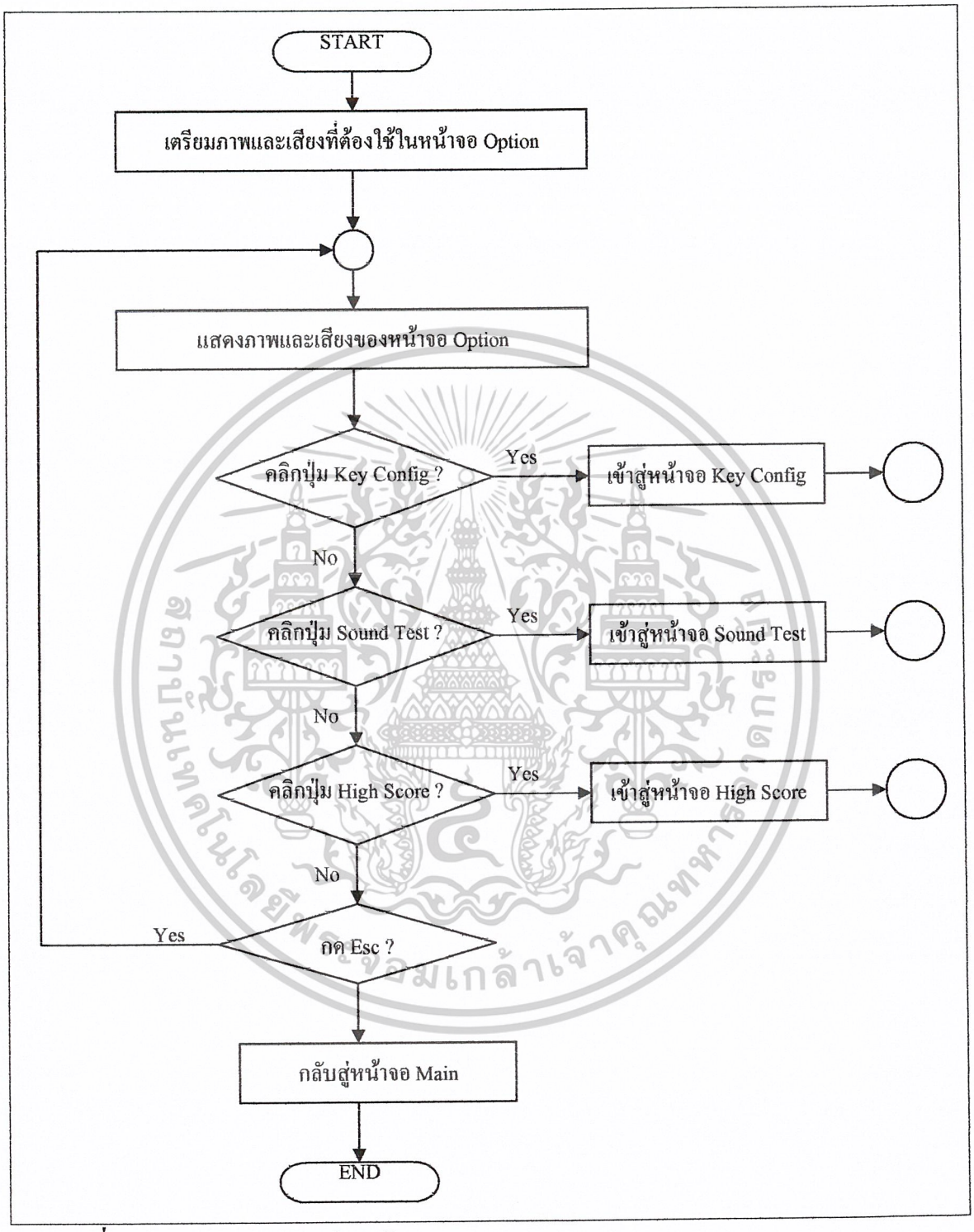
1. ผู้เล่นใส่ข้อมูลเกี่ยวกับเพลงที่จะนำเข้า โดยเพลงที่ต้องการเลือกต้องทำการคัดลอกไปเก็บไว้ในโฟลเดอร์ Import ก่อนเท่านั้น จึงจะสามารถนำเข้าได้ มิฉะนั้นโปรแกรมจะหาเพลงไม่เจอ
2. กดปุ่ม Import เพื่อเริ่มต้นการวิเคราะห์เพลง ซึ่งการวิเคราะห์เพลงจะเป็นการเรียกใช้ฟังก์ชัน FSOUND_DSP_GetSpectrum ของ FMOD เพื่อทำการวิเคราะห์ความถี่ของเพลงนั้น แล้วคัดความถี่ที่มากกว่าค่า range ที่ผู้เล่นกำหนดไว้ ออกมาเป็นโน้ตสำหรับเล่นเกม โดยค่าตัวโน้ตที่ได้มาจากการสุ่มค่าตั้งแต่ค่า 0-5 ซึ่งแทนคีย์ทั้ง 6 คีย์

3.3.7 หน้าจอ Option (case 5) ในหน้าจอนี้มีขั้นตอนในการโหลดและแสดงภาพ background และตัวละครเหมือนกับหน้าจอ Main ภายในหน้าจอนี้มีปุ่มให้เลือกไปเมนูย่อยเพื่อการปรับแต่งค่าซึ่งมี 3 ปุ่ม ได้แก่ ปุ่ม Key Config, ปุ่ม Sound Test และ ปุ่ม High Score ดังรูป

รูปที่ 3.48 แสดงหน้าจอ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

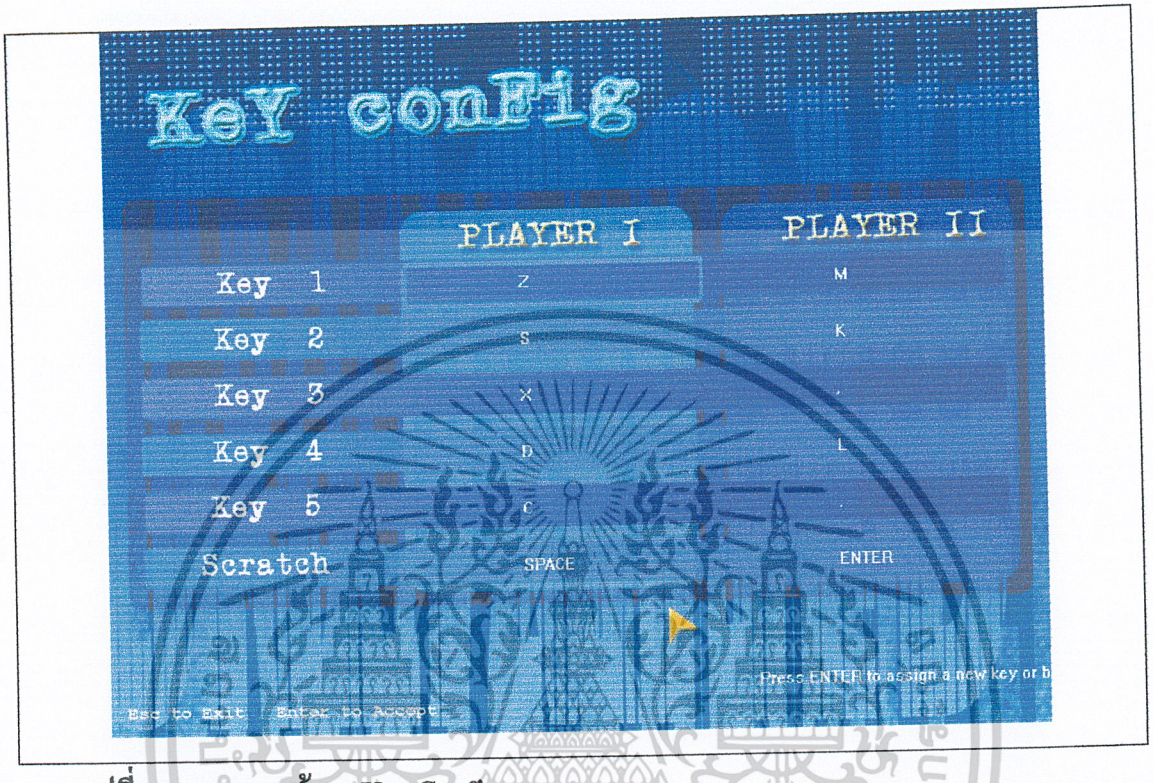
มีแผนผังการทำงานดังนี้



รูปที่ 3.49 แผนผังการทำงานในหน้าจอ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

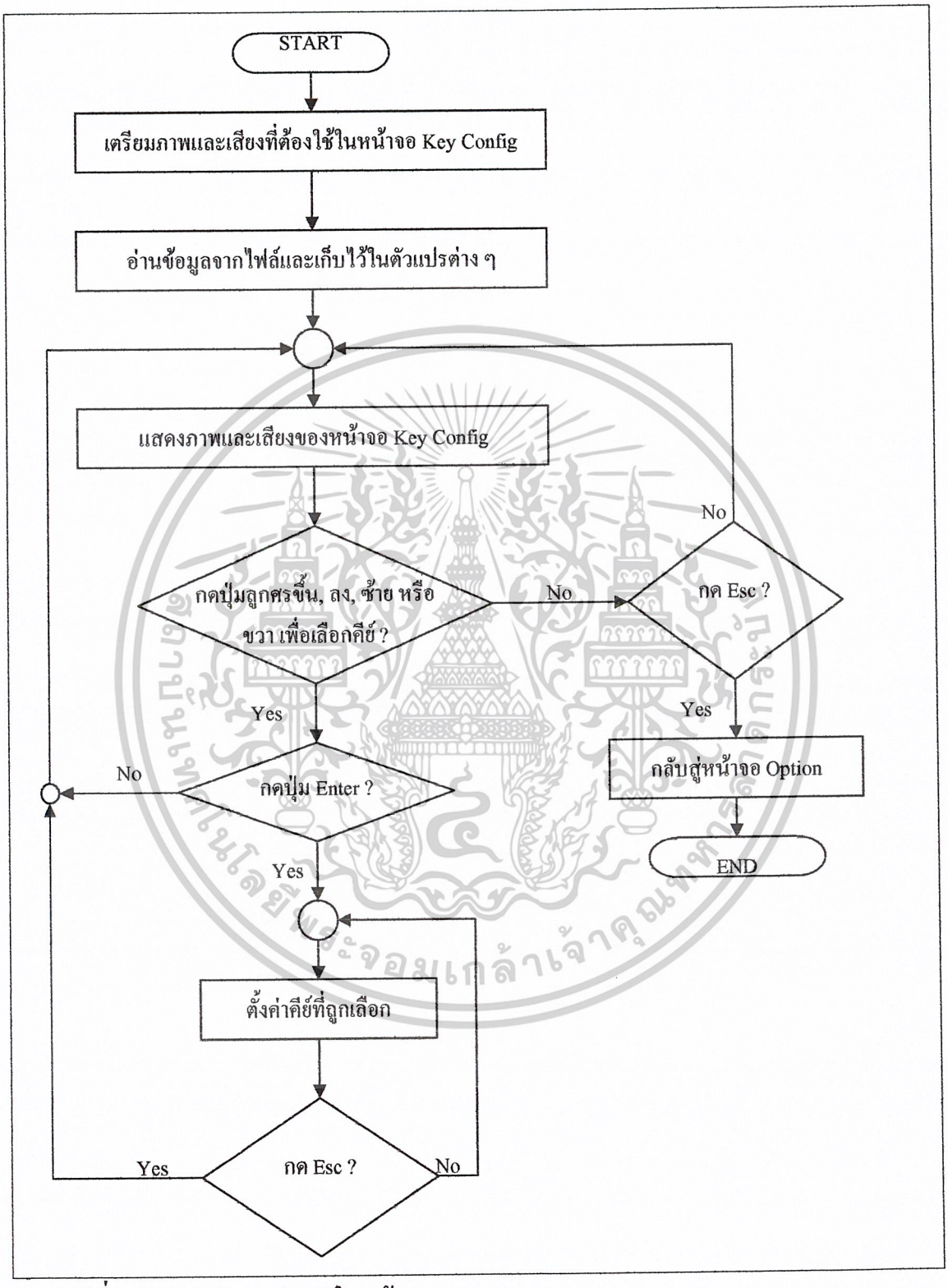
3.3.8 หน้าจอ Key Config (case 6) หน้าจอ Key config มีไว้สำหรับตั้งค่าคีย์บอร์ดที่ใช้สำหรับเล่นเกม ถ้าเป็นการเปิดเกมครั้งแรก หรือไม่มีการเปลี่ยนแปลงการตั้งค่าใด ๆ ค่าเริ่มต้นของคีย์บอร์ดจะเป็นดังรูป



รูปที่ 3.50 แสดงหน้าจอ Key Config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนผังการทำงาน ดังนี้

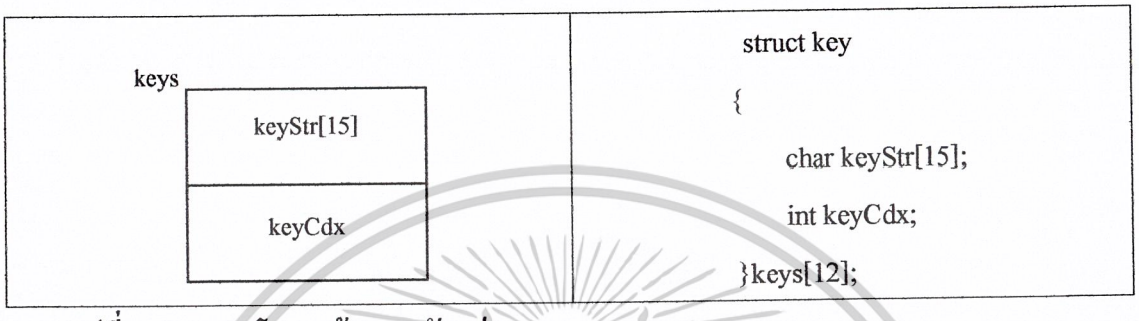


รูปที่ 3.51 แผนผังการทำงานในหน้าจอ Key Config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของหน้าจอนี้ก็คือ

1. อ่านข้อมูลจากไฟล์ที่ชื่อว่า config.txt มาเก็บไว้ในตัวแปร struct ที่ชื่อว่า keys ซึ่งปุ่มที่ผู้เล่นจะต้องใช้ในการเล่นเกมมีทั้งหมด 12 คีย์ นั่นคือ ผู้เล่น 2 คน เล่นคนละ 6 คีย์ โดย 6 คีย์แรกเป็นของผู้เล่นคนที่ 1 และ 6 คีย์หลังเป็นของผู้เล่นคนที่สอง ตัวแปร keys มีโครงสร้างดังต่อไปนี้



รูปที่ 3.52 แสดงโครงสร้างของตัวแปร keys

การอ่านข้อมูลจากไฟล์ จะเรียกใช้ฟังก์ชัน readKeyConfig ถ้าอ่านข้อมูลจากไฟล์ไม่ได้จะออกจากเกมทันที โดยมีการเขียนโปรแกรมดังนี้

```
int readKeyConfig()
{
        ifstream keyConfig ("config.txt");
        if (! keyConfig.is_open())
        {
                SendMessage(g_hWnd, WM_CLOSE, 0, 0);
                return 1;
        }

        for(int i=0;i<12;i++)
        {
                keyConfig >> (keys[i].keyStr);
                keyConfig >> (keys[i].keyCdx);
        }
        keyConfig.close();
        return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แสดงค่าในตัวแปร keys ออกเป็นตัวหนังสือออกทางหน้าจอ เนื่องจากการวาดภาพ, ตัวละคร หรือตัวหนังสือใน CDX ต้องกำหนดตำแหน่งเองทั้งหมด จึงได้กำหนดให้การแสดงคีย์แรกของผู้เล่นคนที่ 1 เริ่มที่ตำแหน่ง (350,210) และ การแสดงคีย์แรกของผู้เล่นคนที่ 2 เริ่มที่ตำแหน่ง (620,210) แต่ละบรรทัดจะแสดงห่างกันในแกน y 48 pixel มีการเขียนโปรแกรมดังนี้

```
int xx = 350;
int yy = 210;
for(int i=0;i<12;i++)
{
        if (yy>450)
        {
                yy=210;
                xx=620;
        }
        Screen->GetBack()->TextXY(xx,yy,RGB(255,255,255),keys[i].keyStr);
        yy = yy+48;
}
```

// ถ้าแสดงคีย์ของผู้เล่นคนที่ 1 ครบหมดแล้ว ให้เริ่มแสดงคีย์ของผู้เล่นคนที่ 2

// เมื่อแสดงคีย์แต่ละบรรทัดเสร็จแล้วให้ขึ้นบรรทัดใหม่นั่นคือบวก yy ด้วย 48

3. เมื่อผู้เล่นต้องการจะเปลี่ยนแปลงค่าใหม่ให้กดลูกศร ขึ้น, ลง, ซ้าย หรือขวา เพื่อให้กรอบไปโฟกัสคีย์ที่ต้องการตั้งค่า แล้วให้กด enter เพื่อเข้าสู่การตั้งค่าค่าจะเห็นว่ากรอบจะกระพริบให้รู้ว่ามีการกด enter แล้ว จากนั้นกดคีย์ที่ต้องการ เสร็จแล้วให้กด Esc เพื่อออกจากการตั้งค่าของคีย์นั้น กรอบก็จะหยุดกระพริบ ถ้าต้องการเปลี่ยนค่าอื่นอีกก็ใช้วิธีเช่นเดิม มีวิธีการเขียนโปรแกรมดังนี้

```
if (entered == false)
{
        tsp[0].setAnimSt(false);
        tsp[0].setFrame(0);
```

// ยังไม่ได้กด enter กรอบไม่ต้องกระพริบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (Input->GetKeyState(CDXKEY_RIGHTARROW)==CDXKEY_PRESS)
{
        if(x>=518)
                x=248;
        else
                x = x+270;
                tsp[0].setPos(x,y);
}
if (Input->GetKeyState(CDXKEY_LEFTARROW)==CDXKEY_PRESS)
{
        if(x<=248)
                x=518;
        else
                x = x-270;
        tsp[0].setPos(x,y);
}
if (Input->GetKeyState(CDXKEY_UPARROW)==CDXKEY_PRESS)
{
        if(y<=197)
                y=432;
        else
                y = y-48;
        tsp[0].setPos(x,y);
}
if (Input->GetKeyState(CDXKEY_DOWNARROW)==CDXKEY_PRESS)
{
        if(y>=432)
                y=197;
        else
                y = y+48;
        tsp[0].setPos(x,y);
}
```

