สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเขียนเกมคอมพิวเตอร์ด้วย Direct X

GAME PROGRAMMING WITH DIRECT X

โดย

นายวิธิวัฒน์ เลิศชัยมงคล

นายสน หาญวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50274
วัน,เดือน,ปี 28 เม.ย. 2547
.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GAME PROGRAMMING WITH DIRECT X

BY

Mr. WITHIWAT LERTCHAIMONGKOL

Mr. SOUN HANWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบเสนอปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การเขียนเกมคอมพิวเตอร์ด้วย Direct X	
นักศึกษา	นาย วิธิวัฒน์ เลิศชัยมงคล	รหัสประจำตัว 42010330
	นาย สน หาญวงศ์	รหัสประจำตัว 42010357
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ . อรลาภ แสงอรุณ	
	อาจารย์ มยุรี เลิศเวชกุล	
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2545	

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(รศ.อรลาภ แสงอรุณ)

.....

(อ.มยุรี เลิศเวชกุล)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การเขียนเกมคอมพิวเตอร์ด้วย Direct X	
นักศึกษา	นาย วิธิวัฒน์ เลิศชัยมงคล	รหัสประจำตัว 42010330
	นาย สน หาญวงศ์	รหัสประจำตัว 42010357
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. อรลาภ แสงอรุณ	
	อาจารย์ มยุรี เลิศเวชกุล	
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2545	

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอการเขียนโปรแกรมเกมคอมพิวเตอร์ โดยเกมที่สร้างจะเป็นเกมแนว Shooting แบบ 2 มิติ โดยมีมุมมองจากด้านบน โดยตัวละครในเกมจะเป็นรถถังแบ่งออกเป็น 2 ฝ่าย คือฝ่ายผู้เล่น และ ฝ่ายศัตรูซึ่งควบคุมโดยคอมพิวเตอร์ โดยฝ่ายผู้เล่นจะต้องทำภารกิจที่แตกต่างกันไป เพื่อที่จะได้รับชัยชนะในแต่ละฉาก ซึ่งในแต่ละฉากจะมีรูปแบบภูมิประเทศของฉากที่ต่างกันออกไป นอกจากนี้ก่อนเข้าสู่ฉากต่อสู้ ผู้เล่นจะสามารถปรับแต่งรถถังของตนให้มีประสิทธิภาพมากยิ่งขึ้น รวมถึงการติด Item สนับสนุนในการรบ และการติดตั้งระบบ Ability เพื่อให้ผู้เล่นแข็งแกร่งยิ่งขึ้นอีกด้วย นอกจากนี้ตัวละครที่เป็นคนขับรถถังจะสามารถเก็บค่าประสบการณ์เพื่อยกระดับความสามารถของตนขึ้นเรื่อย ๆ ในการพัฒนาของตัวละครนี้ ตัวละครจะมีความสามารถพิเศษที่เพิ่มขึ้น ซึ่งผู้เล่นจะสามารถนำความสามารถพิเศษมาประยุกต์ใช้ได้อย่างหลากหลาย

ลักษณะเด่นของตัวเกมนี้ คือผู้เล่นนอกจากจะใช้ ทักษะในการบังคับรถถังแล้วยังต้องใช้ความคิดในการวางแผนต่าง ๆ ได้แก่ การต่อสู้, การปรับแต่งรถถัง, ติดตั้ง Item สนับสนุน รวมถึงระบบ Ability ซึ่งถ้าใช้ได้อย่างมีประสิทธิภาพแล้วย่อมสามารถเอาชนะคู่ต่อสู้ที่มีความแข็งแกร่งกว่าได้

ในการสร้างเกมนี้ได้พัฒนาโดยใช้ Visual C/C++ และ Direct X SDKในส่วนของการพัฒนาโปรแกรม ส่วนของกราฟฟิกและเสียงประกอบจะใช้ซอฟต์แวร์อื่น ๆ เสริมซึ่งได้แก่ พวกโปรแกรมตกแต่งภาพ โปรแกรมวาดภาพ และ โปรแกรมตัดแต่งเสียง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THESIS TITLE	**GAME PROGRAMMING WITH DIRECT X**
STUDENT	**Mr. Witiwat Lertchaimongkol No.42010330**
	Mr. Soun Hanwong No.42010357
ADVISOR	**Miss.Onrlarp Sangaroon**
	Ms.Mayuree Lertwatechakul
COURSE	**Bachelor of Engineering**
DEPARTMENT	**Information Engineering**
YEAR	**2002**

ABSTRACT

This thesis presents game development, the developed game is 2D top view shooting game called " Battle City Extreme " players are grouped into two teams, player team and enemy team.The enemy team is controlled by computer. About each mission has difference environment, the player team must complete each mission. Before the mission you can upgrade his tank, Setup ability system and prepare support item to use in battle.

Highlight of this game is about technique which player can modify and set up ability system, player 's tank with wisdom setup could be win an enemy tank that is more powerful.

