

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาเกมคอมพิวเตอร์แนวเดินยิงมุมมองบุคคลที่หนึ่ง

DEVELOPMENT OF 3D FIRST-PERSON SHOOTING GAME



ชัชพล จัทรภูมิสุวรรณ

นฤเทพ วรงค์สิงห์

สาธิต โชตนะพันธ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เลขหมู่.....
เลขทะเบียน..... 59408
วันเดือนปี..... 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3D FIRST - PERSON SHOOTING



**CHUTCHAPOL CHATPOOMSUWAN
NARUETHEP VARONGSINGHARA
SATIT CHOATNAPHAN**

**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2005**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ

การพัฒนาเกมสามมิติแนวเดินยิงมุมมองบุคคลที่หนึ่ง
DEVELOPMENT OF 3D FIRST-PERSON SHOOTING
GAME

ชื่อนักศึกษา

นายชัชพล ฉัตรภูมิสุวรรณ 45050468
นายนฤเทพ วรงค์สิงหรา 45050485
นายสาธิต โชตินะพันธ์ 45050532

ภาควิชา

คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา

รศ.ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2548

คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ พศ.กฤษฎา บุศรา	
กรรมการ อ.สังกรศรัณย์ ต่องชูผล	
กรรมการและอาจารย์ที่ปรึกษา รศ.ธีรวัฒน์ ประกอบผล	

๙ >

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การพัฒนาเกมสามมิติเดินยิงมุมมองบุคคลที่หนึ่ง	
ชื่อนักศึกษา	นายชัชพล ฉัตรภูมิสุวรรณ	45050468
	นายณฤเทพ วรงค์สิงหรา	45050485
	นายสาริต โขตนะพันธ์	45050532
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2548	
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

บทคัดย่อ

ปัญหาพิเศษนี้เป็นการพัฒนาเกมสามมิติเดินยิงมุมมองบุคคลที่หนึ่ง ในลักษณะการต่อสู้ในสมรภูมิริบ เพื่อหาผู้ชนะ โดยเกมนี้เป็นเกมคอมพิวเตอร์ที่มีภาพแบบสามมิติ มีมุมมองในการเล่นเสมือนว่าผู้เล่นกำลังอยู่ในเกมจริงๆ เดินถืออาวุธ เพื่อโจมตีคู่ต่อสู้ ซึ่งผู้เล่นสามารถเลือกลักษณะของสมรภูมิริบ และลักษณะของตัวผู้เล่นได้ตามความต้องการ โดยรูปแบบการเล่นผู้เล่นสามารถเล่นได้ทั้งในส่วนของผู้เล่นคนเดียวและผู้เล่นหลายคนผ่านระบบเครือข่าย โดยเกมสามมิติเดินยิงมุมมองบุคคลที่หนึ่งนี้ พัฒนาขึ้นมาโดยใช้ Microsoft Visual C++ .NET ร่วมกับ เกมเอนจิน ซึ่งเกมมีภาพเป็นสามมิติสวยงาม มีความสมจริง และสามารถรองรับการเล่นทั้งแบบผู้เล่นคนเดียวและผู้เล่นหลายคนผ่านระบบเครือข่าย

Special Project Title	DEVELOPMENT OF 3D FIRST-PERSON SHOOTING GAME	
Students	Mr.Chutchapol Chatpoomsuwan	45050468
	Mr.Naruethep Varongsinghara	45050485
	Mr.Satit Choatnaphan	44050532
Degree	Bachelor of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Science	
Academic Year	2005	
Special Project Advisor	Assoc.Prof.Teerawat Prakobphon	

ABSTRACT

This special problem concerns with developing shooting game in the battlefield presented in 3D. This game has a realistic environment that players play in an actual battlefield by using their weapons to eliminate all of their enemies. Players are able to select map and character in different battlefields. In addition, this game supports multiplayer mode to increase player's motivation and entertainment. The software project is powered by Microsoft Visual C++ .NET including three engines : Irrlicht engine, RakNet engine and Audiere engine. It entertains players with realistic graphics and sound effects and supports multiplayer mode over the network.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาเกมสามมิติแนวเดินยิงมุมมองบุคคลที่หนึ่งสามารถสำเร็จ ลุล่วงได้ด้วยดี ด้วยความช่วยเหลือ และความร่วมมือจากหลายๆท่าน คณะผู้จัดทำต้องขอขอบพระคุณบุคคลที่มีส่วนช่วยให้การทำงานครั้งนี้เสร็จไปได้ด้วยดี คือ รศ.ธีรวัฒน์ ประกอบผล อาจารย์ที่ปรึกษาปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำแนวคิดต่างๆในการทำปัญหาพิเศษนี้ ดูแลเอาใจใส่ และให้การสนับสนุนทาง ด้านซอฟต์แวร์ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของโครงการปัญหาพิเศษฉบับนี้และทุกๆท่านที่ช่วยตอบกระทู้ทางอินเทอร์เน็ต

นอกจากนี้คณะผู้จัดทำต้องขอขอบบุคคลที่เกี่ยวข้องทุกฝ่าย ที่ทำให้การทำปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี รวมทั้งเพื่อนๆ และพี่ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2549

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ ของการศึกษา.....	1
1.3 ขอบเขตของการศึกษา.....	1
1.4 ประโยชน์ที่ได้รับจากการศึกษา.....	2
1.5 ขั้นตอนของการศึกษา.....	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	3
2.1 Microsoft Visual c++ 6.0.....	3
2.1.1 VC++ Developer Studio.....	3
2.1.2 VC++ Runtime Libraries.....	4
2.1.3 VC++ MFC และ Template Libraries.....	4
2.1.4 VC++ Build Tools.....	4
2.1.5 Active X.....	5
2.1.6 Data Access.....	5
2.1.7 Enterprise Tools.....	5
2.1.8 Graphics.....	5
2.1.9 Tools.....	6
2.2 การเขียนโปรแกรมด้วย Visual C++ บน Window.....	6
2.2.1 ฟังก์ชันและวิธีการสร้างฟังก์ชัน.....	7
2.2.2 การเขียนโปรแกรมเชิงวัตถุ.....	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.3 Direct X.....	19
2.3.1 ประวัติความเป็นมาของ DirectX.....	19
2.3.2 ความรู้เบื้องต้นเกี่ยวกับ DirectX.....	20
2.3.3 หลักการเบื้องต้นเกี่ยวกับ DirectX.....	21
2.3.4 API ของ DirectX.....	22
2.4 OpenGL.....	24
2.4.1 OpenGL Features.....	24
2.4.2 ระบบพิกัดของเบื้องต้นของ OpenGL	25
2.5 ปัญญาประดิษฐ์.....	30
2.5.1 Best-First Search.....	31
2.5.2 A* Algorithm.....	31
2.6 Irrlicht Engine.....	36
2.6.1 Special effects.....	38
2.6.2 Driver.....	39
2.6.3 Platforms.....	39
2.6.4 Materials and Shaders.....	40
2.6.5 Scene Management.....	40
2.6.6 Character Animation.....	41
2.6.7 Supported Formats.....	41
2.7 RakNet Engine.....	42
2.8 Audiere Engine.....	45
บทที่ 3 ขั้นตอนการดำเนินงานวิจัย.....	47
3.1 ขั้นตอนการออกแบบและวิเคราะห์เกม.....	47
3.1.1 การออกแบบรูปแบบ กติกาการเล่น และระบบเกม.....	47
3.1.1.1 การเลือกตัวละคร.....	48
3.1.1.2 การเลือกฉาก.....	48
3.1.2 การออกแบบภาพและกราฟิก.....	49
3.1.3 การออกแบบเสียงและซาวนด์เอฟเฟกต์ประกอบ.....	49
3.2 ขั้นตอนการลงมือปฏิบัติจริง.....	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

3.2.1	รูปแบบหน้าจอในสถานการณ์ต่างๆ.....	50
3.2.2	สร้างโมเดลที่ใช้ในเกมทั้งหมด.....	50
3.2.2.1	โมเดลตัวละครในเกม.....	51
3.2.2.2	โมเดลแผนที่.....	52
3.3	ขั้นตอนการเขียนโปรแกรม.....	53
3.3.1	คลาสหลักที่ใช้ในโปรแกรม.....	54
3.3.2	ฟังก์ชันการทำงานหลักในแต่ละ Engine.....	56
3.3.3	สถานะของเกม.....	58
3.3.4	การสืบทอดคุณสมบัติ.....	61
บทที่ 4	ผลการทดลองและการวิเคราะห์ปัญหา.....	65
4.1	ขั้นตอนการดำเนินการทดสอบการติดตั้งโปรแกรมและ คอมโพเนนต์ที่จำเป็นต้องใช้.....	66
4.2	ขั้นตอนการดำเนินการทดสอบการประมวลผลภายใต้ระบบที่กำหนด.....	66
4.3	ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกม.....	70
4.4	ขั้นตอนการทดสอบการส่งข้อมูล.....	70
4.5	ขั้นตอนการทดสอบการหาข้อผิดพลาดของโปรแกรม.....	72
4.5.1	ปัญหาเสียงขัดข้อง.....	72
4.5.2	ปัญหาโครงสร้างการทำงานของไคลเอนท์เซิร์ฟเวอร์มีความซ้ำซ้อน.....	72
4.5.3	ปัญหาความคมชัดของภาพ.....	73
4.5.4	ปัญหาระบบเครือข่ายล่าช้า.....	73
4.5.5	ปัญหาไม่สามารถกินเนื้อที่หน่วยความจำได้.....	73
4.5.6	ในส่วนการเล่นแบบผู้เล่นคนเดียว ศัตรูสามารถโจมตีผู้เล่นได้ แม้มีสิ่งกีด ขวางอยู่.....	73
4.6	ประเมินประสิทธิภาพของเกม.....	73
บทที่ 5	สรุปผลการดำเนินงานและข้อเสนอแนะ.....	74
5.1	สรุปผลการดำเนินงาน.....	74
5.1.1	การศึกษาและรวบรวมข้อมูล.....	74
5.1.2	การวิเคราะห์และการออกแบบเกม.....	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

5.1.3 การสร้างตัวละคร ภาพ และเสียงต่างๆ.....	74
5.1.4 การพัฒนาโปรแกรม.....	75
5.2 ข้อจำกัดของโปรแกรม.....	75
5.3 ข้อเสนอแนะ.....	76
ภาคผนวก ก คู่มือการติดตั้ง.....	77
ภาคผนวก ข คู่มือการเล่น.....	81
บรรณานุกรม.....	90



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 คลาสหลักที่ใช้ในโปรแกรม.....	54
3.2 ฟังก์ชันการทำงานหลักในแต่ละ Engine.....	56
4.1 ขั้นตอนการทดสอบการติดตั้งโปรแกรมและคอมโพเนนต์ที่จำเป็น.....	66
4.2 ขั้นตอนการรันโปรแกรมภายใต้เครื่องคอมพิวเตอร์ที่กำหนด.....	67
4.3 ขั้นตอนการเลือกฉากต่างๆ.....	67
4.4 ขั้นตอนการเลือกสถานะว่าจะเป็นเซิร์ฟเวอร์ หรือ โคล์เอนท์.....	68
4.5 ขั้นตอนการตัวละครต่างๆ.....	69
4.6 ขั้นตอนการเชื่อมต่อระหว่างโคล์เอนท์กับเซิร์ฟเวอร์.....	69
4.7 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของการเล่น.....	70
4.8 ขั้นตอนการทดสอบการควบคุมตัวละคร.....	70
4.9 ขั้นตอนการทดสอบการส่งข้อมูล.....	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงตัวอย่างของออปเจ็กต์.....	9
2.2 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 1.....	10
2.3 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 2.....	11
2.4 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 3	12
2.5 แสดงคุณสมบัติของแอททริบิวต์.....	13
2.6 แสดงแผนภาพการทำงานของ Direct X.....	22
2.7 ระบบพิกัดคาร์ทีเซียนสองมิติ.....	25
2.8 clipping region ในระนาบคาร์ทีเซียนสองมิติ.....	26
2.9 ตัวอย่างวิวพอร์ต (viewport).....	27
2.10 ระบบพิกัดสามมิติ (แกน z ซึ่ออกจากจอภาพเข้าหาตัวผู้มอง).....	28
2.11 Projection: 3D image --> 2D surface	28
2.12 Orthographic projection clipping volume	29
2.13 Perspective projection clipping volume.....	30
2.14 แสดงตำแหน่งของตัวละคร ศัตรู และ Waypoint ต่างๆ.....	32
2.15 แสดง Adjacent waypoint ที่ตัวละครสามารถเดินไปได้.....	32
2.16 เลือก Adjacent Waypoint ที่มีค่า Heuristic Function น้อยที่สุด.....	33
2.17 เดินไปยัง Waypoint นั้นๆ.....	33
2.18 เริ่มการคำนวณหา Waypoint ที่จะเดินครั้งใหม่.....	34
2.19 แสดง Adjacent waypoint ที่ตัวละครสามารถเดินไปได้.....	34
2.20 เลือก Adjacent Waypoint ที่มีค่า Heuristic Function น้อยที่สุด.....	35
2.21 เดินไปยัง Waypoint นั้นๆ.....	35
2.22 แสดงตัวอย่างตำแหน่งของเวย์พอยต์ (Waypoint).....	36
2.23 แสดงการทำงานแบบไคล์ออนท์เชิร์ฟเวอร์.....	43
3.1 แสดงหน้าจอเลือกตัวละคร.....	48
3.2 แสดงหน้าจอเลือกแผนที่.....	49
3.3 แสดงหน้าจอเมนูหลักของเกม.....	50
3.4 แสดงโมเดลครอยด์.....	51
3.5 แสดงโมเดลครอยด์เก้า.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 แสดงโมเดลอินเตอร์เตอร์.....	52
3.7 แสดงโมเดลแผนที่ที่ 1.....	52
3.8 แสดงโมเดลแผนที่ที่ 2.....	53
3.9 แสดงโมเดลแผนที่ที่ 3.....	53
3.10 โครงสร้างการทำงานของโปรแกรม.....	55
3.11 แสดง State Diagram ของเมนูหลัก.....	58
3.12 แสดง State Diagram ของระบบซิงเกิลเพลย์เซอร์ และมัลติเพลย์เซอร์.....	59
3.13 แสดง State Diagram ของ ServerState และ ClientState.....	60
3.14 แสดงการสืบทอดจากคลาส GameState.....	61
3.15 แสดงการสืบทอดจากคลาส GameEntity.....	62
3.16 แสดงการสืบทอดจากคลาส Graph.....	62
3.17 แสดงการสืบทอดจากคลาส IEventReceiver.....	63
3.18 แสดงการสืบทอดจากคลาส IGameWeaponState และ IAnimationEndCallBack.....	63
3.19 แสดงการสืบทอดจากคลาส IGameEnemyMultiState และ IAnimationCallBack.....	64
4.1 หน้าจอเริ่มต้น.....	66
4.2 หน้าจอเลือกฉาก.....	67
4.3 หน้าจอเลือกสถานะ.....	68
4.4 หน้าจอเลือกตัวละคร.....	68
4.5 หน้าจอใส่ชื่อและไอพี แอดเดรส.....	69
ก.1 แสดงหน้าแรกของการติดตั้ง.....	77
ก.2 แสดงหน้าการป้อนชื่อและบริษัทของผู้ใช้.....	78
ก.3 แสดงหน้าการระบุ Path ที่ต้องการติดตั้ง.....	78
ก.4 แสดงการเลือกที่ใส่โฟลเดอร์ลัด (Shortcut Folder).....	79
ก.5 แสดงรายละเอียดข้อมูลที่จะทำการติดตั้งทั้งหมด.....	79
ก.6 แสดงกระบวนการติดตั้งโปรแกรม.....	80
ก.7 แสดงเมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์.....	80

สารบัญญรูป (ต่อ)

รูปที่	หน้า
ข.1 แสดงหน้าจอเมนูหลักของเกม.....	81
ข.2 แสดงหน้าจอขณะเลือกแผนที่ในโหมดเล่นคนเดียว.....	82
ข.3 แสดงหน้าจอขณะเลือกแผนที่ในโหมดเล่นคนเดียว.....	82
ข.4 ความหมายของสิ่งต่างๆในหน้าจอขณะเล่นเกม.....	83
ข.5 แสดงหน้าจอเมื่อชนะ.....	84
ข.6 แสดงหน้าจอเมื่อแพ้.....	84
ข.7 แสดงหน้าเลือกตัวละครในโหมดมัลติเพลย์เยอร์ของผู้เล่นที่เป็นเซิร์ฟเวอร์.....	85
ข.8 แสดงหน้าจอการเลือกแผนที่ในโหมดมัลติเพลย์เยอร์.....	86
ข.9 แสดงหน้าจอการตั้งชื่อผู้เล่นที่เป็นเซิร์ฟเวอร์.....	86
ข.10 แสดงหน้าจอขณะเล่นเกมในโหมดมัลติเพลย์เยอร์.....	87
ข.11 แสดงหน้าจอการเลือกตัวละครในโหมดมัลติเพลย์เยอร์ของไคลเอนท์.....	87
ข.12 แสดงหน้าจอการระบุชื่อผู้เล่น และ IP Address ของเซิร์ฟเวอร์.....	88
ข.13 แสดงหน้าจอการเมื่อการเชื่อมต่อสำเร็จ.....	88
ข.14 แสดงหน้าจอขณะเล่นเกมเมื่อเป็นผู้เล่นที่เป็นไคลเอนท์.....	89
ข.15 แสดงหน้าจอปรับแต่งระดับความยาก และระดับเสียง.....	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันวงการเกมมีการพัฒนาไปอย่างมาก บางประเทศถึงกับมีองค์กรที่สนับสนุนวงการนี้อย่างเต็มตัว แต่ในขณะที่ประเทศไทยยังไม่มีองค์กรสนับสนุนอย่างเต็มที่เท่าใดนัก ทำให้ในปัจจุบันมีผู้ผลิตซอฟต์แวร์เกมภายในประเทศไม่มากนัก ส่งผลให้การพัฒนาบุคลากรและองค์ความรู้ทางเทคโนโลยีสาขานี้ในประเทศไทยไม่เจริญก้าวหน้าเท่าที่ควร จึงมีความตั้งใจที่จะเป็นส่วนหนึ่งที่จะช่วยกระตุ้นให้ผู้ที่เกี่ยวข้องและเยาวชนที่สนใจในด้านนี้เกิดแรงจูงใจที่จะพัฒนาอุตสาหกรรมเกมคอมพิวเตอร์ภายในประเทศไทยอย่างจริงจังมากขึ้น เพื่อที่จะทำให้ประเทศไทยสามารถผลิตเกมคอมพิวเตอร์ออกมาแข่งขันกับผลิตภัณฑ์ของต่างประเทศได้ และในอนาคตเราจะได้สามารถผลักดันเกมฝีมือคนไทยเข้าไปมีบทบาทในอุตสาหกรรมเกมระดับโลกได้อย่างเต็มภาคภูมิ

ประกอบกับได้ศึกษาพื้นฐานทางด้านทางคอมพิวเตอร์กราฟิกและภาษาคอมพิวเตอร์มาในระดับหนึ่ง จึงเกิดแนวคิดที่จะนำความรู้ที่ได้สั่งสมมานั้นมาประยุกต์ใช้เพื่อพัฒนาเกมคอมพิวเตอร์ขึ้นเอง

1.2 ความมุ่งหมายและวัตถุประสงค์ ของการศึกษา

- 1.2.1 เพื่อศึกษาทฤษฎีและหลักการเขียนโปรแกรมเกม
- 1.2.2 เพื่อส่งเสริมจินตนาการความคิดสร้างสรรค์ให้แก่ผู้เล่นและผู้พัฒนา
- 1.2.3 เพื่อให้เยาวชนไทยสนใจการพัฒนาโปรแกรมเกมคอมพิวเตอร์มากขึ้น
- 1.2.4 เพื่อศึกษาการนำคณิตศาสตร์มาใช้ในการพัฒนาเกม

1.3 ขอบเขตของการศึกษา

- 1.3.1 พัฒนาเกมคอมพิวเตอร์เบื้องต้นในลักษณะของเกมเชิงสนามรบอย่างง่าย
- 1.3.2 เป็นเกมสามมิติแนวเดินยิงมุมมองบุคคลที่หนึ่ง (3D First Person Shooting)
- 1.3.3 มีการแสดงคุณสมบัติต่างๆของอ็อบเจกต์ภายในเกม เช่น ตัวละคร แผนที่ เป็นต้น
- 1.3.4 มีดนตรีและเสียงประกอบในระหว่างเล่นซึ่งจะช่วยเพิ่มความน่าติดตามมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่ได้รับจากการศึกษา

- 1.4.1 เพิ่มพูนประสบการณ์และทักษะในการพัฒนาโปรแกรม
- 1.4.2 ได้เรียนรู้เทคโนโลยีในด้านการประมวลผลภาพสามมิติ และการควบคุมอุปกรณ์อินพุต เอาท์พุต (I/O) ที่เกี่ยวข้องได้แก่เมาส์และคีย์บอร์ด
- 1.4.3 เพื่อส่งเสริมการพัฒนาเกมคอมพิวเตอร์สำหรับคนไทย

1.5 ขั้นตอนของการศึกษา

- 1.5.1 ออกแบบเนื้อหาของเกม
- 1.5.2 ศึกษาเอนจิน (Engine) เพื่อนำมาใช้ในการพัฒนาโปรแกรมเกม
- 1.5.3 จัดการทรัพยากรต่างๆ ไม่ว่าจะเป็นภาพ เสียง ดนตรี เนื้อเรื่องและอื่นๆให้พร้อมเพื่อที่จะ นำมาใช้ในการเขียนโปรแกรม
- 1.5.4 พัฒนาโปรแกรมเกมตามที่ออกแบบไว้
- 1.5.5 พัฒนาโปรแกรมเกมเป็นเวอร์ชันทดลองก่อน เพื่อให้ผู้ทดสอบได้ลองเล่นเพื่อช่วยหาจุดบกพร่องของโปรแกรมเพื่อนำข้อบกพร่องนั้นกลับมาปรับปรุงแก้ไขเกมให้มีความสมบูรณ์มากยิ่งขึ้น
- 1.5.6 ทำเป็นเวอร์ชันสมบูรณ์
- 1.5.7 จัดทำส่วนของการติดตั้งโปรแกรม และบันทึกลงสื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 Microsoft Visual C++ 6.0

Microsoft Visual C++ 6.0 เป็นภาษาการโปรแกรมมิ่งเชิงวัตถุ (Object – Oriented Programming) แบบ GUI (Graphic User Interface) ตัวหนึ่งจากทางบริษัทไมโครซอฟต์ เป็นเครื่องมือพัฒนาโปรแกรมที่มีความสามารถสูงในยุคนี้ Microsoft Visual C++ ได้รับการพัฒนาให้มีความยืดหยุ่น และมีประสิทธิภาพสูงขึ้นมาจากภาษา C++ และได้รับการพัฒนาโปรแกรมในหลายๆ ด้านไม่ว่าจะเป็นการสร้างโปรแกรมทั่วไป , การสร้างโปรแกรมการจัดการฐานข้อมูล , การสร้างโปรแกรมบนระบบเครือข่าย หรือมัลติมีเดีย อย่างครบครัน

ในปัจจุบัน Microsoft Visual C++ ได้รับการพัฒนาจนถึงเวอร์ชันที่ 6 มีลักษณะเป็น IDE(Integrated Development Environment) คือเป็นโปรแกรมซึ่งมีไว้สำหรับเพิ่มความสะดวกในการสร้าง และแก้ไขโปรเจกต์ให้ง่ายขึ้น โดยจะมีเครื่องมืออำนวยความสะดวกในการพัฒนาโปรแกรมต่างๆ ให้เรียกใช้งานใน IDE เช่น โปรแกรมที่ใช้ในการดีบั๊ก, คอมไพเลอร์ และลิงค์เกอร์ เป็นต้น นอกจากนี้ยังมีส่วนรองรับการพัฒนาโปรแกรมบนวินโดวส์ โดยมี MFC (Microsoft Foundation Class) เป็นไลบรารีที่ช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมบนวินโดวส์

นอกจากนี้ Microsoft Visual C++ ยังมีส่วนที่สนับสนุนทางด้านของมัลติมีเดีย เกี่ยวกับรูปภาพ , การเล่นไฟล์ภาพเคลื่อนไหวรวมถึงทางด้านของเสียง (Sound) ทั้งยังมีส่วนของเครื่องมือที่สนับสนุนการใช้งานที่เกี่ยวข้องกับบัพเฟอร์อินพุต เอาท์พุต และริฟฟ์ไฟล์ (Riff files) ที่เรียกกันว่า มัลติมีเดียไฟล์อินพุตเอาท์พุต เซอร์วิส (Multimedia file I/O services) อีกด้วย โดยมีความสามารถ และเครื่องมือต่างๆ ให้ใช้งานมากมายคอมพิวเตอร์ ซึ่งใน Visual C++ นั้นประกอบไปด้วยคอมพิวเตอร์ต่างๆดังนี้

2.1.1 VC++ Developer Studio

ทำหน้าที่เป็นศูนย์กลางในการสร้างแอปพลิเคชัน ซึ่งจะประกอบไปด้วยองค์ประกอบย่อยๆ ภายในดังนี้

Project Manager ทำหน้าที่จัดการกับการสร้างแอปพลิเคชันในลักษณะของโปรเจกต์ (แอปพลิเคชันหนึ่งๆ ประกอบด้วยองค์ประกอบหลายๆ ส่วนซึ่งรวมเรียกว่าโปรเจกต์)

Text Editor ทำหน้าที่ในการเขียนโปรแกรม โดยจะเรียกโปรแกรมที่เขียนว่า Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Resource Editor	ทำหน้าที่ออกแบบหน้าต่างโปรแกรม เช่น เมนู ไอคอนต่าง ๆ และไอคอน
Wizard	เพื่อสร้างแอปพลิเคชันในลักษณะวิซาร์ดต่างๆเช่น AppWizard และ ClassWizard จะช่วยให้สร้างโค้ดพื้นฐานแก แอปพลิเคชันอย่างรวดเร็ว โดยเพียงแค่กำหนดคลาส C++ จัดการกับวินโดว์เมสเสจ และกระทำงานอื่นๆเพิ่มเติมอีก
Compiler	จะทำการคอมไพล์อย่างอัตโนมัติ มีการลิงค์ (Link) ไฟล์ต่างๆ เข้าด้วยกัน
Debugger	เพื่อแก้ไขความผิดพลาดของแอปพลิเคชัน โดยผ่าน Debugger
Online Help	รายละเอียดความช่วยเหลือ (กรณีติดตั้ง MSDN แล้ว)

2.1.2 VC++ Runtime Libraries

คอมไพเนนต์นี้ทำหน้าที่เก็บฟังก์ชันมาตรฐานต่างๆของ ANSI C เช่น ฟังก์ชัน sin ซึ่งสามารถเรียกใช้ภายใน โปรแกรม C หรือ C++

2.1.3 VC++ MFC และ Template Libraries

MFC เป็นไลบรารีคลาส C++ ที่ถูกสร้างมาโดยเฉพาะสำหรับการสร้างแอปพลิเคชันเพื่อใช้งานกับ Windows ซึ่งมีรูปแบบการติดต่อกับผู้ใช้แบบกราฟฟิก หรือที่เรียกว่า GUI (Graphic User Interface) ซึ่งคลาส MFC ทำให้การเขียนโปรแกรมง่ายขึ้น และยังช่วยประหยัดเวลาในการเขียนโค้ด

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

2.1.4 VC++ Build Tools

ประกอบด้วยคอมไพเลอร์ C/C++ , Linker , คอมไพเลอร์ Resource (สำหรับเตรียมการเกี่ยวกับ Resource โปรแกรมต่างๆเช่น เมนู ไอคอนต่าง ๆ และไอคอน เป็นต้น) รวมทั้งเครื่องมืออื่นๆ ที่จำเป็นต่อการสร้างแอปพลิเคชันสำหรับวินโดว์ (การสร้างแอปพลิเคชันแบบ Win32)

2.1.5 Active X

เป็นซอฟต์แวร์ย่อยๆ (Software Component) หรือองค์ประกอบย่อยๆ ที่เพิ่มเข้าไปในแอปพลิเคชันที่สร้างขึ้น ซึ่ง ActiveX นั้นทำให้ไม่จำเป็นต้องสร้างทุกส่วนของแอปพลิเคชันเอง เพียงแต่เลือกใช้อ้องค์ประกอบย่อยๆ ที่เหมาะสมกับงานเพื่อสร้างเป็นแอปพลิเคชันที่สมบูรณ์

2.1.6 Data Access

เป็นคอมโพเนนต์ที่รวมไครเวอร์ฐานข้อมูลชนิดต่างๆ ไว้ให้สามารถสร้างแอปพลิเคชันที่เชื่อมต่อกับฐานข้อมูลได้อย่างสะดวก และนอกจากไครเวอร์แล้วยังประกอบไปด้วยคอนโทรลและเครื่องมืออื่นๆ ที่ช่วยในการสร้างแอปพลิเคชันกับฐานข้อมูลทำได้อย่างรวดเร็ว

2.1.7 Enterprise Tools

เป็นคอมโพเนนต์ที่ประกอบด้วยคอมโพเนนต์ย่อยๆ 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

เป็นคอมโพเนนต์เกี่ยวกับรูปภาพในรูปแบบต่างๆ เช่น metafile, bitmap, cursor และ icon รวมทั้ง video clip ซึ่งคอมโพเนนต์เหล่านี้ใช้ในการเขียนโปรแกรมเกี่ยวกับกราฟฟิกต่างๆ