// ถ้ามีการกดลูกศรขวา กรอบจะเลื่อนไปทางขวา ซึ่งตำแหน่ง x ของกรอบมีได้สองค่าเท่านั้น นั่นคือ 248 และ 518

// ถ้ามีการกดลูกศรซ้าย ให้กรอบจะเลื่อนไปทางซ้าย ซึ่งตำแหน่ง x ของกรอบมีได้สองค่าเท่านั้น นั่นคือ 248 และ 518

// ถ้ามีการกดลูกศรขึ้น กรอบจะเลื่อนขึ้น ซึ่งกำหนดให้ตำแหน่ง y ของกรอบต้องอยู่ระหว่าง 197 และ 432 ถ้ากดขึ้นไปเรื่อย ๆ แล้ว y เกินตำแหน่งที่คีย์แรกของผู้เล่นถูกแสดงอยู่ ก็ให้ y เป็น 432 นั่นคือ กรอบมาโฟกัสที่คีย์สุดท้ายของผู้เล่นคนนั้น

// ถ้ามีการกดลูกศรลง ให้กรอบจะเลื่อนลง ซึ่งกำหนดให้ตำแหน่ง y ของกรอบต้องอยู่ระหว่าง 197 และ 432 ถ้ากดลงไปเรื่อย ๆ แล้ว y มากกว่า 432 ก็กำหนดให้ y = 197 นั่นคือ กรอบมาโฟกัสที่คีย์แรกของผู้เล่นคนนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (Input->GetKeyState(CDXKEY_ENTER)==CDXKEY_PRESS)
        {
                entered = true;
        }
}
else if  (entered == true)
{
        tsp[0].setAnimSt(true);
        if (x==248)
        {
                if (y==197) //key1
                {
                        checkKey(0);
                        Screen->GetBack()->
                        TextXY(350,210,RGB(255,255,255),keys[0].keyStr);
                }
                .....
        }
```

// ถ้าต้องการตั้งค่าคีย์ที่เลือก ก็ให้กด enter

// เมื่อกด enter แล้วให้กรอบกระพริบ

// ถ้าตำแหน่ง x ของกรอบอยู่ที่ 248 หมายถึงว่ากรอบกำลังโฟกัสอยู่ที่การตั้งค่าของผู้เล่นคนที่ 1 โดยการตั้งค่าคีย์ต่าง ๆ จะดูที่ตำแหน่ง y

// checkKey เป็นฟังก์ชันที่ตรวจสอบว่าคีย์กดเป็นคีย์ใด แล้วนำค่านั้นใส่ในตัวแปร keys

// ทำจนครบทุกคีย์

การเปลี่ยนแปลงค่าของผู้เล่นคนที่ 2 ก็ทำได้เช่นเดียวกัน แต่จุดของกรอบจะเริ่มที่ (518,197)

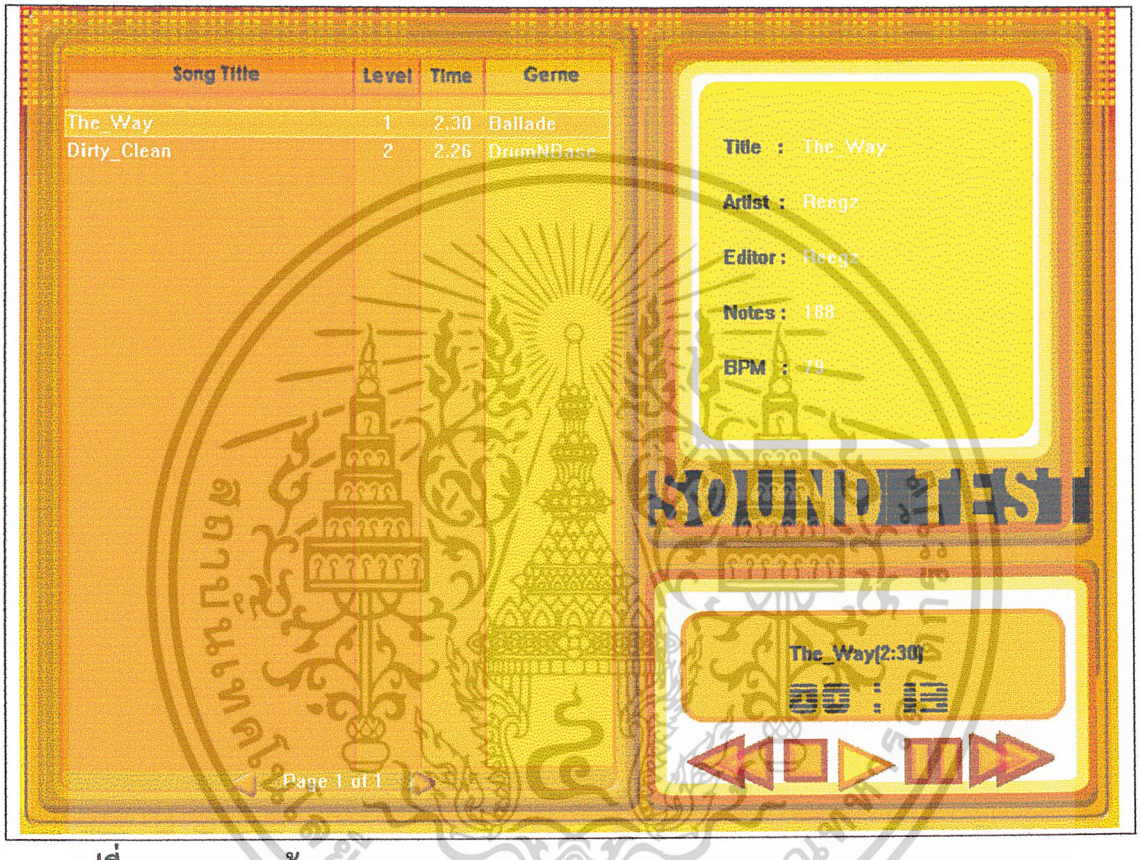
4. การบันทึกค่าที่ได้เปลี่ยนแปลงจะทำเมื่อมีการกด Esc เพื่อกลับไปยังหน้าจอ Option สามารถเขียนโปรแกรมได้ดังนี้

```
if (entered == false && Input-GetKeyState(CDXKEY_ESCAPE)==CDXKEY_PRESS)
        {
                saveKeyConfig();
                state=5;
                showed = false;
                break;
        }
```

// เงื่อนไขนี้คือถ้ามีการกด Esc เพื่อกลับไปยังหน้าจอ Option จะทำการบันทึกการตั้งค่าโดยฟังก์ชัน saveKeyConfig();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

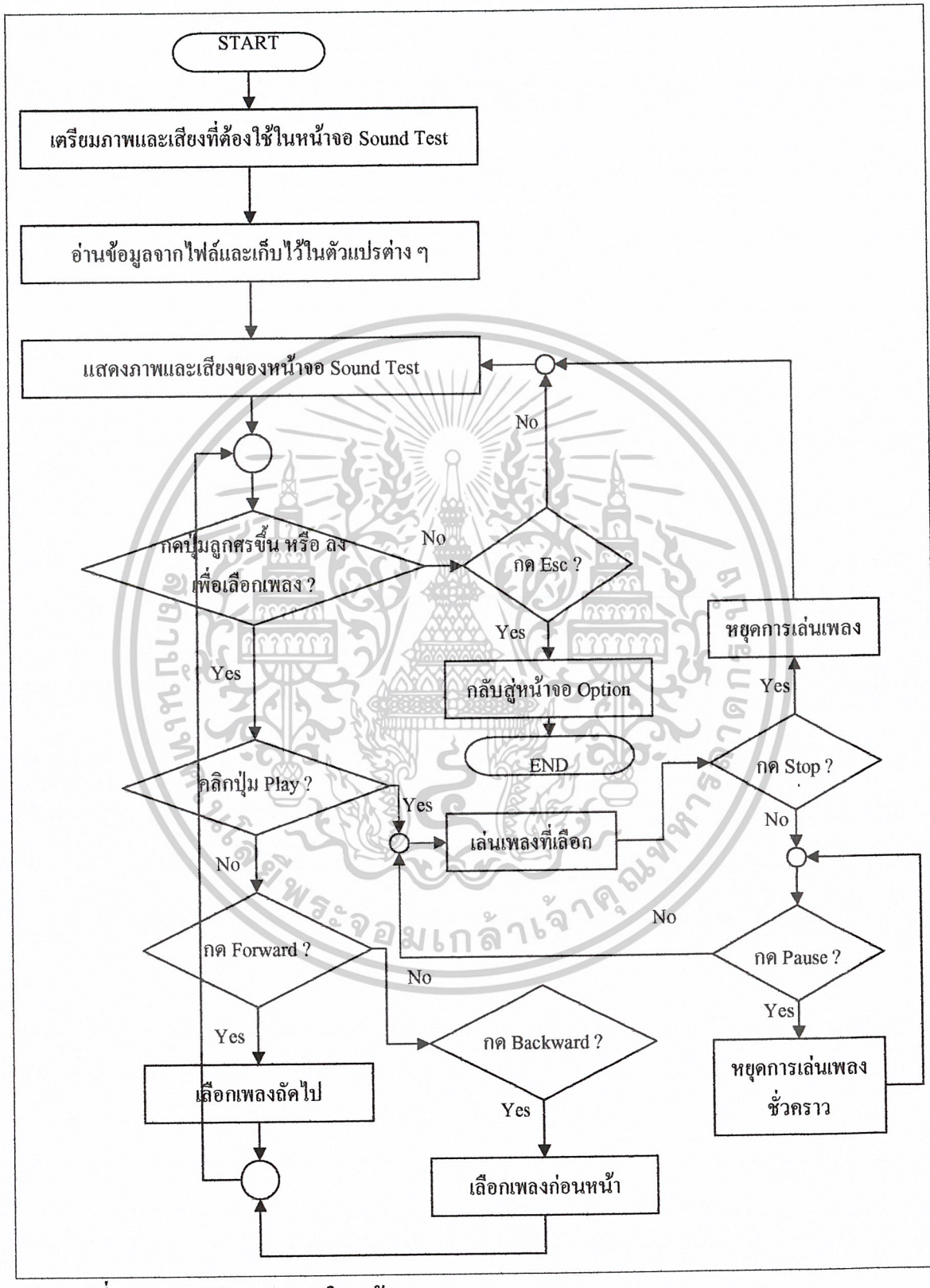
3.3.10 หน้าจอ Sound Test (case 7) เป็นหน้าจอสำหรับให้ผู้เล่นทดลองฟังเพลง โดยเลือกจากรายชื่อเพลงที่แสดงอยู่ทางด้านซ้าย และแสดงรายละเอียดของเพลงทางด้านมุมขวาบน หน้าจอนี้มีลักษณะคล้ายกับหน้าจอ Sound Select แต่มีหน้าที่ที่เพิ่มขึ้นมาคือการเล่นไฟล์เพลงตามที่ผู้เล่นต้องการ โดยการเขียนโปรแกรมในหน้าจอนี้จะรองรับอินพุตการคลิกเมาส์ที่ปุ่มต่าง ๆ ซึ่งมี 5 ปุ่ม ดังรูป



รูปที่ 3.53 แสดงหน้าจอ Sound Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

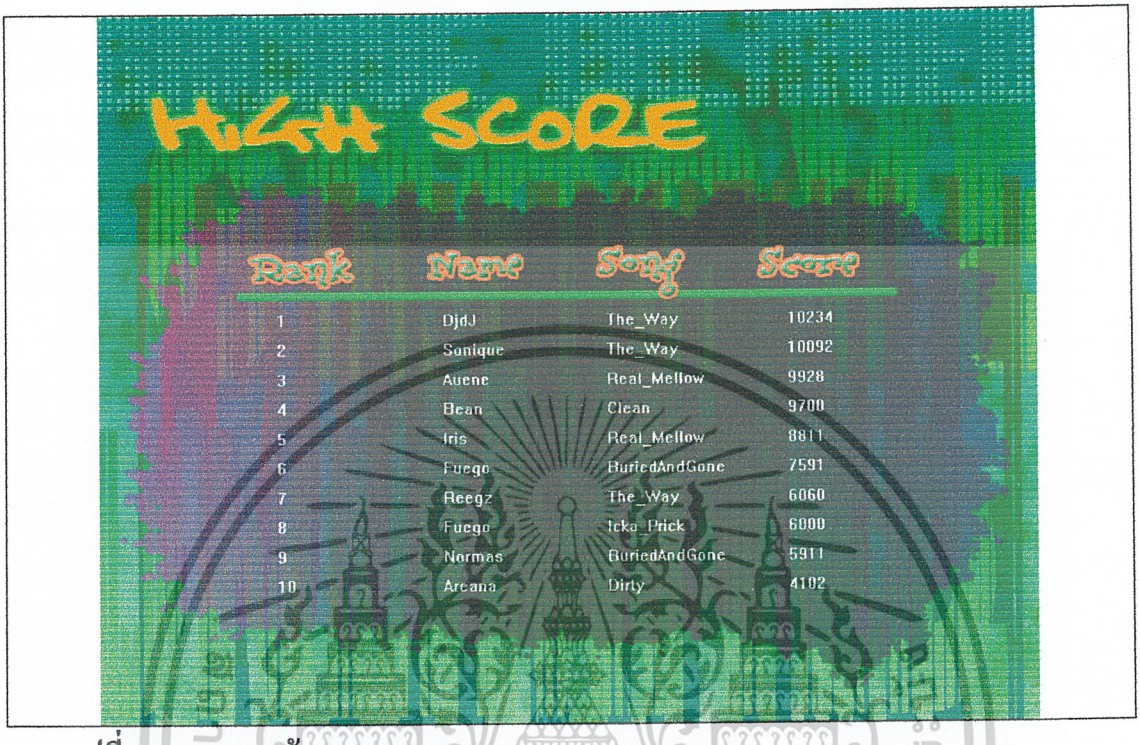
มีแผนผังการทำงานดังนี้



รูปที่ 3.54 แผนผังการทำงานในหน้าจอ Sound Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

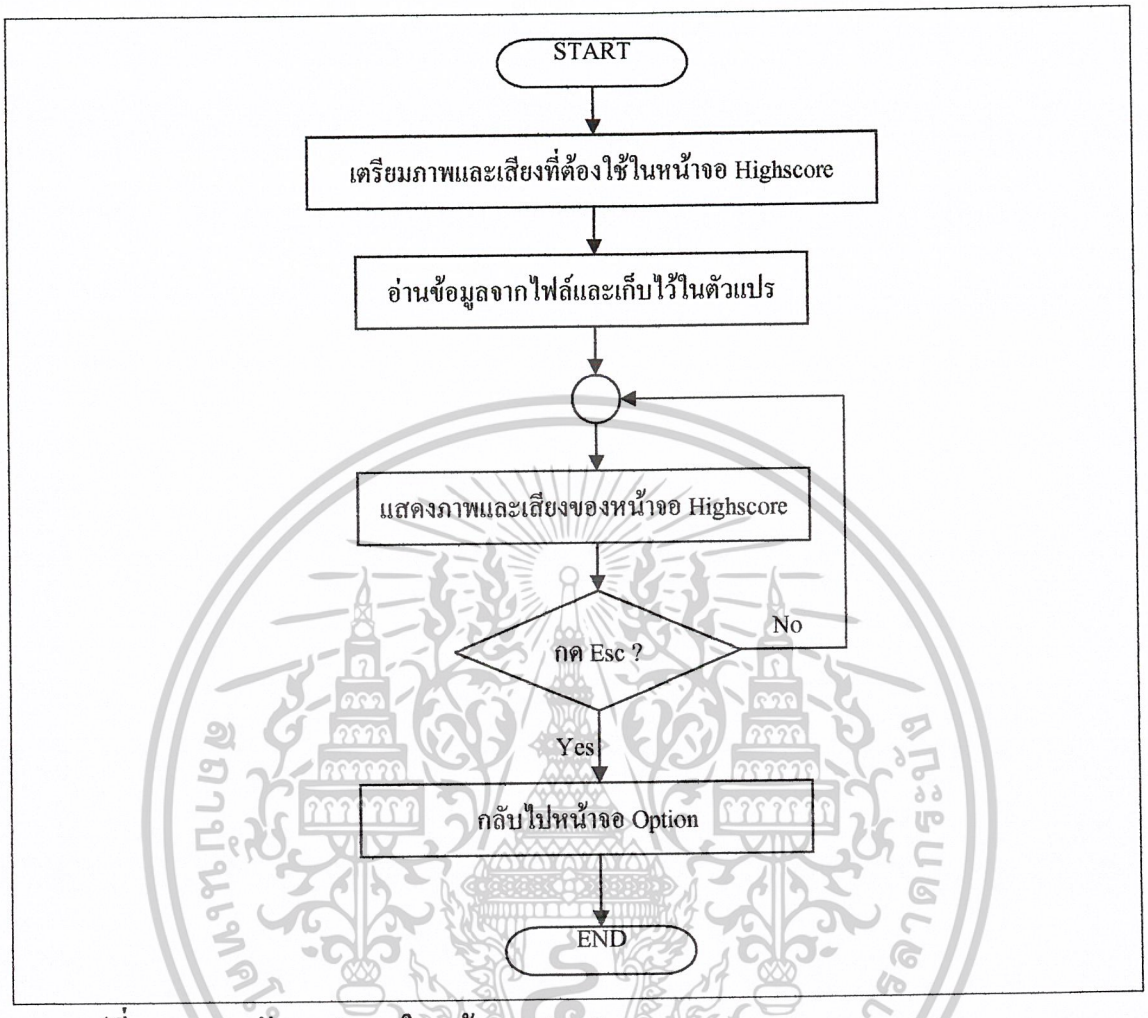
3.3.11 หน้าจอ High Score (case 8) หน้าจอนี้มีไว้สำหรับแสดงผู้ที่มีคะแนนสูงสุด 10 อันดับแรก โดยอ่านข้อมูลจากไฟล์ที่ชื่อว่า highscore.txt แล้วแสดงตัวหนังสือออกทางหน้าจอ ดังรูป



รูปที่ 3.55 แสดงหน้าจอ High Score

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแผนผังการทำงานดังนี้

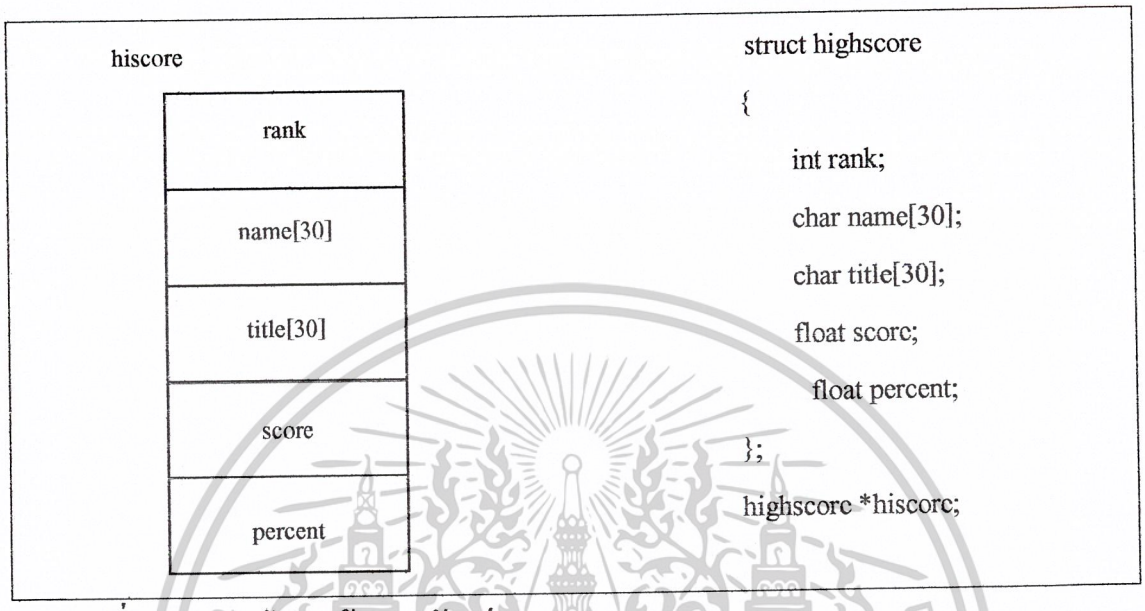


รูปที่ 3.56 แผนผังการทำงานในหน้าจอ High Score

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมของหน้าจอ high score มีวิธีการดังนี้

1. อ่านข้อมูลจากไฟล์ highscore.txt มาเก็บไว้ในตัวแปร struct ที่ชื่อว่า hiscore ซึ่งมีโครงสร้างดังต่อไปนี้



รูปที่ 3.57 แสดงโครงสร้างของตัวแปร hiscore

ตัวแปร rank ใช้สำหรับเก็บตำแหน่งของผู้เล่น
ตัวแปร name สำหรับเก็บชื่อของผู้เล่น
ตัวแปร title สำหรับเก็บชื่อเพลง
ตัวแปร score สำหรับเก็บคะแนนที่ได้
ตัวแปร percent สำหรับเก็บสัดส่วนของคะแนนที่ได้กับคะแนนเต็มของเพลงนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจากไฟล์เขียนโปรแกรมดังนี้