About development of this game, we use Visual C/C++ and Direct X SDK as tools software to develop the game and use other software as supplementary such as Paint, Photoshop etc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงจะไม่สำเร็จด้วยดีหากไม่ได้รับความกรุณาเอื้อเฟื้อจากหลายๆ ท่าน ขอขอบพระคุณ รศ. อรลาภ แสงอรุณ และ อาจารย์มยุรี เลิศเวชกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ให้ความดูแลเอาใจใส่ ให้คำปรึกษาอย่างดีตลอดมา ทั้งทางด้านการดำเนินงาน และการจัดทำปริญญานิพนธ์

ขอขอบคุณคณะครูอาจารย์ทุกท่านที่ช่วยประสิทธิประสาทวิชาความรู้ จนสามารถนำความรู้ที่ได้มาใช้จนสามารถสร้างสรรค์ผลงานได้สำเร็จ

ขอขอบคุณน้องไผ่ และ น้องปอลิน สำหรับภาพประกอบเกมสวยๆ ที่ตั้งใจทำให้

ขอขอบคุณเพื่อนๆ ทุกคนสำหรับคำแนะนำ และ แนวคิดใหม่เพิ่มเติม

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เปิดโอกาสให้ผู้จัดทำได้เข้ามาศึกษาเล่าเรียนวิชาจนมีวันนี้

สุดท้ายนี้ขอขอบคุณสิ่งศักดิ์สิทธิ์ทั้งหลายในสากลโลกที่ได้ดลบันดาลให้ข้าพเจ้าได้กำเนิดมาเป็นลูกของคุณพ่อ คุณแม่ อันเป็นที่เคารพรักยิ่ง ซึ่งท่านได้คอยดูแลอบรม สั่งสอนให้เป็นคนดี มีความสามารถ จนถึงทุกวันนี้

นายวิธิวัฒน์ เลิศชัยมงคล

นายสน หาญวงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูปภาพ	XI
บทที่ 1 บทนำ	
1.1 บทนำ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน และ ลักษณะของตัวเกม	1
1.4 ขั้นตอนการดำเนินงาน	2
1.4.1 ศึกษาหาข้อมูล	2
1.4.2 การวางแผนการดำเนินงาน	2
1.4.3 ออกแบบเกม	3
1.4.4 การสร้างแกนของเกม	3
1.4.5 การออกแบบกราฟฟิก	3
1.4.6 จัดการระบบเสียง	4
1.4.7 ระบบ AI ของเกม	4
1.4.8 การทดสอบและแก้ไข	4
บทที่ 2 หลักการทำงาน และ ทฤษฎีเบื้องต้น	
2.1 หลักการเบื้องต้นของ Direct X	5
2.1.1 สถาปัตยกรรมของ Direct X	6
2.1.2 บทบาทของ COM	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.1.3 COM และไดเร็กเอ็กซ์	9
2.1.4 HAL (Hardware Abstraction Layer)	9
2.1.5 HEL (Hardware Abstraction Layer)	9
2.1.6 ส่วนประกอบของ DirectX	11
2.2 Visual C/C++	12
2.3 แนวคิดพื้นฐานของเกม	12
2.4 Computer Graphic และ Image Processing	13
2.4.1หลักการปรับภาพ Brightness หรือความสว่างของภาพ	13
2.4.2หลักการปรับ Contrast หรือความเข้มของสี	14
2.4.3Bump Mapping ทำภาพพื้นผิว 2 มิติ ให้เป็น 3 มิติ	15
2.5 Artificial intelligence	17
2.5.1 Breadth First Search	17
2.5.2 Depth First Search	19
2.5.3 Bidirectional Breadth First-Search	21
2.5.4 A* algorithm	22
บทที่ 3 ทฤษฎี และการออกแบบ	
3.1 การสร้าง Surface	26
3.2 การแสดงผลภาพ	26
3.3 การสร้างเสียง	27
3.4 การออกแบบในส่วนของตัวเกม	28
3.4.1 รูปแบบรถถัง	28
3.4.1.1 ส่วนตัวรถ	28
3.4.1.2 ส่วนอาวุธ	30
3.4.1.3 ส่วนล้อรถถัง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.4.1.4 เครื่องหลัง	41
3.4.1.5 คอมพิวเตอร์	42
3.4.2 การปรับแต่งรถถัง	43
3.4.3 ไอเท็ม	44
3.4.4 คนขับ	46
3.4.4.1 ค่าสถานะของตัวละคร	46
3.4.4.2 ความสามารถพิเศษ	47
3.4.5 รูปแบบฉาก	50
3.5 การสร้างภาพ และ Effect ของภาพ	51
3.5.1 การสร้างพื้นผิว	53
3.5.2 การสร้างวัตถุ	55
3.5.3 การสร้างภาพรถถัง	55
3.5.4 การสร้างภาพมอนสเตอร์	56
3.5.5 การสร้างผิวน้ำ	56
3.5.6 การคำนวณการเคลื่อนที่ของวัตถุ	57
3.6 การออกแบบระบบเสียงในเกม	58
3.7หลักการทำงานของ Map Engine	59
3.8 โครงสร้างของไฟล์ .MAP	60
3.9 การออกแบบระบบควบคุม (Computer Control)	62
3.9.1 การหาเส้นทาง (Pathfinding)	64
3.9.1.1 ขั้นตอนการค้นหาเส้นทาง	65
3.9.1.2 การคำนวณค่าการตัดสินใจ	66
3.9.2 การโจมตี (Attack)	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 ตัวอย่างผลการดำเนินงาน และ ชิ้นงาน	
4 .1รูปแบบของเกม Battle City Extreme	76
4.2 ฉากเปิดเกม	76
4.3 เมนูหลักของเกม	77
4.4 ร้านขายของ	86
4.5 ฉากแผนที่	86
4.6 ฉากในเกม	87
4.7 โปรแกรม BCX-MapEngine	92
4.8 การค้นหาเส้นทาง	97
บทที่ 5 ผลการดำเนินงาน	
5.2 สรุปผลการดำเนินงาน	99
5.1 ปัญหาในการดำเนินงาน	91
5.3 ข้อจำกัดของโครงงาน	100
5.4 แนวทางในการพัฒนา	101
บรรนาณุกรม	102
ภาคผนวก. ก	103
ภาคผนวก. ข	113