2.1.9 Tools

เป็นคอม โพนেন্টที่เป็นเครื่องมือเสริมการทำงานชนิดต่างๆของ Visual C++ ดังนี้

- API Text Viewer
- MS Info
- MFC Trance Utility
- 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 และได้นำไปใช้ในการพัฒนาโปรแกรมต่างๆมากมาย ไม่ว่าจะเป็นตัวระบบปฏิบัติการหรือ โปรแกรมประยุกต์ ทั้งบนวิน โดวส์และบนระบบปฏิบัติการอื่นๆ เช่น ลินุกซ์ (Linux) และก็ยังมีการพัฒนาตัวแปลภาษาให้มีความสามารถสูงขึ้นๆ ตามความสามารถของระบบปฏิบัติการ เช่น Visual C++ ,Visual Studio, Visual Studio .NET เป็นต้น แต่อย่างไรก็ตามพื้นฐานของภาษา C และการเขียนโปรแกรมแบบการเขียนโปรแกรมเชิงวัตถุ ใน C++ ก็ยังเป็นรากฐานในการพัฒนาภาษาใหม่ๆ ขึ้นมาอีกหลายภาษา เช่น จาวา(Java) , ซีชาร์ป (C#) เป็นต้น

การที่เราจะพัฒนาโปรแกรมภาษา C++ ได้นั้น จะต้องมีการมีโปรแกรมสำหรับแปลภาษาก่อน นั่นก็คือ Turbo C++ ,Borland C++ ,Microsoft C/C++ ,Visual C++ ฯลฯ โปรแกรมแปลภาษาเหล่านี้ สามารถแปลโปรแกรมได้ทั้งภาษา C และภาษา C++ ซึ่งก็ขึ้นอยู่กับว่าเรากำหนดให้นามสกุลของไฟล์เป็นภาษาอะไร ถ้าเขียนโปรแกรมภาษา C ให้กำหนดนามสกุลของไฟล์โปรแกรมเป็น .C แต่ถ้าเขียนโปรแกรมที่มีการประกาศคลาสหรือใช้ความสามารถของการเขียนโปรแกรมเชิงวัตถุด้วย ให้กำหนดนามสกุลของไฟล์โปรแกรมนั้นให้เป็น .CPP

2.2.1 ฟังก์ชันและวิธีการสร้างฟังก์ชัน

- การประกาศฟังก์ชัน

ฟังก์ชันต้นแบบ หรือ Prototype Function นี้มีไว้เพื่อบอกให้ตัวแปลภาษาทราบว่า ในโปรแกรมนี้ยังมีฟังก์ชันชื่อว่า call() อยู่ แต่อยู่ด้านล่างของ main() ดังนั้น เราเขียนโปรแกรมให้เป็นดังนี้

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

หรืออีกวิธีหนึ่งก็คือ ให้ประกาศฟังก์ชัน call() เอาไว้ด้านบนก่อน main() ดังนี้

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

- การคืนค่าฟังก์ชัน

เราสามารถกำหนดให้ฟังก์ชันส่งค่ากลับมายังจุดที่เรียกใช้คำสั่ง return แต่ก่อนที่จะใช้คำสั่ง return นั้นต้องตรวจสอบว่า จะส่งค่าประเภทใดกลับไป ถ้าเป็นค่าจำนวนเต็ม เราจะต้องเปลี่ยนคำว่า void ที่อยู่หน้าฟังก์ชันให้เป็น int ดังตัวอย่างต่อไปนี้

```

int call();

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

int call()
    return 5+6
}

```

จากตัวอย่าง จะเห็นได้ว่าฟังก์ชันแบบ int และคืนค่าให้ 5+6 (ซึ่งก็คือ 11) กลับไปยังจุดที่เรียกใช้ ดังนั้น ถ้าเราไม่เปลี่ยนจาก void ให้มาเป็น int เราก็จะไม่สามารถส่งค่ากลับไปได้

- การส่งค่าไปให้ฟังก์ชัน

เราสามารถส่งค่าให้กับฟังก์ชันได้ เพื่อให้ฟังก์ชันนำเอาค่าที่ส่งไปนี้ไปประมวลผล เราเรียกค่าที่ส่งนี้ว่า “พารามิเตอร์” หรือจะเรียกว่า “อาร์กิวเมนต์” 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;
}

```

จากตัวอย่างจะเห็นได้ว่าฟังก์ชัน call รับพารามิเตอร์ 2 ตัวคือ a กับ b เราเรียกโดยผ่านฟังก์ชัน printf ส่งค่า 10 และ 20 ไปให้กับ call() และมันก็จะคืนค่ากลับมาโดยเอาเลขทั้งสองบวกกัน เพราะฉะนั้น ฟังก์ชัน call() จึงคืนค่ากลับมาเป็น 30 แสดงออกทางจอภาพ โดย printf

2.2.2 การเขียนโปรแกรมเชิงวัตถุ

ธรรมชาติของวัตถุ และทฤษฎีของการเขียนโปรแกรมเชิงวัตถุ

นิยามในเรื่องของการเขียนโปรแกรมเชิงวัตถุ คือ “การมองทุกอย่างให้เป็นวัตถุ” ซึ่ง “วัตถุ” แต่ละอย่างต่างก็มีชื่อเรียกที่แตกต่างกัน เช่น

- โทรศัพท์ จะขี้อะไร แบบไหน เราเรียกวัตถุนี้ว่า โทรศัพท์
- แก้วน้ำ จะเป็นแก้วกระเบื้องหรือแก้วพลาสติก วัตถุนี้ก็คือแก้ว
- กระดาษ จะเป็นเศษกระดาษ หรือ A4 วัตถุนี้ก็คือ กระดาษ
- คน ไม่ว่าจะป็นอาชีพใด สูงต่ำดำขาว ก็คือคนด้วยกันทั้งสิ้น
- รถ ไม่ว่าจะขี้อะไร แบบใด ประเภทใด ก็เรียกวารถ เช่น รถยนต์ รถบรรทุก ฯลฯ

ดังนั้นสามารถสรุปได้ว่า ในโลกนี้ ทุกๆอย่างล้วนแล้วแต่มี “ชื่อที่บอกว่าเป็นสิ่งนั้นคืออะไร หรือเป็นอะไร” ชื่อที่เราสมมติขึ้นมาเพื่อให้เรียกวัตถุต่าง ๆ นั้นก็คือ “คลาส” (Class) นั่นเอง ถ้าเรามองสิ่งของรอบ ๆ ตัว ก็มีหลายคลาส เช่น

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

สาเหตุที่มีคลาส หรือมีชื่อที่เรียกวัตถุตามลักษณะของวัตถุนั้นเพราะว่า ถ้าไม่มีชื่อที่เรียก ในเวลาเราจะอ้างถึงสิ่งๆนั้นก็จะทำได้ยาก พิจารณารูปต่อไปนี้



รูปที่ 2.1 แสดงตัวอย่างของออปเจกต์

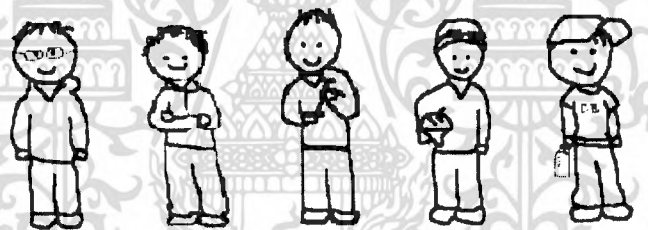
จากรูป จะเห็นว่าเป็นวัตถุที่รูปร่างยาวๆ เรียวๆ มีไส้เป็นถ่าน ปลายด้านหนึ่งจะแหลมๆ ใช้เขียนบนวัสดุรองรับ เช่น กระดาษได้ วัตถุชิ้นนี้เรียกว่า “ดินสอ” และในคลาสของวัตถุใดๆ จะมีอยู่ 2 องค์ประกอบที่ใช้บอกถึงความเป็นวัตถุนั้น คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คุณลักษณะ (Attribute หรือ Properties) ก็คือ ลักษณะที่บอกถึงความเป็นสิ่งๆนั้น
2. การกระทำ (Method หรือ Functions) ก็คือมันทำอะไรได้บ้าง

ยกตัวอย่างเช่น ดินสอ คุณลักษณะของมันก็คือ เรียว ยาว มีไส้เป็นถ่าน การกระทำของดินสอก็คือ เขียน หัก ปา จิ้ม แหวง พูดย่างๆก็คือ ดินสอทำอะไรได้ และเราก็ทำอะไรกับมันได้บ้าง เราสามารถใช้มันทำอะไรได้ตั้งแต่เขียน ทึ่มแหวง ไปจนถึงใช้ทักหู ก็ขึ้นอยู่กับการนำไปใช้งานนั่นเอง เพราะฉะนั้น ถ้าเกิดเราไปพบวัตถุใดๆ ที่มีคุณลักษณะตรงกับคุณลักษณะของดินสอ และยังมี การกระทำที่เหมือนกัน เราจะสรุปได้เลยว่า วัตถุสิ่งนั้นก็คือ “ดินสอ” นั่นเอง

ในการเขียนโปรแกรมคอมพิวเตอร์ก็เช่นกัน เราจะต้องมองสิ่งที่อยู่โปรแกรมของเราให้เป็นเชิงวัตถุเพื่อนที่จะได้ออกแบบวิธีการเก็บและจัดการข้อมูลที่เป็นอิสระต่อกัน แต่ก่อนที่จะเข้าสู่การเขียนโปรแกรมเชิงวัตถุ ในภาษา C+ นั้น คล้ายกับตัวแปรแบบโครงสร้าง (Structure) ในภาษา C การประกาศตัวแปรแบบโครงสร้างนี้ เป็นแนวคิดพื้นฐานที่จะนำมาใช้ในการสร้างคลาสต่อไป ลักษณะการทำงานของตัวแปรแบบโครงสร้างจะช่วยแก้ไขปัญหาในเรื่องของความซับซ้อนของตัวแปรได้ เพราะเป็นการเปลี่ยนวิธีการเขียนโปรแกรม โดยอาศัยชุดของข้อมูลเป็นหลัก จากรูปเป็นตัวอย่างปัญหาในการเก็บข้อมูลของคน 5 คน ดังรูป



สมศักดิ์	มงคล	บันลือ	สมคิด	สาธิต
อายุ : 25	อายุ : 30	อายุ : 36	อายุ : 31	อายุ : 27
สูง : 170	สูง : 165	สูง : 175	สูง : 168	สูง : 172
หนัก : 68	หนัก : 70	หนัก : 65	หนัก : 70	หนัก : 69

รูปที่ 2.2 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 1

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราจะนำเอาข้อมูลมาใส่ลงในตัวแปร age ,height และ weight นั้นเราก็จะอาศัยลำดับในอะเรย์เข้ามาช่วย นั่นคือ คนที่ 1 ก็จะเก็บลงไป ในอะเรย์ช่องที่ 0 ในทุกตัวแปร ดังนี้

```
age[0] = 25;
```

```
height[0] = 170;
```

```
weight[0] = 68;
```

สำหรับคนที่ 2 ก็จะเก็บลงไป ในอะเรย์ช่องที่ 1 ของทุกตัวแปร เช่นกัน ดังนี้

```
age[1] = 30;
```

```
height[1] = 165;
```

```
weight[1] = 70;
```

สำหรับคนที่ 3 ,4 และ 5 ก็จะเก็บลงไป ในอะเรย์ช่องที่ 2,3 และ 4 ของทุกตัวแปร ตามลำดับ เช่นกัน จากตัวอย่างนี้ เราจะสรุปได้ว่า การใช้อะเรย์ในการเก็บข้อมูลที่มีจำนวนมาก และมีตัวแปรหลายๆตัวค่อนข้างจะยุ่งยากพอสมควร เพราะเราใช้ลำดับของอะเรย์เป็นตัวกำหนดลำดับการเก็บข้อมูล ถ้าเราต้องการเรียกดูข้อมูลของคนี่ 1 เราก็จะต้องแสดงข้อมูลที่อยู่ในช่องที่ 0 ของอะเรย์ทุกตัวออกมา รูปแบบของการเก็บข้อมูลในลักษณะนี้ เป็นดังรูป

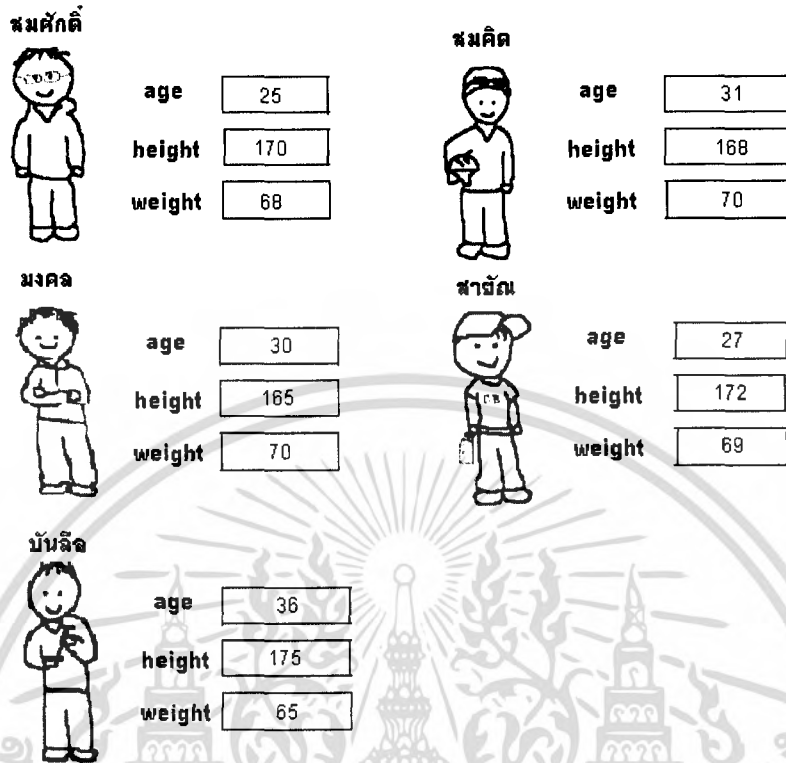


	สมศักดิ์	มงคล	บันลือ	สมคิด	สาธิต
age	25	30	36	31	27
height	170	165	175	168	172
weight	68	70	65	70	69

รูปที่ 2.3 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 2

จะเห็นว่า อะเรย์ 1 ตัวที่เก็บอายุ ก็จะเก็บอายุทุกๆช่อง ตัวแปรที่เก็บส่วนสูง ก็จะเก็บแต่ส่วนสูงทั้งหมด วิธีการลักษณะนี้เสี่ยงต่อการผิดพลาดอย่างมากในเรื่องของลำดับ เพราะตัวแปรแต่ละตัวมีความอิสระต่อกัน การอ้างลำดับผิดในกรณีที่มีปริมาณข้อมูลมากๆ เช่นเก็บข้อมูลเป็นร้อยๆ ข้อมูลขึ้นไป การไล่ลำดับจะต้องไล่ให้ตรงกัน ซึ่งจากปัญหาที่ได้กล่าวไปนี้ เราจะต้องใช้ตัวแปรแบบ โครงสร้างมาช่วยแก้ปัญหา การใช้งานตัวแปร โครงสร้างนี้ จะคล้ายๆกับการที่เราสร้างตัวแปร

ชนิดใหม่ขึ้นมาเป็นของเราเอง โดยพิจารณาจากข้อมูลที่เราต้องการจัดเก็บ รูปแบบของการเก็บข้อมูลแสดงได้ดังรูป



รูปที่ 2.4 แสดงตัวอย่างของแอททริบิวต์ รูปที่ 3

การประกาศตัวแปรโครงสร้าง เราจะใช้คำสั่ง typedef struct ดังนี้

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

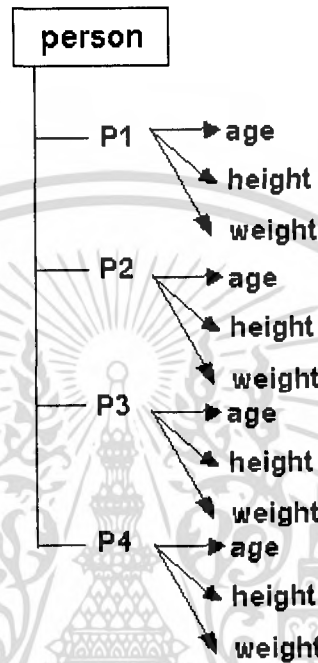
จากข้างต้น เป็นการสร้างตัวแปรแบบโครงสร้างชนิดใหม่ขึ้นมา ชื่อว่า person เราจะถือว่าเป็นการสร้างตัวแปรแบบใหม่ และตัวแปรนี้สามารถเก็บค่าได้ 3 ค่า คือ อายุ (age) , น้ำหนัก (weight) และส่วนสูง (height) เมื่อเราประกาศตัวแปรแบบใหม่นี้เอาไว้ในโปรแกรมของเราแล้ว เราก็สามารถสร้างตัวแปรแบบ person นี้ได้ทันที โดยเขียนโค้ดดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้เราได้ตัวแปร p1,p2,p3 และ p4 เป็นตัวแปรแบบ person แล้ว

สำหรับตัวอย่างนี้ก็เช่นกัน คือ เมื่อเราได้สร้างตัวแปรโครงสร้างชนิดใหม่ขึ้นมาแล้ว โดยชื่อว่า person เราก็สามารถที่จะสร้างตัวแปรแบบ person ขึ้นมาได้ และตัวแปรที่ถูกสร้างขึ้นมาจะ มีคุณลักษณะความเป็นตัวแปรแบบ person ทุกอย่าง คือ ภายในตัวมันเอง สามารถเก็บค่าย่อยๆ ได้ อีก 5 ค่าดังรูป



รูปที่ 2.5 แสดงคุณสมบัติของแอททริบิวต์

ลักษณะของตัวแปรแบบโครงสร้างนี้จะจัดเก็บข้อมูลเป็นหมวดหมู่ และแบ่งแยกชัดเจนมากขึ้นอย่างเห็นได้ชัดว่า age ,height และ weight ก็เป็นของคนๆนั้นเลย เพราะฉะนั้นมันจะไม่ปะปนกันคือ age ของ p1 จะเป็นของ p1 โดยเฉพาะ อาจจะไม่เท่ากับ age ของ p2 ก็ได้ เพราะฉะนั้นด้วยหลักการของโครงสร้างนี้ เมื่อนำมาใช้ในการเก็บข้อมูลอายุ , น้ำหนัก และส่วนสูงของคน 5 คน ดังที่ได้ยกตัวอย่างไว้ในข้างต้น 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;

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 = 31;
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 เป็นข้อมูลเฉพาะส่วนๆ ไป เราสามารถเข้าถึงข้อมูลที่เป็นอายุของคนที่ต้องการได้ โดยอ้างอิงลำดับของคนๆนั้นและตามด้วยชื่อตัวแปรที่เราต้องการ เช่น ต้องการอายุของนายมงคล ซึ่งลำดับที่สอง ก็คือ p[1].age เป็นต้น จะเห็นได้ว่า วิธีการนี้ เป็นการเปลี่ยนแนวคิดและมุมมองจากอะเรย์ให้มาเป็นแนวคิดที่คล้ายๆกับเชิงวัตถุเลย คือ เรามองแต่ละคน เป็นวัตถุมากขึ้น 1 คน ก็จะมีอายุ น้ำหนัก และส่วนสูง หรือมีข้อมูลมากกว่านี้ใน 1 โครงสร้าง ทำให้การจัดการข้อมูลทำได้ง่ายกว่าการใช้อะเรย์แบบในวิธีแรกที่ได้นำเสนอไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• การสร้างคลาส

การสร้างคลาสใน C++ เราจะเขียนคล้ายๆ กับการประกาศตัวแปรแบบโครงสร้าง ดังที่ได้กล่าวไปแล้ว คลาสจะเป็นโครงสร้างที่รวมการทำงานทั้งหมดเข้าด้วยกัน ให้เป็นวัตถุก้อนเดียวกัน ดูตัวอย่างโปรแกรม EASY.CPP ต่อไปนี้

```
#include

class Person{
    int age;
    void setAge(int a){
        age = a;
    }
    void show Age(){
        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()

คลาส Person ที่ยกตัวอย่างมานี้ แสดงให้เห็นถึงการจัดเก็บข้อมูล ก็คือ อายุ และมีฟังก์ชัน setAge ที่ใช้ในการกำหนดค่าอายุ โดยรับพารามิเตอร์จากภายนอก และมีฟังก์ชัน showAge ที่ใช้ในการแสดงค่าอายุออกมาทางจอภาพโดยฟังก์ชัน printf และในฟังก์ชัน main() ก็ได้มีการประกาศออบเจกต์ของคลาส Person นี้ชื่อว่า p เอาไว้ จากนั้นก็จบโปรแกรม โปรแกรมนี้เป็นตัวอย่างที่แสดงให้เห็นถึงการประกาศคลาส ซึ่งมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ในคลาส หนึ่งๆจะมีแค่การสร้างตัวแปรหรือฟังก์ชันเพียงอย่างเดียวก็ได้ หรือเขียนแค่คลาสเพียงอย่างเดียว มีแต่ {...} ไม่ต้องมีตัวแปรหรือฟังก์ชัน ไว้เลยก็ได้

จากโปรแกรม EASY.CPP ในข้างต้นนั้น เราจะมาทำความเข้าใจกันดังนี้ ในคลาส Person มีตัวแปร age ตัวหนึ่งและมีฟังก์ชัน setAge ฟังก์ชันนี้รับค่าพารามิเตอร์ 1 ตัวคือ int a และเมื่อรับเข้ามาแล้ว ก็จะทำการกำหนดให้กับตัวแปร age ที่อยู่ภายในคลาส และอีกฟังก์ชันหนึ่งก็คือฟังก์ชัน showAge ซึ่งเป็นฟังก์ชันที่ใช้ในการแสดงค่า age ออกทางจอภาพ จะเห็นได้ว่าการทำงานจะอยู่ภายในคลาสนั้นๆ จบแค่ภายในคลาสนั้นๆ เมื่อเราประกาศอ็อบเจกต์ของคลาสดังนี้

```
Person p;
```

ก็เท่ากับว่า ตอนนี้อยู่ในคลาส Person มีหายากแล้ว นั่นคือ อ็อบเจกต์ p นั้นเอง และ p นี้ก็มีคุณลักษณะทุกอย่างเช่นเดียวกับคลาส Person นี้เลย นั่นก็คือ p สามารถเรียกใช้ตัวแปร a ,ฟังก์ชัน setAge และ showAge ได้ ให้เราแก้ไข โปรแกรม 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 ออกมาปัญหานั้นก็คือเมื่อเราแปลโปรแกรมนี้โดยใช้ตัวแปรภาษา C++ ใดๆ

จะพบกับข้อความผิดพลาดลักษณะนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
\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 คลาสจะมีการปกป้องข้อมูลไม่ให้กระบวนการภายนอกคลาสใดๆ เข้าถึงสมาชิกในคลาสได้เลย เมื่อเราประกาศออบเจกต์ p ซึ่งออบเจกต์นี้เป็นตัวแปรที่อยู่ภายนอกคลาส ดังนั้น p จึงไม่สามารถเข้าถึงตัวแปร age และฟังก์ชัน setAge กับ showAge ได้ คุณสมบัติดังกล่าวเป็นคุณสมบัติของการเขียนโปรแกรมเชิงวัตถุที่ช่วยให้สมาชิกภายในคลาสไม่ถูกรบกวน ทำให้คลาสมีคุณสมบัติปกป้องข้อมูล หรือ Encapsulation ได้นั่นเอง ซึ่งเป็นสิ่งที่แตกต่างออกไปจากตัวแปรแบบโครงสร้างที่เห็นได้ชัด ถ้าเราจะทำให้ออบเจกต์ p สามารถเรียกใช้สมาชิกในคลาสได้ ผู้เขียนโปรแกรมจะต้องอนุญาตให้สมาชิกในคลาสตัวที่ต้องการมีคุณสมบัติเป็น “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();
}

```

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

จากตัวอย่าง EASY.CPP ในข้างต้น จะเห็นได้ว่า ไม่สามารถเข้าถึงออบเจกต์ภายในคลาสได้ทันที ออบเจกต์จะเรียกใช้ฟังก์ชันหรือตัวแปรในคลาสได้จะต้องกำหนดให้เป็นแบบ public ซึ่งตรงจุดนี้เป็นความสามารถของการเขียนโปรแกรมเชิงวัตถุที่จะช่วยให้เกิดความปลอดภัยของข้อมูลในคลาสมากขึ้น ระดับในการเข้าถึงข้อมูลในคลาสมี 3 ระดับ คือ

- ส่วนตัว หรือ Private
- สาธารณะ หรือ Public
- ปกป้อง หรือ Protect

ถ้าเราสร้างคลาสขึ้นมา โดยไม่ได้ระบุว่าเป็น private, public หรือ protect ใดๆ สมาชิกภายในคลาสนี้จะเป็นแบบ private ทั้งหมด

พิจารณาคลัสต่อไปนี้