```
ifstream highScore ("highscore.txt");
if (!highScore.is_open())
{
        SendMessage(g_hWnd, WM_CLOSE, 0, 0);
}
        for(int i=0;i<10;i++)
{
        highScore >> hiscore[i].rank;
        highScore >> hiscore[i].name;
        highScore >> hiscore[i].title;
        highScore >> hiscore[i].score;
}
highScore.close();
```

2. เมื่ออ่านข้อมูลมาเก็บไว้ในตัวแปร hiscore เรียบร้อยแล้ว ก็ทำการแสดงออกเป็นตัวหนังสือออกทางหน้าจอ โดยเขียนโปรแกรมดังนี้

```
char str[10];
int yy=250;
for(int k=0;k<10;k++)
{
        sprintf(str,"%d",hiscore[k].rank);
        Screen->GetBack()->TextXY(150,yy,RGB(255,255,255),str);
        Screen->GetBack()->TextXY(290,yy,RGB(255,255,255),hiscore[k].name);
        Screen->GetBack()->TextXY(430,yy,RGB(255,255,255),hiscore[k].title);
        sprintf(str,"%.2f\% (%.0f)",hiscore[k].percent,hiscore[k].score);
        Screen->GetBack()->TextXY(585,yy,RGB(255,255,255),str);
        yy=yy+25;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.12 หน้าจอ Loading (case 9) หน้าจอนี้เป็นหน้าจอที่ใช้สำหรับการเตรียมพารามิเตอร์ต่าง ๆ สำหรับหน้าจอการเล่นเกมซึ่งมีเป็นจำนวนมาก หน้าจอ Loadingเป็นดังนี้

รูปที่ 3.58 แสดงหน้าจอ Loading

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

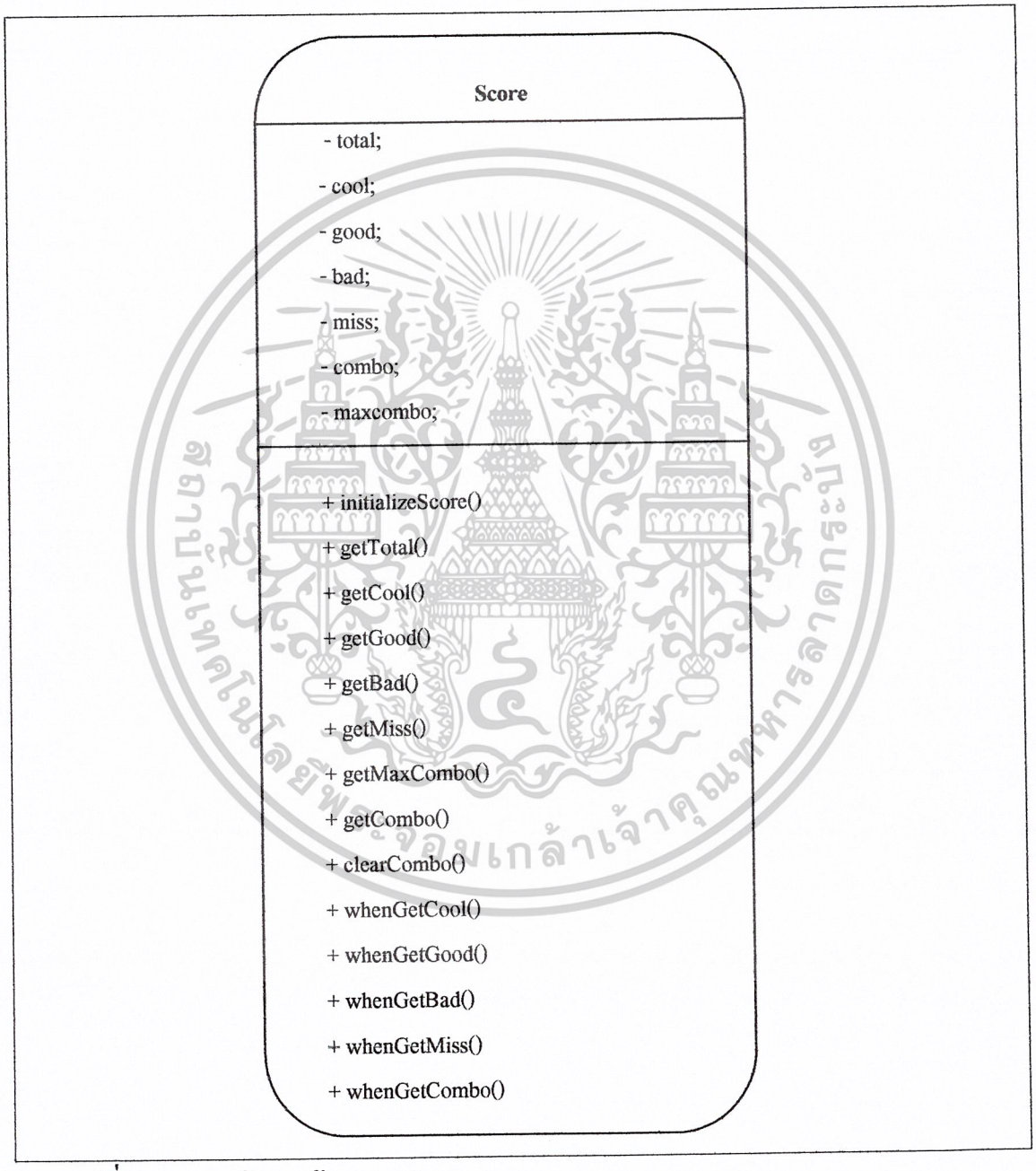
3.3.13 หน้าจอ Game Play (case 10) หน้าจอนี้เป็นส่วนที่สำคัญของเกมเพราะเป็นหน้าจอของการเล่นเกม ซึ่งหน้าจอนี้มีการจัดการเกี่ยวกับตัวละคร และอินพุตต่าง ๆ โดยในส่วนของการจัดการเกี่ยวกับรูปภาพที่นำมาเป็นพื้นหลัง หรือวัตถุในตำแหน่งต่าง ๆ นั้น ใช้วิธีการเช่นเดียวกับวิธีที่อธิบายไปข้างต้น ส่วนทางด้านรายละเอียดอื่นของ หน้าจอ Game Play เป็นดังนี้

รูปที่ 3.59 แสดงหน้าจอ Game Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตัวแปรโครงสร้าง และโครงสร้างคลาสที่ใช้ในการสร้างวัตถุที่จำเป็นสำหรับหน้าจอนี้มีดังนี้

- Class Score ใช้ในการสร้างวัตถุที่แทนตัวผู้เล่นขึ้นมาใช้ในการเก็บคะแนนที่ได้จากการเล่นเกม และมีเมทธอดในการคำนวณคะแนนเตรียมไว้ให้ทำการคำนวณ โดยคลาสนี้มีโครงสร้างดังนี้



รูปที่ 3.60 แสดงโครงสร้างของคลาส Score

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรายละเอียดของแอททริบิว และเมทธอดมีดังนี้

แอททริบิว

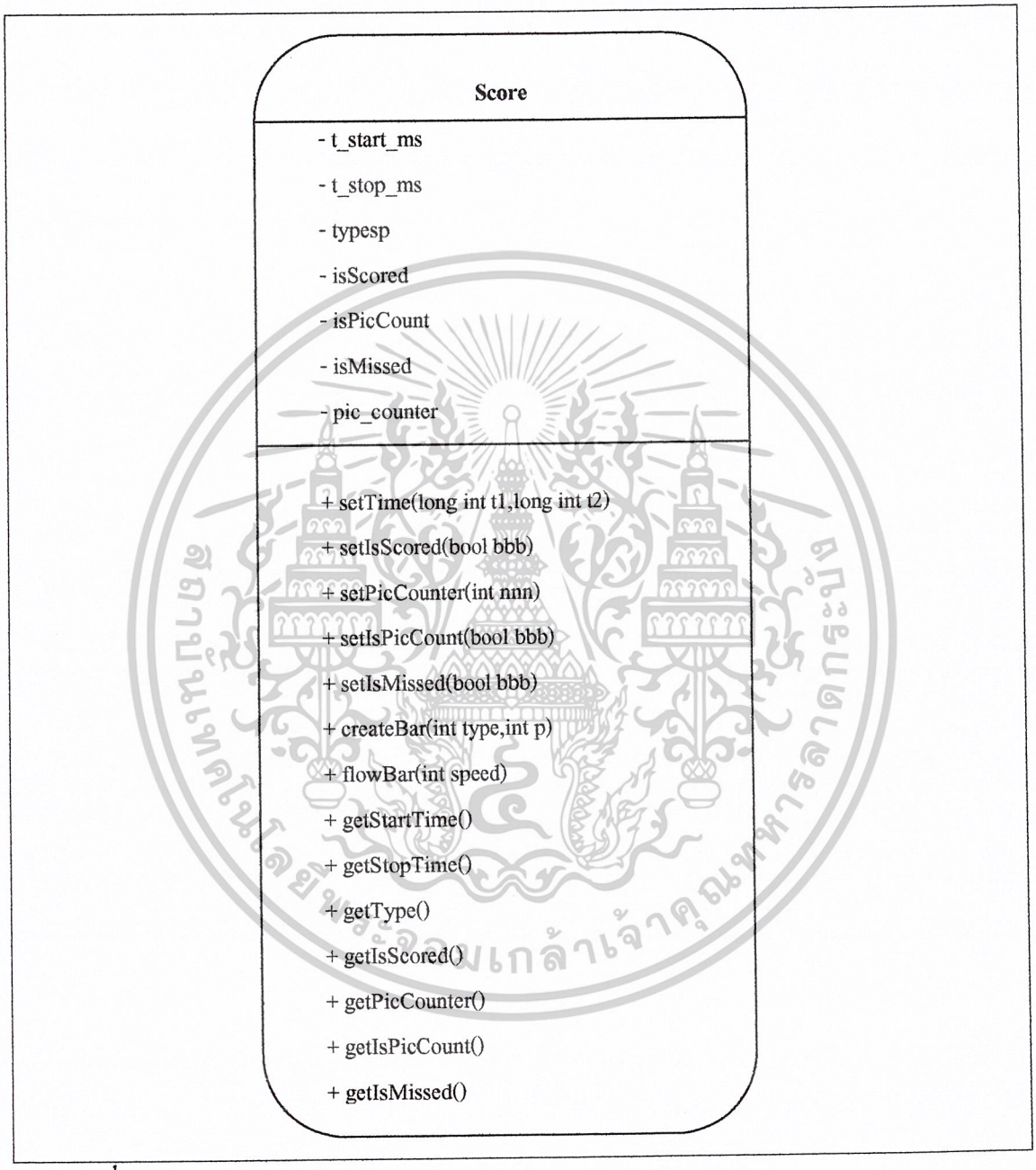
- total ใช้เก็บคะแนนรวมทั้งหมด
- cool ใช้เก็บจำนวนครั้งที่ได้ cool
- good ใช้เก็บจำนวนครั้งที่ได้ good
- bad ใช้เก็บจำนวนครั้งที่ได้ bad
- miss ใช้เก็บจำนวนครั้งที่ได้ miss
- combo ใช้เก็บจำนวนครั้งที่ combo ได้
- maxcombo ใช้เก็บจำนวนครั้งสูงสุดที่ทำ combo ได้

เมธอด

- initializeScore() เป็นการนำค่าคะแนนต่างที่อาจจะค้างอยู่ในแอททริบิวของคลาส Score มาทำการกำหนดค่าเริ่มต้นให้ใหม่ดังนี้
- getTotal() ใช้คืนค่าของแอททริบิว total
- getCool() ใช้คืนค่าของแอททริบิว cool
- getGood() ใช้คืนค่าของแอททริบิว good
- getBad() ใช้คืนค่าของแอททริบิว bad
- getMiss() ใช้คืนค่าของแอททริบิว miss
- getMaxCombo() ใช้คืนค่าของแอททริบิว maxcombo
- getCombo() ใช้คืนค่าของแอททริบิว combo
- clearCombo() ใช้ทำการเคลียร์ค่าของแอททริบิว combo
- whenGetCool() เพิ่มค่าของแอททริบิว cool ขึ้นหนึ่งค่า และเพิ่มค่าคะแนนให้แอททริบิว total
- whenGetGood() เพิ่มค่าของแอททริบิว good ขึ้นหนึ่งค่า และเพิ่มค่าคะแนนให้แอททริบิว total
- whenGetBad() เพิ่มค่าของแอททริบิว bad ขึ้นหนึ่งค่า และเพิ่มค่าคะแนนให้แอททริบิว total
- whenGetMiss() เพิ่มค่าของแอททริบิว miss ขึ้นหนึ่งค่า
- whenGetCombo() เพิ่มค่าของแอททริบิว combo ขึ้นหนึ่งค่า และคำนวณหาค่าสูงสุดของ combo ให้กับแอททริบิว maxcombo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Class SpriteBar จะเป็นคลาสที่สืบทอดมาจากคลาส GUSprite เพื่อเอาไว้ใช้สร้างวัตถุที่เป็น Sprite แต่มีความซับซ้อนในรายละเอียดมากกว่า นั่นคือการสร้างวัตถุ bar ในหน้าการเล่นเกม โดยมีแอททริบิว และเมทธอดเพิ่มขึ้นมา โดยมีโครงสร้างดังนี้



รูปที่ 3.61 แสดงโครงสร้างของคลาส SpriteBar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรายละเอียดของแอททริบิว และเมทธอดมีดังนี้

แอททริบิว

- t_start_ms ใช้เก็บเวลาที่ bar จะเริ่มการไหลลงมาในช่องทางที่กำหนด
- t_stop_ms ใช้เก็บเวลาที่ bar จะไหลถึงยังตำแหน่งที่กำหนด
- typesp ใช้เก็บหมายเลขที่อ้างถึงประเภทของ bar โดยช่องที่กำหนดไว้ในเกมต่อหนึ่งผู้เล่นจะมีทั้งหมด 6 ช่อง ซึ่งจากแอททริบิวนี้ จะใช้เลข 0 ถึง 5 ในการอ้างถึง
- isScored เป็น true เมื่อ bar ตัวนั้น ๆ ถูกคิดคะแนนแล้ว
- isPicCount เป็นแอททริบิวที่ช่วยในการแสดงรูปภาพระหว่างการเล่นเกม ซึ่งรูปภาพที่จะแสดงนั้นคือรูปภาพที่บ่งบอกว่า bar ตัวนั้น ๆ ได้คะแนนในระดับใด และได้ค่า combo เป็นเท่าไร ดังรูป 3.19 เมื่อทำการแสดงรูปจะต้องมีการให้รูปแสดงค้างไว้ชั่วระยะหนึ่ง แล้วจึงหายไปจากหน้าจอ เนื่องจากวัตถุ bar มีการไหลลงมาจากตำแหน่งเริ่มต้นในหน้าจอ (ซึ่งก็คือการเพิ่มค่าแกน y) จึงใช้การนับค่า y ที่เพิ่มมากขึ้น (วัตถุไหลลงตามแกน y) โดยจะมีค่าเป็น true เมื่อมีการนับถึงตำแหน่งที่ต้องการแล้ว
- isMissed รูปที่ใช้แสดงสถานะ miss จะถูกแสดงในกรณี และเงื่อนไขที่ต่างจาก cool, good และ bad จึงต้องมี flag ช่วยในการตรวจสอบว่า bar นั้น ๆ แสดงในสถานะ miss หรือไม่ เพราะหากมีการแสดงสถานะอื่น (ใช้เงื่อนไขคล้ายกันในการตรวจสอบ) ก็จะปรับค่า flag ว่าไม่ต้องแสดงสถานะนี้ โดยค่า true หมายถึงจะต้องแสดงรูป miss
- pic_counter เป็นแอททริบิวใช้นับค่าแกน y ของ bar ที่อธิบายไปในการใช้แอททริบิว isPicCount ข้างต้น