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูป หน้า

รูปที่ 2.1 สถาปัตยกรรมของไดเร็กเอ็กซ์ 6
รูปที่ 2.2 ภาพโดยรวมของ COM 8
รูปที่ 2.3 อินเตอร์เฟซของ COM อ็อบเจ็กต์ 8
รูปที่ 2 .4 การแตกโหนดของการคำตอบแบบ Breadth first search 18
รูปที่ 2.5 การแตกโหนดของการคำตอบแบบ Depth first search 20
รูปที่ 2.6 การแตกโหนดของการคำตอบแบบ Bidirectional Breadth first search 21
รูปที่ 2.7 การแตกโหนดของการคำตอบ A* Algorithm 23
รูปที่ 3.1 รูปการทำงานของเกมคอมพิวเตอร์โดยใช้ GDI 25
รูปที่ 3.2 รูปการทำงานของเกมคอมพิวเตอร์โดยใช้ Direct X 25
รูปที่ 3.3 การสร้าง Surface ของเกม 26
รูปที่ 3.4 การนำภาพมาแสดงผลที่หน้าจอ 26
รูปที่ 3.5 การแสดงผลของเสียงประกอบ 27
รูปที่ 3.6 รูปตัวอย่าง Texture ที่นำมาใช้ในการสร้างพื้นผิว 51
รูปที่ 3.7 รูปตัวอย่างModelของแสงที่นำมาใช้ในการสร้างพื้นผิว3 มิติ 52
รูปที่ 3.8 รูปพื้นผิวก่อนที่จะนำมาทำการเฉลี่ยค่าแสงกับโมเดลแสง 53
รูปที่ 3.9 รูปพื้นผิวหลังจากนำมาทำการเฉลี่ยค่าแสงกับโมเดลแสงแล้ว 53
รูปที่ 3.10 รูปแสดงตัวอย่างการออกแบบวัตถุ 53
รูปที่ 3.11 รูปแสดงการแปลงวัตถุให้เป็นโทนสีเดียว 54
รูปที่ 3.12 รูปแสดงการโย้ภาพ 54
รูปที่ 3.13 รูปแสดงการรวมวัตถุและเงา 54
รูปที่ 3.14 รูปแสดงการใช้เทคนิคสลับจุด 55
รูปที่ 3.15 รูปภาพแสดงการใส่ Texture ลงในชิ้นส่วนของรถถัง 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูป หน้า

รูปที่ 3.16 รูปแสดงการออกแบบมอนสเตอร์ 56
รูปที่ 3.17 รูปแสดงการเปรียบเทียบของผิวน้ำที่ทำให้ใสโดยการเฉลี่ยค่าสี 57
รูปที่ 3.18 รูปแสดงการแพนและปรับระดับของเสียงเอฟเฟ็คท์ 58
รูปที่ 3.19 รูปแสดงระบบโดยรวมของระบบ AI 63
รูปที่ 3.20 รูปแสดงสถานะของพื้นที่ 64
รูปที่ 3.21 รูปแสดงการแตกโหนดในการหาเส้นทาง 65
รูปที่ 3.22 รูปแสดงค่า f(n) h(n) และ g(n) ในแต่ละโหนด 66
รูปที่ 3.23 รูปแสดงการเลือกโหนด 68
รูปที่ 3.24 รูปแสดงกระบวนการแตกโหนด 69
รูปที่ 3.25 รูปแสดงกระบวนการแตกโหนด 70
รูปที่ 3.26 รูปแสดงเส้นทางจากจุดเมต้นไปยังเป้าหมาย 71
รูปที่ 3.27 Flow chart แสดงกระบวนแตกโหนดเพื่อทำการหาเส้นทาง 72
รูปที่ 3.28 State diagram ของการโจมตี 73
รูปที่ 3.29 State diagram ของการเลือกใช้อาวุธ 74
รูปที่ 3.30 Flowchart แสดงกระบวนการเลือกใช้อาวุธ 75
รูปที่ 4.1 รูปแสดงฉากเปิดเกม 76
รูปที่ 4.2 รูปแสดงเมนูไอเท็มในเมนูหลัก 77
รูปที่ 4.3 รูปแสดงเมนูการปรับแต่งชิ้นส่วนของรถถัง 78
รูปที่ 4.4 รูปแสดงเมนูแสดงชิ้นส่วนที่จะนำมาติดตั้ง 79
รูปที่ 4.5 รูปแสดงการปรับแต่งการบรรทุกสิ่งของ 79
รูปที่ 4.6 รูปแสดงเมนูการปรับแต่งสีของรถถัง 80
รูปที่ 4.7 รูปแสดงสล็อตของเครื่องคอมพิวเตอร์ 81
รูปที่ 4.8 รูปแสดงทักษะที่มีเพื่อทำการติดตั้ง 82
รูปที่ 4.9 รูปแสดงเมนูสถานะของตัวละคร 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูป หน้า