```

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

```

สมาชิกทุกตัวในคลาสนี้เป็นแบบ private หมด สมาชิกแบบ private นั้น จะอนุญาตให้กระบวนการที่ดำเนินการอยู่ภายในคลาสนี้ด้วยกันเรียกใช้งานกันได้ ในคลาสนี้ Person จะเห็นว่าไม่มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน setAge และ showAge ซึ่งต่างก็เรียกใช้ตัวแปร age ได้ พุดง่ายๆ ก็คือ ตัวแปรและฟังก์ชันนั้นสามารถเรียกใช้กันภายในคลาสได้อย่างปกติทุกอย่าง แต่ถ้าเกิดเราสร้างออบเจกต์ของคลาสขึ้นมา และเรียกใช้ตัวแปรที่อยู่ในคลาส

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

ไม่สามารถเรียกใช้แบบนี้ได้ ทั้งนี้เป็นเพราะว่าออบเจกต์ p เป็นตัวแปรที่ถูกสร้างขึ้นมาให้มู่อยู่ภายนอกคลาส 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 Direct X

2.3.1 ประวัติความเป็นมาของ DirectX

ในช่วงที่ระบบปฏิบัติการไมโครซอฟต์วินโดวส์เริ่มต้นขึ้นมานั้น สิ่งที่เป็นเอกลักษณ์ของวินโดวส์ ก็คือเรื่องของ GUI (Graphical User Interface) โดยมี GDI (Graphical Device Interface) เป็นเครื่องมือสำหรับการจัดการทางด้านภาพ และเกมทั่วไปก็ยังคงพัฒนาภายใต้ระบบปฏิบัติการเดิมอยู่ ซึ่งก็คือระบบปฏิบัติการ "DOS" นั่นเอง

การเริ่มต้นครั้งแรกสำหรับไมโครซอฟต์ คือ "เครื่องมือ Win-G" สำหรับ Window95 และดูเหมือนว่าเครื่องมือตัวนี้จะไม่ได้รับการตอบรับที่คึกคักจากผู้พัฒนาเกมทั่วโลก ยังผลให้ไมโครซอฟต์เองต้องปรับปรุงรูปแบบใหม่ให้กับเครื่องมือตัวนี้อีกครั้ง

ไมโครซอฟต์ได้ทำการจัดตั้งทีมงานขึ้นมาใหม่เพื่องานกราฟิก งานมัลติมีเดีย งานเน็ตเวิร์ก งานด้านการรับข้อมูล Input และ งานกราฟิก 3 มิติ (ชื่อเทคโนโลยี “RenderGraphic) โดยใช้ชื่อว่า “DirectX”

DirectX 2.0 สามารถเข้ามาจัดการงานด้านเกมได้เป็นอย่างดีการทำงานเร็วขึ้นกว่าระบบปฏิบัติการ DOS ก็เลยทำให้โปรแกรมเมอร์ทั่วโลกได้เริ่มหันมามองเทคโนโลยี ตัวนี้ใหม่อีกครั้ง 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 และยังมีการพัฒนาเวอร์ชันใหม่ๆ ต่อไปจากเทคโนโลยีนี้เองเป็นผลให้เกมบน ระบบปฏิบัติการ DOS ค่อยๆ เลือนหายไปเป็นที่สุด

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

2.3.2 ความรู้เบื้องต้นเกี่ยวกับ DirectX

DirectX เป็น API ของไมโครซอฟท์พัฒนาเพื่อใช้จัดเตรียมอินเตอร์เฟส สำหรับควบคุมฮาร์ดแวร์มัลติมีเดียบนระบบ Microsoft Windows ได้อย่างมีประสิทธิภาพ และเป็นเครื่องมือให้โปรแกรมเมอร์ทำงานกับคำสั่ง และ โครงสร้างข้อมูลในระดับใกล้ฮาร์ดแวร์ โดยไม่ต้องสร้างโค้ดติดต่อในระดับล่างซึ่งวิธี ติดต่อกันจะแตกต่างกันไปตามประเภทของอุปกรณ์ การเขียนโค้ดที่เป็นอิสระจากอุปกรณ์ในลักษณะนี้ ช่วยให้โปรแกรมเมอร์สามารถสร้างซอฟต์แวร์เพื่อทำงานดังกล่าวได้อย่างดี แม้ผู้ใช้จะปรับเปลี่ยนอุปกรณ์ตัวใหม่ และเพิ่มการ์ดเร่งความเร็วแบบสามมิติ เสียง อุปกรณ์อินพุต และ อื่นๆก็ตาม

DirectX ได้รับการออกแบบให้นักพัฒนามีสภาพแวดล้อมคล้ายคลึงกับสภาพแวดล้อมที่มีประสิทธิภาพของ MS-DOS ซึ่งทำงานได้เร็วกว่าโค้ดที่ทำงานบนวินโดวส์ เนื่องจากไม่ต้องสูญเสียประสิทธิภาพจาก API สำหรับจัดการงานมัลติมีเดียของ วินโดวส์รุ่นก่อน แต่อย่างไรก็ตาม การสนับสนุนความสามารถในการทำงานของฮาร์ดแวร์ที่มีอยู่บนระบบ โค้ดที่เขียนขึ้นด้วย DirectX สามารถรันได้เร็วกว่าแอปพลิเคชันของ MS-DOS

เมื่อใดก็ตามที่สร้างอ็อบเจกต์ DirectX ให้กับดีไวซ์นั้น DirectX จะเข้าไปซักถามฮาร์ดแวร์ผ่าน HAI เพื่อดึงเอาข้อมูลเกี่ยวกับดีไวซ์ออกมาที่มีอยู่ในตาราง Cap Bits (Capability Bits) ข้อมูลที่มีอยู่ใน Cap Bits เป็นข้อมูลที่ใช้ออกความสามารถที่ฮาร์ดแวร์สามารถทำได้หรือความสามารถใดที่ HEL ต้องจำลอง

ไมโครซอฟต์แวร์จัดเตรียม Cap Bits ให้รับทราบพีเจอรของฮาร์ดแวร์ที่มีอยู่ใน HAL และพีเจอรที่ต้องจำลองขึ้นโดยซอฟต์แวร์ HEL ดังนั้นวิธีที่ดีที่สุดก็คือเขียนแอปพลิเคชันโดยใช้ระบบ หรือพีเจอรต่ำสุดที่ยอมรับได้ และ Optimize โค้ดที่เขียนให้รันได้เร็วและมีประสิทธิภาพมากที่สุด อีกทางหนึ่งควรเขียนโค้ดเพื่อการทำงานของ HEL ในอนาคตไว้ด้วย การสนับสนุนการใช้พีเจอรเหล่านี้เป็นพีเจอรพิเศษที่จะมีให้เลือกบนเกมได้ เช่น การทำงาน Texture ขั้นสูง การทำพอลีกอนที่มีความซับซ้อนสูง หรือแม้แต่การสร้างแสงแบบไดนามิก เพื่อว่าใครก็ตามที่มีฮาร์ดแวร์ที่มีความสามารถสูงจะสามารถใช้งานพีเจอรดังกล่าวได้ ดังนั้นควรออกแบบให้รองรับพีเจอรต่างๆที่มีได้ทั้งหมด

2.3.3 หลักการเบื้องต้นเกี่ยวกับ DirectX

ก่อนที่จะมี DirectX เกิดขึ้นนั้น นักพัฒนาเกมคอมพิวเตอร์สำหรับ คอสหรือวินโดวส์ จะต้องเขียนเกมคอมพิวเตอร์ให้รู้จักกับฮาร์ดแวร์ ซึ่งมีอยู่มากมายในท้องตลาด ซึ่งในกรณีที่มีฮาร์ดแวร์ตัวใหม่เกิดขึ้น อาจจะทำให้เกิดความไม่สนับสนุนกันระหว่างเกมคอมพิวเตอร์กับฮาร์ดแวร์ตัวใหม่นั้น นักเล่นเกมจะต้องรองจนกว่านักพัฒนาเกมจะทำการอัปเดตเกมนั้นๆ ให้ใช้ความสามารถของฮาร์ดแวร์ใหม่ได้ เมื่อเป็นเช่นนี้ไมโครซอฟต์จึงได้ทำการพัฒนาเทคโนโลยีที่มีความสามารถในการเป็นสื่อกลางติดต่อระหว่างเกมคอมพิวเตอร์ หรือ โปรแกรมมัลติมีเดียต่างๆ กับฮาร์ดแวร์ขึ้นมาโดยใช้ชื่อว่า "DirectX"

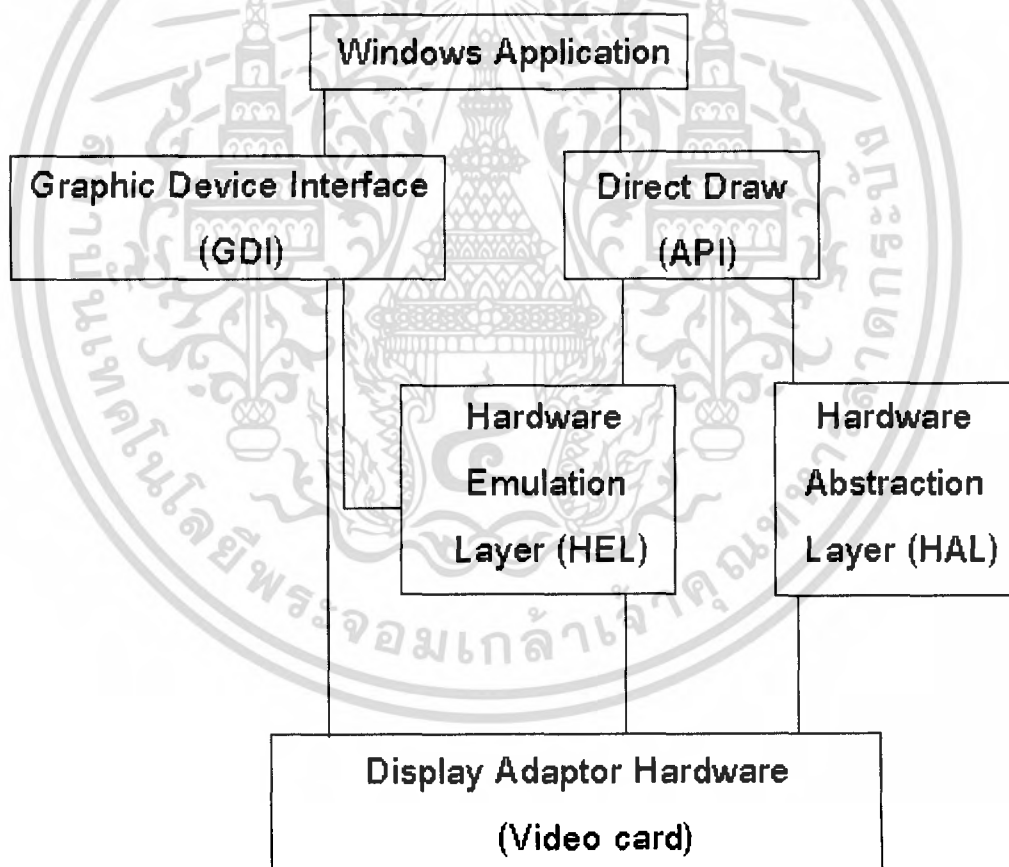
DirectX ตามความหมายจะหมายถึง ไบเบรารีคำสั่ง (Run Time Library) ที่ช่วยทำงานด้านมัลติมีเดีย Graphic โดยตัว DirectX Foundation จะมีส่วนประกอบที่เรียกว่า HAL (Hardware Abstraction Layer) จะใช้ซอฟต์แวร์ในการตรวจสอบความสามารถของฮาร์ดแวร์ที่อยู่ในเครื่องคอมพิวเตอร์นั้นอย่างอัตโนมัติ แล้วนำมากำหนดพารามิเตอร์ของแอปพลิเคชันให้ตรงตามความเหมาะสมระหว่างเกมคอมพิวเตอร์ หรือ โปรแกรมมัลติมีเดียกับไดร์เวอร์ของฮาร์ดแวร์ที่มีส่วนเกี่ยวข้องกับเกมหรือโปรแกรมนั้น ทำให้การประมวลผลโปรแกรมทำได้เร็วขึ้น เนื่องจากขั้นตอนต่างๆ จะถูกนำไปประมวลผลโดยตรง ไม่ต้องอาศัยตัวกลางอย่างเช่น GDI (Graphic Device Interface) ก่อน ทำให้นักพัฒนาเกมคอมพิวเตอร์สามารถเขียนโปรแกรมให้สื่อสารกับ DirectX เท่านั้นก็เพียงพอ นอกจากนี้ DirectX Foundation ยังมีส่วนประกอบที่เรียกว่า HEL (Hardware emulation Layer) ทำให้สามารถใช้โปรแกรมมัลติมีเดีย หรือ โปรแกรมที่เกี่ยวข้องกับ 3D บน Hardware ที่ไม่สนับสนุนการใช้งานทางด้าน 3 มิติ โดยจะทำการจำลองความสามารถบางอย่างที่ฮาร์ดแวร์ตัวนั้น ไม่มี ให้สามารถใช้งานได้กับโปรแกรมที่ต้องการ แม้จะมีข้อเสียอยู่บ้างตรงที่ อาจทำให้ช้าลงบ้างก็ตาม แต่ก็คุ้มค่ากับความสามารถของ DirectX ที่มีอยู่ในปัจจุบัน

2.3.4 API ของ DirectX

DirectX ประกอบไปด้วย API มากมายที่ได้รับการออกแบบเพื่อใช้พัฒนาเกม และการจำลองแบบแบบสามมิติ (3D Simulation) (ส่วนใหญ่ยังไม่ได้พัฒนาให้เป็นแบบสามมิติ) โดย DirectX มี ไลบรารีที่เก็บฟังก์ชันที่ใช้ในการเรนเดอร์แบบสองมิติและสามมิติ , สร้างเสียงแบบปกติ และแบบสามมิติ , ดนตรี , ติดต่อเป็นพิมพ์ , จอยสติ๊ก และอุปกรณ์อินพุตชนิดต่างๆ รวมทั้ง ฮาร์ดแวร์ที่มีความสามารถสร้างปฏิกิริยาสะท้อนกลับ (จอยสติ๊กแบบสั่น) และการเล่นเกมผ่านเครือข่าย สามารถใช้ไลบรารีของคำสั่งต่างๆที่รวมเข้ามาเพื่อสร้างเกม และทำซิมูเลชั่นที่งดงาม

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

DirectDraw เป็นชุด API สำหรับใช้จัดการอุปกรณ์แสดงผล ควบคุมข้อมูลบิตแมปหน่วย ความถี่ ออกนอกพื้นที่สกรีน และสร้างการติดต่ที่รวดเร็วให้กับพีเจอร์ของฮาร์ดแวร์ เช่น Blitting และ Page Flipping ซึ่งเป็นพีเจอร์พื้นฐานที่ Direct3D สามารถทำได้



รูปที่ 2.6 แสดงแผนภาพการทำงานของ Direct X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Cooperative level และ Display mode**

ก่อนที่จะคำสั่งต่างๆใน DirectX ให้แสดงผลภาพได้นั้น เราจะต้องเซต Cooperative level และ Display mode ก่อน โดยการเซต Cooperative level ก็คือการบอก DirectDraw ว่าเราจะทำงานกับ DirectDraw ในลักษณะใด เช่น window mode หรือ fullscreen 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 โดยจะต้องระบุความกว้างความยาวของ 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 ใช้สำหรับเก็บข้อมูลภาพ bitmap เพื่อเตรียมสำหรับการนำมาแสดงผลบนจอภาพต่อไป โดยพื้นที่นี้อาจอยู่บน video memory หรือ system memory ก็ได้ ขึ้นอยู่กับค่า flags ที่ส่งให้ตอนสร้าง โดย surface แบ่งออกเป็น 2 ชนิด คือ Primary Surface และ Off-screen surface

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

Off-screen surface คือ surface ใช้สำหรับเก็บข้อมูลภาพต่างๆ เพื่อ Blitting ลงใน buffer หรือ surface อื่นเพื่อเตรียมแสดงผลต่อไป

- **Blitting และ Flipping**

Blitting คือ การนำภาพต่างๆ จาก source ไปวาดบนบริเวณ destination ที่กำหนด โดยอาจมีการย่อขยายภาพให้พอดีกับขนาด destination ที่กำหนด และสามารถทำการ transparent สีบางส่วนได้ด้วย การทำ Blitting สามารถทำได้หลายครั้ง โดยภาพที่ทำ Blitting จะเรียงซ้อนทับกันเป็นชั้นๆ

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

เป็นที่พักของภาพที่จะรอแสดงผลบนหน้าจอ 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 โดยการสลับในที่นี้ไม่ได้ หมายถึง การสลับที่ข้อมูลภาพแต่เป็นการสลับเฉพาะ pointer ที่ชี้พื้นที่ที่เก็บข้อมูลอยู่เท่านั้น ซึ่งทำได้อย่างรวดเร็ว และไม่ทำให้เกิดการกระพริบ หรือการกระตุกของภาพ

2.4 OpenGL

- เป็นภาษาระดับล่างใช้ในการ rendering และ imaging library
- เป็นส่วนติดต่อระหว่าง graphic hardware กับ application program
- เป็น API ที่ใช้สร้าง color image ของ 3D objects
- เป็นภาษาทางด้านกราฟฟิกในเชิงของการทำงานมากกว่าเชิงอธิบาย
- ไม่ขึ้นกับ OS และ Hardware platform ที่ใช้

2.4.1 OpenGL Features

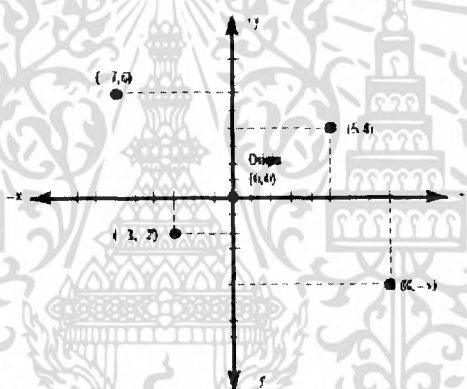
Texture Mapping	เป็นการนำภาพ Mapping เข้ากับ พื้นผิวของ Object
Z-buffering	ความสามารถในการคำนวณระยะทางจากตำแหน่งของ viewer
Double buffering	ทำให้ animation มีความ smooth
Lighting effects	สามารถคำนวณผลกระทบของแสงเมื่อแสงกระทบกับพื้นผิวของวัตถุ จากแหล่งกำเนิดแสง 1 จุด หรือ มากกว่า
Smooth shading	สามารถคำนวณผลกระทบของ shading ที่เกิดขึ้นเมื่อแสงกระทบกับพื้นผิว ของวัตถุ โดย มุมตกกระทบที่ต่างกัน จะให้ผลออกมาที่ต่างกัน
Material properties	ความสามารถในการระบุลักษณะของวัตถุว่า แข็ง หรือ ทุ้อ เช่น เหล็ก หรือ พลาสติก
Alpha blending	สามารถระบุ alpha หรือ opacity เพิ่มเติมจาก RGB ได้
Transformation matrices	สามารถเปลี่ยน ตำแหน่ง , ขนาด และ perspective ของ object ใน 3D coordinate

2.4.2 ระบบพิกัดของเบื้องต้นของ OpenGL

- 2D Cartesian Coordinate
- Coordinate Clipping
- ViewPorts
- 3D Coordinate System
- Projections
- Orthographic Projections
- Perspective Projections

2D Cartesian Coordinate

ระบบพิกัดที่เราคุ้นเคยที่สุดและเป็นมาตรฐานที่สุด ก็คือระบบพิกัดคาร์ทีเซียน (Cartesian) ซึ่งระบบโดยคู่พิกัด (x,y) ตามแนวนอนและแนวตั้งตามลำดับ โดยมีจุดกำเนิด (origin) ที่ (0,0)



รูปที่ 2.7 ระบบพิกัดคาร์ทีเซียนสองมิติ

โดยแกนทั้งสอง (x และ y) นั้นตั้งฉากซึ่งกันและกันและรวมเรียกว่า xy plane หรือระนาบ xy โดยระนาบคือ พื้นที่ราบ นั่นเอง ซึ่งโดยทั่วไปแล้ว เมื่อเส้นตรงสองเส้นตัดกันทำมุมๆหนึ่ง จะเกิดระนาบขึ้นมาโดยปริยาย ซึ่งในระบบที่มีแกนเพียงแกนนั้น จะมีระนาบเพียงหนึ่งเดียวเท่านั้น

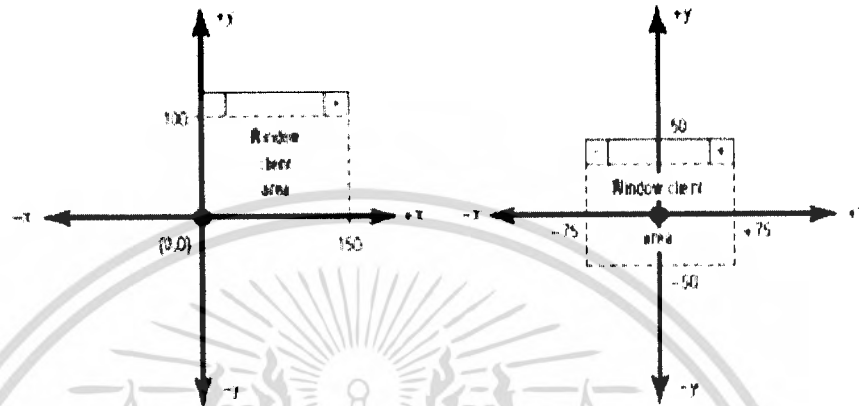
Coordinate Clipping

ก่อนที่เราจะสามารถพล็อตจุด (หรือวาดรูปใดๆ) ลงไปในหน้าต่างได้ เราต้องทำการบอก OpenGL ก่อนว่าจะทำการ “แปลง” คู่พิกัดของสิ่งที่เราต้องการวาด ไปเป็นคู่พิกัดของจอ (physical pixel coordinate) ได้อย่างไร ซึ่งสามารถทำได้ โดยการระบุขอบเขตในระนาบคาร์ทีเซียนที่ใช้หน้าต่าง ซึ่งเรียกว่า clipping region

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในระนาบสองมิตินั้น clipping region ก็คือว่า x, y ต่ำสุดและสูงสุดที่อยู่ภายในหน้าต่างนั่นเอง อีกวิธีหนึ่งที่ทำได้อีกก็คือ การระบุตำแหน่งของจุดกำเนิดในเชิงสัมพันธ์กับหน้าต่าง

รูปต่อไปนี้แสดงถึง clipping region แบบที่นิยมใช้กันสองแบบ



รูปที่ 2.8 clipping region ในระนาบคาร์ทีเซียนสองมิติ

รูปซ้ายนั้น มีพิกัด x ตั้งแต่ 0 ถึง 150 จากซ้ายไปขวา และ y ตั้งแต่ 0 ถึง 100 จากล่างขึ้นบน จุดกลางจมีตำแหน่ง $(75,50)$ ส่วนในรูปซ้ายนั้น x มีคั้งตั้งแต่ -75 ไปถึง 75 และ y มีคั้งตั้งแต่ -50 ไปถึง 50 และจุดกลางมีค่า $(0,0)$

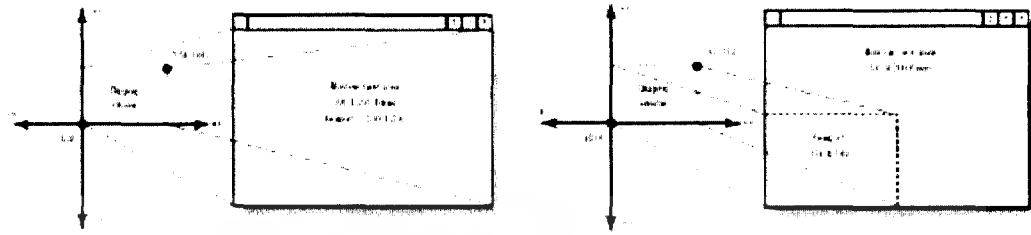
เราสามารถที่จะใช้ OpenGL หรือว่าไลบรารีตัวอื่นๆในการปรับค่านักกลับหัวกลับหางหรือซ้ายเป็นขวาได้ ซึ่งในความเป็นจริงแล้ว ค่าปริยายของพิกัดหน้าต่างนั้น มักจะให้ค่า y วิ่งจากบนลงล่าง มากกว่าล่างขึ้นบน ทั้งนี้เนื่องจาก เป็นลักษณะที่เป็นธรรมชาติมากกว่าสำหรับการแสดงข้อความ (text) แต่ก็ไม่เหมาะสมเท่าใดนักกับการใช้งานหรือวาดกราฟฟิก

Viewports (การ map ค่าจาก drawing coordinate ไปยัง window coordinate)

โดยปกติแล้ว clipping area ที่เราต้องการกำหนดนั้น มักไม่ค่อยตรงกับขนาดจริงของหน้าต่าง (เช่นเรา อาจกำหนดหน้าต่างขนาด 640×480 แต่ว่าต้องการ clipping area จาก $x : [-75,75]$ และ $y : [-50,50]$ ก็ได้) ดังนั้นค่าของจุดใน clipping area ที่เราต้องการวาดนั้นต้องได้รับการ map ไปยังค่าพิกัดจริงของหน้าต่าง และไปยังค่าจริงของอุปกรณ์แสดงผลต่อไป ซึ่งเราเรียกว่า map เช่นนี้ว่า viewport