เมทธอด

- setTime(long int t1,long int t2) ใช้ในการกำหนดค่าให้กับแอททริบิว t_start_ms และ t_stop_ms
- setIsScored(bool bbb) กำหนดค่าให้กับแอททริบิว isScored
- setPicCounter(int nnn) กำหนดค่าให้กับแอททริบิว PicCounter
- setIsPicCount(bool bbb) กำหนดค่าให้กับแอททริบิว isPicCount
- setIsMissed(bool bbb) กำหนดค่าให้กับแอททริบิว isMissed
- createBar(int type,int p) สร้างวัตถุ bar ขึ้นมาในหน้าจอ (กำหนดรูปภาพ, ตำแหน่งตามหมายเลขประเภทที่ถูกส่งเข้ามา)
- flowBar(int speed) ทำให้ bar เคลื่อนที่ไหลลงตามแกน y ในหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- getStartTime() ใช้คืนค่าของแอททริบิว t_start_ms
- getStopTime() ใช้คืนค่าของแอททริบิว t_stop_ms
- getType() ใช้คืนค่าของแอททริบิว typesp
- getIsScored() ใช้คืนค่าของแอททริบิว isScored
- getPicCounter() ใช้คืนค่าของแอททริบิว PicCounter
- getIsPicCount() ใช้คืนค่าของแอททริบิว isPicCount
- getIsMissed() ใช้คืนค่าของแอททริบิว isMissed

2. โครงสร้างการทำงานของหน้าจอการเล่นเกมตามกติกาที่เราได้รู้มาแล้วนั้นมีดังนี้

- การกำหนดประเภทและตำแหน่งของวัตถุ bar ตามที่อ่านได้จากไฟล์ของเพลงนั้น ๆ ที่ได้เลือกเล่น

- อ่านจากไฟล์เพลงที่เลือกเล่นมาทำการสร้าง bar โดยใช้เมทธอด CreateBar() ใน Class SpriteBar ตามจำนวนและประเภทที่อ่านได้จากไฟล์
- โดยถ้าหากมีการเล่นแบบ 2 ผู้เล่น จะทำการสร้าง bar เพิ่ม 2 เท่าจากแบบ 1 ผู้เล่น

```
for(วนลูปตามจำนวนวัตถุ bar ทั้งหมด)
{
    // โหลดวัตถุทั้งหมดขึ้นมาแล้วซ่อนไว้
    .....
    if(ถ้ามีผู้เล่น 2 คน)
    {
        // โหลดวัตถุทั้งหมด
        ขึ้นมาแล้วซ่อนไว้
        .....
    }
}
```

- สร้างรูปภาพที่จะต้องแสดงในหน้าจอการเล่นเกมนอกเหนือจากที่ได้ประกาศไว้ตอนแรก ซึ่งรูปภาพเหล่านี้จะต้องนำมาใส่เงื่อนไข เพื่อหาเวลา และจังหวะที่จะต้องแสดงตามกติกาของเกมซึ่งได้แก่รูปภาพ cool, good, bad, miss ,digit และ combo
- โดยถ้าหากมีการเล่นแบบ 2 ผู้เล่น จะทำการสร้างรูปภาพเพิ่ม 2 เท่าจากแบบ 1 ผู้เล่น

```
//โหลดรูปภาพอื่น ๆ ที่เกี่ยวข้อง
.....
if(ถ้ามีผู้เล่น 2 คน)
{
    //โหลดรูปภาพอื่น ๆ ที่เกี่ยวข้อง
    .....
}
```

- รูปที่สร้างมานำไปซ่อนไว้บริเวณนอกหน้าจอ เพื่อรอจังหวะที่จะแสดงในหน้าจอ โดยในที่นี่หน้าจอจะถูกกำหนดเป็นขนาด 800x600 เราจึงนำรูปภาพที่สร้างมาไปแอบไว้ที่บริเวณ x = -50 และ y = -50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการไหลวัตถุ bar (เคลื่อนที่ตามแนวแกน y) ตามจังหวะที่ต้องแสดง โดยขั้นตอนของการไหลจะมี 3 ขั้นตอน ดังนี้

  - ขั้นตอนที่ 1 ให้ทำการย้ายรูปภาพวัตถุ bar ที่ซ่อนเอาไว้มาเตรียมอยู่ในตำแหน่งแกน x และแกน y ที่ต้องการให้เป็นจุดเริ่มไหล ตามเวลาเริ่มไหลที่กำหนดไว้
  - ขั้นตอนที่ 2 เราจะใช้เป็นขั้นตอนวิธีหลักในการไหลของวัตถุ bar โดยเราจะทำการแบ่งช่วงของเส้นทางที่จะต้องแสดงแล้วไหลไปตามแกน y ออกเป็นช่วง ๆ จำนวนช่วงขึ้นอยู่กับความต้องการที่จะให้ได้จุดที่ละเอียดมากแค่ไหน วิธีการทำให้ไหลโดยการเมื่อถึงเวลาที่กำหนดให้ bar ไหลไปยังจุดนั้น ๆ ตามที่ได้คำนวณไว้ (กำหนด (x,y) ให้รูปภาพไปแสดงที่จุดนั้นเลย เมื่อกำหนดโดยการเปลี่ยนค่า y ไปเรื่อย ๆ รูปภาพก็จะไหลลงเรื่อย ๆ) โดยมีตัวอย่างดังนี้ โดยให้จุดที่ต้องการเริ่มต้นคือจุด y = 0 (ค่า x คงที่) และจุดที่จะให้หยุด แล้วหายไปคือจุดที่ y = 472
  - ขั้นตอนที่ 3 จะใช้เป็นขั้นตอนสำรองเมื่อวัตถุ bar ไม่ไหลไปตามจุดต่าง ๆ ที่กำหนด เราจะทำการบังคับให้วัตถุ bar นั้น ๆ ไหลลงตามแกน y ทีละหนึ่งไปเรื่อย ๆ จนถึงจุดที่กำหนดให้หยุด (y = y + 1)

โดยจากขั้นตอนวิธีจะได้โครงสร้างของการโปรแกรมดังนี้

```
for(วนลูปตามจำนวนวัตถุ bar ทั้งหมด)
{
    if(ถ้าเวลาเริ่มของ bar ตรงกับเวลาปัจจุบันของเพลงที่กำลังเล่นอยู่)
    {
        if(ถ้าวัตถุยังถูกซ่อนอยู่)
            //.ให้มาอยู่ในตำแหน่งเตรียมพร้อม
        if(ถ้ามีผู้เล่น 2 คน)
        {
            // ทำการเตรียมพร้อมสำหรับวัตถุของผู้เล่นที่ 2 ด้วย
            if(ถ้าวัตถุยังถูกซ่อนอยู่)
                //.ให้มาอยู่ในตำแหน่งเตรียมพร้อม
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//main movement algorithm
for(วนลูปตามจำนวนวัตถุ bar ทั้งหมด)
{
        if(ถ้าวัตถุถูกตระเตรียมสำหรับการเคลื่อนที่แล้ว)
        {
                for(ค้นหาข้อมูลว่าในเวลาปัจจุบันวัตถุ bar นั้น ๆ ต้องอยู่ที่ตำแหน่งไหน)
                {
                        if((ถ้าค้นหาพบว่าเวลานี้ต้องอยู่ที่ตำแหน่งไหน)
                        {
                                if(ถ้าวัตถุยังไม่ถึงยังตำแหน่งนั้นให้ทำ  ถ้าเลยไปแล้วไม่
                                สนใจ)
                                {
                                        // ให้ bar เลื่อนไปยังตำแหน่งที่ระบุไว้
                                        if(ถ้ามีผู้เล่น 2 คน)
                                                // ให้ bar เลื่อนไปยังตำแหน่งที่ระบุไว้
                                }
                        }
                }
                //additional movement algorithm
                // ให้ bar เคลื่อนที่ตามแนวแกน y 1 หน่วย ( y = y + 1)
                if(ถ้ามีผู้เล่น 2 คน)
                // ให้ bar ของผู้เล่น 2 เคลื่อนที่ตามแนวแกน y 1 หน่วย ( y = y + 1)
        }

}
```

- เมื่อวัตถุ bar ไหลลงมายังจุดที่ต้องหยุดแล้ว ตามกติกาคือต้องกดปุ่ม KeyInput ที่กำหนดไว้ให้ตรงกับจังหวะของดนตรี และจังหวะของวัตถุ bar ที่ไหลมาถึง โดยถ้าตรงจังหวะมากก็จะได้คะแนนมากขึ้นตามไปด้วย ซึ่งส่วนนี้จะเป็นส่วนของการจับจังหวะของการกดปุ่ม KeyInput กับจังหวะของดนตรี และจังหวะการไหลมาถึงของวัตถุ bar จากนั้นจึงทำการคำนวณระดับที่กดว่าใกล้เคียงมากน้อยกับเวลาที่ต้องกดจริง ๆ มากแค่ไหน แล้วทำการคำนวณคะแนนที่จะได้ออกมาบันทึกผลเก็บไว้ โดยขั้นตอนวิธีของส่วนนี้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จังหวะที่ได้ระบุไว้ในกติกาคือ
  Cool : กดได้ใกล้เคียงกับจังหวะที่ต้องกดในช่วงที่ไม่เกิน 0.2 วินาที
  Good : กดได้ใกล้เคียงกับจังหวะที่ต้องกดในช่วงที่ไม่เกิน 0.3 วินาที
  Bad : กดได้ใกล้เคียงกับจังหวะที่ต้องกดในช่วงที่ไม่เกิน 0.5 วินาที
  Miss : กดเกินมากกว่าไป 0.5 วินาที

```
//calculate score here!!
    for(วนลูปตามจำนวนวัตถุ bar ทั้งหมด)
    {
        if(ถ้าวัตถุ bar ไหลมายังตำแหน่งที่กำหนดให้หยุด)
        {
            if(ถ้ายังไม่มีการคิดคะแนน)
            {
                // ถ้าไม่ตรงกับจังหวะที่กำหนดไว้แล้ว
                    ก็ให้แสดงรูปภาพ Miss
                // คำนวณคะแนนสำหรับการ Miss
            }
            // นำ bar ที่คิดคะแนนแล้วไปซ่อนเอาไว้
        }
        // แสดงรูปภาพต่าง ๆ ที่เกี่ยวข้อง
        If(ถ้ามีผู้เล่น 2 คน)
        {
            // แสดงรูปภาพต่าง ๆ ที่เกี่ยวข้องของผู้เล่นที่ 2
        }

        //when press 1st button
        if((ถ้ากดปุ่มที่ 1 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
        {
            if(จังหวะที่ได้ไม่เกิน 0.1 วินาที)
            {
                // ปรับค่าที่เกี่ยวข้อง และคำนวณคะแนน
                // แสดงรูปภาพ Cool
            }
            else if(จังหวะที่ได้ไม่เกิน 0.2 วินาที)
            {
                // ปรับค่าที่เกี่ยวข้อง และคำนวณคะแนน
                // แสดงรูปภาพ Good
            }
```

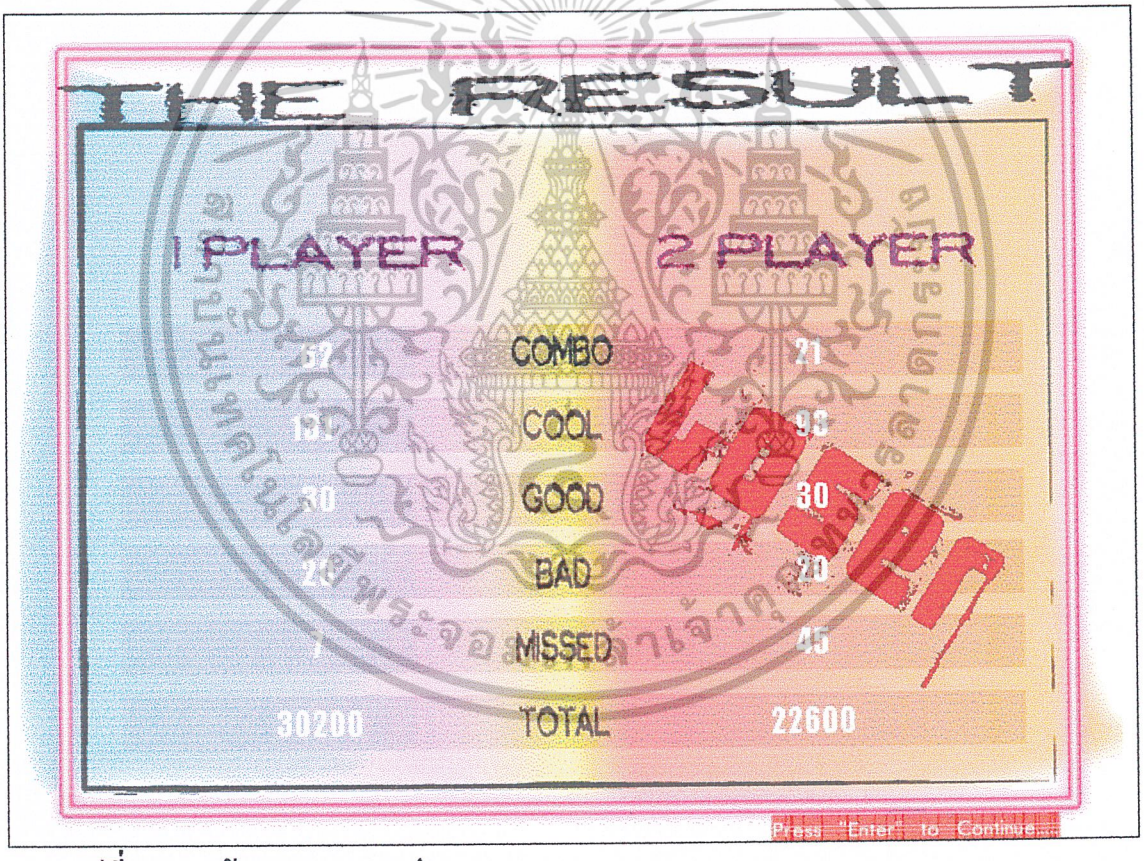
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if(จังหวะที่ได้ไม่เกิน 0.3 วินาที)
        {
                // ปรับค่าที่เกี่ยวข้อง และคำนวณคะแนน
                // แสดงรูปภาพ Bad
        }
        if(ถ้ามีผู้เล่น 2 คน)
        {
                // ดำเนินการอย่างเดียวกันทั้ง 3 กรณีกับผู้เล่นคนที่ 2
        }
}
//when press 1st button
if((ถ้ากดปุ่มที่ 2 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
{
        // ดำเนินอย่างการเดียวกันกับกรณีแรกสำหรับการกดปุ่มที่ 2
}
if((ถ้ากดปุ่มที่ 3 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
{
        // ดำเนินอย่างการเดียวกันกับกรณีแรกสำหรับการกดปุ่มที่ 3
}
if((ถ้ากดปุ่มที่ 4 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
{
        // ดำเนินอย่างการเดียวกันกับกรณีแรกสำหรับการกดปุ่มที่ 4
}
if((ถ้ากดปุ่มที่ 5 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
{
        // ดำเนินอย่างการเดียวกันกับกรณีแรกสำหรับการกดปุ่มที่ 5
}
if((ถ้ากดปุ่มที่ 6 และ bar ตัวนั้นยังไม่ได้คิดคะแนน)
{
        // ดำเนินอย่างการเดียวกันกับกรณีแรกสำหรับการกดปุ่มที่ 6
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อสิ้นสุดเกม โดยอาจสิ้นสุดได้ 2 กรณี คือด้วยการทำงานมาจนถึงจบเพลง และการออกจากเกมระหว่างการเล่นด้วยการกดปุ่ม ESCAPE ซึ่งเมื่อเกมสิ้นสุดแล้วจะมีการเปลี่ยนหน้าจอไปที่หน้าจอของการสรุปคะแนนทั้งหมด

3.3.14 หน้าจอ Result (case 11) ในหน้าจอ Result จะมีการแสดงภาพฉากหลัง และมีการแสดงผลสรุปคะแนนของเกมที่ได้เล่นไปในหน้าจอการเล่นเกมที่ผ่านมา โดยถ้าคะแนนของผู้เล่นคนไหนมีคะแนนติดหนึ่งในสิบของอันดับคะแนนสูงสุดที่ได้บันทึกไว้ ก็จะมีการบันทึกคะแนนและรายละเอียดของผู้เล่นคนนั้น เช่น ชื่อ เพลงที่เล่น คะแนนที่ได้ และเปอร์เซ็นต์ของคะแนน (เนื่องจากแต่ละเพลงมีจำนวนของ bar ไม่เท่ากันทำให้คะแนนเต็มที่เป็นไปได้ของแต่ละเพลงไม่เท่ากันไปด้วย จึงใช้วิธีการคำนวณคะแนนสูงสุดเป็นแบบอัตราส่วนต่อจำนวนของ bar ทั้งหมด)