รูปที่ 4.10 รูปแสดงเมนูการปรับแต่งระบบเกม 84
รูปที่ 4.11 รูปแสดงการเซฟเกม 85
รูปที่ 4.12 รูปแสดงเมนูซื้อขายแลกเปลี่ยน 86
รูปที่ 4.13 รูปแสดงแผนที่โลก 86
รูปที่ 4.14 รูปแสดงตัวอย่างฉากของเกม 87
รูปที่ 4.15 รูปแสดงการอธิบายค่าพารามิเตอร์ของฉาก 88
รูปที่ 4.16 รูปแสดงการสนทนาในเกม 89
รูปที่ 4.17 รูปแสดงค่าตัวเลขที่แสดงถึงความเสียหายของรถถัง 89
รูปที่ 4.18 รูปแสดงมอนสเตอร์ในฉาก 90
รูปที่ 4.19 รูปแสดงตัวอย่างการใช้ทักษะเสริมในการต่อสู้ 91
รูปที่ 4.20 รูปแสดงการเลือกใช้ไอเท็มของผู้เล่น 91
รูปที่ 4.21 รูปแสดงเมนูกำหนดค่าเริ่มต้นของฉาก 92
รูปที่ 4.22 รูปแสดงเมนูเลือกวัตถุต้นแบบ 93
รูปที่ 4.23 รูปแสดงตัวอย่างการใช้งานโปรแกรม BCX-MapEngine V1.4 94
รูปที่ 4.24 รูปแสดงเมนูในส่วนออกแบบ 96
รูปที่ 4.25 รูปแสดง การเดินเข้าโจมตีศัตรู 97
รูปที่ 4.26 รูปแสดง การเส้นทางในการเดินหาเป้าหมาย 98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง หน้า

ตารางที่ 1.1 ตารางการดำเนินงาน 3
ตารางที่ 3.1 ตารางการกำหนดค่าพารามิเตอร์ของตัวถังรถประเภทเกราะหนัก 29
ตารางที่ 3.2 ตารางการกำหนดค่าพารามิเตอร์ของตัวถังรถประเภทเกราะเบา 29
ตารางที่ 3.3 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทมีด 35
ตารางที่ 3.4 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทปืน 35
ตารางที่ 3.5 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทปืนกล 36
ตารางที่ 3.6 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทโล่ห์ 36
ตารางที่ 3.7 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทระเบิด 36
ตารางที่ 3.8 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทจรวด 37
ตารางที่ 3.9 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทล้อยาง 38
ตารางที่ 3.10 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถ ประเภทล้อตีนตะขาบ 39
ตารางที่ 3.11 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทล้อกึ่งผสม 39
ตารางที่ 3.12 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทขากล 40
ตารางที่ 3.13 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถ ประเภทโฮเวอร์คราฟท์ 40
ตารางที่ 3.14 ตารางการกำหนดค่าพารามิเตอร์ของเครื่องหลัง 41
ตารางที่ 3.15 ตารางการกำหนดค่าพารามิเตอร์ของเครื่องคอมพิวเตอร์ 42
ตารางที่ 3.16 ตารางแสดงรายละเอียดสิ่งของประเภทซ่อมแซม 44
ตารางที่ 3.17 ตารางแสดงรายละเอียดสิ่งของประเภทกับดัก 45
ตารางที่ 3.18 ตารางแสดงรายละเอียดสิ่งของประเภทสนับสนุน 45
ตารางที่ 3.19 ตารางแสดงรายละเอียดสิ่งของประเภทกระสุน 45
ตารางที่ 3.20 ตารางแสดงรายละเอียดสิ่งของประเภทอื่น ๆ 46
ตารางที่ 3.21 ตารางแสดงรายละเอียด Command Ability 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตาราง หน้า

ตารางที่ 3.22 ตารางแสดงรายละเอียด Fighting Ability 48
ตารางที่ 3.23 ตารางแสดงรายละเอียด Status Ability 49
ตารางที่ 3.24 ตารางแสดงรายละเอียด Support Ability 49