Viewport คือพื้นที่ในส่วน of หน้าต่าง โปรแกรมที่ใช้ในการวาด clipping area ซึ่งอันที่จริงแล้วก็เป็นเพียง mapping ของ clipping area ไปยัง window

coordinate อย่างง่าย ๆ เท่านั้น ซึ่งโดยปกติแล้ว viewport จะมีค่าเป็นพื้นที่ทั้งหมดของ หน้าต่าง แต่ว่าไม่ใช่สิ่งที่จำเป็นแต่อย่างใด เราสามารถระบุ viewport เป็นอะไรก็ได้ที่เราต้องการ เช่น ล่างซ้ายของ clipping area เป็นต้น



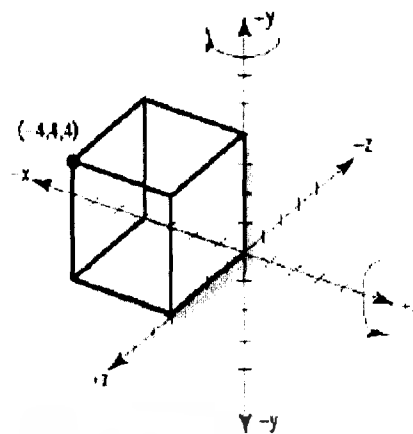
รูปที่ 2.9 ตัวอย่างวิวพอร์ต (viewport)

จากรูป ซ้ายมือ เรากำหนดขนาดของหน้าต่างไว้เป็น 300x200 และ กำหนดให้ viewport มีขนาดเป็น 300x200 ด้วยเช่นกัน ซึ่งถ้าหากเรากำหนด clipping area ไว้เป็น $x : [0,150]$ และ $y : [0,100]$ ระบบก็จะทำการ map ค่าพิกัดไปยังค่าพิกัดของหน้าต่างจริงให้เอง (ซึ่งนั่นหมายความว่า การเปลี่ยนแปลงพิกัดใน clipping area ก็จะถูก map โดยการเพิ่มเป็นสองเท่า และถูกวาดลงหน้าต่าง) ส่วนรูปขวามือนั้น เรากำหนดขนาดหน้าต่างไว้เป็น 300x200 และ clipping area เป็น $x : [0,150]$ และ $y : [0,100]$ เท่ากันกับภาพซ้าย แต่ว่าเรากำหนด view เป็นขนาด 150x100 ซึ่งผลที่ได้นั้น ทำให้ภาพที่เราเห็นเต็ม clipping area นั้น จะปรากฏบนมุมล่างซ้ายของหน้าต่างเท่านั้น (เพราะว่าการเปลี่ยนแปลงขนาด 1 pixel ใน clipping area ก็จะมีค่าเท่ากับ 1 pixel ในหน้าต่างด้วย)

เราสามารถใช้ viewport นี้ในการเพิ่ม / ลดขนาดภาพในหน้าต่างได้และสามารถใช้ในการแสดงเฉพาะส่วนของ clipping area ได้เช่นกัน

3D Coordinate System

สำหรับระบบพิกัดสามมิตินั้นก็เหมือนกับสองมิติ เพียงแต่เพิ่มแกน z ที่ตั้งฉากกับระนาบ xy เข้ามาอีกแกนหนึ่ง ซึ่งในระบบการแสดงผลของภาพนั้น ก็คือระยะใกล้ไกล (+ z ใช้ออกจากจอภาพมาหาผู้ใช้) ดังรูป

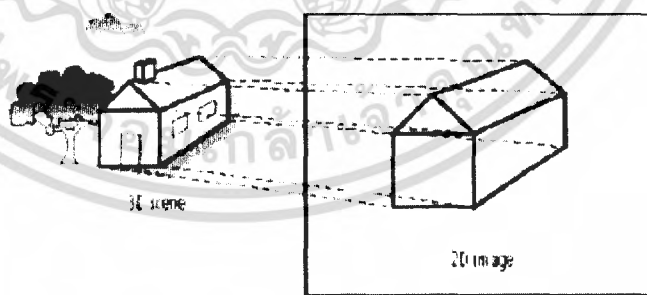


รูปที่ 2.10 ระบบพิกัดสามมิติ (แกน z ชี้ออกจากจอภาพเข้าหาตัวผู้มอง)

Projection: จาก 3D สู่ 2D

แม้ว่าเราจะสามารถกำหนดและวาดสิ่งที่ต้องการได้โดยใช้พิกัดสามมิติ แต่ว่าจอภาพก็ยังคงเป็นสองมิติอยู่นั่นเอง ดังนั้นการที่จะแสดงผลออกให้เราเห็นทางหน้าต่างของโปรแกรมได้นั้น ต้องทำการแปลงพิกัดมาเป็นสองมิติเสียก่อนถึงจะระบุตำแหน่ง pixel ได้ ซึ่งสามารถทำได้โดยใช้วิธีการทางคณิตศาสตร์ (ตรีโกณมิติและวิธีการทางเมตริกซ์) ซึ่งแม้ว่าเราจะไม่ต้องเข้าใจคณิตศาสตร์ตรงนี้มากเท่าไหร่นัก ความเข้าใจที่ลึกซึ้งของ 3D maths จะช่วยให้เราใช้งาน OpenGL ได้อย่างมีประสิทธิภาพมากขึ้น กับงานหลายประเภทมากขึ้นด้วย

สำหรับคอนเซปต์แรกที่เราต้องทำความเข้าใจ ก็คือสิ่งที่เรียกว่า projection ซึ่งเป็นการนำเอาพิกัดสามมิติที่เราได้กระทำการระบุตอนที่วาดไป แปลง (project) ลงบนระนาบสองมิติ



รูปที่ 2.11 Projection: 3D image --> 2D surface

ซึ่งใน OpenGL นั้น จะมี projection อยู่สองแบบที่เราต้องทำความเข้าใจ นั่นก็คือ orthographic และ perspective ซึ่งเราทำการระบุ projection หมายความว่า

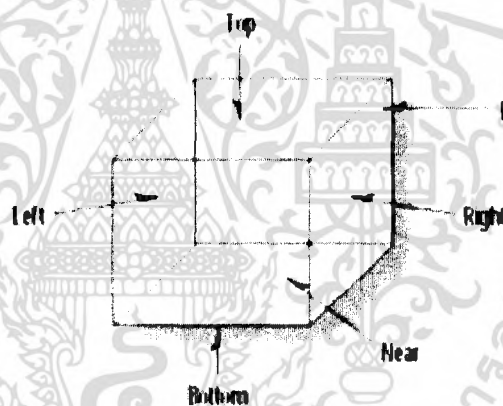
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราได้ทำการระบุ viewing volume และกำหนดว่า viewing volume นี้จะถูก transform ยังไงลงในระนาบสองมิติ

- **Orthographic Projections**

ใช้เมื่อทำการระบุ square (หรือ rectangular) viewing volume ซึ่งอะไรก็ตามนอกเหนือจาก volume นี้จะไม่ถูกวาด นอกเหนือจากนั้น วัตถุทุกอย่างที่มีขนาดเท่ากันจะถูกวาดด้วยขนาดเท่ากัน ไม่ว่าจะอยู่ใกล้หรือไกลต่างกันยังไงก็ตาม ซึ่ง projection ในลักษณะนี้ มักจะถูกใช้ในงานจำพวกงานออกแบบเขียนแบบหรือกราฟสองมิติ ตัวอย่างในการใช้งานกับโปรแกรมสามมิติก็เช่นการเพิ่มตัวอักษร (เช่น frame rate) ลงไปบนหน้าจอ เป็นต้น

เราทำการระบุ orthographic projection โดยการระบุระนาบสำหรับ clipping ดังนี้ :far, near, left, right, top, และ bottom ดังรูป

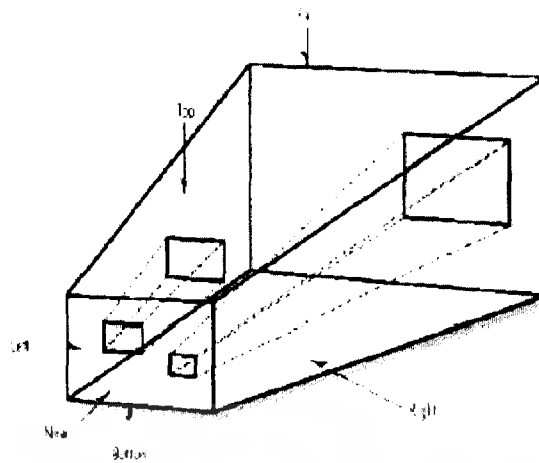


รูปที่ 2.12 Orthographic projection clipping volume

ซึ่งวัตถุทุกอย่างที่ระบุตำแหน่งให้อยู่ภายใน clipping volume นี้จะถูก project ไปเป็นพิกัดสองมิติ (ตาม orientation) สำหรับแสดงผลต่อไป

- **Perspective Projections**

Projection แบบที่สอง และเป็นแบบที่คุ้นเคยกันมากกว่า ก็คือแบบ Perspective ซึ่งจะเพิ่มระยะใกล้ไกลของวัตถุในรูปเข้าไปด้วย ซึ่งเหมือนกับการคิดว่า clipping volume มีลักษณะเป็นพรางมิดหัวตัด (เรียกว่า frustum) ซึ่งทำให้วัตถุที่อยู่ใกล้กว่า มีขนาดใกล้เคียงกับขนาดที่ถูกกำหนดไว้มากกว่าวัตถุที่อยู่ไกล ดังรูป



รูปที่ 2.13 Perspective projection clipping volume

2.5 ปัญญาประดิษฐ์ (Artificial Intelligence)

หรือ ปัญญาของเครื่องจักร (Machine Intelligence) โดยทั่วไป คือการศึกษาเกี่ยวกับความฉลาดของสิ่งมีชีวิต โดยเฉพาะมนุษย์ เพื่อที่จะสร้างความฉลาดเทียมให้กับสิ่งที่ไม่มีชีวิต เช่น หุ่นยนต์ เครื่องจักร และคอมพิวเตอร์ ขึ้นมาได้ โดยทั่วไปแล้ว เราถือกันว่าปัญญาประดิษฐ์เป็นสาขาหนึ่งของศาสตร์ด้านวิทยาการคอมพิวเตอร์ แต่นักวิทยาศาสตร์หลาย ๆ ท่านเชื่อว่าการศึกษาด้านปัญญาประดิษฐ์นั้น เมื่อมองให้ลึกซึ้งแล้ว ครอบคลุมกว้างไกลกว่าศาสตร์ด้านวิทยาการคอมพิวเตอร์ โดยได้โยงไปถึงสาขาอื่นๆ เช่น ปรัชญา จิตวิทยา และ ชีววิทยาด้วย เนื่องจากศาสตร์เหล่านี้ ได้มีการศึกษาเรื่องกระบวนการการคิด การให้เหตุผลหรือการอนุมานและการทำงานของสมองมานานแล้ว ซึ่งจริง ๆ แล้ว นับว่าความเชื่อนี้ถูกต้อง เนื่องจากแนวคิดหลาย ๆ อย่างในศาสตร์นี้ได้มาจากการปรับปรุงเพิ่มเติมจากศาสตร์อื่นๆ เช่น

- การเรียนรู้ของเครื่อง นั้นมีเทคนิคการเรียนรู้ที่เรียกว่า การเรียนรู้ต้นไม้ตัดสินใจ ซึ่งประยุกต์เอาเทคนิคการอุปนัยของ จอห์น สจวร์ต มิลล์ นักปรัชญาชื่อดังของอังกฤษ มาใช้
- เครื่องข่ายประสาทเทียมก็นำเอาแนวคิดของการทำงานของสมองของมนุษย์ มาใช้ในการแก้ปัญหาการแบ่งประเภทของข้อมูล และแก้ปัญหาอื่นๆ ทางสถิติ เช่น การวิเคราะห์ความถดถอยหรือ การปรับเส้นโค้ง

อย่างไรก็ตาม เนื่องจากปัจจุบันวงการปัญญาประดิษฐ์ มีการพัฒนาส่วนใหญ่โดยนักวิทยาศาสตร์คอมพิวเตอร์ อีกทั้งวิชาปัญญาประดิษฐ์ ก็ต้องเรียนที่ภาควิชาคอมพิวเตอร์ของคณะวิทยาศาสตร์หรือคณะวิศวกรรมศาสตร์ เราจึงถือเอาง่าย ๆ ว่า ศาสตร์นี้เป็นสาขาของวิทยาการคอมพิวเตอร์นั่นเอง

แต่ในที่นี้จะนำปัญญาประดิษฐ์มาใช้เพื่อทำให้ศัตรูสามารถค้นหาผู้เล่น และ โจมตีได้

ถูกต้อง โดยใช้ความรู้ของ Best-First Search และ A* Path Finding เข้ามาเกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 Best-First Search

เป็นวิธีการค้นหาแบบใช้ฟังก์ชันฮิวริสติก เพื่อค้นหาเส้นทางที่ดีที่สุด โดยมี อัลกอริทึมดังต่อไปนี้

1. OPEN = { initial state }
2. UNTIL (พบ goal state หรือ ไม่มี state อยู่ใน OPEN) DO
 - 2.1 ดึง state ที่มีค่า heuristic ต่ำที่สุดออกจาก OPEN
 - 2.2 สร้าง successors ของ state นั้น
 - 2.3 FOR EACH successor DO
 - 2.3.1 IF successor ยังไม่เคยถูกสร้างขึ้นมาก่อน THEN
เติม successor นั้นใน OPEN และจำ parent state ไว้
 - 2.3.2 IF successor นั้นเคยถูกสร้างขึ้นมาแล้วและ path ใหม่ดีกว่าเดิม
THEN เปลี่ยน parent state และ update cost ใหม่

2.5.2 A* Algorithm

Best-First Search เป็น algorithm ง่ายของ A* ซึ่งใน A* เราแยก Heuristic Function ของ state s ใดๆ ออกเป็น 2 ส่วน ได้แก่

$$f(s) = g(s) + h'(s)$$

โดยที่ g คือ ฟังก์ชันที่คำนวณ cost จาก initial state ถึง current state

h' คือ ฟังก์ชันที่ประมาณ cost จาก current state ถึง goal state

f จึงเป็นฟังก์ชันที่ประมาณ cost จาก initial state ถึง goal state

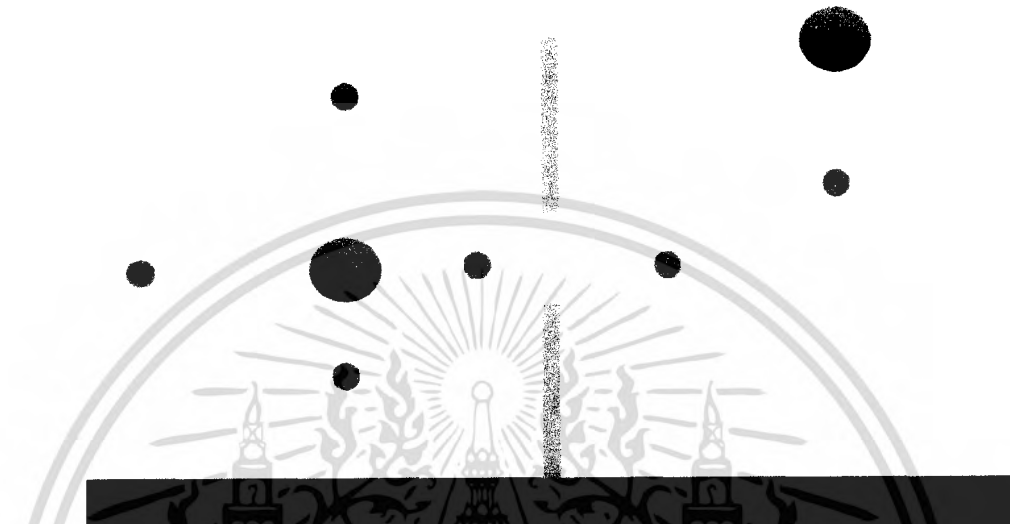
จุดเด่นของ A* ที่แตกต่างจาก Best-First Search

- การใช้ g ร่วมในการประมาณค่า f ทำให้การเลือก state ไม่เพียงพิจารณาแต่ค่า state ต่ำเท่าไร (ซึ่งประมาณจาก h') แต่ยังรวมไปถึงว่า path ที่นำไปสู่ state ต่ำแค่ไหนด้วย
- โดยการใช้ g state ที่ประมาณว่าใกล้ goal state มากที่สุดอาจจะไม่ถูกกระจายก็ได้
- โดยปกติ A* จะหาคำตอบซึ่ง path ที่นำไปสู่คำตอบเสีย cost น้อยที่สุด
- ถ้าให้ g ของทุก state เป็น 0 state ที่ใกล้ goal state จะถูกกระจายก่อน
- ถ้าต้องการหาคำตอบโดยให้ step น้อยที่สุด ก็ให้ g ของทุก step เป็น 1

ตัวอย่างขั้นตอนการทำงานของ A* Pathfinding Algorithm

NooB !!

Algorithm Concept



รูปที่ 2.14 แสดงตำแหน่งของตัวละคร ศัตรู และ Waypoint ต่างๆ

NooB !!

Algorithm Concept

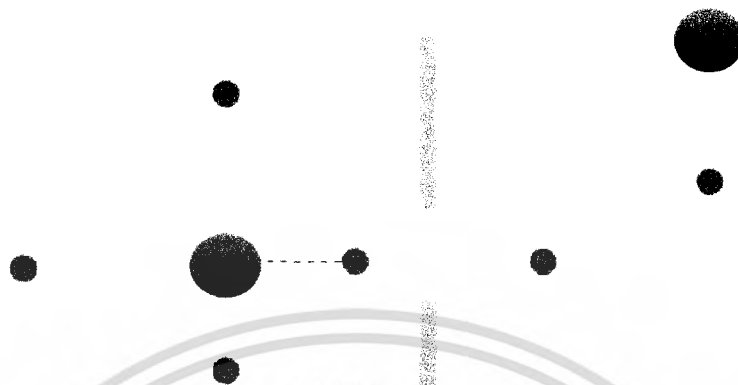


รูปที่ 2.15 แสดง Adjacent waypoint ที่ตัวละครสามารถเดินไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NooB !!

Algorithm Concept



รูปที่ 2.16 เลือก Adjacent Waypoint ที่มีค่า Heuristic Function น้อยที่สุด

NooB !!

Algorithm Concept

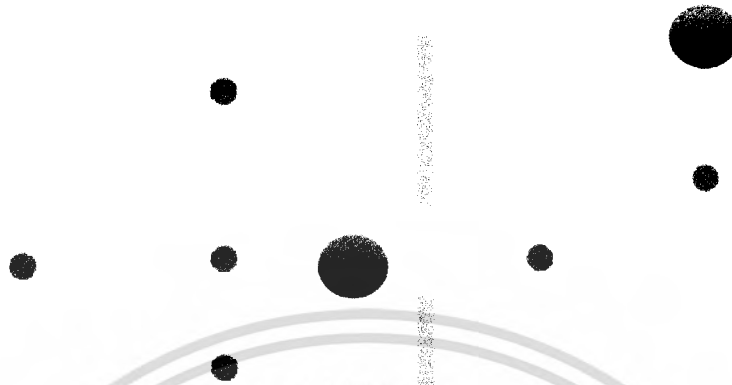


รูปที่ 2.17 เดินไปยัง Waypoint นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NooB !!

Algorithm Concept



รูปที่ 2.18 เริ่มการคำนวณหา Waypoint ที่จะเดินครั้งใหม่

NooB !!

Algorithm Concept

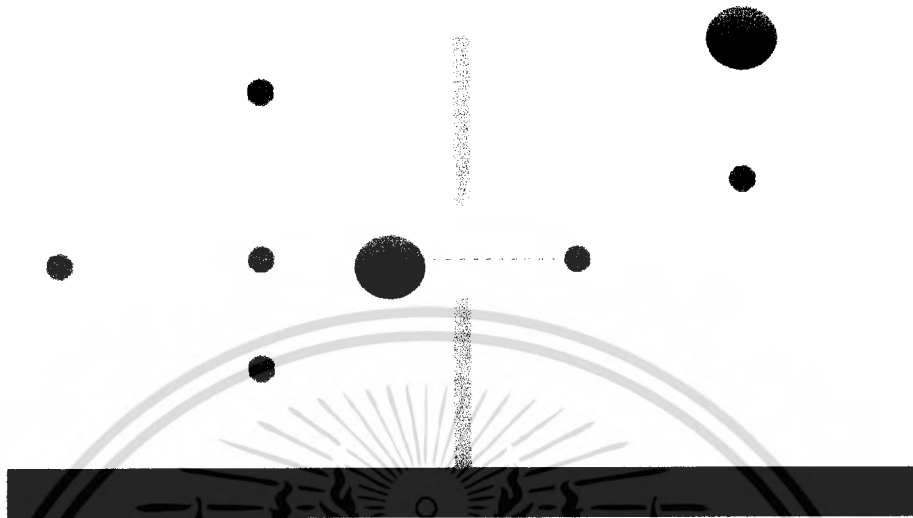


รูปที่ 2.19 แสดง Adjacent waypoint ที่ตัวละครสามารถเดินไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NooB !!

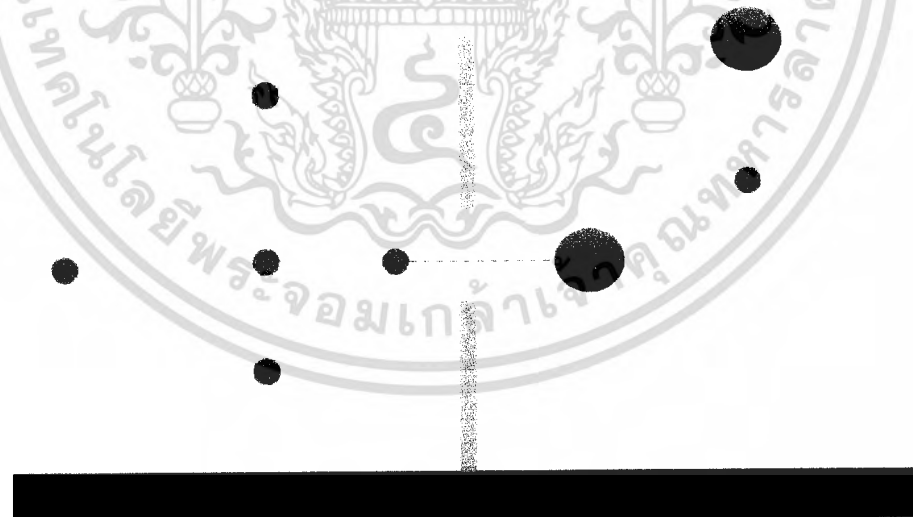
Algorithm Concept



รูปที่ 2.20 เลือก Adjacent Waypoint ที่มีค่า Heuristic Function น้อยที่สุด

NooB !!

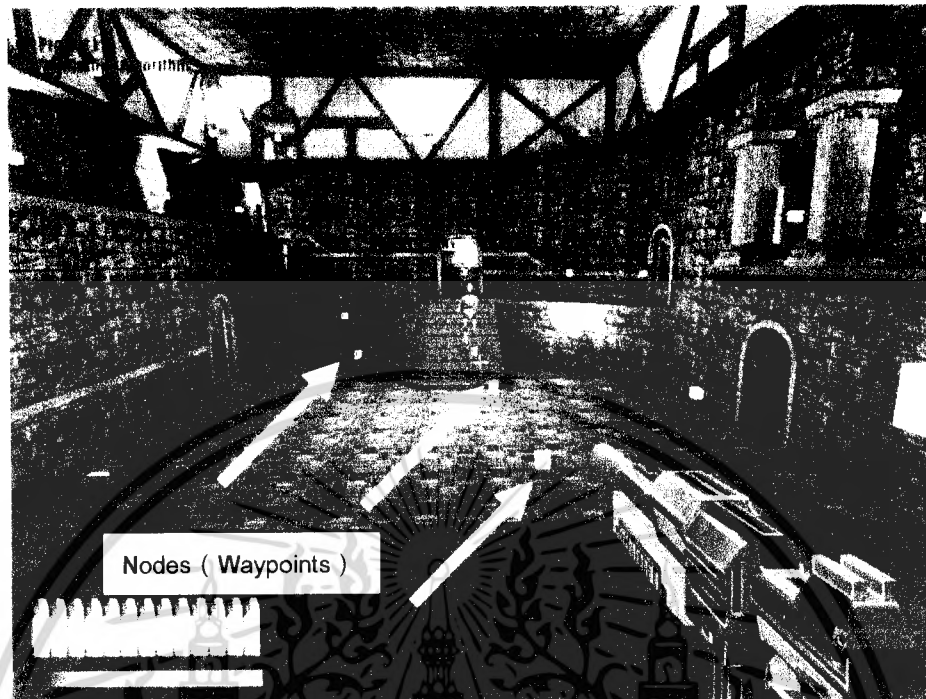
Algorithm Concept



รูปที่ 2.21 เดิน ไปยัง Waypoint นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาและเลือกเส้นทางที่เหมาะสมของศัตรูภายในเกม



รูปที่ 2.22 แสดงตัวอย่างตำแหน่งของเวย์พอยต์ (Waypoint)

ภายในเกมนั้น โหนดที่ใช้ในการพิจารณา คือ โหนดที่เราได้ทำสร้างและกำหนดพิคคไว้ ซึ่ง Node ที่ได้กล่าวมานั้น คือ กล้องสี่เหลี่ยมจัตุรัสสีขาว (Waypoint) ในรูปประกอบด้านบน ซึ่งศัตรูที่เป็นระบบคอมพิวเตอร์จะใช้มันในการตัดสินใจเพื่อเลือกเส้นทางที่เหมาะสมสำหรับ ตามหาและ โจมตีผู้เล่น

2.6 Irrlicht Engine

Irrlicht Engine เป็น 3D engine ที่ถูกพัฒนาขึ้นโดยใช้ภาษา C++ ซึ่งมีความสามารถในการทำงานได้ทุก platform ซึ่งมี API ระดับสูงที่ใช้สำหรับสร้างเกมได้ทั้ง 3D และ 2D หรือใช้สามารถนำไปใช้ในงานด้านกราฟิก อื่นๆ ได้ ซึ่งอนจิ้นได้รวบรวมเอาความสามารถต่างๆ ไว้ด้วยกันอันได้แก่ การทำงานแบบ

- Dynamic Shadow
- Particle Systems
- Character Animation
- Indoor & Outdoor Technology
- Collision Detection

ซึ่งเข้าควบคุมและออกแบบได้ง่ายผ่านทางอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งรายละเอียดความสามารถต่างๆมีดังนี้

- มีประสิทธิภาพในการ render ภาพทั้งใน Direct3D และ OpenGL
- ทำงานได้โดยไม่ขึ้นกับ platform เช่น Window95 98 NT 2000 XP หรือ Linux
- มี Library ให้ใช้เป็นจำนวนมาก ทั้งยังสามารถเพิ่ม Library อื่นๆ ได้อย่างยืดหยุ่น ทั้งยังสนับสนุนการทำงานของ vertex & pixel shader อีกด้วย
- สามารถทำงานผสมผสานระหว่าง feather ตัวเองและ feather นอก ได้อย่างมีประสิทธิภาพ เช่นสามารถ import utilities ต่างๆมาใช้ได้
- มีระบบ Character animation คือ สามารถนำ animation ต่างๆมาประกอบเข้ากันได้
- มี effect หลากหลายให้ใช้
- สนับสนุนการพัฒนาโดยใช้ VC++ และ .NET ทุกภาษาเช่น C# , VisualBasic และ Delphi.NET
- มีความเป็นอิสระทั้งในด้าน Platform และ driver ทั้งยังมี software ที่ช่วยสำหรับการ render อย่างมีประสิทธิภาพ (software renderer)ซึ่งมีความสามารถในการทำ z-buffer gouraud shading,alpha-blending และ transparency ทั้งยังสามารถวาดภาพ 2D ได้อย่างรวดเร็ว
- มีระบบ GUI ให้ผู้ใช้สามารถใช้งานได้ง่าย
- มี Function ช่วยในการวาดภาพแบบ 2 มิติ เช่น alpha blending และการผสมผสานระหว่างภาพ 3D และ 2D
- มี Tutorial ที่สอนในเรื่องการใช้ API ที่เข้าใจได้ง่าย
- สนับสนุนการเขียนโดยใช้ C++ และภาษาเชิง object ได้เกือบทุกภาษา
- สามารถ import ไฟล์ข้อมูลโดยตรงจาก Maya (.obj) , 3DStudio(.3ds) COLLADA(.dae) , DeleD(.dmf), Milkshape(.ms3d) Quake 3 levels (.bsp) , Quake 2 models(.md2) , Microsoft DirectX (.X) ได้
- สามารถ import texture ได้ทั้งในสกุล
 - Windows Bitmap (.bmp)
 - Portable Network Graphics (.png)
 - Adobe Photoshop (.psd)
 - JPEG File Interchange Format (.jpg)
 - Truevision Targe (.tga)
 - ZSoft Painbrush (.pcx)
- มีระบบ collision detection ที่ตอบสนองได้รวดเร็วและใช้งานง่าย
- มี Library สนับสนุน function ทางคณิตศาสตร์ ที่ทำงานกับภาพ 3D เพื่อให้ทำงานได้รวดเร็วยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถอ่านข้อมูลจากไฟล์ ที่ถูกบีบอัดไว้ได้โดยตรงเช่น (.zip)
- มี XML parser ที่รวดเร็ว
- สนับสนุน ข้อมูลแบบ Unicode
- ทำงานได้กับทั้ง Microsofts VisualStudio6.0, VisualStudio.NET 7.0-8.0, Metrowerks Codewarrior, and Bloodshed Dev-C++ with g++3.2-4.0.
- ตัวเองจะ open source และฟรี ซึ่งสามารถเข้าไป debug หรือแก้ไขได้ตามใจชอบ ซึ่ง licence จะเป็นของ zlib-licence ไม่ใช่ของ GPL หรือ LGPL ซึ่งทั้ง GPL และ LGPL เป็น licence ที่ยอมให้คนอื่นเข้าไปปรับปรุงแก้ไขได้ แต่หากมีการแก้ไข จำเป็นต้องแจ้งให้เจ้าของทราบ