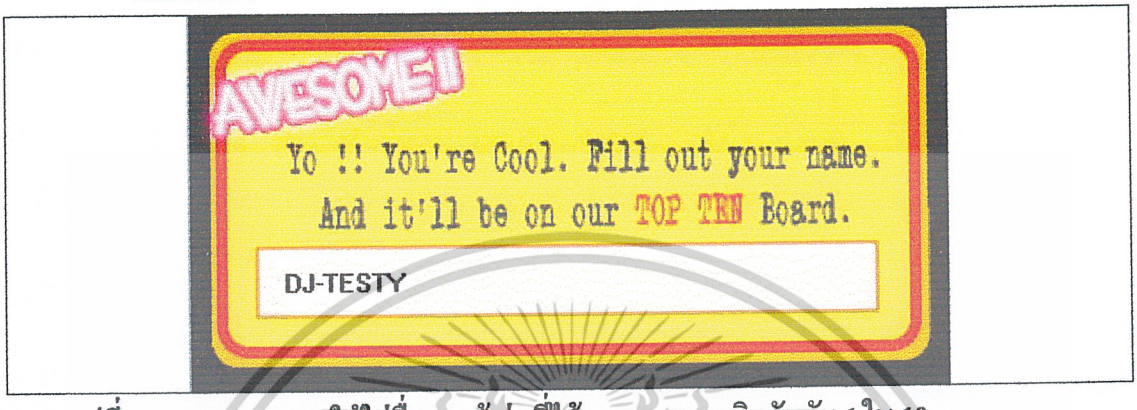
รูปที่ 3.62 หน้าจอของการสรุปผลคะแนน

- หน้าจอ result จะแสดงผลสรุปของคะแนนที่เล่นได้ และแสดงรูปภาพที่บ่งบอกว่าผู้เล่นฝ่ายไหนเป็นฝ่ายแพ้ โดยจากหน้าจอนี้ให้ทำการกดปุ่ม Enter เพื่อเปลี่ยนสถานะไปหน้าเริ่มเกม

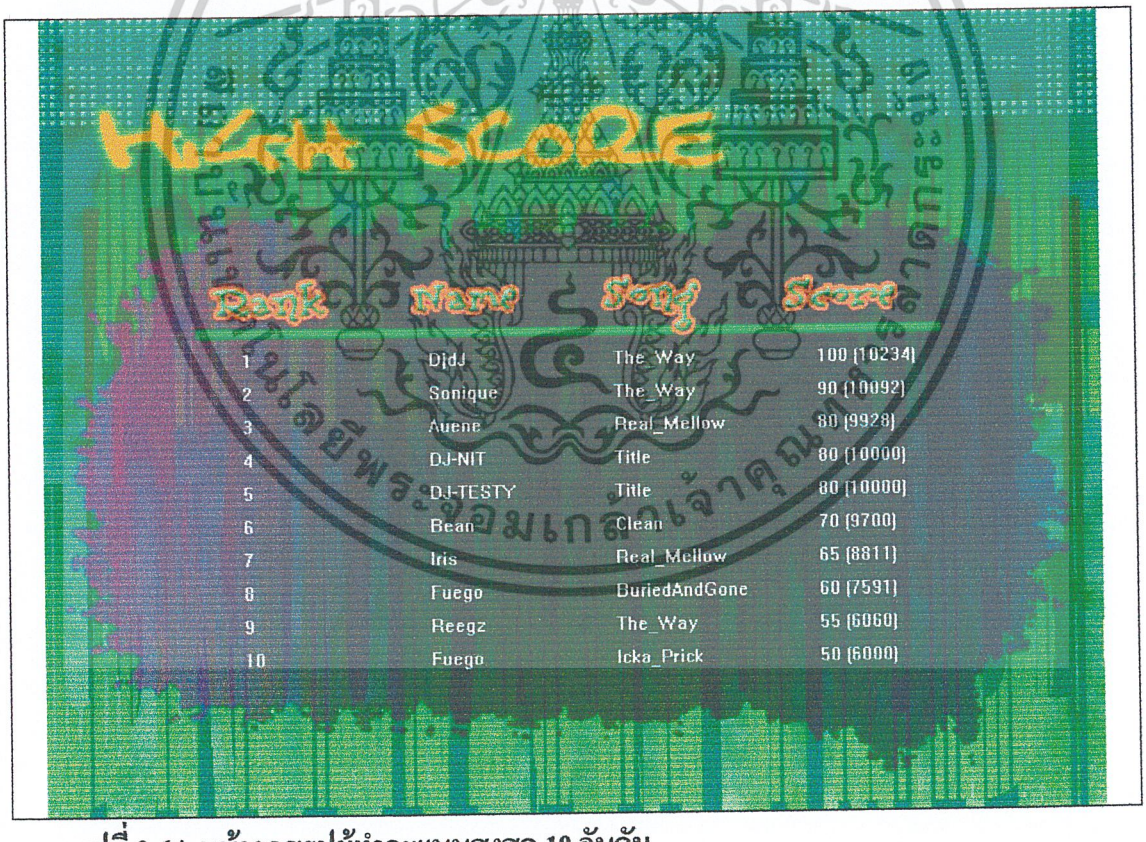
- คะแนนที่ได้จากการเล่นนั้นเมื่อนำไปเปรียบเทียบกับคะแนนสูงสุดในไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

hiscore.txt และจัดอันดับคะแนน หากได้คะแนนสูงสุดเป็นอันดับที่ 1 ถึง 10 โดยเปรียบเทียบจากคะแนนเดิมของผู้เล่นอื่นที่เคยทำอันดับสูงสุดได้ แล้วมีการเก็บบันทึกไว้ผลคะแนน และรายละเอียดเกมของผู้เล่นคนใหม่ที่สามารถทำคะแนนได้สูงสุดเป็นอันดับที่ 1 ถึง 10 นั้นจะถูกบันทึกลงไปในอันดับที่สามารถทำได้ แล้วบันทึกลงไฟล์ hiscore.txt



รูปที่ 3.63 Dialog Box ให้ใส่ชื่อของผู้เล่นที่ได้คะแนนสูงสุดติดอันดับ 1 ใน 10



รูปที่ 3.64 หน้าจอสรุปผู้ทำคะแนนสูงสุด 10 อันดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางคะแนนสูงสุด 10 อันดับ ประกอบด้วยรายละเอียดต่าง ๆ 4 คอลัมน์ดังนี้

- หมายเลขอันดับ (Rank)
- ชื่อของผู้เล่น (Name)
- เพลงที่เล่นได้คะแนน (Song)
- คะแนนที่เล่นได้ (Score)

ที่คอลัมน์ของ คะแนนที่เล่นได้ (Score) นั้นจะมีตัวเลขอยู่ 2 ตัวได้แก่ ตัวเลขที่อยู่นอกวงเล็บ และตัวเลขที่อยู่ในวงเล็บ คะแนนที่อยู่ในวงเล็บจะเป็นคะแนนที่ผู้เล่นสามารถทำคะแนนสูงสุดได้ติด 1 ใน 10 อันดับ ส่วนตัวเลขที่อยู่นอกวงเล็บจะเป็นเปอร์เซ็นต์ที่คำนวณอัตราส่วนของคะแนนที่ได้กับคะแนนสูงสุดของเพลงนั้น ๆ เนื่องจากแต่ละเพลงที่มีให้ในเกมจะมีจำนวนของตัวโน้ต (bar) ไม่เท่ากันทำให้เมื่อคิดคะแนนที่ได้จากเพลงหนึ่งที่มีตัวโน้ตมากกว่า จะทำให้ได้คะแนนมากกว่าเพลงอีกเพลงหนึ่งที่มีตัวโน้ตน้อยกว่า ทำให้ความผิดพลาดในการเปรียบเทียบคะแนนเกิดขึ้นได้ เราจึงต้องคิดเปอร์เซ็นต์เปรียบเทียบคะแนน เพื่อการจัดอันดับที่ถูกต้องเอาไว้ด้วย ดังตัวอย่างเช่น

- เพลงที่ 1 จำนวนตัวโน้ต 500 ตัว ดังนั้นจะมีคะแนนเต็ม 500 x 200 = 100,000 คะแนน (คะแนนสูงสุดคิดได้จากการนำเอาจำนวนของตัวโน้ตไปคูณกับคะแนนในระดับสูงสุด หรือระดับ Cool โดยจะได้คะแนน 200 คะแนนสำหรับตัวโน้ตหนึ่งตัวโน้ต)

  ผู้เล่น A ทำคะแนนได้ 82,000 คะแนน

  ดังนั้นผู้เล่น A ทำคะแนนได้ $\frac{82,000}{100,000} \times 100 = 82$ เปอร์เซ็นต์

- เพลงที่ 2 จำนวนตัวโน้ต 450 ตัว ดังนั้นจะมีคะแนนเต็ม 450 x 200 = 90,000 คะแนน

  ผู้เล่น B ทำคะแนนได้ 81,000 คะแนน

  ดังนั้นผู้เล่น A ทำคะแนนได้ $\frac{81,000}{90,000} \times 100 = 90$ เปอร์เซ็นต์

จะเห็นได้ว่าถึงแม้ว่าผู้เล่น A จะได้คะแนนมากกว่าแต่อันดับของผู้เล่น B จะสูงกว่าเนื่องมาจากเปอร์เซ็นต์ของคะแนนที่ได้สูงกว่า

3.4 ขั้นตอนการทดสอบเกม

เป็นขั้นตอนสุดท้ายของการทำเกม โดยขั้นตอนนี้เป็นการตรวจสอบความผิดพลาดและประสิทธิภาพโดยรวมของเกมทั้งหมด ว่าสามารถโต้ตอบและแสดงผลได้ตามที่ออกแบบไว้หรือไม่ ถ้าตรวจสอบพบความผิดพลาดจะได้นำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขให้มีความถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและการวิเคราะห์ปัญหา

การทดลองและการวิเคราะห์ปัญหาที่จะกล่าวถึงในบทนี้ จะเป็นขั้นตอนของการทดสอบและผลที่ได้จากการทดสอบในแต่ละส่วนของเกมทั้งหมด โดยผลการทดสอบที่ได้นี้ จะถูกนำไปวิเคราะห์ถึงปัญหาและแนวทางในการแก้ปัญหาเพื่อการพัฒนาต่อไปในอนาคต โดยทางผู้จัดทำหวังเป็นอย่างยิ่งว่าผลการทดลองและแนวทางในการแก้ปัญหาที่ได้จากการทดสอบในครั้งนี้ จะเป็นประโยชน์และแนวทางต่อผู้ที่มีความสนใจจะศึกษาและค้นคว้าเพื่อจะนำไปพัฒนาต่อ ให้ได้เห็นถึงปัญหา และข้อดีข้อเสียของปัญหาโดยผู้ที่จะศึกษาไม่จำเป็นต้องทำการทดสอบเพื่อหาปัญหาด้วยตนเองอีกครั้ง

คุณสมบัติของระบบที่นำมาทดสอบ

1. ระบบปฏิบัติการ Microsoft Windows XP Professional
2. BIOS: ASUS - 42302e31
3. หน่วยประมวลผลกลาง Intel(R) Pentium(R) 4 CPU 2.53GHz
4. หน่วยความจำหลักขนาด 256MB
5. VGA Card ของ NVIDIA GeForce4 MX 440 หน่วยความจำบนการ์ดขนาด 64 MB
6. ระบบเสียง Sound Card ของ C-Media Wave Device
7. หน่วยความจำสำรองขนาด 20 GB โดยมีพื้นที่เหลือว่าง 9.29 GB

ขั้นตอนการดำเนินการทดสอบ

1. ทำการติดตั้งตัวโปรแกรมและคอมโพเนนต์ที่จำเป็นต้องใช้
2. ทำการประมวลผลภายใต้ระบบที่กำหนด
3. ทดสอบความสมบูรณ์ของเกม
4. ทดสอบหาข้อผิดพลาด
5. ประเมินประสิทธิภาพของเกม

จุดประสงค์ของการดำเนินการทดสอบ

1. หลังจากการติดตั้งตัวโปรแกรมแล้ว สามารถใช้งานโปรแกรมได้
2. การประมวลผลคำสั่งในหน้าจอต่าง ๆ ต้องเป็นไปตามที่กำหนด
3. ข้อผิดพลาดที่เกิดขึ้นจะต้องไม่มี หรือมีน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ขั้นตอนการดำเนินการทดสอบการติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็นต้องใช้

ตารางที่ 4.1 ขั้นตอนการดำเนินการทดสอบการติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็น

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
การติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็น	ทำการรันโปรแกรมโดยดับเบิ้ลคลิกที่ไฟล์ “Setup.exe” ในแผ่นซีดี	1. สามารถติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็นได้สมบูรณ์ 2. ผู้ใช้สามารถรันโปรแกรมได้

4.2 ขั้นตอนการดำเนินการทดสอบการประมวลผลภายใต้ระบบที่กำหนด

ตารางที่ 4.2 ขั้นตอนการดำเนินการทดสอบการประมวลผลภายใต้ระบบที่กำหนด

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
รันโปรแกรมภายใต้ระบบเครื่องคอมพิวเตอร์ที่กำหนด	ทำการรันโปรแกรมโดยดับเบิ้ลคลิกที่ไฟล์ “Groove Unlimited.exe” ในโฟลเดอร์ปลายทางที่ได้ทำการติดตั้ง	โปรแกรมสามารถทำงานได้โดยหน้าจอ Main
ทดสอบลำดับของการเปลี่ยนสถานะหน้าจอ	คลิกที่ปุ่มต่าง ๆ เพื่อเข้าสู่หน้าจอทุกส่วนของเกม	ลำดับของการเปลี่ยนสถานะหน้าจอถูกต้องตามที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกม

4.3.1 หน้าจอการเลือกเพลง

ตารางที่ 4.3 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการเลือกเพลง

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
การเปลี่ยนลำดับของการเลือกเพลง	กดปุ่มลูกศรขึ้นลงของคีย์บอร์ด	- แถบสีที่เลือกรายชื่อเพลงเลื่อนขึ้นลงตามการกดปุ่ม - รายละเอียดของเพลงแสดงที่หน้าจอด้านขวามือถูกต้อง
ทดสอบการเปลี่ยนหน้าของรายชื่อเพลง	- รันโดยใช้ไฟล์รายชื่อเพลงที่มีจำนวนเพลง 100 เพลง - กดปุ่มเลื่อนหน้าบริเวณตรงกลางด้านล่างของหน้าจอ	- รายชื่อเพลงแสดง 23 เพลงต่อหนึ่งหน้าจอ - หน้าจอเปลี่ยนไปแสดงรายชื่อเพลงในหน้าถัดไปได้ถูกต้อง เลขหน้ามีการเปลี่ยนแปลงที่ถูกต้อง
ทดสอบการกดปุ่มต่าง ๆ ที่เหลือของหน้าจอ	กดปุ่ม Play, Easy, Medium, Hard, 1 player และ 2 player	ปุ่มที่ถูกกดเปลี่ยนรูปไปอยู่ในลักษณะของการถูกกดถูกต้องทุกปุ่ม เมื่อกด Play มีการเปลี่ยนหน้าจอเข้าสู่หน้าจอการเล่นเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 หน้าจอการเล่นเกม

ตารางที่ 4.4 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการเล่นเกม

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบจำนวนของตัวโน๊ตและความเร็วการไหล เมื่อเลือกเล่นในระดับความยากต่าง ๆ	- เลือกเล่นระดับง่าย - เลือกเล่นระดับกลาง - เลือกเล่นระดับยาก	- จำนวนโน้ตน้อย ความเร็วการไหลช้า - จำนวนโน้ตเพิ่มขึ้นมา ความเร็วการไหลเร็วขึ้น - จำนวนโน้ตมาก ความเร็วการไหลเร็วมาก
ทดสอบการเล่นแบบ 1 ผู้เล่น และแบบ 2 ผู้เล่น	- เลือก 1 ผู้เล่น - เลือก 2 ผู้เล่น	- ตัวโน๊ตถูกสร้างและอยู่ที่ตำแหน่งของผู้เล่นที่ 1 ถูกต้อง - ตัวโน๊ตถูกสร้างและอยู่ทั้งตำแหน่งของผู้เล่นที่ 1 และ ผู้เล่นที่ 2 ถูกต้อง แถบพลังของแต่ละฝ่ายมีการเปลี่ยนแปลงตามกติกาที่ได้วางเอาไว้ในการเล่นแบบ 2 ผู้เล่นได้ถูกต้อง
ทดสอบจำนวนโน้ตที่ถูกสร้างขึ้นของเพลงหนึ่ง ๆ	สุ่มเลือกเล่นเพลง 5 เพลง เพลงละ 2 ครั้ง ทำการนับจำนวนโน๊ตที่ไหลลงมา	จำนวนโน้ตสูญหายไปบ้างประมาณ 1- 2 ตัวโน๊ต จากการทดสอบทั้งหมด *
ทดสอบการเล่นตามกติกา	เล่นเกมตั้งแต่เริ่มจนจบเกมตามปรกติ	จำนวนโน๊ต ความเร็วการไหลถูกต้องตามกติกาที่ได้เลือกเล่น จังหวะการกดปุ่มคีย์บอร์ดสัมพันธ์กับจังหวะของดนตรี และการไหลมาถึงจุดคิดคะแนน (เส้นสีแดง) เป็นไปอย่างถูกต้อง รูปภาพบ่งบอกสถานะต่าง ๆ แสดงอย่างปรกติ
ทดสอบการกดปุ่มต่าง ๆ ของคีย์บอร์ดที่ใช้ภายในหน้าจอ	กดปุ่มคีย์บอร์ดตามที่ได้ตั้งค่าเอาไว้ใน Option/Key Config	ปุ่มทุกปุ่มทำงานได้ปรกติ และถูกต้องตามการกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 หน้าจอสรุปผลคะแนน

ตารางที่ 4.5 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการสรุปผลคะแนน

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบผลลัพธ์ของคะแนนที่ได้	ผู้ทดสอบคนหนึ่งทำการเล่นเกม และผู้ทดสอบคนที่เหลือทำการจดบันทึกว่าในเกมนั้น ๆ ได้คะแนนในระดับใดเป็นเท่าไรบ้าง โดยถือเอาตามกติกา จดผลคะแนนรวม แล้วนำมาเปรียบเทียบกับคะแนนที่ได้จากหน้าจอนี้	ผลลัพธ์ของระดับคะแนนทุกช่วง คะแนนรวม และข้อมูลทุกอย่างแสดงออกมาถูกต้อง เมื่อมีการเล่น 2 ผู้เล่น ฝ่ายที่แพ้จะมีรูปภาพแสดงอยู่ที่กึ่งกลางของผลลัพธ์ว่า “Loser” ได้ถูกต้อง