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

1.1 บทนำ

ในอดีต หลายคนอาจจะคิดว่าเกมคอมพิวเตอร์เป็นเรื่องของเด็กๆ แต่ปัจจุบันนี้ด้วยเทคโนโลยีที่พัฒนาขึ้นอย่างมากทำให้กลุ่มคนที่เล่นเกมมีปริมาณเพิ่มมากขึ้นมาก จึงทำให้ผู้ผลิตเกมคอมพิวเตอร์ ผลิตเกมคอมพิวเตอร์ออกจำหน่ายและสามารถสร้างผลกำไรได้อย่างมหาศาล อุตสาหกรรมเกมคอมพิวเตอร์จึงขยายใหญ่ขึ้น จนทำให้เกมคอมพิวเตอร์เป็นสินค้าส่งออกที่สร้างผลกำไรให้แก่ประเทศผู้ส่งออกได้อย่างดี

แต่น่าเสียดายที่อุตสาหกรรมเกมคอมพิวเตอร์ในบ้านเรานั้น ไม่มีผู้ให้ความสนับสนุนเท่าใดนัก อุตสาหกรรมเกมคอมพิวเตอร์ในบ้านเราจึงมีการพัฒนาที่ค่อนข้างช้า อย่างไรก็ดียังมีผู้ทำงานด้านนี้อยู่บ้าง แต่ผู้ที่ทำงานด้านนี้มักขาดการสนับสนุนในด้านกำลังทรัพย์ ในอนาคตเชื่อว่าหากมีผู้สนับสนุนอุตสาหกรรมด้านนี้มากขึ้น อาจมีผู้พัฒนาเกมคอมพิวเตอร์ที่สามารถติดตลาด และเป็นที่ยอมรับของนักเล่นเกมได้ อาจไม่ถึงขั้นเทียบเท่ากับต่างประเทศที่ทำอุตสาหกรรมเกมคอมพิวเตอร์มาเป็นสิบๆ ปี แต่เชื่อว่าคนไทยเราจะมีเกมสายเลือดไทย เช่นเดียวกับที่เรามีเพลงไทย และ หนังไทยเป็นของเราเอง แล้วสามารถพัฒนาต่อไปจนสามารถเทียบเท่าเกมของประเทศอื่น ๆ ทั่วโลกได้

1.2 วัตถุประสงค์

1. ศึกษาการสร้างเกม และ ฝึกฝนการเขียนเกมด้วยไดเร็กเอ็กซ์ (Direct X) บนฐานของโปรแกรม Visual C++
2. สามารถสร้างเกมที่มีระบบที่สามารถการตอบโต้กับผู้เล่นได้ ในขอบเขตที่เหมาะสม
3. สร้างรูปแบบเกมที่ผู้เล่นนอกจากจะใช้ ทักษะในการบังคับแล้วยังต้องใช้ความคิด ในการวางแผน และ ดำเนินเนื้อเรื่องอีกด้วย

1.3 ขอบเขตของโครงงาน และ รูปแบบของตัวเกม

- ลักษณะมุมมองของเกมเป็นแบบ มุมมองด้านบน (Top View) โดยที่รถถังมีการเคลื่อนที่ได้ 8 ทิศทาง

  มีระบบการเลื่อนฉาก ซึ่งสามารถดึงภาพบางส่วนมาแสดงผลบนหน้าจอได้
- มีฉากในเกมทั้งหมด 10 ฉาก ซึ่งจะประกอบด้วยสภาพภูมิประเทศต่างๆ ได้แก่ ป่า ภูเขา ทะเลทราย แม่น้ำ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในรถถังแต่ละคันจะมีตัวละครที่เป็นคนขับซึ่งตัวละครจะสามารถยกระดับความสามารถของตนเองด้วยการเก็บค่าประสบการณ์ ตัวละครจะมีความสามารถใหม่ๆ เมื่อมีค่าประสบการณ์เพิ่มขึ้น
- รถถังสามารถซื้อและติดตั้งอาวุธก่อนเข้าฉากต่อสู้ได้ นอกจากนี้ยังรวมถึงการติดตั้งระบบความสามารถ (Ability) ของตัวละคร รวมทั้ง การติดสิ่งของ (Item) สนับสนุนอีกด้วย
- ใช้ คีย์บอร์ด (Keyboard) เมาส์ (Mouse) และ (จอยสติ๊ก) Joystick เป็นอุปกรณ์ควบคุมในเกม
- ระบบการควบคุมอย่างฉลาด (AI) ของเกมจะใช้ทั้งในส่วนของการควบคุมการทำงานของระบบเกม และ การควบคุมตัวรถถัง
- สำหรับการนำเนื้อหาความรู้วิชา Artificial Intelligence (AI) มาใช้จะใช้ในส่วนของการหาเส้นทางที่เหมาะสม การเรียนรู้แบบง่ายๆ และ การตัดสินใจจะเป็นแบบดิสิชั่นทรี (Decision Tree)