2.6.1 Special effects

ใน Irrlicht engine จะมี special effect มากมายให้เลือกใช้ ซึ่งใช้ได้ง่าย ส่วนใหญ่แค่ switch ไปมาว่าจะเปิดใช้หรือไม่แค่นั้น ซึ่ง effect ที่รองรับมี

ดังต่อไปนี้

- Realistic water surfaces
- Dynamic lights
- Dynamic shadows using the stencil buffer
- Billboards
- Bump mapping
- Parallax mapping
- Transparent objects
- Light maps
- Customizable Particle systems for snow, smoke, fire, ...
- Sphere mapping
- Texture animation
- Skyboxes
- Fog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 Driver

Irrlicht Engine สนับสนุน 5 API ซึ่งในนั้นมี 4 ตัวที่เอนจินนี้รองรับมากกว่าเอนจินอื่นๆ

- Direct3D 8.1
- Direct3D 9.0
- OpenGL 1.2
- The Irrlicht Engine software renderer
- A null device

ลักษณะเด่นอื่นจุดที่เอนจินนี้สนับสนุนคือ ผู้พัฒนาโปรแกรม ไม่จำเป็นต้องรู้ว่า API ที่เอนจิน ใช้ติดต่อเป็นเอนจิน ตัวใด มันสามารถจัดการได้เอง ซึ่งผู้พัฒนาเพียงแค่เป็นผู้ระบุว่าจะใช้ API ตัวไหนแค่นั้นพอ และนี่คือเหตุผล 3 ประการที่ทำให้เอนจินนี้ ไม่สนใจ API ตัวใดตัวหนึ่งเท่านั้น

- 1) **Performance** เพื่อประสิทธิภาพ เนื่องจาก API แต่ละตัวจะเก่งคนละด้าน เช่น OpenGL จะแสดงผลภาพได้ดีกว่า แต่ถ้าใช้ Direct3D จะทำให้แสดงผลได้รวดเร็วกว่า เป็นต้น
- 2) **Platform independence** เนื่องจาก Direct3D จะไม่สามารถทำงานได้บน Mac และ Linux ในขณะที่ OpenGL สามารถทำงานได้ ซึ่งในกรณีของ OpenGL ก็เช่นเดียวกัน ซึ่งเพราะเหตุนี้เอง Irrlicht Engine จึงได้พยายามรองรับ API ที่หลากหลายเพื่อให้ทำงานได้บนทุก platform
- 3) **Driver problems** เป็นปัญหาบ่อยๆที่เกิดขึ้นที่ user จะต้องเจอ เมื่อใช้ 3D software ซึ่งปัญหาส่วนใหญ่เกิดจากการไม่อัปเดตของ driver และไม่สนับสนุน driver version เก่าๆ ซึ่งในเอนจินนี้จะให้ผู้ใช้ สามารถเลือก driver ที่จะใช้ได้เอง เมื่อเกิดปัญหาขึ้น

2.6.3 Platforms

Irrlicht Engine ไม่ขึ้นกับ Platform ใด Platform หนึ่ง มันจึงสามารถทำงานได้บน Platform มากกว่า 1 Platform ทั้ง

- Windows 98, ME, NT 4, 2000, XP, XP64
- Linux

Platforms อื่นๆ เช่น MacOS ได้ถูกออกแบบไว้แล้ว แต่ยังไม่ได้นำไป implement โปรแกรมเมอร์จะเขียน Code สำหรับเกมของเค้าเพียงแค่ครั้งเดียวเท่านั้น มันก็สามารถรันได้บนทุกๆ Platform โดยไม่จำเป็นต้องเปลี่ยนแปลง Code แม้แต่บรรทัดเดียว

2.6.4 Materials and Shaders

มันมีความสามารถในการสร้าง environment ที่มีความสมจริงได้อย่างรวดเร็ว ซึ่งในเอนจินนี้จะมี material จำนวนมากให้เลือกใช้

ซึ่ง material บางตัวจะมาจาก fixed function pipeline (ตัวอย่างเช่นการทำ light mapped) และบางอย่างต้องอาศัย programmable pipeline (normal mapped/parallax per pixel lighted materials for example) ซึ่งในปัจจุบัน 3D Hardware ได้มีให้แล้ว มันยังสามารถที่จะผสม materials ทั้ง 2 ประเภท ใน scene ได้โดยไม่มีปัญหาใดๆ และ เมื่อมีการใช้ material ที่ต้องการ feature ที่ Hardware ไม่สามารถทำได้

เอนจินจะจัดหา fall back materials ไว้ให้ แต่อย่างไรก็ตาม ถ้า built in materials มีไม่เพียงพอ มันจะไปใส่ที่จะทำการเพิ่ม material ใหม่ๆ เข้าไปใน irrlicht engine ในตอน runtime โดยไม่ต้องทำการ modify หรือ recompile ตัวเอนจินใดๆ ทั้งสิ้น shader languages ที่รองรับในปัจจุบันได้แก่

- Pixel and Vertex Shaders 1.1 to 3.0
- ARB Fragment and Vertex Programs
- HLSL
- GLSL

2.6.5 Scene Management

การ Render ใน Irrlicht Engine จะทำโดยการใช้ hierarchical scene graph ซึ่ง Scene nodes จะถูกยึดติดกับ nodes อื่นๆ ตามการเคลื่อนที่ของมัน และคัดเลือก children node ของพวกมัน ไปยัง the viewing frustum และสามารถทำการ collision detection บาง scene node สามารถถูกใช้ เป็นตัวอย่างของระดับชั้นของกล้อง, ภายในอาคาร, ภายนอกอาคาร ลักษณะโครงกระดูก การเคลื่อนไหวของน้ำ เป็นต้น

ในกรณีนี้ irrlicht engine สามารถทำการผสม indoor scene กับ outdoor scene ได้อย่างกลมกลืน และให้ programmer สามารถควบคุมสิ่งต่างๆ ใน scene ได้อย่างเต็มที่ อีกทั้ง Programmer ยังสามารถทำการเพิ่มเติมสิ่งต่างๆ ได้เอง เพราะว่าโปรแกรมเมอร์สามารถเพิ่ม node ของเค้าได้เอง รวมถึง mesh and texture loaders, GUI elements อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.6 Character Animation

ในปัจจุบันมี Character Animation ที่ถูกใช้อยู่ 2 ประเภท คือ

Morphing target animation : Meshes เป็นการ interpolated แบบ linear จาก frame หนึ่ง ไปยัง frame ถัดไป ซึ่งวิธีนี้ถูกใช้ในเกมตระกูล Quake ส่วนใน irrlicht engine จะใช้มัน โดยการ import file สกุล .md2

Skeletal animation : พื้นผิวจะถูกจัดการ โดยการเชื่อมต่อกัน ซึ่ง Irricht Engine จะทำ โดยการ โหลด file สกุล .ms3d หรือ .x ซึ่งมันง่ายมากที่เราจะยึดติด objects ใดๆเข้ากับโมเดลที่เรา สร้างขึ้นมา อย่างเช่น การติดอาวุธเข้ากับมือของโมเดล ซึ่งมันจะเคลื่อนที่ตามการเคลื่อนที่ของมือ ด้วย ซึ่งการยึดติด object นี้สามารถทำได้โดยการเขียน code เพียงแค่บรรทัดเดียวเท่านั้น

Programmer ไม่จำเป็นต้องรู้เกี่ยวกับการทำงานเหล่านี้ ถ้าเค้าไม่ต้องการ เค้าเพียงแค่ ทำการ โหลด file เข้าไปในเอนจินและใช้จินตนาการเพื่อวาดมันก็เพียงพอแล้ว

2.6.7 Supported Formats

มี file formats มากมายที่ได้รับการรองรับและ สามารถถูก load เข้าสู่เอนจินได้โดยตรง โดยไม่ต้องถูก Convert เพื่อเปลี่ยนเป็น format ใดๆ สำหรับการใช้งานกับ irrlicht engine ซึ่งเป็นการช่วยประหยัดเวลาที่ใช้ในการพัฒนาเกม ได้ทางหนึ่ง จำนวน file format ที่ irrlicht engine รองรับ ได้เพิ่มมากขึ้นเรื่อยๆ หากคุณต้องการให้ irrlicht engine สามารถ โหลด file format ใดๆที่ยังไม่มีการ รองรับในปัจจุบัน คุณสามารถส่งเมลล์มาคุยกับทางเราได้

Textures file formats ที่ Irrlicht Engine รองรับในปัจจุบัน คือ

- Adobe Photoshop (.psd)
- JPEG File Interchange Format (.jpg)
- Portable Network Graphics (.png)
- Truevision Targa (.tga)
- Windows Bitmap (.bmp)
- Zsoft Paintbrush (.pcx)

Mesh file formats ที่ Irrlicht Engine รองรับในปัจจุบัน คือ

- 3D Studio meshes (.3ds)
- Alias Wavefront Maya (.obj)
- Cartography shop 4 (.csm)
- COLLADA (.xml, .dae)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DeleD (.dmf)
- FSRad oct (.oct)
- Microsoft DirectX (.x)
- Milkshape (.ms3d)
- My3DTools 3 (.my3D)
- Pulsar LMTools (.lmts)
- Quake 3 levels (.bsp)
- Quake 2 models (.md2)

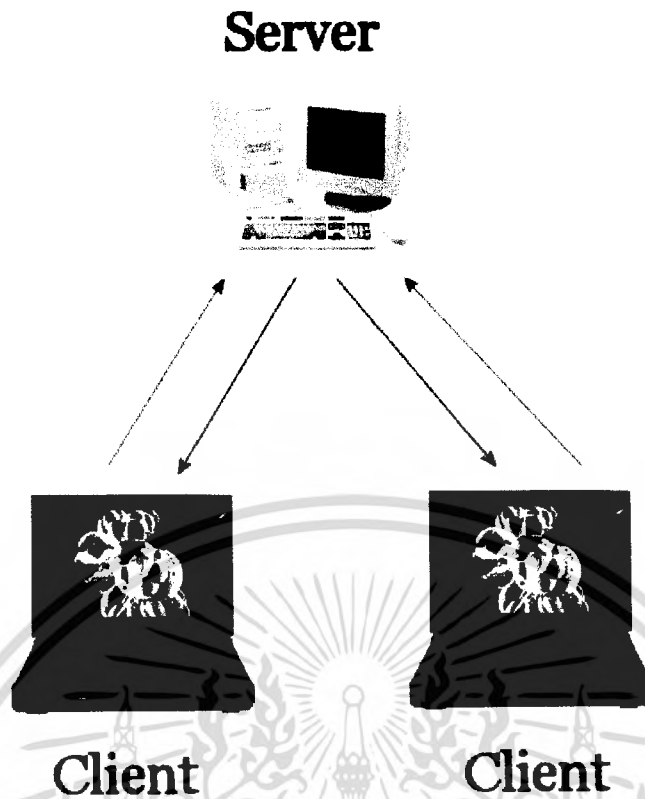
นอกจากนั้น ยังมีตัว exporter สำหรับ 3D Packages ที่เป็นที่นิยม (เช่น Blender, 3DSMax, Gile[s], ...) ซึ่งถูกรวมไว้อยู่ใน SDK แล้ว

2.7 RakNet Engine

เป็นเอพีไอที่สนับสนุนการทำงานบนเครือข่าย ทั้ง Berkeley Sockets และ Windows systems “Winsock” โดยมีจุดประสงค์เพื่อให้แอปพลิเคชันต่างๆสามารถติดต่อสื่อสารกันได้ ทั้งบนเครื่องคอมพิวเตอร์เดียวกัน และในวงแลน หรือแม้แต่บนระบบอินเทอร์เน็ต โดยเอพีไอนี้จะสนับสนุนการทำงานของเกมออนไลน์อย่างเต็มรูปแบบ โดยมีความสามารถเด่นหลายประการด้วยกัน ได้แก่

- มีประสิทธิภาพสูง โดยสามารถส่งรับข้อมูลจำนวนมากได้ โดยใช้แบนด์วิดท์เพียงเล็กน้อย
- มีความเร็วสูง
- มีฟังก์ชันพื้นฐานที่จำเป็นๆรองรับไว้อยู่แล้ว

ระบบบนเครือข่ายสามารถแบ่งประเภทออกได้เป็น 2 ประเภทหลักๆ ได้แก่ แบบเครื่องต่อเครื่อง (Peer to Peer) และระบบไคลเอนท์เซิร์ฟเวอร์ (Client / Server) ซึ่งทั้ง 2 ระบบนี้จะมีวิธีการ Implement ที่แตกต่างกันออกไป ทั้งวิธีการและโปรโตคอล (Protocol) ที่ใช้ โดย RakNet จะสนับสนุนการทำงานของระบบทั้ง 2 แบบนี้อย่างเต็มรูปแบบ



รูปที่ 2.23 แสดงการทำงานแบบไคลเอนท์เซิร์ฟเวอร์

โดยปกติแล้วระบบ Client / Server จะมีการทำงานที่มีประสิทธิภาพสูงกว่าระบบ

Peer to Peer

ซึ่งปัจจุบันมีการเข้ารหัสของแพ็คเกจ (Packet) หลากหลายรูปแบบด้วยกัน โดยใช้โปรโตคอลทั้ง UDP และ TCP โดยทั้ง UDP และ TCP ก็จะมีข้อดีแตกต่างกันออกไป

- TCP มีข้อดีคือ เป็นโปรโตคอลที่ใช้ในการส่งไฟล์ที่ดี แต่จะทำงานได้ไม่ดีเมื่อใช้ส่งข้อมูลของเกมที่มีลักษณะเป็นสตรีม (Stream) ดังนั้นเวลาส่งข้อมูล ผู้พัฒนาจำเป็นต้องแบ่งข้อมูลออกเป็นแพ็คเกจเอง

- UDP เป็นโปรโตคอลที่เหมาะสมสำหรับการใช้งานกับเกมมากกว่า TCP เพราะข้อมูลจะมีการจัดการการส่งแบบเป็นแพ็คเกจทำให้ง่ายต่อการแบ่งข้อมูลสำหรับส่ง แต่อย่างไรก็ตาม UDP ก็ยังคงมีปัญหาเช่นกัน ได้แก่

- ไม่สามารถรับรองได้ว่าแพ็คเกจที่ส่งจะถึงผู้รับ
- ไม่มีการรับรองลำดับก่อนหลังของการส่งข้อมูล
- UDP มีการรับรองความถูกต้องของข้อมูล แต่ไม่มีการควบคุมหรือป้องกันการคุกคามของแฮกเกอร์ หรือการลักลอบการเปลี่ยนแปลงข้อมูลระหว่างส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่จำเป็นต้องมีการเชื่อมต่อก่อนก็สามารถทำงานได้ ซึ่งเหมือนจะดีในความเป็นจริงแล้วหากไม่มีการป้องกันในเรื่องนี้จะเสี่ยงต่อการถูกแฮก
- การส่งข้อมูลจะไม่มีกระบวนการควบคุมกระบวนส่ง (Flow Control) ซึ่งเป็นเหตุให้ในบางครั้งส่งข้อมูลได้อย่างไม่มีประสิทธิภาพ

จากที่กล่าวมาทั้งหมดจะเห็นข้อดีและข้อเสียของทั้ง TCP และ UDP ซึ่ง RakNet ได้นำข้อดีของทั้ง 2 โพรโตคอลมารวมไว้ด้วยกันโดยจะจัดการปัญหาเบื้องต้นของ UDP ให้ทำให้ผู้พัฒนาโปรแกรมไม่จำเป็นต้องจัดการเรื่องเหล่านี้เอง โดยสิ่งหลักๆที่ RakNet จะจัดการให้ ได้แก่

- การส่งข้อมูลไปยังผู้รับ จนกว่าข้อมูลจะถึงผู้รับ
 - มีการจัดการเรื่องลำดับการส่งแพคเกจให้อัตโนมัติ เพื่อเพิ่มประสิทธิภาพการทำงาน
 - มีระบบการป้องกันการเปลี่ยนแปลงข้อมูลระหว่างส่ง หากมีการแก้ไขก่อนถึงมือผู้รับจะ
 - มีการแจ้งข้อผิดพลาดก่อน
 - มีการจัดการเรื่องความเร็วในการส่ง และป้องกันการคุกคามจากภายนอก
 - มีการจัดการเรื่องการควบคุมกระบวนส่งข้อมูล (Flow Control)
- นอกจากนี้ RakNet ยังมีข้อดีเหนือกว่า API ทั่วไป ได้แก่

- RakNet จะเพิ่มข้อมูลการควบคุมเพียงไม่กี่ไบต์ลงในชุดข้อมูลที่ส่ง เมื่อเทียบกับ API อื่นๆ
- RakNet จะนำการใช้งาน แบนวิดธ์ ที่ไม่จำเป็นออก
- RakNet สามารถทำการเชื่อมต่อและยกเลิกการเชื่อมต่อได้อย่างรวดเร็ว
- RakNet มีความสามารถในการจัดการกับปัญหาที่เกิดขึ้นได้เป็นอย่างดี
- ง่ายต่อการใช้งาน

ความสามารถหลักของ RakNet

ความสามารถหลักของ RakNet สนับสนุนทั้งระบบ Peer to Peer และ Client / Server โดยการเรียกใช้จากฟังก์ชันที่ RakNet เตรียมไว้ให้

- IO Completion ports

ในระบบปฏิบัติการวินโดวส์ จะมีกลไกในการเรียกใช้ IO Port กับการ์ดการเชื่อมต่อ (Network Card) ซึ่งสามารถเขียนและอ่านข้อมูลจากหน่วยความจำได้โดยตรง ซึ่งความนี้จะคอยจัดการกับเทรด (Thread) ที่มีการร้องขอข้อมูลขณะที่กำลังรออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Waiting) ซึ่งเทรดเหล่านี้จะถูกเก็บอยู่ใน คอมมอนพูล (Common pool) ซึ่งทั้งไคลเอนท์ และเซิร์ฟเวอร์จะใช้พูลร่วมกัน (Share Pool) ทำให้จัดการกับไคลเอนท์จำนวนมากได้อย่างมีประสิทธิภาพ

- **Remote Procedure Calls**

ทำให้สามารถเรียกฟังก์ชันต่างๆบนระบบอื่นๆได้ โดยเรียกใช้จากบิตสตรีม (BitStream) โดยสามารถใช้กับข้อมูลได้ทุกขนาดทุกประเภท ซึ่งการใช้ RPC นี้ทำให้จัดการกับแพคเกจได้ง่ายกว่าการที่จะต้องมาสร้างตัวแยกประเภทแพคเกจเอง

- **Timestamping**

ไทม์แสตมป์ (Timestamps) คือการแปะชุดข้อมูลบางอย่างติดไปกับแพคเกจที่ส่ง เพื่อให้ทราบความถูกต้องของการส่งเมื่อข้อมูลไปยังผู้รับ รวมทั้งยังควบคุมความแตกต่างของคล็อก (Clock) บนเครื่องอื่นๆบนระบบได้อีกด้วย

- **Global Data Compression**

RakNet สามารถวิเคราะห์การจราจรบนเครือข่าย และสามารถบีบอัดข้อมูลที่ส่งผ่านเครือข่ายให้มีขนาดเล็กลงเพื่อให้ข้อมูลส่งได้เร็วขึ้น แต่การทำแบบนี้จะทำให้ฝั่งผู้รับและฝั่งผู้ส่งต้องใช้ CPU ในการประมวลผลมากขึ้นในการถอดรหัส

- **Automatic Memory Synchronization**

ทำหน้าที่ตรวจสอบหาข้อมูลในหน่วยความจำ ซึ่งหากมีการเปลี่ยนแปลงจะมีการจัดการโดยการส่งแพคเกจไปยังรีโมทซิสเต็ม (Remote System) เพื่อแก้ไขข้อมูลให้ถูกต้องตรงกัน

2.8 Audiere Engine

ออดีเยร์ (Audiere) คือ แอปพลิเคชันโปรแกรมอินเตอร์เฟซ (Application Program Interface) ทางด้านเสียงระดับสูง มันสามารถเล่นไฟล์ได้หลายรูปแบบ ไม่ว่าจะเป็นไฟล์

- Ogg Vorbis
- MP3
- FLAC
- uncompressed WAV
- AIFF
- MOD
- S3M
- XM
- IT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ ออดีโอ เอาต์พุต (audio output) ของ ออเดียร์ สนับสนุน ไดรেকซาวด์ (DirectSound) หรือ วินเอ็มเอ็ม (WinMM) บน ระบบปฏิบัติการ วินโดวส์ (Windows), โอเอสเอส (OSS) บนลินุกซ์ (Linux) และ ซิกวิน(Cygwin) และ เอสจีแอล เอแอล (SGI AL) บน อิริก (IRIX)

ออเดียร์ เป็น โอเพน ซอส (Open Source) ที่มี ลิขสิทธิ์ แบบ แอลจีพีแอล (LGPL) ซึ่งหมายถึง สามารถใช้ในทางธุรกิจได้ แต่ ไม่อนุญาตให้เปลี่ยนแปลง ซอสโค้ด (Source Code) ถ้าคุณต้องการ เปลี่ยนแปลง จะต้องชำระเงิน

ออเดียร์ สามารถทำงานได้บนหลายระบบ ไม่ว่าจะเป็น วินโดวส์, ลินุกซ์ ไอสามแปดหก (Linux-i386), ซิกวิน และ อิริก ด้วย คอมไพเลอร์ (Compiler) หลัๆ อย่างน้อย 3 ตัว