4.3.4 หน้าจอสอนการเล่นเกม

ตารางที่ 4.6 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการสอนการเล่นเกม

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบลำดับของการสอนการเล่นเกม	เปรียบเทียบผลจากการรันโปรแกรมในหน้าจอของการสอนการเล่นเกม กับการเขียนโปรแกรม	ลำดับของการสอน รูปภาพ และเสียงเพลงในหน้าจอนี้ถูกต้องตามที่ได้กำหนดเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 หน้าจอนำเพลงเข้าสู่เกม

ตารางที่ 4.7 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการนำเพลงเข้าสู่เกม

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบเพลงที่ได้จากการนำเข้า	ทำการนำเข้าเพลงแล้วเปิดไฟล์ที่ได้จากการนำเข้ามาเปรียบเทียบกัน แล้วลองเล่นเกมด้วยเพลงที่ถูกนำเข้านี้	ขั้นตอนการนำเข้า ทำการจับเสียงตัวอย่างที่มีความถี่สูงกว่าเสียงอื่น ๆ ขึ้นมาเก็บเป็นตัวโน๊ตแล้วทำการสุ่ม Key ที่จะกดให้ได้อย่างถูกต้องเหมาะสมทั้งเพลง เมื่อได้ทดสอบเล่นแล้วจังหวะโน๊ตที่ผ่านขั้นตอนการนำเข้าเพลงสัมพันธ์กับจังหวะของเพลงจริง ๆ ปานกลางถึงดี

4.3.6 หน้าจอตั้งค่าตัวเลือก

ตารางที่ 4.8 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการตั้งค่าตัวเลือก

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการกดปุ่มต่าง ๆ	กดปุ่มเลือก Key Config, Sound Test และ High Score ดูความเปลี่ยนแปลงที่เกิดขึ้น	การเปลี่ยนหน้าจอจากการกดปุ่มเลือกเมนูต่าง ๆ เปลี่ยนได้อย่างถูกต้อง โดยเมื่อกดปุ่ม Key Config หน้าจอจะเปลี่ยนไปยังหน้าจอของการปรับตั้งค่าคีย์บอร์ด กดปุ่ม Sound Test หน้าจอจะเปลี่ยนไปยังหน้าจอของการทดสอบฟังเพลง กดปุ่ม High Score หน้าจอจะเปลี่ยนไปยังหน้าจอของคะแนนสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 หน้าจอการปรับตั้งค่าคีย์บอร์ด

ตารางที่ 4.9 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในการปรับตั้งค่าคีย์บอร์ด

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการเปลี่ยนค่าคีย์บอร์ดในหน้าจอ	ให้ทำการเลือกที่ปุ่มที่ต้องการจะเปลี่ยน แล้วพิมพ์ค่าตัวอักษรที่ต้องการลงไปดูการเปลี่ยนแปลง	ตัวอักษรที่ต้องการเปลี่ยน ถูกเปลี่ยนอย่างถูกต้องตามที่ได้กดปุ่มคีย์บอร์ดของตัวอักษรนั้น ๆ ภาพแสดงสถานะการเลือกของแต่ละปุ่มทำงานได้อย่างปรกติทุกปุ่ม
ทดสอบการเปลี่ยนค่าคีย์บอร์ดในไฟล์	หลังจากการเปลี่ยนแปลงที่หน้าจอแล้วให้กดปุ่ม S เพื่อทำการบันทึกค่าที่เปลี่ยนแปลงลงยังไฟล์ แล้วลองเข้าไปยังหน้าจอเล่นเกมเพื่อทดสอบเล่นเกมโดยการใช้ปุ่มคีย์บอร์ดตามที่ได้ปรับตั้งไว้	คีย์บอร์ดที่ใช้เล่นเกมเปลี่ยนไปตามที่ได้ปรับตั้งค่าไว้ในหน้าจอการปรับตั้งค่าคีย์บอร์ดอย่างถูกต้อง เล่นเกมได้ปรกติด้วยปุ่มที่ปรับตั้งค่าไว้

4.3.8 หน้าจอทดสอบฟังเพลง

ตารางที่ 4.10 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมส่วนของการทดสอบฟังเพลง

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการทำงานของการทดสอบฟังเพลง	ทดสอบเลือกเพลง แล้วทำการกดปุ่ม เล่น(play) หยุด(stop) หยุดชั่วคราว (pause) เล่นถัดไป (Next) เล่นก่อนหน้า (Back) แล้วดูผลที่เกิดขึ้นจากการทำงานกับการกดปุ่มนั้น ๆ	หลังจากการเลือกเพลงแล้วกดเล่นสามารถทำงานในทุกหน้าที่ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.9 หน้าจอคะแนนสูงสุด 10 อันดับ

ตารางที่ 4.11 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของคะแนนสูงสุด 10 อันดับ

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการอ่านไฟล์ “hiscore.gro” ขึ้นมาแสดงในหน้าจอคะแนนสูงสุด	เปิดหน้าจอคะแนนสูงสุด ดูผลของผู้เล่นที่ทำคะแนนสูงสุด 10 อันดับแล้วนำไปเปรียบเทียบกับข้อมูลในไฟล์ “hiscore.gro”	ข้อมูลที่ได้จากการอ่านไฟล์ขึ้นมาแสดง ตรงกับไฟล์จริง การแสดงค่าต่าง ๆ ในแต่ละแถว และคอลัมน์ทำได้ถูกต้อง
ทดสอบการบันทึก และจัดอันดับเมื่อมีผู้เล่นทำคะแนนได้สูงสุด 1 ใน 10 อันดับ ลงไฟล์ “hiscore.gro”	ทำการเตรียมไฟล์ “hiscore.gro” จากระดับคะแนน 0 คะแนน ทั้ง 10 อันดับ จากนั้นทำการเล่นเกมหลาย ๆ ครั้ง ให้ได้คะแนนสูงสุดต่าง ๆ กันหลาย ๆ คะแนน โดยแต่ละครั้งที่ผู้เล่นทำได้คะแนนสูงกว่าคะแนนอันดับสุดท้าย ในไฟล์ (ได้คะแนนติดอันดับ 1 ใน 10 ของอันดับคะแนนสูงสุด) เกมจะมีการบันทึกข้อมูลเพลงที่ผู้เล่นเล่น และให้ผู้เล่นพิมพ์ชื่อตัวเองเพื่อทำการบันทึก	หลังจากการทดสอบหน้าจอของการใส่ชื่อผู้เล่น และหน้าจอแสดงคะแนนสูงสุด 10 อันดับทำงานปรกติ อันดับของระดับคะแนนที่ได้จากการเล่นเกมที่ติด 1 ใน 10 ของอันดับคะแนนสูงสุดถูกเรียง และแสดงไว้อย่างถูกต้อง และจากการทดสอบหลาย ๆ ครั้ง ผลลัพธ์ของหน้าจอแสดงคะแนนสูงสุด 10 อันดับ แสดงข้อมูลล่าสุดได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ทดสอบหาข้อผิดพลาด

ในขั้นตอนของการทดสอบเพื่อหาข้อผิดพลาดจะไม่มีรูปแบบที่ตายตัวชัดเจน ที่จะทดสอบให้ได้ข้อผิดพลาดที่เกิดขึ้น โดยวิธีการทดสอบที่ดีที่สุดคือการทดสอบให้ครอบคลุมทุกส่วนของโปรแกรม และทดสอบหลาย ๆ ครั้งให้ได้ผลของการทดสอบที่แน่นอน โดยจากการทดสอบทั้งหมดของโปรแกรมทำให้เราสามารถหาข้อผิดพลาดได้ 2 จุดดังนี้

1) ปัญหาของการที่ตัวโน๊ตที่ควรจะไหลลงมาจากหน้าจอมีจำนวนไม่ครบตามที่ควรจะเป็น โดยหายไปประมาณ 1 – 2 ตัว ในหน้าจอการเล่นเกม

โดยปัญหานี้อาจจะเกิดจากการที่เมื่อวนลูปเกมด้วยลูป while(1) ทำให้คำสั่งต่าง ๆ ภายในลูปมีการขึ้นอยู่กับ ความเร็วของหน่วยประมวลผลกลาง และส่งผลกระทบโดยตรงกับการเปรียบเทียบเวลาในการสั่งให้ตัวโน๊ตไหลลงมา เนื่องจากเวลาที่ใช้ในการเปรียบเทียบ ในการจับจังหวะต่าง ๆ ของตัวโน๊ตในการไหล การสร้าง การคิดคะแนน กับเวลาของเพลงมีหน่วยเป็นมิลลิวินาที (Millisecond) ทำให้ตรวจสอบเงื่อนไขนี้ได้ยากเนื่องจากหน่วยมิลลิวินาทีปรับเพิ่มค่ารวดเร็วมาก

2) ปัญหาของการที่เมื่อใส่ค่าคีย์บอร์ดใหม่ลงไปในช่องรับค่า โดยค่าเก่ายังคงค้างอยู่ทำให้เห็นเป็นตัวอักษร 2 ตัว ซ้อนกันอยู่ ในหน้าของการปรับตั้งค่าคีย์บอร์ด

โดยปัญหานี้อาจจะเกิดจากการที่บัฟเฟอร์ของการแสดงผลออกทางหน้าจอของ DirectX จะต้องใช้ 2 บัฟเฟอร์ ได้แก่ Front Buffer และ Back Buffer เมื่อต้องการจะแสดงผลก็ให้นำข้อมูลไปใส่ไว้ที่ Back Buffer แล้วทำการสลับบัฟเฟอร์มาแสดงผลที่หน้าจอ แล้วมีการนำข้อมูลใส่ลงไปผิดบัฟเฟอร์ทำให้การสลับข้อมูลขึ้นมาแสดงผิดพลาด

4.5 ประเมินประสิทธิภาพของเกม

จากการทดสอบที่ผ่านมาทั้งหมดทำให้เราสามารถประเมินประสิทธิภาพโดยรวมของเกมได้ว่าเกมนี้สามารถทำงานได้ตามที่ออกแบบไว้สมบูรณ์พอสมควร โดยมีข้อผิดพลาดเล็กน้อยที่เกิดขึ้นโดยการทดสอบจากระบบที่กำหนดไว้ข้างต้นซึ่งอาจจะสามารถเปลี่ยนแปลง เพิ่มหรือลดประสิทธิภาพได้ ถ้าหากมีการเปลี่ยนระบบที่ใช้ในการประมวลผลของเกม และเนื่องจากเกมนี้มีการติดต่อกับเสียงและภาพที่ค่อนข้างมากทำให้ระบบที่จะนำเกมนี้ไปประมวลผลต้องค่อนข้างทำงานได้ดีและมีประสิทธิภาพสูงพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

5.1.1 การศึกษาและรวบรวมข้อมูล

การศึกษาและรวบรวมข้อมูลที่จำเป็นต้องใช้ในการพัฒนาเกมทำได้โดยการศึกษาและรวบรวมข้อมูลทำได้โดยศึกษาความเป็นไปได้ที่จะพัฒนาเกมนี้บนเครื่องคอมพิวเตอร์ได้หรือไม่ เมื่อทำการวิเคราะห์และตัดสินใจได้แล้วก็ทำการรวบรวมข้อมูลเพื่อสร้างกฎและกติกาที่จะนำมาเป็นแนวทางในการพัฒนาเกม พร้อมทั้งทำการศึกษาขั้นตอนวิธีการต่าง ๆ ในการพัฒนาเกม โดยอาจศึกษาจากเนื้อหาวิชา หรือคู่มือวิธีใช้ของเครื่องมือที่จำเป็นต้องใช้ หรืออาจจะศึกษาจากตัวเกมต่าง ๆ ที่มีอยู่ จากนั้นรวบรวมความรู้ทั้งหมดเพื่อนำไปใช้เป็นโครงสร้างของเกม เพื่อง่ายต่อการออกแบบหน้าจอส่วนติดต่อผู้ใช้ ออกแบบโครงสร้างและวิธีการในการพัฒนา และการเลือกใช้เครื่องมือต่าง ๆ ที่เหมาะสมและสนับสนุนในการพัฒนา

5.1.2 การวิเคราะห์และการออกแบบเกม

การออกแบบเกมต้องทำให้เกมมีความสวยงามดึงดูดผู้เล่น มีกติกาที่ชัดเจน สามารถใช้งานได้ง่าย โดยการวิเคราะห์และออกแบบ จะนำข้อมูลต่าง ๆ ที่รวบรวมไว้มาประกอบกับความรู้ความสามารถที่มีอยู่ รวมไปถึงความรู้ที่ได้มาจากการศึกษาค้นคว้าเพิ่มเติม จากนั้นทำการออกแบบกฎและกติกา วางโครงสร้างและออกแบบหน้าจอส่วนติดต่อผู้ใช้ รวมถึงขั้นตอนวิธีการต่าง ๆ ในการพัฒนาเกมให้เสร็จตามระยะเวลาที่กำหนด แล้วจึงทำการแบ่งหน้าที่ และมอบหมายส่วนงานต่าง ๆ ให้กับสมาชิก

5.1.3 การสร้างตัวละคร ภาพ และเสียงต่าง ๆ

ขั้นตอนนี้สามารถทำควบคู่ไปได้กับการเขียนโปรแกรม โดยเกมกรู๊ฟ อันลิมิเต็ดนี้ได้ใช้เครื่องมือหลายอย่างที่ช่วยในการออกแบบและสร้างตัวละคร ภาพ และเสียงต่าง ๆ รวมทั้งการสร้างภาพเคลื่อนไหวภายในเกม โดยเครื่องมือที่ใช้ในการออกแบบและสร้างงานกราฟิกได้แก่ โปรแกรมสร้างรูปภาพ Adobe Illustrator และ โปรแกรมตกแต่งรูปภาพ Adobe Photoshop ส่วนเครื่องมือที่ใช้ในการออกแบบและสร้างงานด้านเสียงเอฟเฟกต์และเสียงประกอบ ได้แก่ โปรแกรมการสร้าง และตัดต่อเสียง Fruity Loop และโปรแกรมการตัดต่อเสียง Adobe Audition เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 การพัฒนาโปรแกรม

ขั้นตอนนี้จะเป็นขั้นตอนที่สำคัญและใช้ระยะเวลานานที่สุดในการพัฒนาเกม เพราะว่าขั้นตอนนี้จะเป็นการนำความรู้ที่เรารวบรวมได้ทั้งหมด มารวมกับโครงสร้าง รูปแบบ และกติกาของเกมที่เราจะพัฒนา รวมไปถึงภาพ และเสียงที่เราได้จัดเตรียมไว้ และเมื่อเริ่มต้นพัฒนาก็ต้องมีการค้นคว้าหาความรู้ใหม่ ๆ มากมายนอกเหนือจากที่ค้นคว้าไว้ในตอนแรก มีการวิจัยและพัฒนาอัลกอริธึมที่ใช้ในการควบคุมกระบวนการของเกมให้เป็นไปตามกฎและกติกาที่วางเอาไว้ และทำการทดสอบการทำงานของเกมที่พัฒนาไปเรื่อย ๆ จึงต้องใช้ระยะเวลาในขั้นตอนนี้มากพอสมควร ซึ่งขั้นตอนการทำงานในส่วนนี้มีซอร์ฟแวร์ และเครื่องมือที่ใช้ในการพัฒนาดังนี้ Microsoft Visual C++ 6.0, DirectX SDK, CDX Game Engine และ FMOD Sound System

5.2 ข้อจำกัดของโปรแกรม

เกมกรู๊ฟ อันลิมิเต็ด เป็นเกมที่พัฒนาบนพื้นฐานโครงสร้างของ Microsoft Windows และทำงานที่ความละเอียด 800 x 600 pixels โดยที่เกมนี้ทำงานร่วมกับภาพ และเสียงค่อนข้างมาก รวมไปถึงการตรวจสอบเงื่อนไขบางเงื่อนไขภายในเกมโดยขึ้นอยู่กับหน่วยประมวลผลกลาง เช่นการตรวจสอบค่าเวลาที่มีหน่วยเป็นมิลลิวินาที ทำให้เกมนี้ต้องทำงานอยู่บนระบบคอมพิวเตอร์ที่มีประสิทธิภาพสูงพอสมควร โดยรายละเอียดของข้อจำกัดของโปรแกรมมีดังต่อไปนี้

1. ระบบเครื่องคอมพิวเตอร์ที่ใช้รันเกมต้องมีประสิทธิภาพที่สูง เนื่องจากเกมจำเป็นต้องมีการประมวลผลของภาพกราฟิก และเสียง ค่อนข้างมาก
2. เครื่องคอมพิวเตอร์ที่ใช้ควรเป็นเครื่อง Personal Computer หรือเครื่อง Notebook Computer ที่มีการใช้พลังงานจากไฟฟ้า เนื่องจากเครื่อง Notebook Computer ที่มีการใช้พลังงานจากแบตเตอรี่นั้นทางด้านของระบบปฏิบัติการนั้นจะลดประสิทธิภาพที่จะใช้ประมวลทางด้านภาพกราฟิกและเสียงลงเพื่อประหยัดพลังงานของแบตเตอรี่
3. เกมกรู๊ฟ อันลิมิเต็ด ต้องการ DirectX เวอร์ชัน 8 ขึ้นไป
4. การพิมพ์ข้อความในเกมกรู๊ฟ อันลิมิเต็ดห้ามมีการเว้นวรรค หากมีการเว้นวรรค เกมจะทำการใช้ตัวอักษรขีดล่าง (underscore) แทนที่ให้ทันที
5. การนำเพลงเข้าสู่เพลง (Import Song) ต้องนำเพลงที่มีนามสกุล mp3, wav และ ogg เข้าสู่เพลงเท่านั้น และต้องนำเพลงไปเก็บไว้ในโฟลเดอร์ music ก่อน จึงรันเกมแล้วเข้าเมนู Import Song กรอกรายละเอียดให้ครบทุกช่อง แล้วจึงกดปุ่ม Import
6. ห้ามมีการเคลื่อนย้ายโฟลเดอร์ต่าง ๆ จากตำแหน่งที่ติดตั้งเกม เพราะว่าเกมมีการเรียกใช้ข้อมูลที่อยู่ในโฟลเดอร์เหล่านั้น โดยหากเกมหาข้อมูลที่จำเป็นต้องใช้ในการทำงานไม่เจอก็จะเกิดความผิดพลาดได้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เพลงที่นำสู่เพลงเข้าไม่ควรมีความยาวมากเกินไป เนื่องจากเพลงที่ยาวอาจก่อให้เกิดความเบื่อหน่ายระหว่างการเล่นเกม และเพลงที่ยาวอาจทำให้การสร้างตัวโน้ตที่ได้จากการวิเคราะห์ในเมนู Import Song มีจำนวนมากเกินไปไม่เพียงพอกับตัวแปรที่ประกาศเตรียมเอาไว้
8. จำนวนโน้ตที่เป็นไปได้มากที่สุดต่อ 1 เพลง คือ 500 ตัวโน้ต ดังนั้น การสร้างโน้ตในการนำเข้าเพลงของเพลงที่มีขนาดยาวเกินไปอาจมีตัวโน้ตเกินกว่าที่กำหนด