1.4 ขั้นตอนการดำเนินงาน

มีขั้นตอนดังนี้

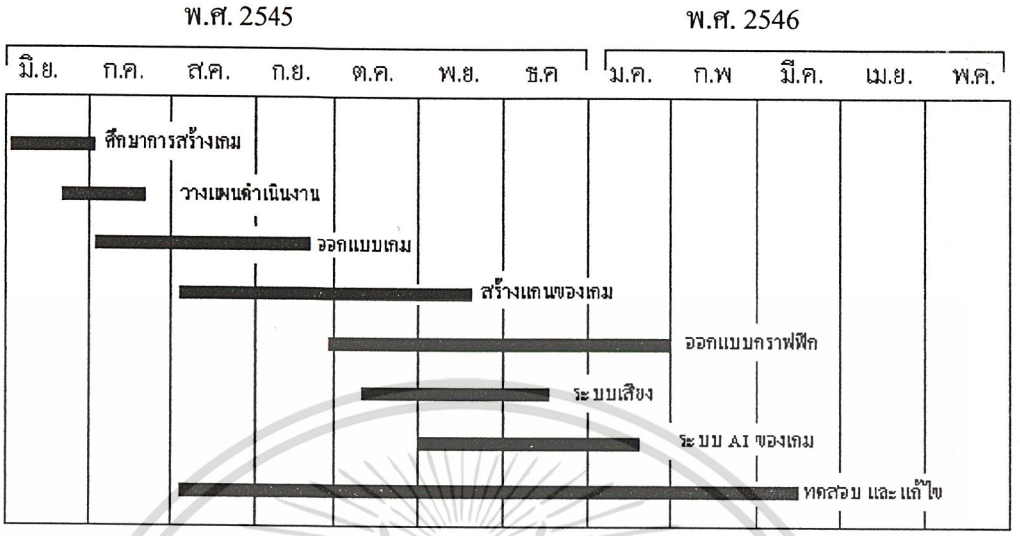
1.4.1 ศึกษาหาข้อมูล

ส่วนใหญ่จะเป็นการศึกษาข้อมูลจากหนังสือ และ เว็บไซท์ต่าง ๆ เนื้อหาที่ศึกษาจะเกี่ยวกับ การทำงานของ Direct X อัลกอริทึมต่าง ๆ ในการเขียนเกม การสร้างภาพกราฟฟิก และการปรับแต่งเสียง

1.4.2 การวางแผนการดำเนินงาน

ในส่วนนี้จะเป็นส่วนของการแบ่งงาน และการกำหนดตารางเวลาการดำเนินงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตารางที่ 1.1 ตารางการดำเนินงาน

1.4.3 ออกแบบเกม

ในขั้นตอนนี้จะเป็นการออกแบบ

- รูปแบบข้อมูลที่นำมาใช้ในเกม
- ระบบกราฟฟิก และ เสียงเบื้องต้น
- อัลกอริทึมต่าง ๆ
- เนื้อเรื่องของเกม
- รูปแบบเมนู เริ่มเกม การ Save/Load เกม การซื้อ-ติดตั้งอาวุธ ระบบ Ability และ การติดตั้ง Item
- ระบบของเกม และ อื่น ๆ

1.4.4 การสร้างตัวเกม

ในขั้นตอนนี้จะเป็นส่วนของการเขียนโปรแกรมเพื่อสร้างแกนหลักของเกม

1.4.5 การออกแบบกราฟฟิก

ในขั้นตอนนี้จะเป็นส่วนของการออกแบบกราฟฟิกต่าง ๆ ที่จะนำมาใช้ในเกม โดยการใช้งานโปรแกรมกราฟฟิกต่าง ๆ เช่น Photo Shop , Paint Shop Pro และ Illustrator เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.6 จัดการระบบเสียง

ในส่วนของเสียงนี้ส่วนใหญ่จะเป็นการหาเสียงจากแผ่นซีดี (CD) และ เว็บไซท์เพื่อนำมาปรับปรุง แก้ไขให้ได้เสียงที่ต้องการ โดยใช้โปรแกรมปรับแต่งเสียง Cool Wave Editor

1.4.7 ระบบ AI ของเกม

ในขั้นตอนนี้จะเป็นการนำการกระทำ (Action) และ สิ่งที่ได้รับจากสิ่งแวดล้อม (Percept) ต่าง ๆ ที่สร้างไว้ในตัวเกมมาใช้ เพื่อสร้างการตัดสินใจ รวมถึงการเรียนรู้ให้แก่คอมพิวเตอร์ในการที่จะควบคุมรถถังแต่ละคัน นอกจากนี้ยังใช้เกี่ยวกับการควบคุมการทำงานทั่วไปของเกมเพื่อสร้างความสนุกสนานและสมจริงในตัวเกมให้มากที่สุด