ออเดียร์ ไม่ขึ้นกับระบบปฏิบัติการ ดังนั้นการทำงานในสถาปัตยกรรมที่ต่างๆ กัน ทำได้โดย การเปลี่ยนแปลง โค้ด (Code) เพียงเท่านั้น

คุณสมบัติ

- เป็น แอปพลิเคชัน โปรแกรมอินเตอร์เฟซ ที่ใช้งานได้ง่าย
- สนับสนุนรูปแบบไฟล์ แบบ Uncompressed WAV*, Uncompressed AIFF*, Ogg Vorbis*, FLAC*, MP3, MOD, S3M, IT, XM (* สนับสนุนการ seeking)
- สนับสนุนการ สตรีมมิ่ง (Streaming) และ บัฟเฟอร์ (buffered) เกี่ยวกับเสียง
- สนับสนุนการปรับแต่ง เปลี่ยนแปลง และ ขยายความถี่ของเสียง
- สนับสนุน สตรีมของ ไฟล์ ที่กำหนดขึ้นเอง
- สนับสนุน Python, Delphi, Java, XPCOM (JavaScript in Mozilla) ไบนารี (bindings)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนการดำเนินงานวิจัย

ลำดับขั้นตอนในการดำเนินงานวิจัยทั้งหมดแบ่งออกเป็น 3 ส่วน ได้แก่ ส่วนของการวิเคราะห์และออกแบบเกม ส่วนของการลงมือปฏิบัติจริง และส่วนของการทดสอบเกม

1) ขั้นตอนการวิเคราะห์และออกแบบเกม

เป็นการวิเคราะห์ความต้องการความต้องการจะสร้างเกมในรูปแบบใด ไม่ว่าจะ เป็น เนื้อหาภายในเกม กติกาการเล่น ระบบการเล่น แนวทางการเล่น รูปแบบตัวละคร ฉาก เสียง เนื้อหาของเกม

2) ขั้นตอนการลงมือปฏิบัติการจริง

จะเป็นการเลือกฉากที่จะนำมาใช้จริงในเกม โดยฉากและตัวละครภายในเกม จะทำการอิมพอร์ตเข้ามาจากแหล่งอื่น

3) ขั้นตอนการพัฒนาโปรแกรม

- ด้านการพัฒนาโปรแกรมจะเกี่ยวข้องกับการเขียนคำสั่งต่างๆ โดยมีการนำ จากตัวละคร และเสียงประกอบ ที่ได้สร้างไว้มาใช้
- วางโครงสร้างของ โปรแกรม และการทำงานของ เกม
- กำหนดสถานะต่างๆของเกม

3.1 ขั้นตอนการออกแบบและวิเคราะห์เกม

3.1.1 การออกแบบรูปแบบ กติกาการเล่น และระบบเกม

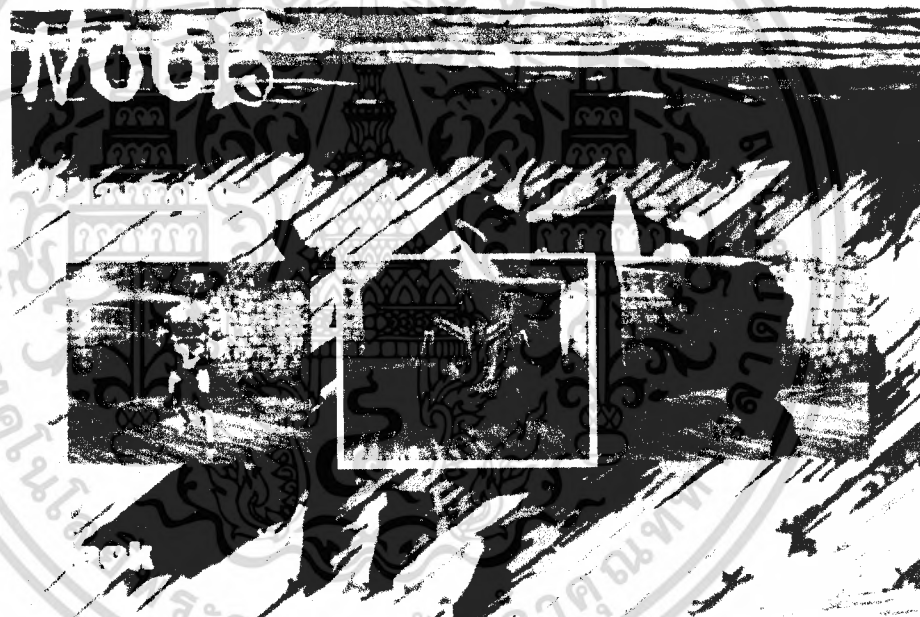
NooB (นู๊ป) เป็นเกมแนวเดินยิง 3 มิติมุมมองบุคคลที่ 1 ซึ่งสนับสนุนทั้งการเล่นคนเดียวโดยมีคู่ต่อสู้เป็นระบบคอมพิวเตอร์ และผู้เล่นจริงๆ โดยมีจุดมุ่งหมายคือล้มฝ่ายตรงข้ามให้ได้ โดยในการเล่นคนเดียวจะมีรูปแบบในการเล่นคือขั้นแรกผู้เล่นต้องทำการเลือกฉากที่ต้องการจะทำจะเล่นก่อน โดยตัวเกมในโหมดการเล่นคนเดียว จะมีเป้าหมายคือต้องทำลายคู่ต่อสู้ในฉากให้หมด โดยที่เราจะต้องไม่ถูกคู่ต่อสู้ยิงจนพรั่งหมดเสียก่อน โดยวิธีการทำลายคู่ต่อสู้ในฉากทำได้โดย ทำการเลื่อนเป้ายิงที่อยู่กลางหน้าจอไปยังคู่ต่อสู้ในฉาก แล้วทำการคลิกเมาส์ปุ่มซ้ายเพื่อยิง ถ้ากระสุนหมดสามารถทำการเติมกระสุนได้โดย คลิกเมาส์ปุ่มขวา จำนวนคู่ต่อสู้ในแต่ละระดับความยากจะมีจำนวนไม่เท่ากัน โดยถ้าระดับความยากยิ่งมาก

เท่าไรหรือจำนวนคู่ต่อสู้ในฉากนั้นๆ ก็จะมากขึ้นตามลำดับ ซึ่งเราต้องทำการค้นหาในฉากเพื่อทำการทำลายคู่ต่อสู้ให้หมด

การเล่นกับผู้เล่นจริงๆ ผ่าน ระบบเครือข่ายคอมพิวเตอร์ จะมีรูปแบบในการเล่นคือ ขั้นแรกผู้เล่นต้องทำการเลือกตัวละครที่ต้องการจะเล่นเสียก่อน โดยเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นโฮสต์จะต้องทำการเลือกฉากที่ต้องการจะเล่นด้วย โดยเป้าหมายของเกมในโหมดนี้คือต้องทำการทำลายผู้เล่นทุกคนในฉากให้หมด โดยคนที่เหลือรอดเป็นคนสุดท้ายจะเป็นผู้ชนะ

3.1.1.1 การเลือกตัวละคร (Character Selection)

ภายในเกมจะมีตัวละครให้เลือก 3 ตัว ได้แก่ ครอบ ครอบเคก้า และ ฮันเตอร์ ซึ่งตัวละครทั้งสามตัว จะมีอาวุธ และความเร็วในการเคลื่อนที่ ที่แตกต่างกันออกไป



รูปที่ 3.1 แสดงหน้าจอเลือกตัวละคร

3.1.1.2 การเลือกฉาก (Map Selection)

หลังจากเลือกตัวละครเสร็จ จะเข้าสู่การเลือกฉาก โดยภายในเกมจะมีสามฉากให้เลือก ซึ่งแต่ละฉากจะมีความซับซ้อนของทางเดินแตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงหน้าจอเลือกแผนที่

3.1.2 การออกแบบภาพและกราฟิก

รูปแบบของภาพในเกม มีลักษณะเป็น 3 มิติ โดยจะพยายามใช้แสงเงา และ เอฟเฟกต์ต่างๆที่ Irrlicht Engine มีเตรียมมาให้ เพื่อให้ภาพที่ออกมาดูสมจริงที่สุด ซึ่งตัวละครอาวุธ แผนที่ และองค์ประกอบอื่นๆจะใช้โปรแกรม 3ds max 7 ช่วยในการออกแบบ แต่ในส่วนการออกแบบหน้าจอจะใช้โปรแกรม Adobe Photoshop CS และ Adobe Illustrator CS เข้าช่วย

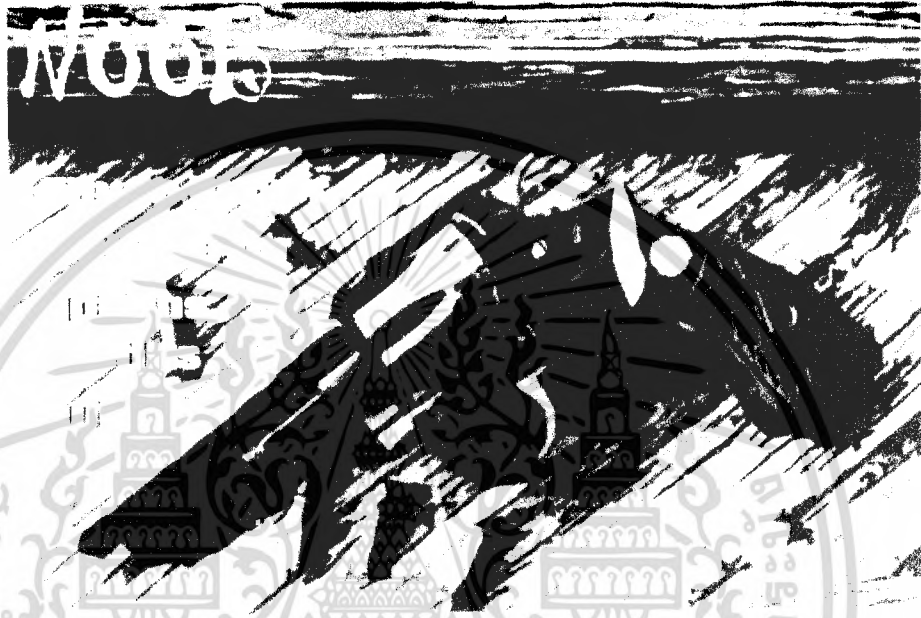
3.1.3 การออกแบบเสียงและซาวนด์เอฟเฟกต์ประกอบ

ไฟล์เสียงที่ใช้เป็นเอฟเฟกต์ในเกมมี 2 ประเภทคือ WAVE และ MP3 ส่วนไฟล์เพลงเป็นแบบ MP3 โดยเสียงเอฟเฟกต์จะเสียงประกอบภายในเกม เช่น เสียงยิงปืน เสียงตะโกน เสียงฆ่าตัว เสียงระเบิด เป็นต้น ส่วนเสียงประกอบจะเป็นทำนองเพลงประกอบขณะเล่นเกม เพื่อเพิ่มบรรยากาศในการเล่น โดยโปรแกรมที่นำมาใช้สำหรับการสร้างและตัดต่อไฟล์เสียงและซาวนด์เอฟเฟกต์ ได้แก่ Nero Wave Editor และ Adobe Audition

3.2 ขั้นตอนการลงมือปฏิบัติจริง

3.2.1 รูปแบบหน้าจอในสถานการณ์ต่างๆ

รูปแบบหน้าจอภายในเกมจะแบ่งได้ออกเป็นหลายๆส่วนด้วยกัน ไม่ว่าจะเป็น หน้าจอเข้าเกม หน้าจอรอโหลดเกม หน้าจอเมนู หรือหน้าจอแสดงผลอื่นๆ



รูปที่ 3.3 แสดงหน้าจอเมนูหลักของเกม

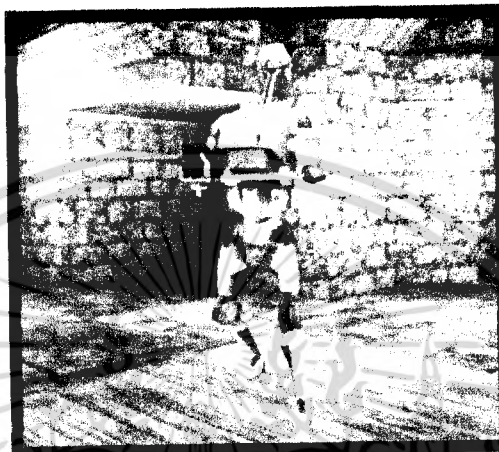
3.2.2 สร้างโมเดลที่ใช้ในเกมทั้งหมด

ส่วนประกอบต่างๆภายในเกมล้วนแต่โมเดลที่สร้างขึ้นทั้งสิ้น ซึ่งโมเดลเหล่านี้ส่วนใหญ่สร้างมาจากโปรแกรม 3ds max 7 ประกอบกับการนำโมเดลที่มีอยู่แล้วมาประยุกต์ใช้ โดยโมเดลภายในเกมจะประกอบด้วย 3 ส่วนหลักด้วยกัน ได้แก่ โมเดลตัวละครในเกม โมเดลแผนที่ โมเดลอาวุธและอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 โมเดลตัวละครในเกม (Character)

ตัวละครภายในเกมเหมือนอย่างที่กล่าวเอาไว้ข้างต้น คือผู้เล่นจะสามารถเลือกตัวละครได้ ซึ่งโมเดลที่จะนำเข้ามาใช้ในเกมนี้ไฟล์สกุล md2 และ pk3 ของเกมควาก (Quake)



รูปที่ 3.4 แสดงโมเดลทรอยด์



รูปที่ 3.5 แสดงโมเดลทรอยด์เก้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงโมเดลฮันเตอร์

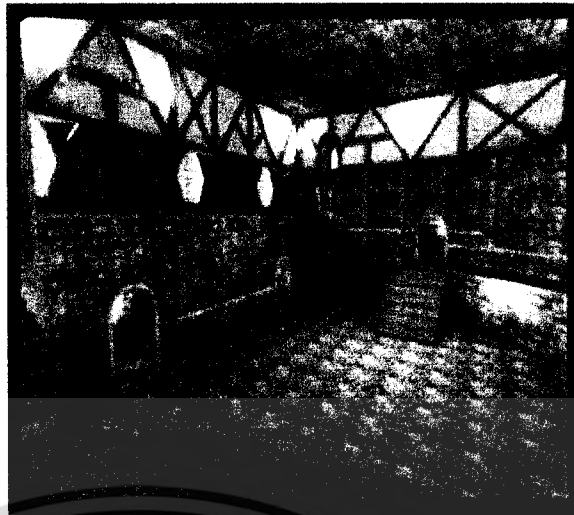
3.2.2.2 โมเดลแผนที่ (Map)

ผู้เล่นสามารถเลือกแผนที่ที่ต้องการเล่นได้อย่างอิสระ ซึ่งแผนที่ในเกมนี้ จะใช้แผนที่ของเกม Quake 3 ซึ่งจะมี Irrlicht Engine เป็นผู้สนับสนุนการทำงาน



รูปที่ 3.7 แสดงโมเดลแผนที่ที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงโมเดลแผนที่ที่ 2



รูปที่ 3.9 แสดงโมเดลแผนที่ที่ 3

3.3 ขั้นตอนการเขียนโปรแกรม

ในขั้นตอนการเขียนโปรแกรมนั้น ได้ใช้ Visual C++ .NET สำหรับการพัฒนาโดยใช้ Engine ได้แก่

- Irrlicht Engine สำหรับดูแลเรื่องกราฟฟิก (Graphic Engine)
- RakNet Engine สำหรับดูแลเรื่องเน็ตเวิร์ค (Network Engine)
- Audiere Engine สำหรับดูแลเรื่องเสียง (Sound Effect Engine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 คลาสหลักที่ใช้ในโปรแกรม

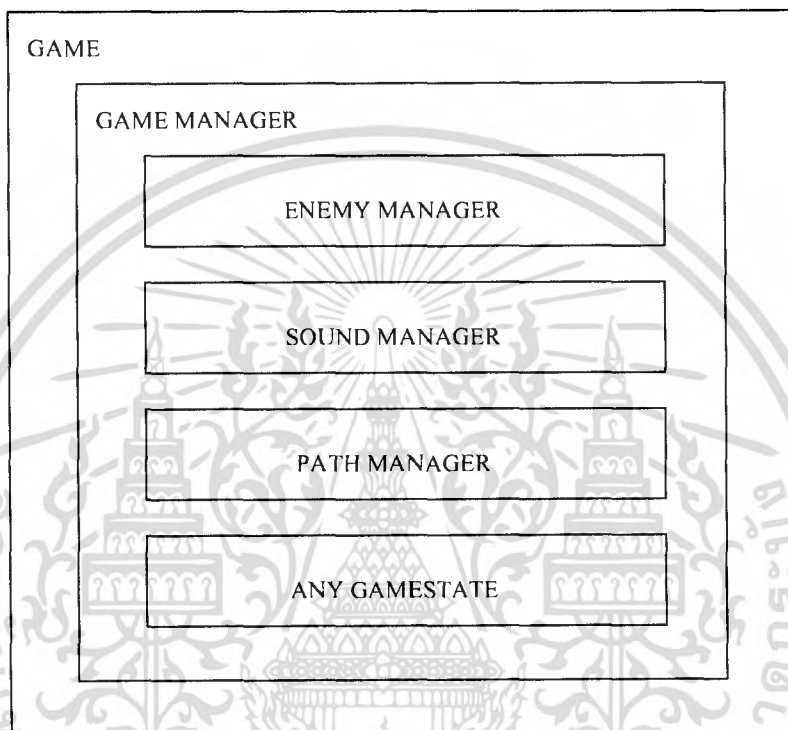
ตาราง 3.1 คลาสหลักที่ใช้ในโปรแกรม

ชื่อคลาส	บทบาท และหน้าที่
Game.cpp	เป็นคลาสที่ครอบทุกอย่างไว้ โดยมีหน้าที่เริ่มการทำงานของเกม
GameCharacterManager.cpp	ดูแลเรื่องการเลือกตัวละคร
GameEnemy.cpp	บ่งบอกถึงข้อมูลและบทบาท ศัตรูตัวในเกม
GameEnemyManager.cpp	ดูแลเรื่องการจัดการกับศัตรู เช่นการเพิ่มศัตรู หรือลบ ศัตรูออกจากเกม
GameEnemyMultiManager.cpp	เช่นเดียวกับ GameEnemyManager.cpp แต่ดูแลในโหมด Multiplayer
GameEnemyOwnedStates.cpp	ดูแลเรื่องการเปลี่ยนสถานะต่างๆของตัวละคร เช่น การเดิน การยิง การตาย
GameEnemyOwnedMultiStates.cpp	เช่นเดียวกับ GameEnemyOwnedStates.cpp แต่ดูแลในโหมด Multiplayer
GameEntity.cpp	เป็นคลาสแม่ ที่บ่งบอกลักษณะออบเจกต์ทุกอย่างในเกม
GameManager.cpp	เป็นคลาสที่สำคัญที่สุด โดยจะดูแลการทำงานของเกมนทั้งระบบ
GameMapState.cpp	เป็นคลาสในขั้นตอนการเลือกแผนที่
GameOptionsState.cpp	ควบคุมระดับความยากของ และระดับเสียงของเกม
GamePather.cpp	ดูแลเรื่องการทำงานของระบบ AI (A* Path finding Algorithm)
GamePathManager.cpp	ดูแลเฉพาะด้านการเรียกใช้งานจาก GamePather.cpp
GamePlayer.cpp	บ่งบอกถึงข้อมูลและบทบาท ผู้เล่นในเกม
GamePlayState.cpp	ควบคุมการทำงานระหว่างเล่นเกม ไม่ว่าจะเป็นการ กดคีย์ หรือคลิกเมาส์
GamePlayMultiState.cpp	ควบคุมการทำงานระหว่างเล่นเกม ในโหมดมัลติเพลย์เยอร์
GameSoundManager.cpp	ดูแลและควบคุมการทำงานของเสียง และเอฟเฟกต์
GameState.cpp	เป็นคลาสแม่ที่เกือบทุกคลาสจะต้องสืบทอดจากคลาสนี้ เพื่อให้คลาสอื่นๆมี คุณสมบัติในการเปลี่ยนสถานะได้
GameStateLevel01.cpp	เป็นคลาสที่มีข้อมูลของ Waypoint และการทำงานในแผนที่ที่ 1
GameStateLevel02.cpp	เป็นคลาสที่มีข้อมูลของ Waypoint และการทำงานในแผนที่ที่ 2
GameStateLevel03.cpp	เป็นคลาสที่มีข้อมูลของ Waypoint และการทำงานในแผนที่ที่ 3
GameWeapon.cpp	เป็นข้อมูลของอาวุธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคลาส	บทบาท และหน้าที่
GameWeaponOwnedStates.cpp	ดูแลการเปลี่ยนสถานะของอาวุธ { Idle , Attack }
NoobClient.cpp	เป็นคลาสที่ทำงานเมื่อเข้าสู่โหมดมัลติเพลเยอร์ แบบเป็นไคลเอนท์
NoobServer.cpp	เป็นคลาสที่ทำงานเมื่อเข้าสู่โหมดมัลติเพลเยอร์ แบบเป็นเซิร์ฟเวอร์

โดยมีแผนภาพการทำงานหลักๆดังนี้



รูปที่ 3.10 โครงสร้างการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ฟังก์ชันการทำงานหลักในแต่ละ Engine

ตาราง 3.2 ฟังก์ชันการทำงานหลักในแต่ละ Engine

IRRLICHT ENGINE	
ชื่อฟังก์ชัน	หน้าที่การทำงาน
IrrlichtDevice::createDevice()	เป็นฟังก์ชันหลักที่จำเป็นที่สุดของการใช้ Irrlicht Engine โดยจะเป็นการกำหนดข้อมูลเบื้องต้นของการใช้งาน ไม่ว่าจะเป็นประเภทของการแสดงผล และขนาดหน้าจอ
IVideoDriver:: getVideoDriver()	เป็นฟังก์ชันเรียก Video Driver ของ Irrlicht ซึ่งจะจัดการการแสดงผล ทั้งยังใช้ระบุจุดเริ่มต้น และจุดสุดท้ายของการแสดงผล
ISceneManager:: getSceneManager()	เป็นฟังก์ชันเรียก Scene Manager ของ Irrlicht ซึ่งจะมีหน้าที่จัดการและควบคุมออบเจกต์อะไรก็ตามที่ใส่เข้ามาในฉาก (Scene) เช่น ตัวละคร แผนที่ หรือวัตถุต่างๆ เป็นต้น
IGUIEnvironment:: getGUIEnvironment()	เป็นฟังก์ชันเรียก Scene Manager ของ Irrlicht ซึ่งจะมีหน้าที่จัดการและควบคุม GUI ที่ใส่เข้ามาในฉาก เช่น EditBox หรือปุ่มต่างๆ เป็นต้น
IAnimatedMesh:: getMesh()	เป็นการกำหนดโมเดลที่เราจะใช้
IAnimatedMeshSceneNode:: addAnimatedMeshSceneNode()	เป็นฟังก์ชันสร้างโหนดจาก Mesh ที่เราได้กำหนดไว้
IAnimatedMeshSceneNode:: setMaterialTexture()	เป็นการเซต Texture ให้กับโหนดที่เราสร้างขึ้น
IrrlichtDevice:: getFileSystem()::addZipFileArchive()	เป็นคำสั่งเพิ่มไฟล์ต่างๆเข้าระบบ ไฟล์ของ Irrlicht เพื่อให้สามารถเรียกใช้ ได้ภายหลัง
ISceneManager:: addCameraSceneNode()	เป็นคำสั่งกำหนดมุมมองกล้อง
ISceneManager:: createFlyStraightAnimator()	เป็นฟังก์ชันที่กำหนดให้โมเดลเคลื่อนไหวยจากจุดหนึ่งไปยังอีกจุดหนึ่งเป็นเส้นตรง รวมทั้งกำหนดความเร็วในการเคลื่อนที่ด้วย
ISceneManager:: createCollisionResponseAnimator()	เป็นฟังก์ชันที่ใช้ตรวจสอบ (Detect) วัตถุรอบข้าง เกี่ยวกับการชน (Collision)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

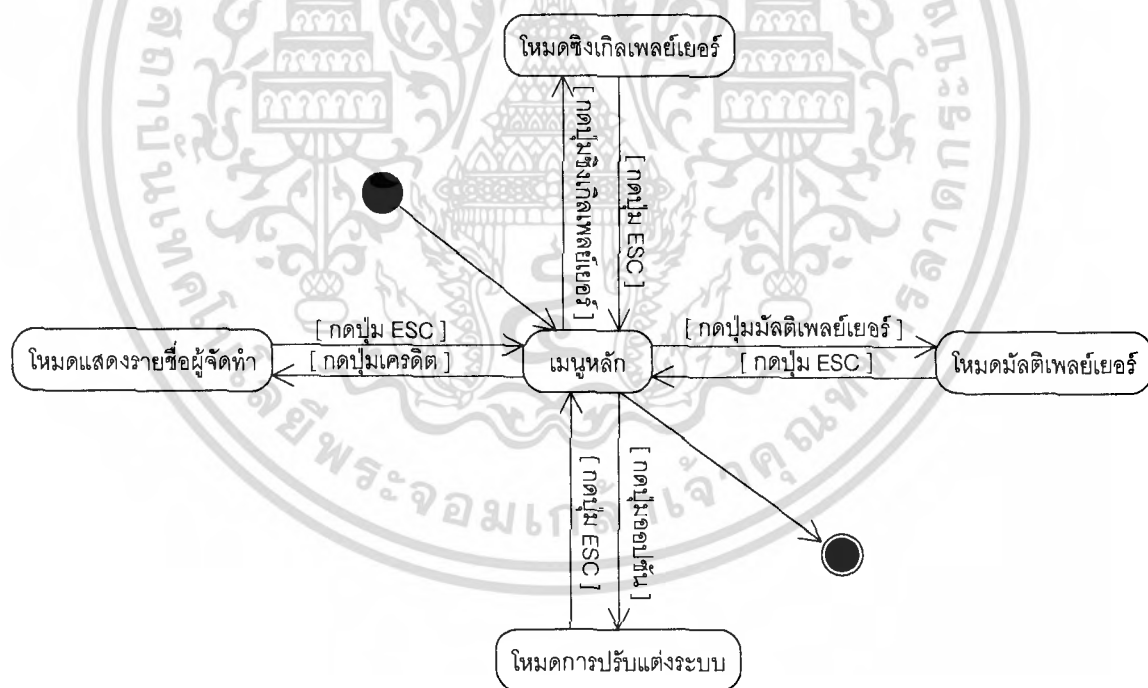
RAKNET ENGINE	
ชื่อฟังก์ชัน	หน้าที่การทำงาน
RakClientInterface:: GetRakClientInterface()	เป็นฟังก์ชันที่เก็บรวบรวมเอา หน้าที่การทำงานทั้งหมดของเครื่องที่เป็นไคลเอนท์ไว้ด้วยกัน
RakServerInterface:: GetRakServerInterface()	เป็นฟังก์ชันที่เก็บรวบรวมเอา หน้าที่การทำงานทั้งหมดของเครื่องที่เป็นเซิร์ฟเวอร์ไว้ด้วยกัน
RakClientInterface::Start()	เป็นฟังก์ชันที่สั่งให้ไคลเอนท์อินเทอร์เน็ตเฟสทำงาน โดยจะต้องระบุพอร์ตที่ใช้ทั้งฝั่งไคลเอนต์และเซิร์ฟเวอร์ รวมทั้ง IP Address ของฝั่งเซิร์ฟเวอร์ด้วย
RakServerInterface::Start()	เป็นฟังก์ชันที่สั่งให้เซิร์ฟเวอร์อินเทอร์เน็ตเฟสทำงาน โดยจะต้องระบุพอร์ตทางฝั่งเซิร์ฟเวอร์ไว้ด้วย
RakClientInterface::Receive()	เป็นฟังก์ชันคอยดักจับแพคเกจที่เซิร์ฟเวอร์ส่งมา
RakServerInterface::Receive()	เป็นฟังก์ชันคอยดักจับแพคเกจที่ไคลเอนท์ส่งมา
RakClientInterface::Disconnect()	ระงับการเชื่อมต่อของฝั่งไคลเอนท์
RakServerInterface::Disconnect()	ระงับการเชื่อมต่อของฝั่งเซิร์ฟเวอร์
RakClientInterface:: DestroyRakClientInterface()	ยกเลิกการทำงานของไคลเอนท์อินเทอร์เน็ตเฟส
RakServerInterface:: DestroyRakServerInterface()	ยกเลิกการทำงานของเซิร์ฟเวอร์อินเทอร์เน็ตเฟส
BitStream::Write()	เป็นคำสั่งเขียนข้อมูลลงแพคเกจที่จะส่ง
BitStream::Read()	เป็นคำสั่งอ่านข้อมูลจากแพคเกจที่ได้รับ
RakClientInterface::Send()	เป็นคำสั่งที่ใช้ส่งแพคเกจที่ได้สร้างไว้ของฝั่งไคลเอนท์
RakServerInterface::Send()	เป็นคำสั่งที่ใช้ส่งแพคเกจที่ได้สร้างไว้ของฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUDIERE ENGINE	
ชื่อฟังก์ชัน	หน้าที่ การทำงาน
AudioDevicePtr::AudioDevicePtr()	เป็นคำสั่งเริ่มการทำงานของ Audiere Engine
OutputStream::OpenSound()	เป็นคำสั่งระบุเสียงที่เราต้องการเปิด
OutputStream::setVolume()	เป็นฟังก์ชันเซตระดับเสียง
OutputStream::play()	เป็นฟังก์ชันให้เริ่มทำการเล่นเสียงที่ระบุไว้
OutputStream::setRepeat()	เป็นฟังก์ชันที่บอกว่าเสียงที่เล่นนั้น เมื่อเล่นจบให้เล่นซ้ำหรือไม่
OutputStream::stop()	เป็นฟังก์ชันเพื่อหยุดเล่นเสียง

3.3.3 สถานะของเกม (State Diagram)

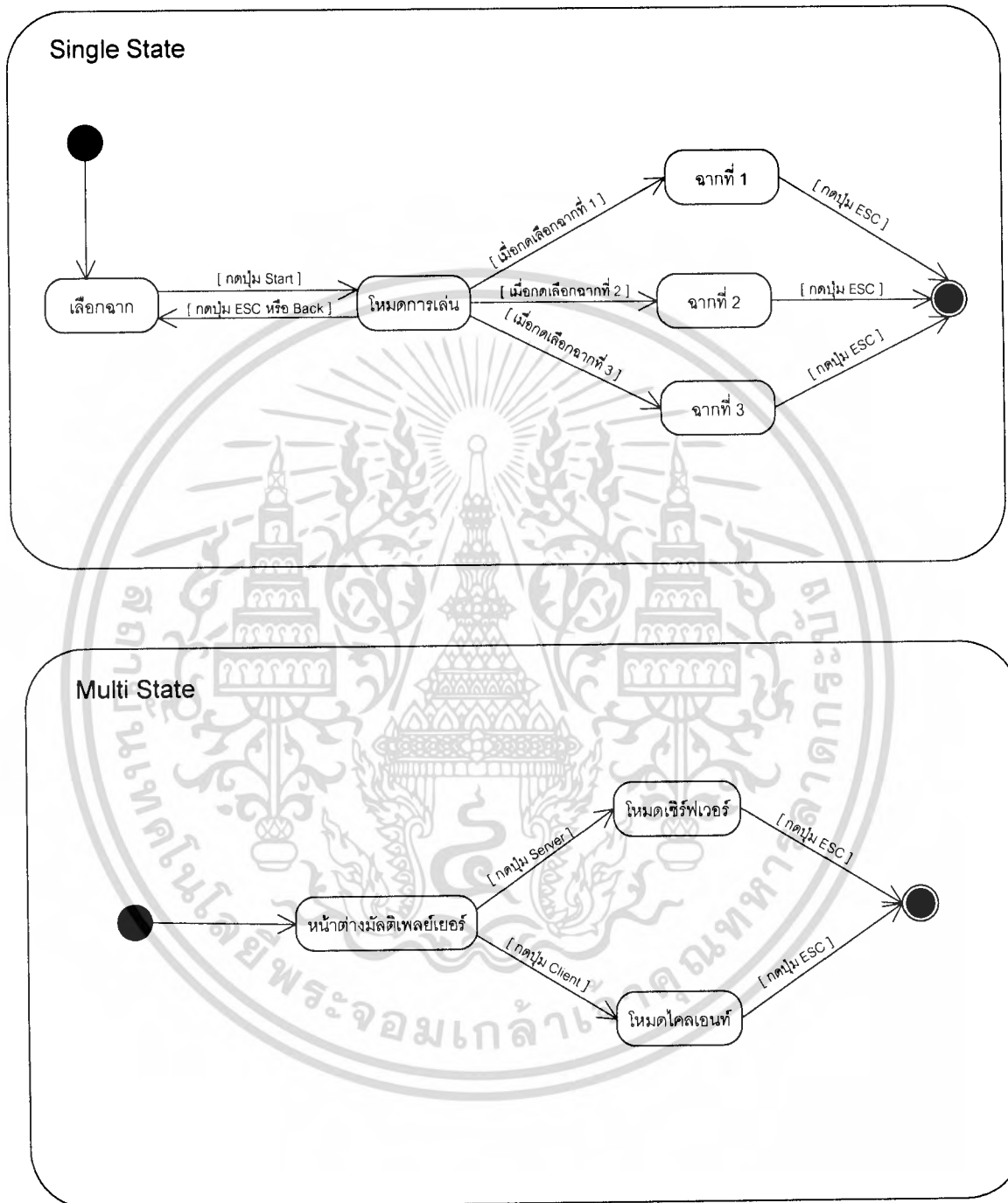
State Diagram ของเมนูหลัก