5.3 ข้อเสนอแนะ

เนื่องจากเกมกรู๊ฟ อันลิมิเต็ดเป็นเกมที่มีกฎและกติกาค่อนข้างซับซ้อน และปลีกย่อยมากมาย จึงทำให้ในการพัฒนาจำเป็นต้องมีอัลกอริธึมในการควบคุมกระบวนการของเกมที่ดี และต้องมีการพัฒนาโปรแกรมที่มีโครงสร้างซับซ้อนมากขึ้น ด้วยเหตุนี้ทำให้การออกแบบอัลกอริธึมของเกมต้องทำด้วยความรอบคอบ และต้องคอยตรวจสอบผลการทำงานของเกมไปทุกระยะ เพื่อตรวจหาข้อผิดพลาดที่อาจเกิดขึ้น นอกจากนี้ยังสามารถนำเกมนี้ไปพัฒนาโดยการสร้างกฎและกติกาใหม่ ๆ หรือความสามารถพิเศษใหม่ ๆ เพื่อให้ดูมีความเหมาะสมมากขึ้นได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

นิรุธ อำนวยศิลป์. 2547. **ซีดีสื่อการสอนเขียนเกมโดยใช้ CDX Game Engine และ Visual C++.** [CD-ROM]. กรุงเทพฯ : บริษัทไทยเดฟ จำกัด

นิรุธ อำนวยศิลป์. 2004. **C++ และแนวคิด OOP.** [online]. Available : http://www.thaidev.com/cpage.php

ประภาพร ช่างไม้. 2545. **คู่มือการเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น.** พิมพ์ครั้งที่ 1. นนทบุรี : อินโฟเพรส.

ยุทธนา ลีลาศวัฒนกุล. 2544. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ 6.0 ฉบับโปรแกรมเมอร์.** กรุงเทพฯ : อินโฟเพรส.

Brett Paterson and Andrew Scott. 2004. **FMOD Sound System Tutorial.** [online]. Available : http://www.fmod.org

File Processing: C++ File Streaming. [online]. Available : http://www.functionx.com/cpp/articles/filestreaming.htm

The C++ Resources Network. 2001. **Input/Output with files.** [online]. Available : http://www.cplusplus.com/doc/tutorial/tut6-1.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
การติดตั้งและใช้งานโปรแกรม

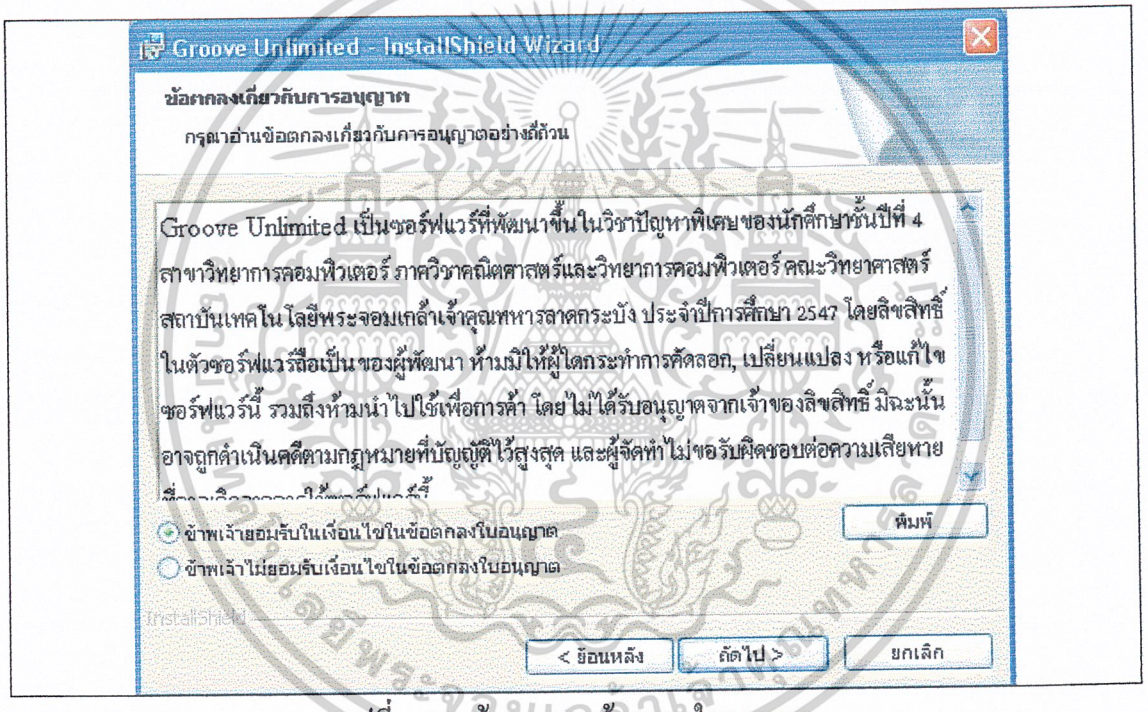
1. ขั้นตอนการติดตั้งเกมกรู๊ฟ อันลิมิเต็ด (Groove Unlimited)

1.1 ดับเบิ้ลคลิกที่ไฟล์ setup.exe ในแผ่น CD จะเข้าสู่หน้าจอการเริ่มติดตั้งเกม

รูปที่ ก-1 หน้าจอแรกของการ setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

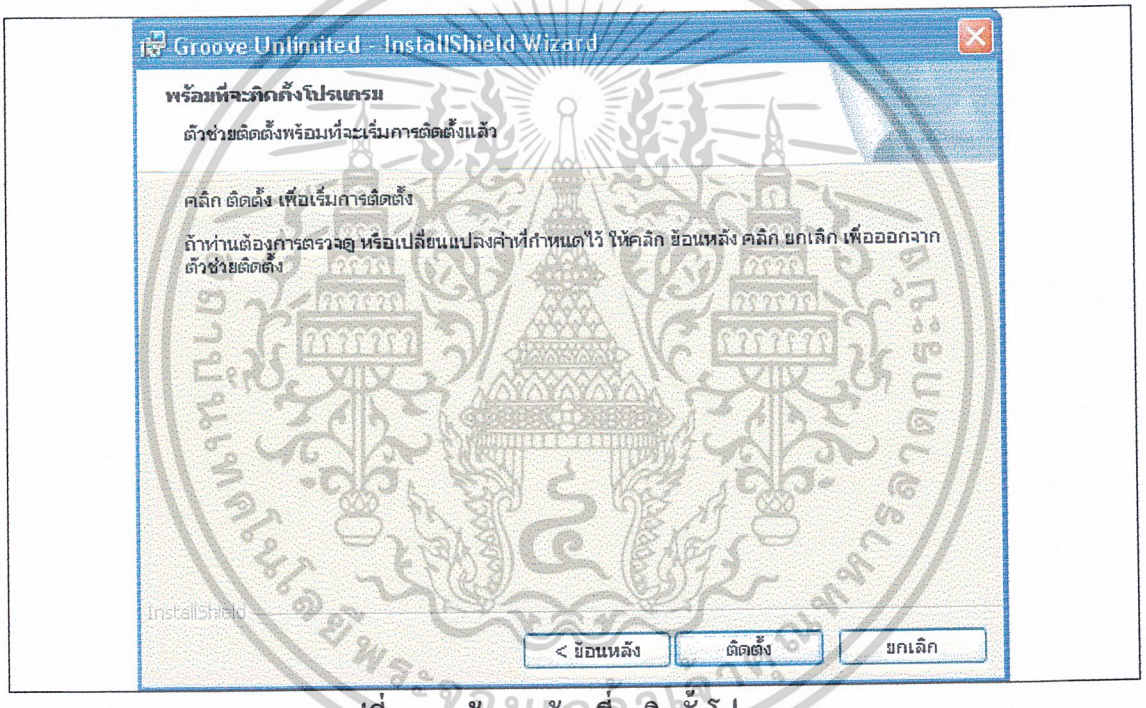
1.2 อ่านคำเตือนเกี่ยวกับลิขสิทธิ์แล้วคลิกเลือกยอมรับตามข้อตกลง แล้วกดปุ่มถัดไป จนมาถึงหน้าพร้อมที่จะติดตั้งโปรแกรม



รูปที่ ก-2 หน้าจอแสดงข้อตกลงในอนุญาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

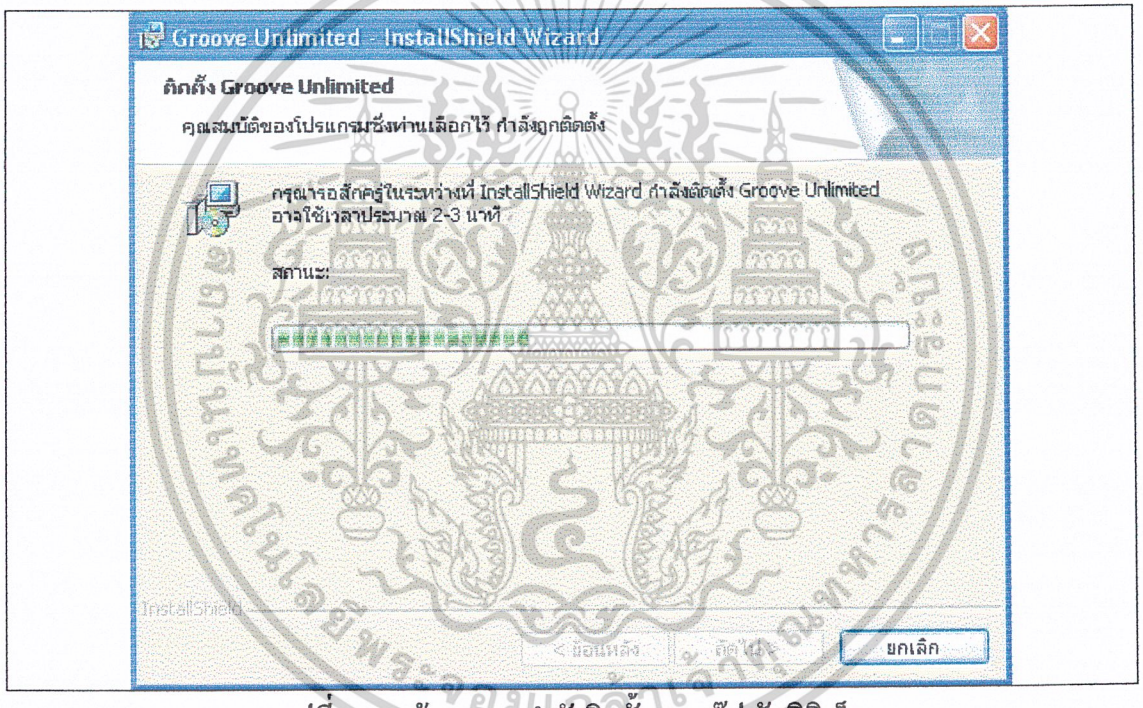
1.3 กดปุ่มติดตั้งเพื่อทำการติดตั้งเกมกรูฟ อันลิมิเต็ด



รูปที่ ก-3 หน้าจอพร้อมที่จะติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 โปรแกรมจะทำการติดตั้งเกมกรูฟอันลิมิเต็ดลงใน Directory ปลายทาง



รูปที่ ก-4 หน้าจอขณะกำลังติดตั้งเกมกรูฟ อันลิมิเต็ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้งานเกมกรู๊ฟ อันลิมิเต็ด

เมื่อทำการเข้าโปรแกรมเกมกรู๊ฟ อันลิมิเต็ดแล้วจะพบกับหน้าจอต่าง ๆ ซึ่งแต่ละหน้าจอจะมีการทำงานดังนี้

2.1 หน้าจอ Main

ในหน้าจอ Main มีการแสดงภาพฉากหลัง มีการแสดงตัวละครเพื่อใช้เป็นปุ่ม 5 ปุ่ม ได้แก่ ปุ่ม Game Start, ปุ่ม Training, ปุ่ม Import Song, ปุ่ม Option และ ปุ่ม Exit โดยแต่ละปุ่มมีการทำงานดังนี้

2.1.1 Game Start – เป็นปุ่มสำหรับเริ่มเล่นเกม เมื่อเลือกแล้วจะเข้าไปสู่หน้าเลือกเพลง (sound select) เพื่อนำเพลงที่เลือกไปเล่นเกมในหน้าเล่นเกม (play game)

2.1.2 Training – เป็นปุ่มเข้าสู่การสอนเล่นเกม เมื่อเลือกแล้วจะเข้าสู่หน้าสอนการเล่นเกมกรู๊ฟ อันลิมิเต็ด

2.1.3 Import Song – เป็นปุ่มที่ใช้ในการนำเข้าเพลงจากภายนอกเข้าสู่เกม เมื่อกดแล้วจะเข้าสู่หน้าจอการนำเข้าเพลง

2.1.4 Option – เมื่อกดแล้วจะเข้าสู่หน้าจัดการปรับตั้งค่าต่าง ๆ ภายในเกม

2.1.5 Exit (กดปุ่ม ESC) – เป็นปุ่มที่ใช้ในการออกจากเกม

2.2 หน้าจอ Sound Select

หน้าจอนี้มีหน้าที่ในการแสดงรายชื่อเพลงและรายละเอียดของเพลง จะแสดงรายชื่อเพลงทั้งหมดที่มีในเกมทางด้านตารางซ้ายมือของจอ และแสดงรายละเอียดของเพลงที่เลือกทางด้านช่องขวาบน ผู้เล่นสามารถคลิกปุ่มซ้าย ขวา เพื่อเปลี่ยนหน้าของรายชื่อเพลงได้ ในหนึ่งหน้าจอแสดงได้ 23 เพลง เมื่อผู้เล่นเลือกเพลงเรียบร้อยแล้วคลิกปุ่ม play เพื่อเข้าสู่หน้าจอของ Game Play และเริ่มเล่นเกม

2.2.1 Play – เป็นปุ่มสำหรับเริ่มเล่นเกมตามรายละเอียดของเกมที่เลือก

2.2.2 Easy – เป็นปุ่มสำหรับเลือกเล่นเกมในระดับง่าย

2.2.3 Medium – เป็นปุ่มสำหรับเลือกเล่นเกมในระดับปานกลาง

2.2.3 Hard – เป็นปุ่มสำหรับเลือกเล่นเกมในระดับปานยาก

2.2.4 1 player – เป็นปุ่มสำหรับเลือกเล่นเกมโดยมีผู้เล่น 1 คน

2.2.5 2 player – เป็นปุ่มสำหรับเลือกเล่นเกมโดยมีผู้เล่น 2 คน

2.2.6 กดปุ่ม ESC เพื่อออกจากหน้าจอ Sound Select กลับสู่หน้าจอหลักของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หน้าจอ Game Play

หน้าจอนี้เป็นส่วนที่สำคัญของเกมเพราะเป็นหน้าจอของการเล่นเกม ซึ่งหน้าจอนี้มีการจัดการเกี่ยวกับตัวละคร และอินพุตต่าง ๆ เมื่อเริ่มเกมจะมีตัวโน้ตไหลลงมาให้เราใช้ปุ่มที่ตั้งค่าไว้ใน Option กดเพื่อให้ตรงจังหวะที่ตัวโน้ตไหลลงมาตามช่องต่าง ๆ ส่วนทางด้านรายละเอียดอื่นของ หน้าจอ Game Play มีดังนี้

2.3.1 ปุ่มที่กดจะมี 6 ปุ่ม กดตามจังหวะเสียงเพลงโดยดูจากสัญลักษณ์ของโน้ตที่ไหลมาถึงเส้นสีแดงด้านล่าง

2.3.2 เมื่อกดปุ่มทั้ง 6 ปุ่มตามจังหวะที่ต้องกดจะมีข้อความบอกสถานะแจ้งขึ้นมาว่าตัวโน้ตตัวนั้นผู้เล่นกดปุ่มได้ตรงกับจังหวะที่ต้องกดมากแค่ไหน อยู่ระดับคะแนนระดับใด

2.3.3 แถบพลังของ player 1 และ player 2 จะเป็นแถบสีที่บอกพลังของแต่ละฝ่าย

2.3.4 เล่นจนกระทั่งจบเพลงเพื่อเข้าสู่หน้าจอ Result หรือกดปุ่ม ESC เพื่อออกจากหน้าจอ Game Play ไปสู่หน้าจอ Result ก่อนจบเพลงได้

2.4 หน้าจอ Result

จะเป็นหน้าจอที่สรุปผลคะแนนของผู้เล่นว่าเมื่อจบเกมได้คะแนนเท่าไร และได้คะแนนย่อยในแต่ละระดับเป็นอย่างไรบ้าง เมื่อมีการเล่นแบบ 2 player จะมีการแสดงรูปภาพบอกว่าฝ่ายแพ้คือฝ่ายไหน โดยกดปุ่ม ENTER เพื่อออกจากหน้าจอ Sound Select กลับสู่หน้าจอหลักของเกม หรือไปยังหน้าจอ High Score ถ้าคะแนนของผู้เล่นคนนั้นติดอันดับของอันดับคะแนนสูงสุด 10 อันดับที่ได้มีการบันทึกไว้