1.4.8 การทดสอบและแก้ไข

ในขั้นตอนนี้จะเป็นการหาข้อผิดพลาดของตัวโปรแกรม แล้วทำการแก้ไข รวมถึงการรับข้อติชมต่าง ๆ จากผู้ทดสอบเล่นเกม เพื่อนำมาพัฒนาตัวเกมให้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการทำงาน และ ทฤษฎีเบื้องต้น

2.1 หลักการเบื้องต้นของ Direct X

ในอดีตนั้นเกมถูกพัฒนาบนระบบปฏิบัติการดอส (MS-DOS) ซึ่งมีข้อดีคือผู้พัฒนานั้นสามารถที่จะเข้าถึงการทำงานของฮาร์ดแวร์ในระดับต่ำ (low-level) ได้โดยตรง ซึ่งทำให้แอพพลิเคชันมีประสิทธิภาพการทำงานที่สูง แต่มีข้อเสียคือ ผู้พัฒนาแอพพลิเคชันต้องพัฒนาให้แอพพลิเคชันของตนสามารถรองรับการทำงานกับอุปกรณ์ของบริษัทต่างๆ ที่มีอยู่มากมายในท้องตลาด เช่น การ์ดแสดงผลและการ์ดเสียงที่มีอยู่มากมายหลายยี่ห้อ เพื่อให้ครอบคลุมกลุ่มผู้ใช้งานให้ได้มากที่สุด ซึ่งเป็นงานที่ยากลำบากและสิ้นเปลืองงบประมาณเป็นอย่างมาก

ต่อมาบริษัทไมโครซอฟต์ได้เปิดตัวระบบปฏิบัติการวินโดว์ (Windows) ออกมา ซึ่งระบบปฏิบัติการ Windows นี้เป็นระบบปฏิบัติการแบบ GUI (Graphical User Interface) มีข้อดีคือผู้พัฒนาแอพพลิเคชันไม่จำเป็นต้องพัฒนาแอพพลิเคชันของตนให้สนับสนุนอุปกรณ์ของบริษัทต่างๆ อีกต่อไป เพราะเมื่ออุปกรณ์เหล่านั้นถูกพัฒนามาเพื่อใช้งานกับระบบปฏิบัติการ Windows แล้วผู้พัฒนาแอพพลิเคชันเพียงแต่พัฒนา โดยยึดรูปแบบมาตรฐานของระบบปฏิบัติการ Windows ก็เพียงพอ ซึ่งเป็นคุณสมบัติแบบ “device- independent” แต่ระบบปฏิบัติการ Windows นั้นเป็นระบบปฏิบัติการที่มีการแสดงผลด้านกราฟิกที่ช้า ดังนั้นในยุคแรกๆ ของระบบปฏิบัติการ Windows ผู้พัฒนาแอพพลิเคชันทางด้านเกมจึงยังคงยึดติดอยู่กับระบบปฏิบัติการ MS-DOS อยู่ ซึ่งแอพพลิเคชันทางด้านเกมที่ทำงานอยู่บนระบบปฏิบัติการ Windows นั้นก็พอมีอยู่บ้าง แต่จะเป็นพวกที่ไม่ต้องการการแสดงผลทางด้านกราฟฟิกที่รวดเร็ว เช่น เกมหมากกระดานและเกมแนวผจญภัย

ทางบริษัทไมโครซอฟต์ได้เล็งเห็นว่าแอพพลิเคชันทางด้านเกมนั้นมีผู้ใช้งานกันอย่างแพร่หลายจึงมีแนวคิดที่จะให้ผู้พัฒนาแอพพลิเคชันทางด้านเกมและมัลติมีเดียหันมาพัฒนาบนระบบปฏิบัติการ Windows ดังนั้นบริษัทไมโครซอฟต์จึงพัฒนา Direct X ขึ้นมา ซึ่งเป็นไลบรารีทางด้านมัลติมีเดียที่มีจุดมุ่งหมายหลักๆ ดังต่อไปนี้