รูปที่ 3.11 แสดง State Diagram ของเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

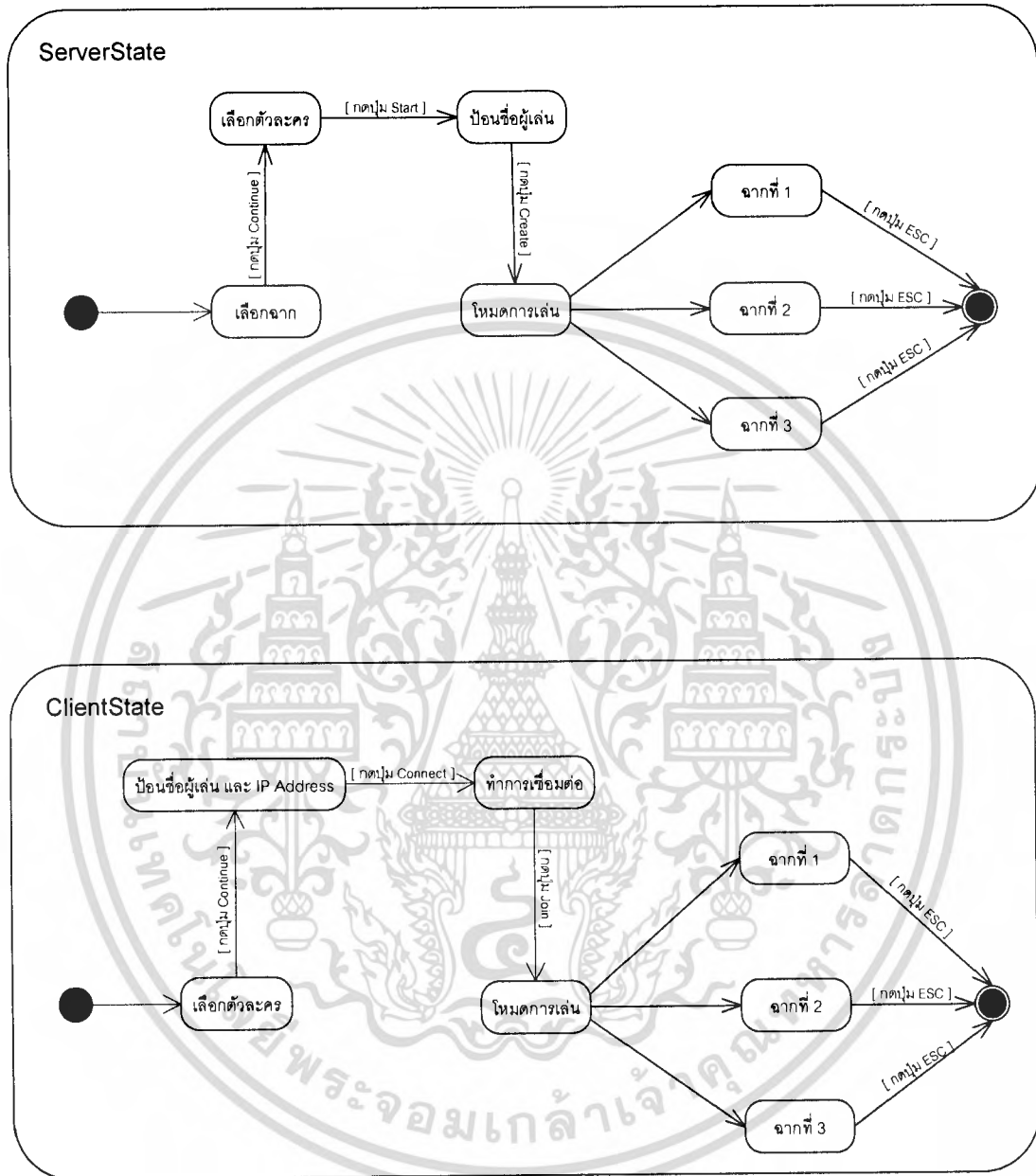
State Diagram ของ Single State และ Multi State



รูปที่ 3.12 แสดง State Diagram ของระบบซิงเกิลเพลย์เยอร์ และมัลติเพลย์เยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

State Diagram ของ ServerState และ ClientState

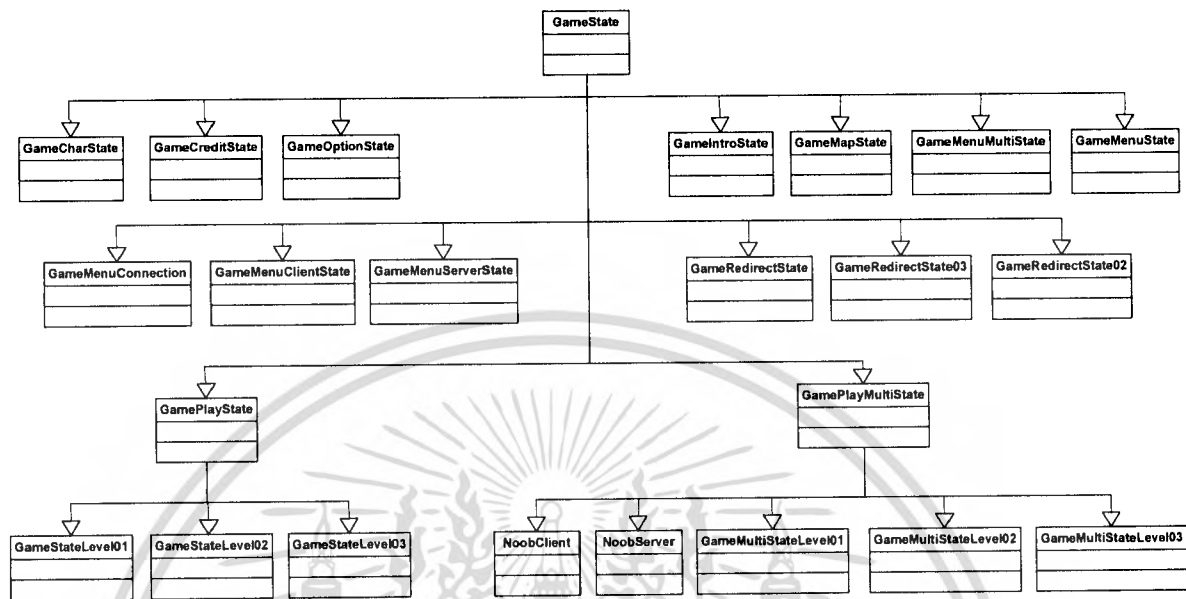


รูปที่ 3.13 แสดง State Diagram ของ ServerState และ ClientState

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การสืบทอดคุณสมบัติ

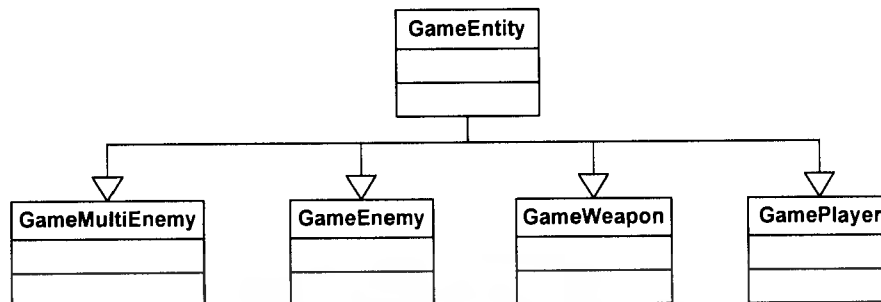
การสืบทอดคุณสมบัติของคลาส GameState



รูปที่ 3.14 แสดงการสืบทอดจากคลาส GameState

จากภาพ เป็นการสืบทอดคุณสมบัติการมีพฤติกรรมการทำงานของคลาส GameState โดยในคลาสลูก GamePlayState ก็ยังทำหน้าที่เป็นคลาสแม่ให้กับคลาสอื่นๆอีก ไม่ว่าจะเป็น GameStateLevel01, 02 และ 03 ส่วนคลาส GamePlayMultiState ก็ทำหน้าที่เป็นคลาสแม่ให้กับคลาส NoobClient, NoobServer, GameMultiStateLevel01, 02 และ 03 เป็นต้น

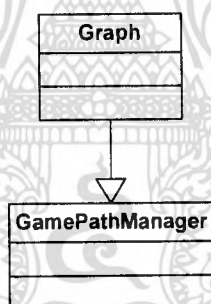
การสืบทอดคุณสมบัติของคลาส GameEntity



รูปที่ 3.15 แสดงการสืบทอดจากคลาส GameEntity

จากภาพ เป็นการสืบทอดคุณสมบัติการมีพฤติกรรมการทำงานของคลาส GameEntity เป็นคลาสแม่ให้กับคลาส GameMultiEnemy, GameEnemy, GameWeapon, GamePlayer

การสืบทอดคุณสมบัติของคลาส Graph

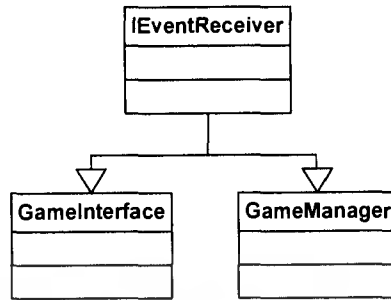


รูปที่ 3.16 แสดงการสืบทอดจากคลาส Graph

จากภาพ เป็นการสืบทอดคุณสมบัติการมีพฤติกรรมการทำงานของคลาส Graph เป็นคลาสแม่ให้กับคลาส GamePathManager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

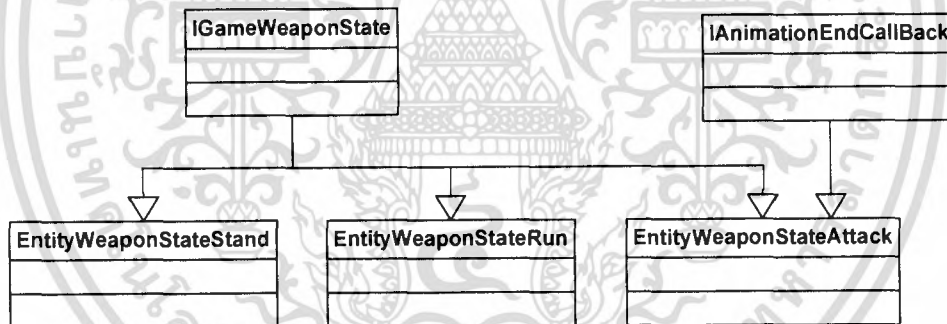
การสืบทอดคุณสมบัติของคลาส IEventReceiver



รูปที่ 3.17 แสดงการสืบทอดจากคลาส IEventReceiver

จากภาพ เป็นการสืบทอดคุณสมบัติการมีพฤติกรรมการทำงานของคลาส IEventReceiver เป็นคลาสแม่ให้กับคลาส GameInterface, GameManager

การสืบทอดคุณสมบัติของคลาส IGameWeaponState และ IAnimationEndCallBack

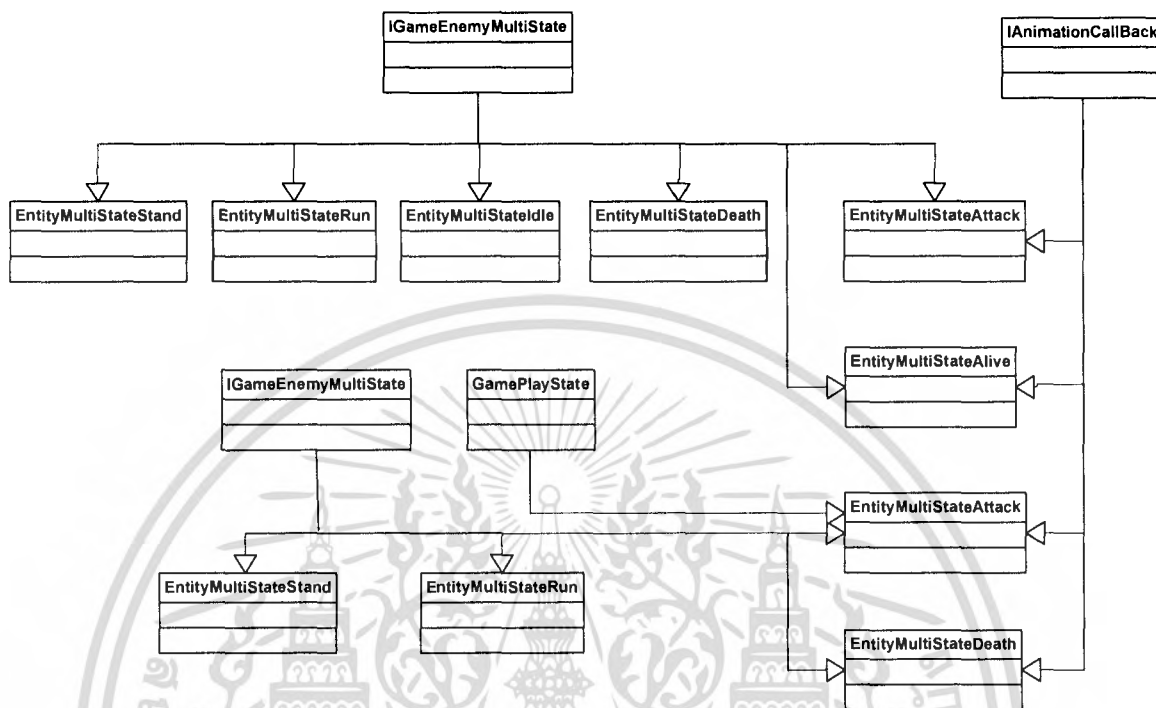


รูปที่ 3.18 แสดงการสืบทอดจากคลาส IGameWeaponState และ IAnimationEndCallBack

จากภาพ เป็นการสืบทอดคุณสมบัติการมีพฤติกรรมการทำงานของคลาส IGameWeaponState และ IAnimationEndCallBack โดยคลาส IGameWeaponState เป็นคลาสแม่ให้กับคลาส EntityWeaponStateStand, EntityWeaponStateRun และ EntityWeaponStateAttack โดยคลาสลูก EntityWeaponStateAttack ยังนั้นรับการถ่ายทอดคุณสมบัติมาจากคลาสแม่ IAnimationEndCallBack ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสืบทอดคุณสมบัติของคลาส IGameEnemyMultiState, IAnimationEndCalBack และ
GamePlayState



รูปที่ 3.19 แสดงการสืบทอดจากคลาส IGameEnemyMultiState และ IAnimationCallBack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและการวิเคราะห์ปัญหา

การทดลองและการวิเคราะห์ปัญหาที่จะกล่าวถึงในบทนี้ จะเป็นขั้นตอนของการทดสอบ และผลที่ได้จากการทดสอบในแต่ละส่วนของเกมทั้งหมด โดยผลการทดสอบที่ได้นี้ จะถูกนำไปวิเคราะห์ถึงปัญหาและแนวทางในการแก้ปัญหาเพื่อการพัฒนาต่อไปในอนาคต โคนทางผู้จัดทำหวังเป็นอย่างยิ่งว่าผลการทดลองและแนวทางในการแก้ปัญหาที่ได้จากการทดสอบในครั้งนี้ จะเป็นประโยชน์และแนวทางต่อผู้ที่มีความสนใจจะศึกษาและค้นคว้าเพื่อจะนำไปศึกษาต่อ ให้ได้เห็นถึงปัญหา และข้อดีเสียของปัญหาโคนผู้ศึกษาไม่จำเป็นต้องทำการทดสอบเพื่อหาปัญหาด้วยตนเองอีกครั้ง

คุณสมบัติของระบบที่จะนำมาทดสอบ

1. ระบบปฏิบัติการ Microsoft Windows XP
2. หน่วยประมวลผลกลางที่มีความเร็ว 1.0 Ghz
4. หน่วยความจำหลักขนาด 320 MB
5. VGA Card ของ NVIDIA Geforce4 MX 440 หน่วยความจำบนการ์ดขนาด 64 MB
6. ระบบเสียง Sound Card ของ Creative Vibra 128

ขั้นตอนการดำเนินการทดสอบ

- ทำการติดตั้งตัวโปรแกรมและซอฟต์แวร์ที่จำเป็นต้องใช้
- ทำการรันและประมวลผลภายใต้ระบบที่กำหนด
- ตรวจสอบการใช้คำสั่งของปุ่มกดต่างๆและการทำงานของปุ่มกดต่างๆ
- ตรวจสอบการทำงานของโปรแกรมโดยการใช้เมาส์และคีย์บอร์ด
- ตรวจสอบการเชื่อมต่อผ่านระบบเครือข่าย
- ตรวจสอบการหา Error

จุดประสงค์ของการดำเนินการทดสอบ

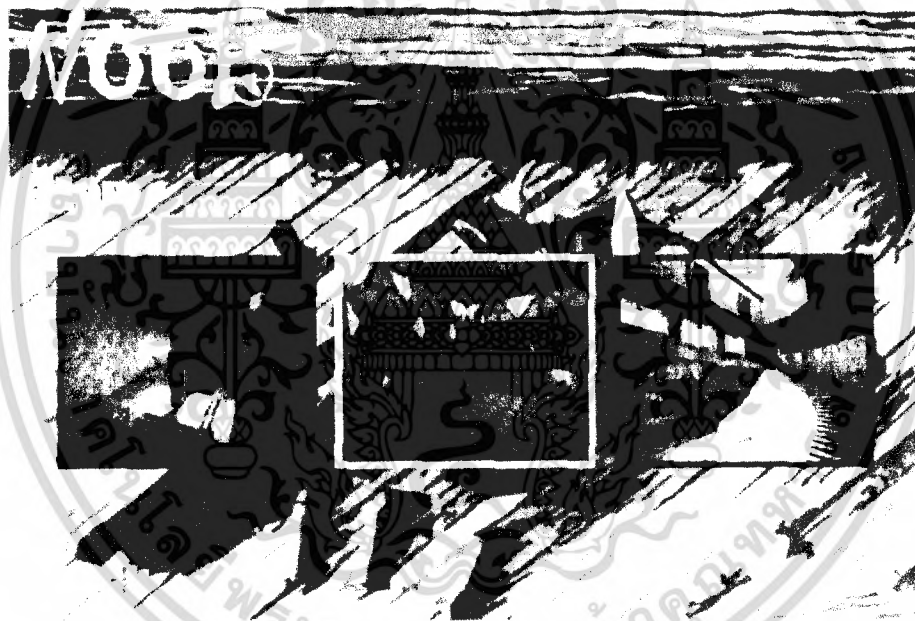
- หลังจากติดตั้งตัวโปรแกรมแล้วสามารถใช้โปรแกรมได้
- การประมวลผลของโปรแกรมจะต้องทำความเร็วในระดับที่ยอมรับได้
- การตอบโต้ของคอมพิวเตอร์จะต้องเป็นไปตามที่กำหนดไว้
- การใช้คำสั่งของปุ่มกดต่างๆเป็นไปตามรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.2 ขั้นตอนการรันโปรแกรมภายใต้เครื่องคอมพิวเตอร์ที่กำหนด

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
รันโปรแกรมภายใต้ระบบเครื่องคอมพิวเตอร์ที่กำหนด	ทำการรันโปรแกรมโดยดับเบิลคลิกที่ไฟล์ “Noob.exe” ในโฟลเดอร์ปลายทางที่ได้ทำการติดตั้ง	โปรแกรมสามารถทำงานได้โดยหน้าจอ Main
ทดสอบลำดับของการเปลี่ยนสถานะ	คลิกที่ปุ่มต่างๆ เพื่อเข้าสู่หน้าจอส่วนต่างๆ ของเกม	ลำดับของการเปลี่ยนสถานะหน้าจอถูกต้องตามที่ได้ออกแบบไว้

4.2.2 เมื่อคลิกที่ปุ่ม “Single Player” จะปรากฏหน้าจอเลือกฉาก



รูปที่ 4.2 หน้าจอเลือกฉาก

ตาราง 4.3 ขั้นตอนการเลือกฉากต่างๆ

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการเลือกฉาก	คลิกที่ฉากต่างๆ แล้วกดปุ่ม Start	เมื่อเริ่มจะตัดเข้าฉากที่ได้ทำการเลือกไว้ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 เมื่อคลิกที่ปุ่ม “Multiplayer” จะปรากฏหน้าต่างให้เลือกที่จะเป็น Server หรือ Client



รูปที่ 4.3 หน้าจอเลือกสถานะ

ตาราง 4.4 ขั้นตอนการเลือกสถานะว่าจะเป็นเซิร์ฟเวอร์ หรือ ไคลเอนท์

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
สถานะเซิร์ฟเวอร์	คลิกที่ปุ่ม Server	จะไปยังหน้าจอเลือกตัวละคร , ฉาก และหน้าจอใส่ชื่อของผู้เล่นที่เป็น เซิร์ฟเวอร์ ได้อย่างถูกต้อง
สถานะไคลเอนท์	คลิกที่ปุ่ม Client	จะไปยังหน้าจอเลือกตัวละคร และหน้าจอใส่ชื่อของผู้เล่นที่เป็นไคลเอนท์ ได้อย่างถูกต้อง

4.2.3 หน้าจอเลือกตัวละคร



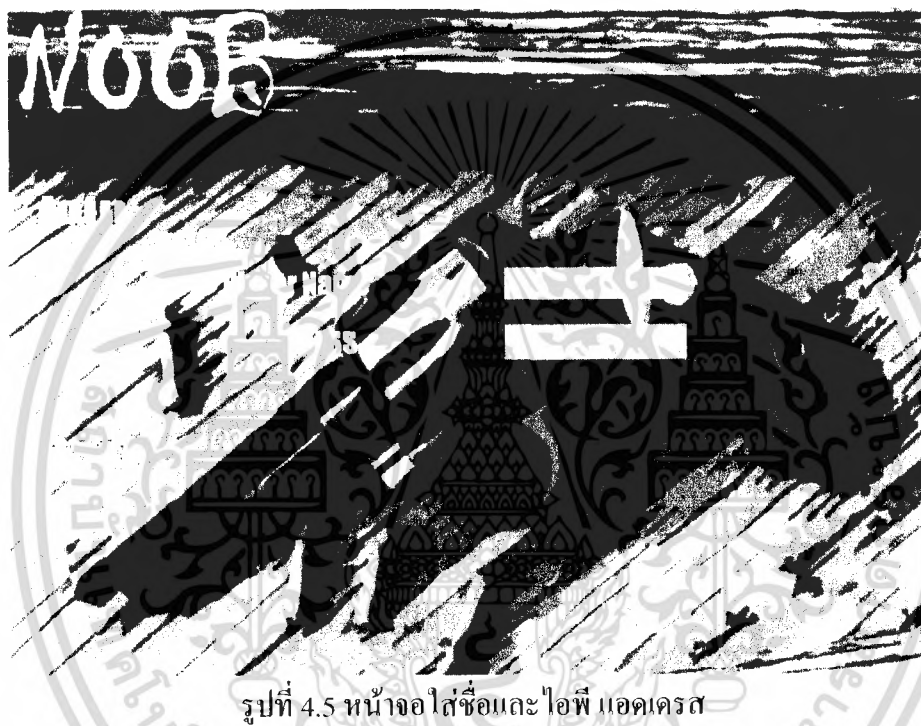
รูปที่ 4.4 หน้าจอเลือกตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.5 ขั้นตอนการเลือกตัวละครต่างๆ

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการเลือกตัวละครต่างๆ	คลิกตัวละครต่างๆ แล้วกดปุ่ม Start	ตัวละครที่ปรากฏเป็นไปตามที่ได้ทำการเลือกไว้

4.2.4 หน้าจอการใส่ชื่อของผู้เล่นไคล์เอนท์ และ ไอพี แอดเดรส ของเครื่องเซิร์ฟเวอร์



รูปที่ 4.5 หน้าจอใส่ชื่อและไอพี แอดเดรส

ตาราง 4.6 ขั้นตอนการเชื่อมต่อระหว่างไคล์เอนท์กับเซิร์ฟเวอร์

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการเชื่อมต่อ	ใส่ ไอพี ของเครื่อง เซิร์ฟเวอร์ ที่ต้องการเชื่อมต่อ	ทำการเชื่อมต่อกับเครื่องเซิร์ฟเวอร์ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกม

ตารางที่ 4.7 ขั้นตอนการดำเนินการทดสอบความสมบูรณ์ของเกมในส่วนของทางเลือกเมนู

การทดสอบ	ขั้นตอนการทดสอบ	ผลการทดสอบ
ทดสอบการเลือกปุ่มต่างๆของหน้าจอ	เลื่อนลูกศรไปวางไว้เหนือปุ่ม Single Player, Multiplayer, Options, Credits, Quit	ปุ่มต่างๆ จะมีลักษณะเปลี่ยนไป เมื่อลูกศรวางอยู่เหนือปุ่ม
ทดสอบการกดปุ่มต่างๆของหน้าจอ	กดปุ่ม Single Player, Multiplayer, Options, Credits, Quit	เมื่อกดปุ่มแล้วหน้าจอจะถูกเปลี่ยนเป็นหน้าจอของแต่ละโหมด

ตารางที่ 4.8 ขั้นตอนการทดสอบการควบคุมตัวละคร

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
การเดินหน้า	กดปุ่ม W	เกมแสดงผลของการเดินหน้า
การถอยหลัง	กดปุ่ม S	เกมแสดงผลของการถอยหลัง
การสไลด์ไปทางซ้าย	กดปุ่ม A	เกมแสดงผลของการสไลด์ไปทางซ้าย
การสไลด์ไปทางขวา	กดปุ่ม D	เกมแสดงผลของการสไลด์ไปทางขวา
การยิง	คลิกเมาท์ซ้าย	จำนวนกระสุนลดลง
การเพิ่มจำนวนกระสุน	คลิกเมาท์ขวา	จำนวนกระสุนเต็ม

4.4 ขั้นตอนการทดสอบการส่งข้อมูล

เนื่องจากการส่งข้อมูลผ่านเครือข่ายในเกม จะถูกจัดให้อยู่ในรูปของแพ็คเกจ (Packets) ทุกครั้งที่มีการส่ง ซึ่งแต่ละแพ็คเกจจะมีหน้าที่การทำงานที่แตกต่างกันออกไป ดังนั้นการทดสอบระบบจึงต้องทดสอบทุกแพ็คเกจที่ใช้ในเกมดังนี้

ตารางที่ 4.9 ขั้นตอนการทดสอบการส่งข้อมูล

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
แพ็คเกจการเชื่อมต่อ	ทดลองให้โคล์เอนท์ทำการเชื่อมต่อเข้าสู่เซิร์ฟเวอร์	เซิร์ฟเวอร์มีการตอบสนองถูกต้องโดยการแสดงคำว่า "connection : 1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
แพคเกจข้อมูลของศัตรูก่อนหน้า	ให้ไคล์เอนท์ที่สอง เชื่อมต่อเข้าสู่เซิร์ฟเวอร์	การแสดงผลตำแหน่งการขึ้น ทั้งในเซิร์ฟเวอร์ ไคล์เอนท์ แรก และไคล์เอนท์ที่สอง ถูกต้อง
แพคเกจตำแหน่ง	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์จับตำแหน่ง การขึ้นโดยการกดปุ่ม W AS หรือ D เพื่อเคลื่อนที่	ตำแหน่งของไคล์เอนท์และ เซิร์ฟเวอร์เคลื่อนที่ถูกต้อง
แพคเกจการหมุน	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์จับเมาส์	การหมุนตัวละครของไคล์เอนท์ และเซิร์ฟเวอร์ถูกต้อง
แพคเกจการโจมตี	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์กดคลิกเมาส์ ซ้ายเพื่อยิงใส่ศัตรู	ผู้ยิงทำทำยิงถูกต้อง
แพคเกจการถูกโจมตี	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์กดคลิกเมาส์ ซ้ายเพื่อยิงใส่ศัตรู	ผู้ถูกยิงพลั้งลด
แพคเกจตำแหน่งกระสุนกระทบ	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์กดคลิกเมาส์ ซ้ายเพื่อยิงพื้นที่ใดๆ	ถ้ากระสุนกระทบตัวละครจะมี สีสีแดงระเบิดจะเป็นสีแดง แต่ ถ้ากระทบฉากจะมีสีขาว
แพคเกจการตาย	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์ยิงศัตรูจนตาย โดยการคลิกเมาส์ซ้าย	ทั้งไคล์เอนท์และเซิร์ฟเวอร์ แสดงผู้ตายได้ถูกต้อง
แพคเกจการฟื้นคืนชีพ	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์คลิกเมาส์ขวา ขณะตาย	ทั้งไคล์เอนท์และเซิร์ฟเวอร์ แสดงการฟื้นคืนชีพได้ถูกต้อง
แพคเกจการฆ่า	ให้ไคล์เอนท์หรือ เซิร์ฟเวอร์ยิงศัตรูจนตาย โดยการคลิกเมาส์ซ้าย	ทั้งไคล์เอนท์และเซิร์ฟเวอร์ แสดงชื่อผู้ฆ่าและผู้ตายได้ ถูกต้อง
แพคเกจยกเลิกการเชื่อมต่อ	กด ESC	ทั้งไคล์เอนท์และเซิร์ฟเวอร์ ลบตัวละครนั้นๆจากเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบ	วิธีการทดสอบ	ผลการทดสอบ
แพ็คเกจข้อมูลของไคล์เอนท์	ทดลองให้ไคล์เอนท์ทำการเชื่อมต่อเข้าสู่เซิร์ฟเวอร์	เซิร์ฟเวอร์บันทึกข้อมูลของไคล์เอนท์โดยเห็นได้จากการทราบชื่อผู้เล่นที่เป็นไคล์เอนท์
แพ็คเกจแผนที่	ทดลองให้ไคล์เอนท์ทำการเชื่อมต่อเข้าสู่เซิร์ฟเวอร์	ไคล์เอนท์เชื่อมต่อกับภาคใต้ตรงกับที่เซิร์ฟเวอร์ได้สร้างเกม (Create) ไว้

4.5 ปัญหาข้อผิดพลาดและข้อเสนอแนะ

ในขั้นตอนของการทดสอบเพื่อหาข้อผิดพลาดจะไม่มีรูปแบบที่ตายตัวชัดเจน ที่จะทดสอบให้ได้ข้อผิดพลาดจะไม่มีรูปแบบที่ตายตัวชัดเจน ที่จะทดสอบให้ได้ข้อผิดพลาดที่เกิดขึ้น โดยวิธีการทดสอบที่ดีที่สุดคือการทดสอบให้ครอบคลุมทุกส่วนของโปรแกรม และทดสอบหลายๆ ครั้งให้ได้ผลการทดสอบที่แน่นอน โดยจากการทดสอบทั้งหมดโปรแกรมทำให้เราสามารถหาข้อผิดพลาดได้ดังนี้

4.5.1 ปัญหาเสียงขัดข้อง

ปัญหานี้จะเกิดขึ้นเมื่อผู้เล่นเล่นเกมไปนานๆ เสียงปืน เสียงเติมกระสุนจะหายไป แต่ทำนองเพลงประกอบฉากยังคงอยู่ ซึ่งสาเหตุคาดว่าเกิดจากฟังก์ชัน OpenSound ของเอนจิน Audiere มีปัญหา เนื่องจากการคืนเนื้อที่หน่วยความจำ ซึ่งแนวทางการแก้ไขอาจจะต้องมีการตรวจสอบการคืนเนื้อที่หน่วยความจำในส่วนการทำงานของเสียงทั้งหมด

4.5.2 ปัญหาโครงสร้างการทำงานของไคล์เอนท์เซิร์ฟเวอร์มีความซ้ำซ้อน

ปัญหาเกิดจากการเขียนโค้ด ซึ่งผู้ออกแบบได้แยกคลาสไคล์เอนท์และเซิร์ฟเวอร์ไว้ โดยไม่ข้องเกี่ยวกับทั้งๆที่มีการทำงานบางอย่างเหมือนกัน ดังนั้นแนวทางแก้ไขควรจะสร้างคลาสแม่ไว้โดยใส่กระบวนการทำงานที่จำเป็นในทั้งไคล์เอนท์และเซิร์ฟเวอร์ไว้ในคลาส จากนั้นให้คลาสไคล์เอนท์และเซิร์ฟเวอร์มาสืบทอดจากคลาสแม่นี้ จะทำให้การเขียนโค้ดไม่ซ้ำซ้อน และแก้ไขได้สะดวกมากยิ่งขึ้น

4.5.3 ปัญหาความคมชัดของภาพ

ปัญหานี้เกิดจากสกุลของภาพเนื่องจากภาพบางภาพ เช่น ปุ่มต่างๆ จำเป็นต้องมีการทำ Transparent กับสีที่เฉพาะเจาะจง ดังนั้นหากภาพไม่คมชัดพอจะทำให้การทำ Transparent ไม่สมบูรณ์ ผลที่เกิดขึ้นคือเห็นรอยต่างๆบริเวณขอบปุ่ม ดังนั้นแนวทางแก้ไขควรใช้สกุลภาพที่มีความคมชัดสูงมากๆ เช่น สกุล Bitmap (bmp) เป็นต้น

ปัญหาอีกประการหนึ่งคือ ขนาดของภาพที่นำไปใส่ไว้ในฉาก ทั้งส่วนสูงและความกว้างของภาพต้องอยู่ในรูปของ 2 ยกกำลัง X โดย X คือจำนวนเต็มบวกใดๆ มิฉะนั้นภาพที่แสดงจะมีคุณภาพต่ำ

4.5.4 ปัญหาระบบเครือข่ายล่าช้า

ปัญหานี้จะสังเกตได้อย่างชัดเจนเมื่อมีการเชื่อมต่อตั้งแต่ 2 การเชื่อมต่อขึ้นไป ซึ่งสาเหตุเกิดจากการจัดการแพคเกจ และเงื่อนไขการส่งแพคเกจ โดยแนวทางการแก้ไขคือต้องหาวิธีการในการส่งแพคเกจให้มีประสิทธิภาพมากขึ้น เช่น การส่งข้อมูลการเดินทางของผู้เล่น ไม่จำเป็นต้องส่งข้อมูลไปทุกครั้งที่มีการเดินทุกๆหนึ่งพิกัด ควรส่งไปเมื่อมีระยะทางห่างจากจุดเริ่มต้นพอสมควร เพื่อลดความหนาแน่นบนเครือข่าย โดยการคำนวณหา ค่า Distance ระหว่างจุดเริ่มต้นกับจุดสิ้นสุด (โดยให้ Distance (d) เท่ากับรากที่สองของ $x^2 + y^2 + z^2$)

4.5.5 ปัญหาไม่สามารถคืนเนื้อที่หน่วยความจำได้