2.5 หน้าจอ Training

เป็นหน้าจอที่จะช่วยสอนวิธีการเล่นเกมกรู๊ฟ อันลิมิเต็ดเบื้องต้น โดยหน้าจอนี้จะไม่สามารถทำอะไรได้นอกจากการกดปุ่ม ESC เพื่อจบการสอน โดยในหน้าจอนี้จะสอนโดยการใช้ข้อความอธิบายการเล่นเกม โดยมีรูปภาพประกอบเป็นขั้นตอนเบื้องต้น โดยละเอียด

2.6 หน้าจอ Import Song

หน้าจอนี้ใช้สำหรับนำเพลงประเภท WAV, MP หรือ OGG เข้าสู่เกม ซึ่งเป็นเพลงที่ผู้เล่นต้องการเล่น และไม่ได้เตรียมให้อยู่แล้วในเกม ผู้เล่นสามารถนำเพลงเข้ามาได้ โดยการคัดลอกไฟล์เพลงที่ต้องการนำเข้ามาใส่ไว้ในโฟลเดอร์ music ภายใน Directory ที่ติดตั้งตัวเกมไว้ โดยชื่อเพลงต้องเป็นภาษาอังกฤษและไม่มีการเว้นวรรคภายในชื่อ จากนั้นให้เข้ามาในเกมในส่วนของ Import Song ใส่ข้อมูลในทุก ๆ ช่องให้ครบ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ช่อง filename ให้ใส่ชื่อและนามสกุลของไฟล์ที่ต้องการ Import
- ช่อง Title ให้ใส่ชื่อเพลงโดยชื่อเพลงในช่อง Title นี้ต้องเป็นชื่อเดียวกับชื่อของเพลงที่นำมาใส่ไว้ในโฟลเดอร์ music
- ช่อง Artist เป็นช่องที่ให้ใส่ชื่อศิลปินเจ้าของเพลงที่ต้องการจะ Import
- ช่อง range ให้ใส่ตัวเลขระหว่าง 0 ถึง 1 โดย range จะเป็นช่วงความถี่ที่เป็นระดับขอบเขตล่างสำหรับการวิเคราะห์ โดยถ้าใส่เลขที่มีค่าใกล้ 0 ช่วงความถี่ที่จะนำมาวิเคราะห์ก็จะต่ำ ดังนั้นเสียงที่มีเสียงเบาก็อาจถูกนำไปสร้าง bar ได้ และถ้าใส่เลขที่มีค่าใกล้ 1 ช่วงความถี่ที่จะนำมาวิเคราะห์ก็จะสูง ทำให้เกมจะนำเสียงเด่น ๆ ขึ้นมาสร้าง bar เป็นต้น
- ช่อง Type เป็นประเภทของไฟล์ที่จะนำมา Import โดยให้เติมได้เฉพาะ MP3, OGG และ WAV ได้เท่านั้น

จากนั้นกดปุ่ม Import เกมจะทำการวิเคราะห์ไฟล์เพลงเพื่อสร้างไฟล์ที่จะนำไปเป็นรูปแบบที่เกมสามารถนำมาเล่นได้ โดยเมื่อมีการวิเคราะห์จนจบเพลงแล้ว ให้ทำการกดปุ่ม Enter เพื่อบันทึกข้อมูลที่ได้จากการวิเคราะห์เพื่อนำไปเล่นในเกมได้ โดยหลังจากการกดปุ่ม Enter จะเข้าไปสู่หน้า Sound Select เพื่อทำการเล่นเกม หรือระหว่างการวิเคราะห์เพลงในหน้า Import Song สามารถกดปุ่ม ESC เพื่อยกเลิกการวิเคราะห์เพลง แล้วกลับสู่หน้าหลักของเกมได้

2.7 หน้าจอ Option

ภายในหน้าจอนี้มีปุ่มให้เลือกไปเมนูย่อยเพื่อการปรับแต่ค่าซึ่งมี 3 ปุ่ม ได้แก่ ปุ่ม Key Config, ปุ่ม Sound Test และ ปุ่ม High Score โดยมีรายละเอียดดังนี้

2.7.1 Key Config - เป็นปุ่มที่ใช้ในการปรับตั้งค่าคีย์บอร์ดที่จะใช้ในการเล่นเกม

2.7.2 Sound Test - เป็นปุ่มที่ผู้เล่นจะสามารถกดเพื่อเข้าไปลองฟังตัวอย่างเพลงที่มีอยู่ในเกมได้

2.7.3 High Score - เป็นปุ่มที่ผู้เล่นจะสามารถเข้าไปดูรายชื่อ และรายละเอียดผู้เล่นที่มีอันดับคะแนน (เปอร์เซ็นต์ของคะแนน) มากที่สุด 10 อันดับ

2.7.4 กดปุ่ม ESC เพื่อออกจากหน้าจอ Option กลับสู่หน้าจอหลักของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 หน้าจอ Key Config

หน้าจอ Key config มีไว้สำหรับปรับตั้งค่าคีย์บอร์ดที่ใช้สำหรับเล่นเกม โดยมีขั้นตอนการทำงานดังนี้

- ใช้เมาส์คลิกเลือกตำแหน่งปุ่มของแต่ละผู้เล่นที่ต้องการเปลี่ยนแปลง ให้ตำแหน่งที่ต้องการกระพริบ
- กดปุ่มคีย์บอร์ดที่ต้องการ
- คลิกเมาส์ที่ตำแหน่งเดิมอีกครั้ง จนการกระพริบหายไป
- ทำตามขั้นตอนตามข้างต้นจนครบทุกปุ่มของผู้เล่นแต่ละคน หรือตามที่ต้องการจะเปลี่ยนแปลง
- กดปุ่มคีย์บอร์ด "S" เพื่อทำการบันทึกข้อมูลที่ได้เปลี่ยนแปลง
- กดปุ่ม ESC เพื่อออกจากหน้าจอ Key config กลับสู่หน้าจอหลักของเกม

2.9 หน้าจอ Sound Test

เป็นหน้าจอสำหรับให้ผู้เล่นทดลองฟังเพลง โดยเลือกจากรายชื่อเพลงที่แสดงอยู่ในตารางด้านซ้าย และแสดงรายละเอียดของเพลงทางด้านกรอบมุมขวาบน หน้าจอนี้มีลักษณะคล้ายกับหน้าจอ Sound Select แต่มีหน้าที่ที่เพิ่มขึ้นมาคือการเล่นไฟล์เพลงตามที่ผู้เล่นต้องการ โดยการเขียนโปรแกรมในหน้าจอนี้จะรองรับอินพุตการคลิกเมาส์ที่ปุ่มต่าง ๆ ซึ่งมี 5 ปุ่ม ดังนี้

2.9.1 - เป็นปุ่มที่ใช้ในการเริ่มเล่นตัวอย่างเพลงที่มีอยู่ภายในเกม

2.9.2 - เป็นปุ่มที่ใช้ในการหยุดการเล่นเพลงที่กำลังเล่นอยู่ชั่วคราว

2.9.3 - เป็นปุ่มที่ใช้ในการหยุดเพลงที่กำลังเล่นอยู่

2.9.4 - เป็นปุ่มที่ใช้ในการเลือกเพลง และเล่นเพลงลำดับก่อนหน้า

2.9.5 - เป็นปุ่มที่ใช้ในการเลือกเพลง และเล่นเพลงลำดับถัดไป

2.9.6 กดปุ่ม ESC เพื่อออกจากหน้าจอ Sound Test กลับสู่หน้าจอหลักของเกม

2.10 หน้าจอ High Score

หน้าจอนี้มีไว้สำหรับแสดงผู้ที่มีคะแนนสูงสุด 10 อันดับแรก โดยอ่านข้อมูลจากไฟล์ที่ชื่อว่า highscore.gro แล้วแสดงตัวหนังสือออกทางหน้าจอ โดยจากกหน้าจอนี้สามารถกดปุ่ม ESC เพื่อออกจากหน้าจอ High Score กลับสู่หน้าจอหลักของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

กฎและกติกาการเล่น

1. กฎและกติกาของเกมกรู๊ฟ อันลิมิเต็ด (Groove Unlimited)

1.1 กฎและกติกาโดยรวมของการเล่นเกม

ในหน้าจอ Game Play นั้น จะมีการแบ่งส่วนสำหรับผู้เล่นออกเป็น 2 ส่วน แต่ละส่วนสำหรับผู้เล่น 1 คน โดยเกมนี้สามารถเล่นพร้อมกันได้สูงสุด 2 ผู้เล่น วิธีการเล่นเริ่มจากการเลือกเพลงที่ต้องการที่จะเล่นในหน้าจอ Sound Select และเลือกระดับความยากง่ายที่เกมได้จัดเตรียมไว้ให้ 3 ระดับ และประเภทของการเล่น 3 ประเภท ซึ่งแต่ละส่วนจะมีการคิดคะแนนและรายละเอียดของวิธีการเล่นที่แตกต่างกัน โดยมีภาพรวมของการเล่นคือ เกมจะทำการอ่านไฟล์ตัวโน้ตของเพลงที่ผู้เล่นได้เลือกเอาไว้มาทำการสร้างตัวโน้ตเตรียมก่อนเข้าสู่หน้า Game Play จากนั้นเมื่อเข้าสู่หน้า Game Play เกมจะทำการปล่อยให้ตัวโน้ตไหลลงมาตามช่องต่าง ๆ ซึ่งมีทั้งหมด 6 ช่องตามเวลาที่ได้วิเคราะห์เอาไว้ตามจังหวะของเพลง ๆ นั้น เมื่อตัวโน้ตไหลมาถึงเส้นสีแดง (ตามจังหวะของเพลง) ให้ผู้เล่นกดปุ่มของช่องนั้น ๆ ตามที่ได้ปรับตั้งค่าเอาไว้ในหน้าจอ Key Config ให้ตรงตามจังหวะของเพลงมากที่สุด โดยมีช่วงระดับในการคิดคะแนนเทียบระหว่างจังหวะในการกดกับจังหวะของเพลง ต่างกัน 4 ระดับ ดังนี้

- Cool จะได้รับคะแนน 200 คะแนน โดยจังหวะในการกดเมื่อเทียบกับจังหวะของเพลงไม่ช้าและไม่เร็วเกินกว่า 300 มิลลิวินาที
- Good จะได้รับคะแนน 100 คะแนน โดยจังหวะในการกดเมื่อเทียบกับจังหวะของเพลงจะช้าหรือเร็วเกินกว่า 300 มิลลิวินาที แต่ไม่เกิน 450 มิลลิวินาที
- Bad จะได้รับคะแนน 50 คะแนน โดยจังหวะในการกดเมื่อเทียบกับจังหวะของเพลงจะช้าหรือเร็วเกินกว่า 450 มิลลิวินาที แต่ไม่เกิน 600 มิลลิวินาที
- Miss จะไม่ได้รับคะแนน โดยจังหวะในการกดเมื่อเทียบกับจังหวะของเพลงจะช้าหรือเร็วเกินกว่า 600 มิลลิวินาที

โดยการกดได้ในระดับ Cool, Good และ Bad ได้ต่อเนื่องกันโดยไม่ตกในระดับ Miss จะมีการนับจำนวนครั้งของการกดได้ต่อเนื่อง (Combo) โดยจะทำการนับไปเรื่อย ๆ และเลิกนับเมื่อมีการ Miss และเกมจะเก็บค่า Combo รวมในครั้งนั้น ๆ เอาไว้เปรียบเทียบกับครั้งอื่น ๆ จนจบเกมเพื่อหาค่า Combo ที่มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 กติกาของแต่ละประเภทการเล่นเกม

เกมกรู๊ฟ อันลิมิเต็ด ได้แบ่งกติกาของการเล่นออกเป็นประเภทได้ 3 ประเภทดังนี้

- 1 Player เป็นการเล่นสำหรับ 1 ผู้เล่นในการคิดคะแนนจะคิดจากคะแนนรวมที่ได้จากการกดได้ในระดับต่าง ๆ โดยจะมีรายละเอียดของการกดได้ประเภทใด ๆ บ้าง อยู่บริเวณตรงกลางด้านล่างของหน้าจอ Game Play และมีแถบพลังชีวิตอยู่ในบริเวณเหนือขึ้นมาเล็กน้อย โดยแถบพลังชีวิตจะเพิ่มขึ้นเมื่อมีการกดได้ในระดับ Cool และ Good และจะลดลงเมื่อกดได้ในระดับ Miss โดยแถบพลังสำหรับการเล่นแบบ 1 ผู้เล่นนี้มีไว้เพียงเป็นสัญลักษณ์บอกถึงพลังชีวิตของผู้เล่น ซึ่งไม่ได้มีผลอะไรถ้าหากพลังชีวิตหมด และการเล่นแบบ 1 ผู้เล่นจะจบลงเมื่อเพลงจบ หรือมีการกดปุ่ม ESC ในระหว่างการเล่นเกม
- 2 Player เป็นการเล่นสำหรับ 2 ผู้เล่นในการคิดคะแนนจะคิดจากคะแนนรวมที่ได้จากการกดได้ในระดับต่าง ๆ โดยมีรายละเอียดของการเล่นบอกในระหว่างการเล่น และมีแถบพลังชีวิตรวมถึงการปรับเพิ่มลดของระดับพลังชีวิตเช่นเดียวกับการเล่น 1 ผู้เล่น โดยแถบพลังสำหรับการเล่นแบบ 2 ผู้เล่นนี้มีไว้เพียงเป็นสัญลักษณ์บอกถึงพลังชีวิตของผู้เล่น ซึ่งไม่ได้มีผลอะไรถ้าหากพลังชีวิตหมด และการเล่นแบบ 2 ผู้เล่นจะจบลงเมื่อเพลงจบ หรือมีการกดปุ่ม ESC ในระหว่างการเล่นเกม โดยผู้ชนะคือผู้ที่ทำคะแนนรวมได้มากกว่า
- VS MODE เป็นการเล่นสำหรับ 2 ผู้เล่นในการคิดคะแนนจะคิดจากคะแนนรวมที่ได้จากการกดได้ในระดับต่าง ๆ โดยมีรายละเอียดของการเล่นบอกในระหว่างการเล่นเช่นเดียวกับการเล่น 1 ผู้เล่น แต่พลังชีวิตจะถูกคิดคำนวณจากการกดของฝ่ายตรงข้าม เหมือนกับการปล่อยพลังใส่กันทางด้านดนตรี โดยแถบพลังชีวิตจะเพิ่มขึ้นเมื่อมีการกด Combo ได้มากกว่า 15 ครั้งติดต่อกัน และจะลดลงเมื่อฝ่ายตรงข้ามสามารถกดได้ในระดับ Cool และ Good และการเล่นแบบ 2 ผู้เล่นจะจบลงเมื่อเพลงจบ หรือมีการกดปุ่ม ESC ในระหว่างการเล่นเกม หรือพลังชีวิตของฝ่ายใดฝ่ายหนึ่งหมดลง โดยผู้ชนะคือผู้ที่เหลือพลังชีวิตมากกว่า

1.3 ระดับความยากง่ายของเกม

เกมกรู๊ฟ อันลิมิเต็ด ได้แบ่งระดับความยากง่ายออกเป็น 3 ระดับดังนี้

- Easy เป็นระดับที่ง่ายที่สุด โดยมีความเร็วและจำนวนตัวโน้ตปรกติสำหรับเพลงหนึ่ง ๆ
- Medium เป็นระดับความยากง่ายปานกลาง โดยมีความเร็วปรกติ แต่มีจำนวนตัวโน้ตที่เพิ่มมากขึ้น
- Hard เป็นระดับที่ยากที่สุด โดยมีความเร็วและจำนวนตัวโน้ตที่เพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 การคิดคะแนนสำหรับ 10 อันดับผู้เล่นสูงสุด

เนื่องจากการคิดคะแนนรวมของแต่ละเพลงที่เล่น ขึ้นอยู่กับจำนวนโน้ตของเพลงนั้น ๆ ทำให้การคิดคะแนนเปรียบเทียบของผู้เล่นที่เล่นกันคนละเพลงว่าคะแนนของใครสูงกว่า ก็ย่อมที่จะมีการได้เปรียบเสียเปรียบกันเกิดขึ้นจึงไม่สามารถที่จะนำคะแนนที่ได้มาเปรียบเทียบกันเพื่อหา 10 อันดับสูงสุด เกมกรู๊ฟ อันลิมิเต็ดจึงได้วิธีคิดเปอร์เซ็นต์ขึ้นมาใช้ในการเปรียบเทียบแทน โดยการคิดเปอร์เซ็นต์มีวิธีดังนี้

1.4.1 นำจำนวนตัวโน้ตไปคิดหาคะแนนรวมสูงสุดที่เป็นไปได้ของเพลงนั้น ๆ โดยคะแนนสูงสุดที่เป็นไปได้ คือการที่ผู้เล่นได้คะแนนในระดับ Cool ในการกดสำหรับทุก ๆ ตัวโน้ตของเพลงนั้น ๆ ซึ่งการได้ระดับ Cool 1 ครั้งจะได้รับคะแนนเพิ่ม 200 คะแนน ดังนั้นคะแนนรวมสูงสุดที่เป็นไปได้คือ

$$\text{คะแนนรวมสูงสุด} = \text{จำนวนตัวโน้ต} \times 200$$

1.4.2 นำคะแนนที่ได้เมื่อจบการเล่นเกมในเพลงนั้น ๆ ไปเปรียบเทียบกับคะแนนรวมสูงสุดแล้วคิดออกมาเป็นเปอร์เซ็นต์ได้ดังนี้

$$\text{เปอร์เซนต์คะแนน} = \frac{\text{คะแนนที่ได้}}{\text{คะแนนสูงสุด}} \times 100$$

1.4.3 จากนั้นนำเปอร์เซ็นต์ที่ได้ไปเปรียบเทียบกับคะแนนของ 10 อันดับสูงสุด แล้วทำการเปรียบเทียบว่าติดอันดับหรือไม่ ถ้าติดจะอยู่ในอันดับที่เท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้