ปัญหานี้จะเกิดโหมดจึงเกิดเพลย์เยอร์ เนื่องจากแต่ละฉากในเกมจะมีการเซตจุดเวย์พอยต์ (Waypoint) และจุดเชื่อมต่อไว้ (Adjacent Node) เพื่อให้ศัตรูสามารถเดินมาต่อสู้กับผู้เล่นได้ตามเวย์พอยต์ที่กำหนดไว้ ซึ่งปัญหาที่เกิดขึ้นเกิดเมื่อผู้เล่นเปลี่ยนฉากแล้ว Adjacent Node ไม่ถูกเคลียร์ ทำให้เกิดปัญหากับการเดินของศัตรู

4.5.6 ในส่วนการเล่นแบบผู้เล่นคนเดียว ศัตรูสามารถโจมตีผู้เล่นได้ แม้มีสิ่งกีดขวางอยู่

โดยปัญหานี้อาจจะเกิดจากการที่ ศัตรูใช้การตรวจสอบระยะห่างระหว่างตำแหน่งของศัตรูถึงตำแหน่งของผู้เล่น หากระยะห่างอยู่ในขอบเขตที่สามารถโจมตีได้ ศัตรูก็จะโจมตีเลย โดยไม่ตรวจสอบว่ามีสิ่งกีดขวางกั้นระหว่างผู้เล่นกับตัวศัตรูหรือไม่

4.6 ประเมินประสิทธิภาพของเกม

จากเกมทดสอบที่ผ่านมาทั้งหมดทำให้เราสามารถประเมินประสิทธิภาพโดยรวมของเกมได้ว่า เกมนี้สามารถทำงานได้ตามที่ออกแบบไว้สมบูรณ์พอสมควร โดยมีข้อผิดพลาดเล็กน้อยที่เกิดขึ้น โดยการทดสอบได้จากระบบที่กำหนดไว้ข้างต้นซึ่งอาจจะสามารถเปลี่ยนแปลง เพิ่มหรือลดประสิทธิภาพได้ ถ้าหากมีการเปลี่ยนระบบที่ใช้ในการประมวลผลของเกม และเนื่องจากเกมนี้มีการติดต่อกับเสียงและภาพที่ค่อนข้างมากทำให้ระบบที่จะนำไปประมวลผลค่อนข้างทำงานได้ดีและมีประสิทธิภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

5.1.1 การศึกษาและรวบรวมข้อมูล

การศึกษาและรวบรวมข้อมูลที่จำเป็นต้องใช้ในการพัฒนาเกมทำได้โดยการศึกษาและรวบรวมข้อมูลทำได้โดยศึกษาความเป็นไปได้ที่จะพัฒนาเกมนี้นบนเกมนี้นบนเครื่องคอมพิวเตอร์ได้หรือไม่ เมื่อทำการวิเคราะห์และและตัดสินใจได้แล้วก็ทำการรวบรวมข้อมูล เพื่อสร้างกฎและกติกาที่จะนำมาเป็นแนวทางในการพัฒนาเกม พร้อมทั้งทำการศึกษาค้นคว้าวิธีการต่างๆ ในการพัฒนาเกม โดยอาจศึกษาจากเนื้อหาวิชา หรือคู่มือวิธีใช้ของเครื่องมือที่จำเป็นต้องใช้ หรืออาจจะศึกษาจากตัวเกมต่างๆ ที่มีอยู่ จากนั้นรวบรวมความรู้ทั้งหมดเพื่อนำไปใช้เป็น โครงสร้างของเกม เพื่อง่ายต่อการออกแบบหน้าจอส่วนติดต่อผู้ใช้ ออกแบบโครงสร้างและวิธีการในการพัฒนา และการเลือกใช้เครื่องมือต่างๆ ที่เหมาะสมและสนับสนุนในการพัฒนา

5.1.2 การวิเคราะห์และการออกแบบเกม

การออกแบบเกมต้องทำให้เกมมีความสวยงามดึงดูดผู้เล่น มีกติกาที่ชัดเจน สามารถใช้งานได้ง่าย โดยการวิเคราะห์และออกแบบ จะนำข้อมูลต่างๆ ที่รวบรวมไว้มาประกอบกับความรู้ความสามารถที่มีอยู่ รวมไปถึงความรู้ที่ได้มาจากการศึกษาค้นคว้าเพิ่มเติม จากนั้นทำการออกแบบกฎและกติกา วางโครงสร้างและออกแบบหน้าจอติดต่อผู้ใช้ รวมถึงขั้นตอนวิธีต่างๆ ในการพัฒนาเกมให้เสร็จตามระยะเวลาที่กำหนด แล้วจึงทำการแบ่งหน้าที่ และมอบหมายส่วนงานต่างๆ ให้กับสมาชิก

5.1.3 การสร้างตัวละคร ภาพ และเสียงต่างๆ

ขั้นตอนนี้สามารถทำควบคู่ไปได้กับการเขียนโปรแกรม โดยเกมนี้ใช้ได้ใช้เครื่องมือหลายอย่างที่จะช่วยในการออกแบบและสร้างตัวละคร ภาพ และเสียงต่างๆ รวมทั้งการสร้างภาพเคลื่อนไหวภายในเกม โดยเครื่องมือที่ใช้ในการออกแบบและสร้างงานกราฟฟิก ได้แก่ โปรแกรมตกแต่งรูปภาพ Adobe Photoshop ส่วนเครื่องมือที่ใช้ในการออกแบบและสร้างงานด้านเสียงเอฟเฟกต์และเสียงประกอบ ได้แก่ โปรแกรมการตัดต่อเสียง MP3 Cutter Joiner

5.1.4 การพัฒนาโปรแกรม

ขั้นตอนนี้จะเป็นขั้นตอนที่สำคัญและใช้เวลานานที่สุดในการพัฒนาเกม เพราะว่าขั้นตอนนี้จะเป็นการนำความรู้ที่เรารวบรวมได้ทั้งหมด มารวมกับโครงสร้าง รูปแบบ และกติกาของเกมที่เราจะพัฒนา รวมไปถึงภาพ และเสียงที่เราได้จัดเตรียมไว้ และเมื่อเริ่มต้นพัฒนาก็ต้องมีการค้นคว้าความรู้ใหม่ๆ มากมายนอกเหนือจากที่ค้นคว้าไว้ในตอนแรก มีการวิจัยและพัฒนาอัลกอริทึมที่ใช้ในการควบคุมกระบวนการของเกมให้เป็นไปตามกฎและกติกาที่วางเอาไว้ และทำการทดสอบการทำงานของเกมที่พัฒนาไปเรื่อยๆ จึงต้องใช้ระยะเวลาในขั้นตอนนี้มากพอสมควร ซึ่งขั้นตอนการทำงานในส่วนนี้มีซอฟต์แวร์ และเครื่องมือที่ใช้ในการพัฒนาดังนี้

- Microsoft Visual C++ .NET
- Irrlicht Game Engine
- Raknet Network Engine
- Audiere Sound Engine

5.2 ข้อกำหนดของโปรแกรม

เกมนี้จะเป็นเกมที่พัฒนาบนพื้นฐาน โครงสร้างของ Microsoft Windows และทำงานที่ความละเอียด 1024 x 768 พิกเซล โดยที่เกมนี้ทำงานร่วมกับภาพ และเสียงค่อนข้างมาก รวมไปถึงการตรวจเงื่อนไขภายในเกมโดยขึ้นกับหน่วยประมวลผลกลาง เช่นการตรวจสอบเงื่อนไขบางเงื่อนไขภายในเกมโดยขึ้นอยู่กับหน่วยประมวลผลกลาง ทำให้เกมนี้ต้องทำงานอยู่บนระบบคอมพิวเตอร์ที่มีประสิทธิภาพสูงพอสมควร โดยรายละเอียดของข้อกำหนดของโปรแกรมหาดังต่อไปนี้

1. ระบบเครื่องคอมพิวเตอร์ที่ใช้รันเกมต้องมีประสิทธิภาพที่สูง เนื่องจากเกมจำเป็นต้องมีการประมวลผลของภาพกราฟฟิก และเสียง ค่อนข้างมาก
2. เกมนี้จำเป็นต้อง DirectX เวอร์ชัน 9 ขึ้นไป
3. ห้ามมีการเคลื่อนย้ายไฟล์เครื่องต่างๆ จากตำแหน่งที่ติดตั้งเกม เพราะว่าเกมมีการเรียกใช้ข้อมูลที่อยู่ในไฟล์เครื่องเหล่านั้น โดยหากเกมหาข้อมูลที่เป็นต้องใช้ในการทำงานไม่เจอ ก็จะทำให้เกิดความผิดพลาดได้
4. เกมนี้ต้องการการเชื่อมต่อ ที่สนับสนุนการประมวลผลด้าน 3 มิติ

5.3 ข้อเสนอแนะ

เนื่องจากเกมญี่ปุ่น เป็นเกมที่มีกฎและกติกาค่อนข้างซับซ้อน และปลื้กย้อยมากมาย จึงทำให้ในการพัฒนาจำเป็นต้องมีอัลกอริทึมในการควบคุมกระบวนการของเกมที่ดี และต้องมีการพัฒนาโปรแกรมที่มีโครงสร้างซับซ้อนมากขึ้น ด้วยเหตุนี้ทำให้การออกแบบอัลกอริทึมของเกมต้องทำด้วยความรอบคอบ และต้องคอยตรวจสอบผลการทำงานของเกมไปทุกระยะ เพื่อตรวจหาข้อผิดพลาดที่อาจเกิดขึ้น นอกจากนี้ยังสามารถนำเกมนี้ไปพัฒนาโดยการสร้างกฎและกติกาใหม่ๆ หรือความพิเศษใหม่ๆ เพื่อให้ดูมีเหมาะสมมากขึ้นได้ตามต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] บุทธนา ลีลาศวัฒนกุล , เริ่มต้นการเขียนโปรแกรมด้วยภาษา C++ , บริษัท ดวงกลมสมัย จำกัด
- [2] วิวัฒน์ อุดมปิณฑทรัพย์ , 3ds max Reference Autodesk VIZ 2005 , บริษัท เอส.พี.ซี บั๊คส์ จำกัด
- [3] Nikolaus Gebhardt. 2002. **Irrlicht Graphic Engine**. [Online].
Available : <http://irrlicht.sourceforge.net>
- [4] **Rakkarsoft**. [Online]. Available _ : <http://www.rakkarsoft.com>
- [5] Chad Austin , Matt Campbell , Jacky Chong , Theo Reed , Richard Schaaf , Ben Scott.
Audiere Sound Engine [Online]. Available : <http://audiere.sourceforge.net>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก คู่มือการติดตั้ง

คุณสมบัติขั้นต่ำของเครื่องคอมพิวเตอร์ที่ต้องการ

1. ระบบปฏิบัติการ Microsoft Windows XP Professional
2. หน่วยประมวล CPU 1.0 GHz
3. หน่วยความจำหลักขนาด 256 MB
4. VGA Card หน่วยความจำบนการ์ดขนาด 64 MB
5. หน่วยความสำรองต้องมีพื้นที่ว่างเหลือไม่ต่ำกว่า 50 MB
6. ต้องสนับสนุน OpenGL หรือ DirectX 9.0 ขึ้นไป
7. หากมีการเล่นแบบมัลติเพลเยอร์ จะต้องใช้ Network(TCP/IP ตั้งแต่ 128kbps ขึ้นไป)

วิธีการติดตั้ง NooB มีดังต่อไปนี้

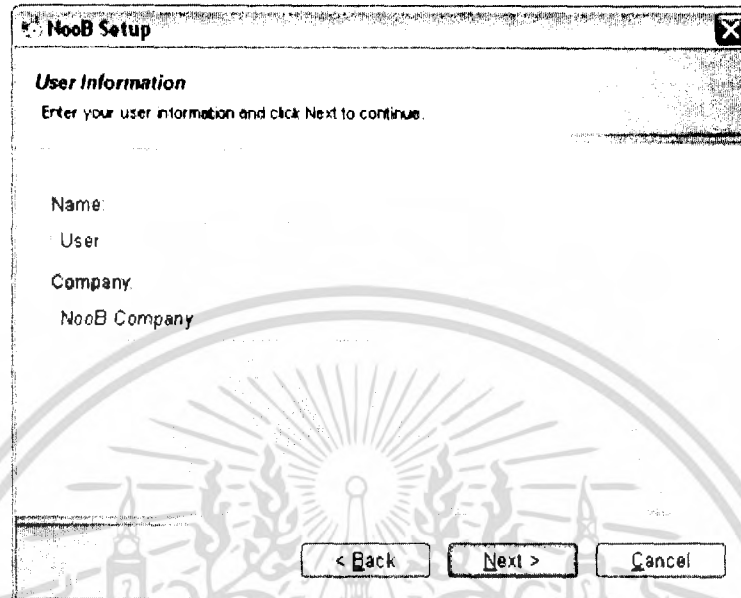
1. รันไฟล์ Setup.exe จะปรากฏหน้าจอ ดังภาพ



รูปที่ ก.1 แสดงหน้าแรกของการติดตั้ง

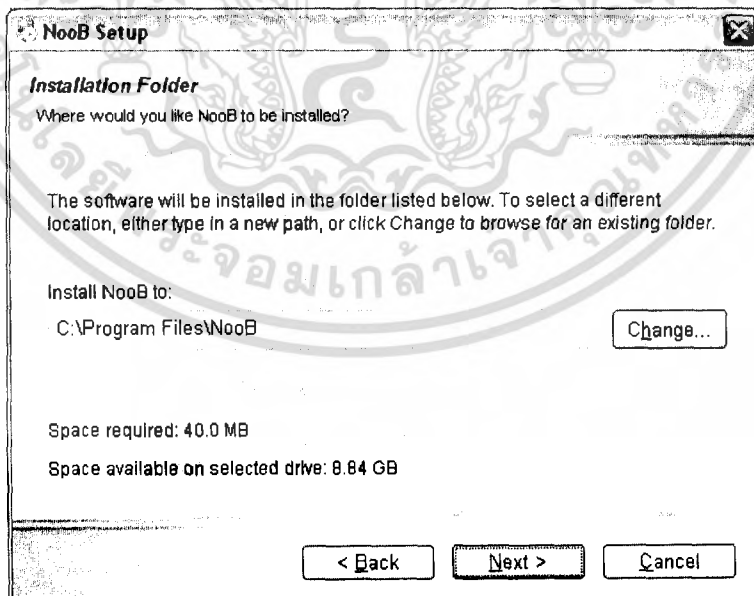
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อกด Next แล้วจะเข้าสู่ขั้นตอนถัดไปคือการป้อน ชื่อผู้ใช้ และชื่อบริษัท



รูปที่ ก.2 แสดงหน้าการป้อนชื่อและบริษัทของผู้ใช้

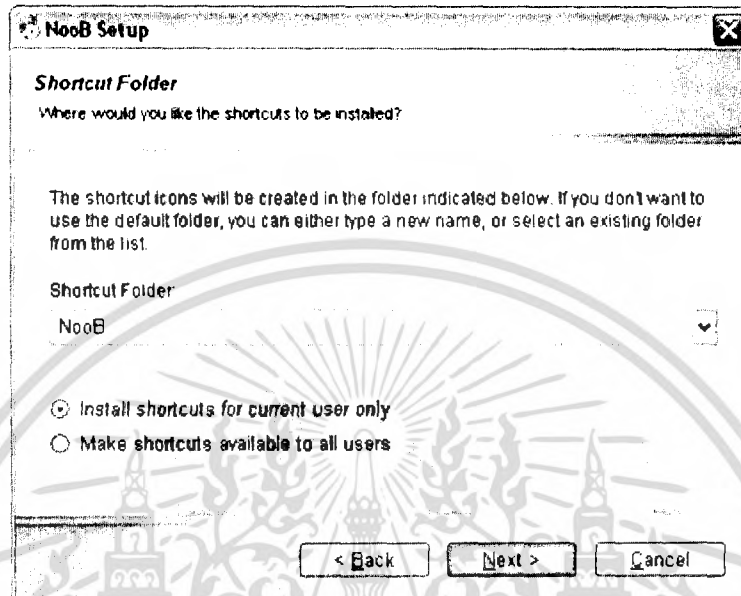
3. เมื่อป้อนข้อมูลเสร็จแล้วกด Next จะเป็นขั้นตอนการเลือก Path ที่ต้องการลง โปรแกรม โดยเนื้อที่ของ โปรแกรมที่ใช้ คือ 40 MB



รูปที่ ก.3 แสดงหน้าการระบุ Path ที่ต้องการติดตั้ง

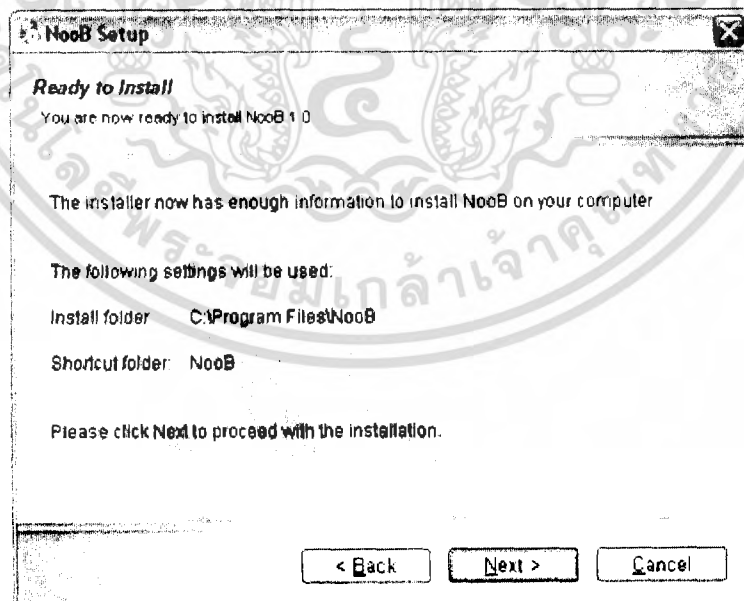
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขั้นต่อไปเป็นการเลือกที่ใส่โฟลเดอร์ลัด (Shortcut Folder) และเลือกผู้ใช้ที่สามารถใช้ได้



รูปที่ ก.4 แสดงการเลือกที่ใส่โฟลเดอร์ลัด (Shortcut Folder)

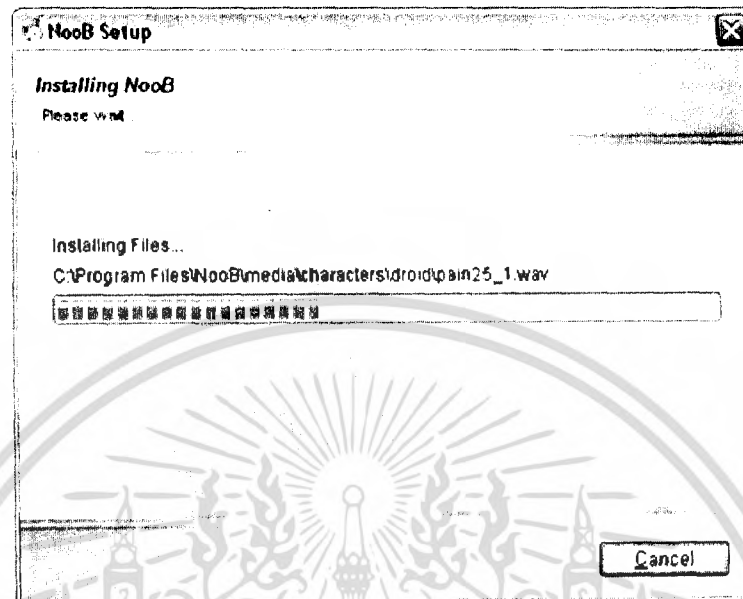
5. ขั้นตอนต่อไปเป็นการแสดงรายละเอียดข้อมูลที่จะทำการติดตั้งทั้งหมด ดังภาพ



รูปที่ ก.5 แสดงรายละเอียดข้อมูลที่จะทำการติดตั้งทั้งหมด

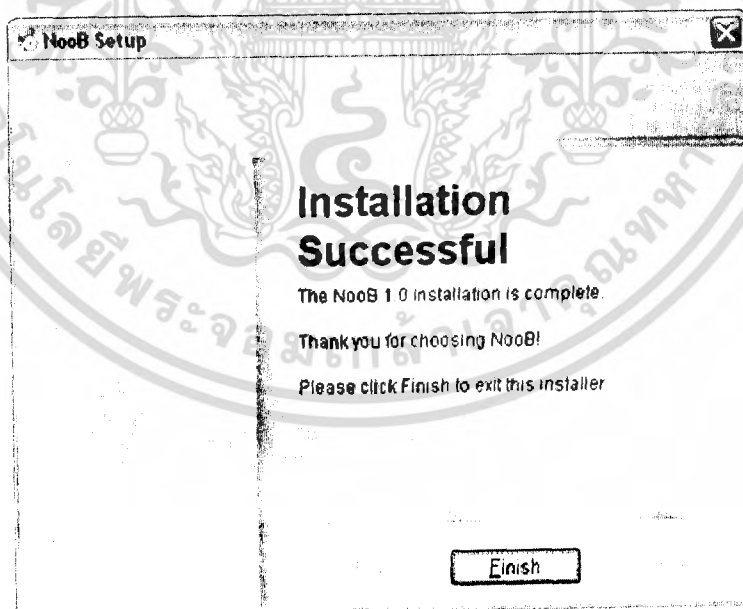
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากนั้นจะเป็นการเริ่มการติดตั้ง



รูปที่ ก.6 แสดงกระบวนการติดตั้งโปรแกรม

7. เมื่อติดตั้งเสร็จสมบูรณ์จะปรากฏหน้าจอ ดังภาพ



รูปที่ ก.7 แสดงเมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คู่มือการเล่น

ทำการเข้าเกมจาก file ที่ชื่อ NooB.EXE จะปรากฏหน้าจอดังรูป

- คลิกที่ปุ่ม Single Player เพื่อเข้าสู่การเล่นคนเดียว
- คลิกที่ปุ่ม Multiplayer เพื่อเข้าสู่การเล่นผ่านระบบเครือข่าย
- คลิกที่ปุ่ม Options เพื่อเข้าสู่การปรับแต่งระดับความยากและระดับเสียงในเกม
- คลิกที่ปุ่ม Credits เพื่อดูรายชื่อผู้พัฒนาเกม
- คลิกที่ปุ่ม Quit เพื่อทำการออกจากเกม



รูปที่ ข.1 แสดงหน้าจอเมนูหลักของเกม

การทำงานในโหมด Single Player (ซิงเกิลเพลย์เยอร์)

เป็นโหมดสำหรับการเล่นคนเดียว โดยการเล่นในโหมดนี้จะมีระดับความยากให้เลือกอยู่ด้วยกัน 4 ระดับได้แก่ Easy (ง่าย) , Medium (ปานกลาง) , Hard (ยาก) , Impossible (ยากมาก) ซึ่งโดยค่ามาตรฐานของเกมจะเป็น ปานกลาง (Medium) ซึ่งความยากของเกมนั้นจะส่งผลถึง ความรุนแรงของอาวุธ และความเร็วของ คู่ต่อสู้

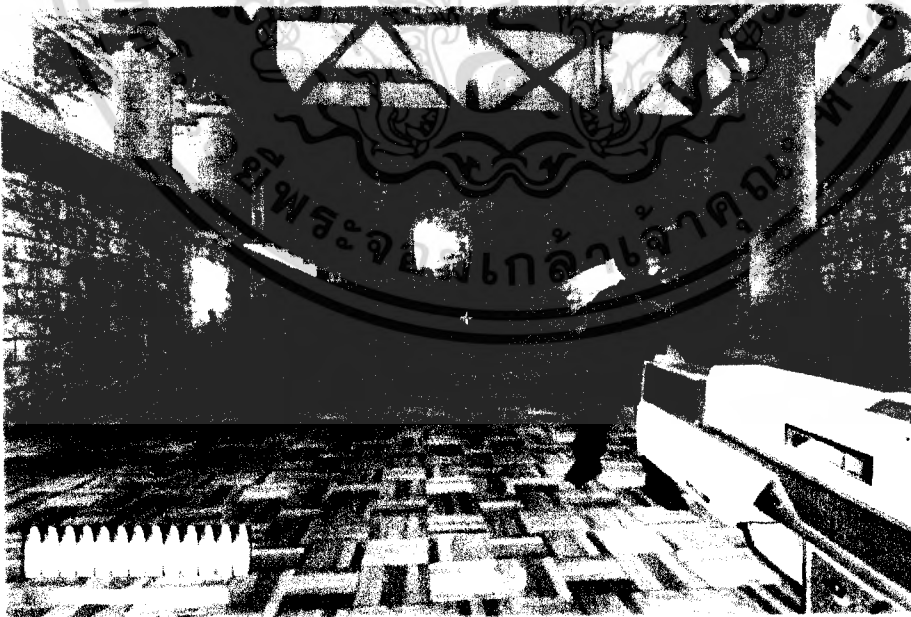
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกโหมด Single Player แล้วจะเข้าสู่หน้าจอการเลือกแผนที่ โดยในเกมจะมีให้เลือกด้วย 3 แผนที่ ซึ่งแต่ละฉากจะมีลักษณะภูมิประเทศและความกว้างแตกต่างกันไป



รูปที่ ข.2 แสดงหน้าจอขณะเลือกแผนที่ในโหมดเล่นคนเดียว

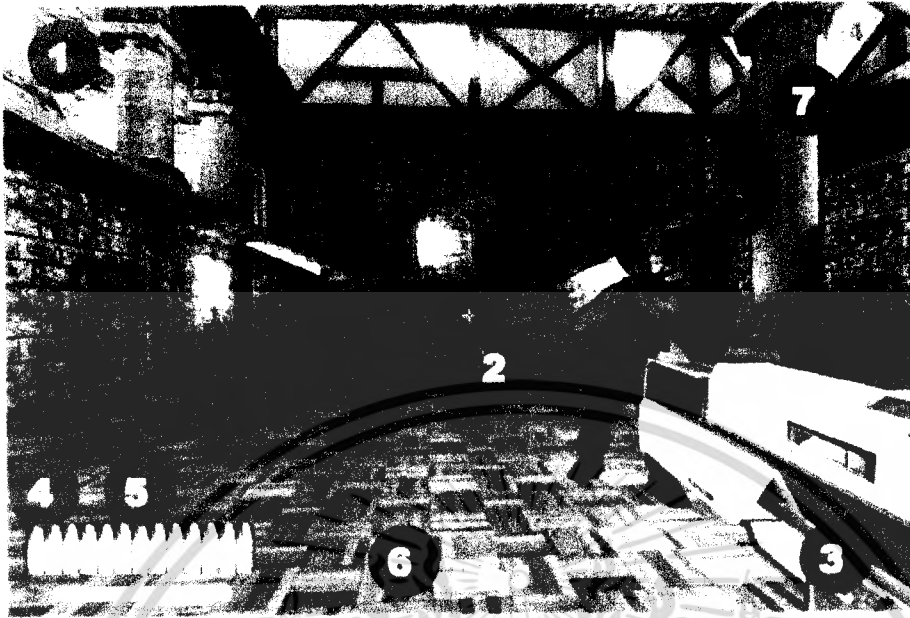
หลังจากเลือกแผนที่แล้วจะเข้าสู่เกม โดยจุดมุ่งหมายของการเล่นในโหมดนี้คือพยายามจัดการศัตรูให้หมดทุกตัวในแผนที่



รูปที่ ข.3 แสดงหน้าจอขณะเลือกแผนที่ในโหมดเล่นคนเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรายละเอียดของหน้าจอเป็นดังนี้



รูปที่ ข.4 ความหมายของสิ่งต่างๆในหน้าจอขณะเล่นเกม

หมายเลข 1	ฉากหรือแผนที่
หมายเลข 2	ศัตรู
หมายเลข 3	อาวุธของผู้เล่น
หมายเลข 4	กระสุนที่มี
หมายเลข 5	พลังชีวิตของผู้เล่น
หมายเลข 6	จำนวนชีวิตที่เหลืออยู่
หมายเลข 7	จำนวนศัตรูที่ยังเหลืออยู่ในแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสามารถจัดการคู่ต่อสู้ได้ทั้งหมด จะปรากฏหน้าจอดังนี้



รูปที่ ข.5 แสดงหน้าจอเมื่อชนะ

แต่หากไม่สามารถเอาชนะคู่ต่อสู้ได้จะปรากฏหน้าจอดังภาพ



รูปที่ ข.6 แสดงหน้าจอเมื่อแพ้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในโหมด Multi Player (มัลติเพลย์เยอร์)

เป็นโหมดสำหรับการเล่นหลายคน จุดมุ่งหมายของโหมดนี้คือการจัดการผู้เล่นให้
เกมโดยในโหมด Multiplayer นี้จะเล่นได้จำเป็นต้องมีผู้เล่น 2 ประเภทด้วยกัน ได้แก่ ผู้เล่นที่เป็น
เซิร์ฟเวอร์ และผู้เล่นที่เป็นไคลเอนท์

ผู้เล่นที่เป็นเซิร์ฟเวอร์ (Server)

ผู้เล่นที่เป็นเซิร์ฟเวอร์จะเป็นผู้สร้างเกมเพื่อให้ผู้เล่นอื่นๆที่เป็นไคลเอนท์ ทำการ
เชื่อมต่อเข้ามา โดยรายละเอียดและขั้นตอนการสร้างเกมมีดังนี้

- เมื่อกดเข้าสู่โหมด Multiplayer แล้วจะเข้าสู่ฉากเลือกตัวละคร โดยจะมีตัวละคร ให้
เลือก 3 ตัวดังที่กล่าวไปแล้วข้างต้น



รูปที่ ข.7 แสดงหน้าเลือกตัวละครในโหมดมัลติเพลย์เยอร์ของผู้เล่นที่เป็นเซิร์ฟเวอร์

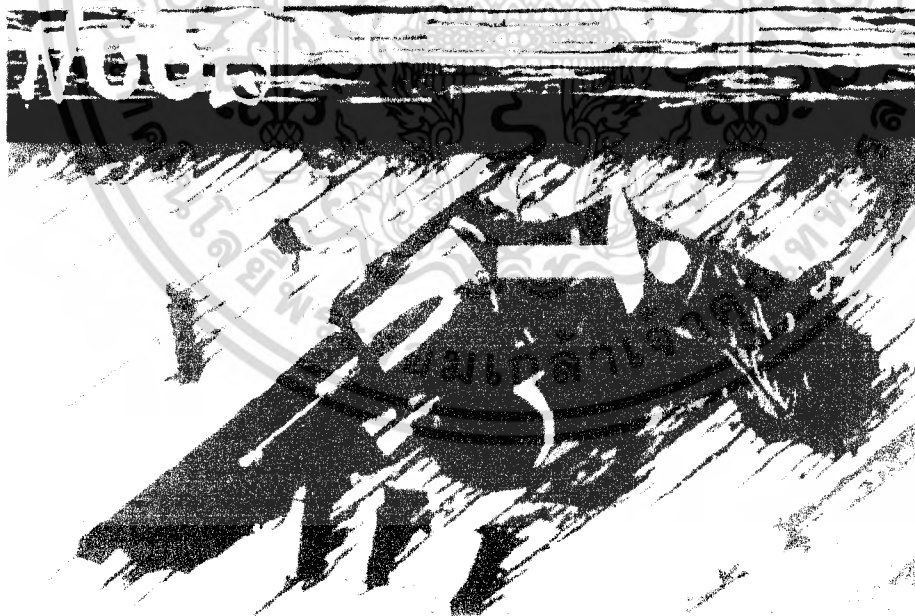
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากเลือกตัวละครเรียบร้อยแล้วจะเป็นขั้นตอนการเลือกแผนที่ตั้งภาพ



รูปที่ ข.8 แสดงหน้าจอการเลือกแผนที่ในโหมดมัลติเพลย์เยอร์

- หลังจากนั้นจะเป็นขั้นตอนการระบุชื่อผู้เล่นที่เป็นเซิร์ฟเวอร์โดย Port (พอร์ต) ที่ใช้ในเกมจะเป็น 10000



รูปที่ ข.9 แสดงหน้าจอการตั้งชื่อผู้เล่นที่เป็นเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

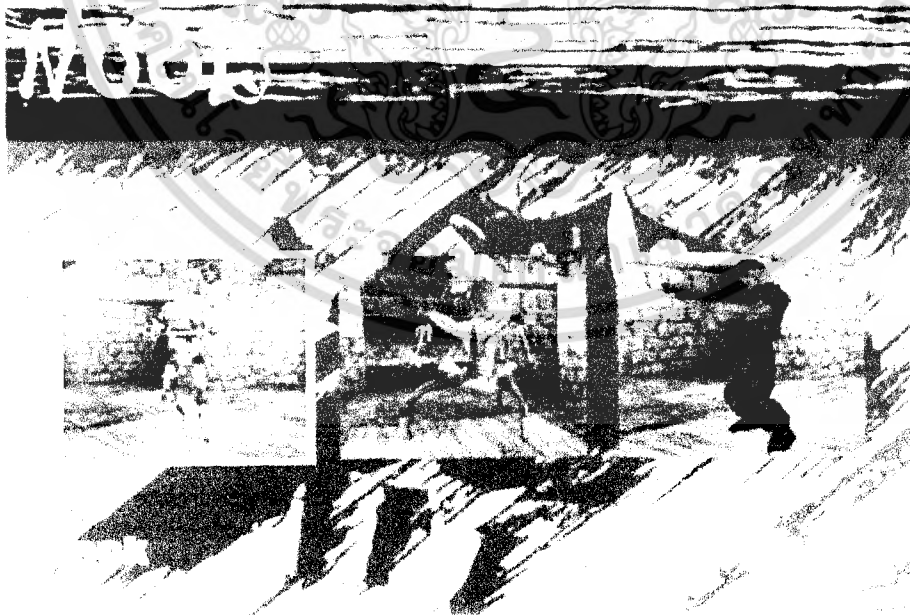
- หลังจากที่ตั้งชื่อผู้เล่นกด Create จะเป็นการเข้าสู่โหมดการเล่น และรอการเชื่อมต่อ



รูปที่ ข.10 แสดงหน้าจอขณะเล่นเกมในโหมดมัลติเพลย์เซอร์

ผู้เล่นที่เป็นไคลเอนท์ (Client)

ผู้เล่นที่เป็นไคลเอนท์จะเป็นผู้เล่นที่จะทำการเข้ามาเชื่อมต่อกับเครื่องที่เป็นเซิร์ฟเวอร์ โดยขั้นตอนการเชื่อมต่อมีดังต่อไปนี้



รูปที่ ข.11 แสดงหน้าจอการเลือกตัวละครในโหมดมัลติเพลย์เซอร์ของไคลเอนท์

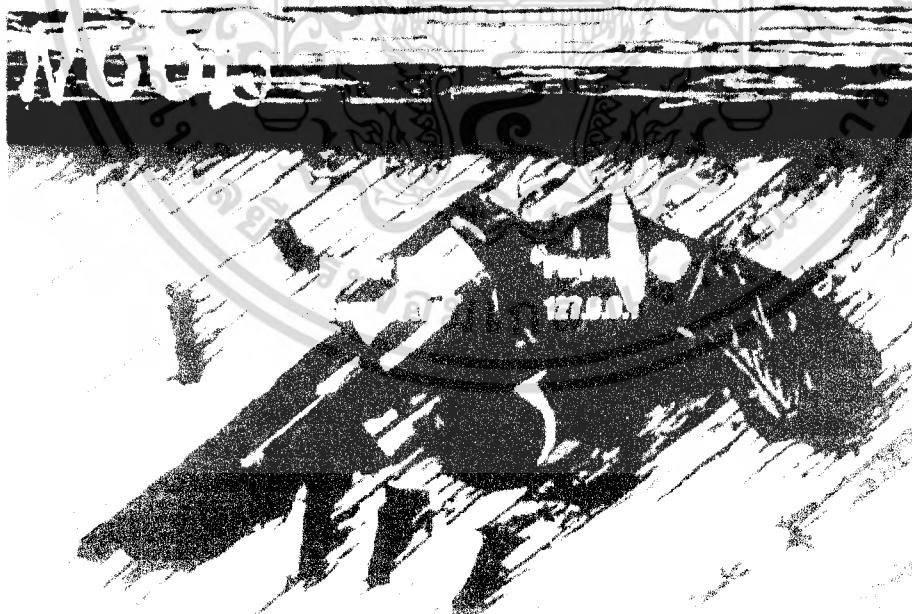
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากนั้นจะเข้าสู่ขั้นตอนการการระบุชื่อผู้เล่น และระบุ IP Address ของเซิร์ฟเวอร์ที่ต้องการเชื่อมต่อ



รูปที่ ข.12 แสดงหน้าจอการระบุชื่อผู้เล่น และ IP Address ของเซิร์ฟเวอร์

- เมื่อกดปุ่ม Connect แล้ว จะเป็นการเริ่มเชื่อมต่อกับเซิร์ฟเวอร์เบื้องต้น โดยถ้าสามารถเชื่อมต่อ ได้ จะมีปุ่ม Join ปรากฏขึ้น เพื่อให้สามารถเข้าเล่นเกมได้



รูปที่ ข.13 แสดงหน้าจอการเมื่อการเชื่อมต่อสำเร็จ

- จากนั้นจะเข้าสู่หน้าจอขณะเล่นเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.14 แสดงหน้าจอขณะเล่นเกมเมื่อเป็นผู้เล่นที่เป็นโคลอนท์

การทำงานในโหมด Options

เราสามารถปรับระดับเสียงและระดับความยากของเกมในโหมด ซิงเกิลเพลย์เออร์ ได้ที่ โหมดนี้



รูปที่ ข.15 แสดงหน้าจอปรับแต่งระดับความยาก และระดับเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้