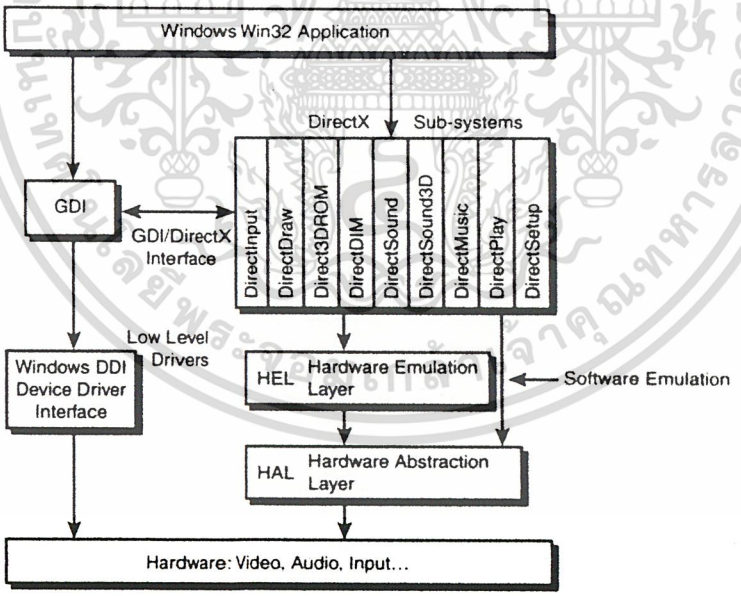
- Direct X ประกอบด้วยไลบรารีที่ทำงานในระดับต่ำ ซึ่งทำงานได้รวดเร็วและไม่มีข้อจำกัด ในการพัฒนาแอพพลิเคชันทางด้านเกม
- โครงสร้างของ Direct X ต้องยกภาระในเรื่องฮาร์ดแวร์จากผู้พัฒนาแอพพลิเคชันไปสู่ผู้ผลิตฮาร์ดแวร์ ซึ่งผู้ผลิตฮาร์ดแวร์ต้องเป็นผู้สร้างไดรเวอร์สำหรับผลิตภัณฑ์ของตนและให้ผู้พัฒนาแอพพลิเคชันนั้นสามารถใช้ความสามารถล่าสุดที่มีอยู่ในฮาร์ดแวร์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Direct X นั้นต้องอนุญาตให้ผู้พัฒนาแอพพลิเคชันสามารถพัฒนาแอพพลิเคชันออกมาในรูปแบบของ Windows application ที่สามารถทำงานบนเดสท็อปได้และสามารถทำงานร่วมกับฟังก์ชันต่างๆ ที่มีอยู่บนระบบปฏิบัติการ Windows ได้
- แอพพลิเคชันที่ถูกพัฒนาโดยใช้ Direct X นั้นต้องมีประสิทธิภาพที่อยู่สูงกว่าหรืออย่างน้อยที่สุดต้องเทียบเท่ากับประสิทธิภาพของแอพพลิเคชันที่ทำงานบนระบบปฏิบัติการ MS-DOS

2.1.1 สถาปัตยกรรมของไดเร็กเอ็กซ์

ไดเร็กเอ็กซ์ได้สร้างขึ้นมาบนพื้นฐานของคอมโพเน็นต์ โดยใช้โครงสร้างของ HAL (Hardware Abstraction Layer) ซึ่งสามารถซ่อนลักษณะเฉพาะของดีไวซ์ (Device) ที่เกี่ยวพันอยู่กับฮาร์ดแวร์ และเนื่องจากไดเร็กเอ็กซ์ได้รับการออกแบบให้มีความสามารถในการพัฒนาได้เพิ่มมากขึ้นในอนาคต ดังนั้นจึงสามารถรับรองความสามารถของฮาร์ดแวร์เร่งความเร็วชนิดใหม่ที่เข้ามา โดยใช้การทำงานผ่าน HEL (Hardware Emulation Layer)



รูปที่ 2.1 สถาปัตยกรรมของไดเร็กเอ็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

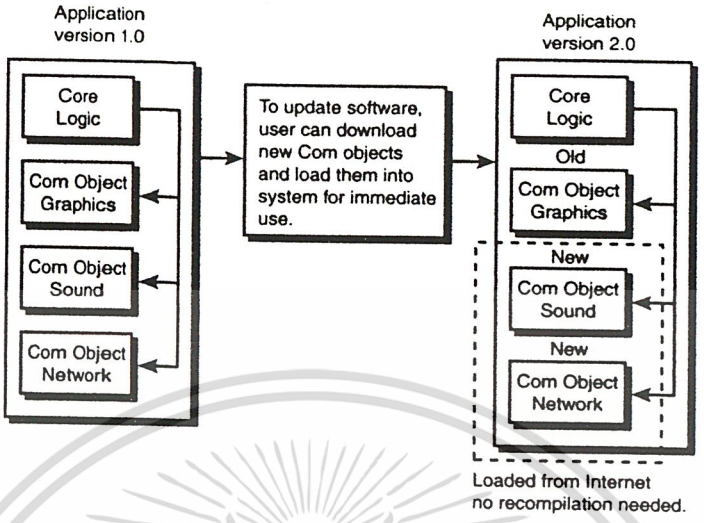
2.1.2 บทบาทของ COM

COM (Component Object Model) ได้กล่าวถึงรายละเอียดเกี่ยวกับวิธีการติดต่อสื่อสารกันระหว่างอ็อบเจ็กต์ โดยข้ามกันระหว่างโพรเซสและข้ามผ่านเน็ตเวิร์ค ระบบที่มีการเรียกใช้ COM ใช้งานจะถูกรับรองเป็นอ็อบเจ็กต์อินสแตนซ์ (Object Instance) ทุกๆ COM อ็อบเจ็กต์จะต้องยึดติดกันจนเป็นโครงสร้างแบบไบนารี ซึ่งมีความคล้ายคลึงกับโครงสร้างของ virtual table ของ C++ ที่ถูกคอมไพล์แล้ว ดังนั้นจะทำให้ทุกภาษาสามารถที่จะสร้าง COM อ็อบเจ็กต์ได้ด้วยการจัดโครงสร้างให้เหมือนกับแบบไบนารีหลังจากการคอมไพล์

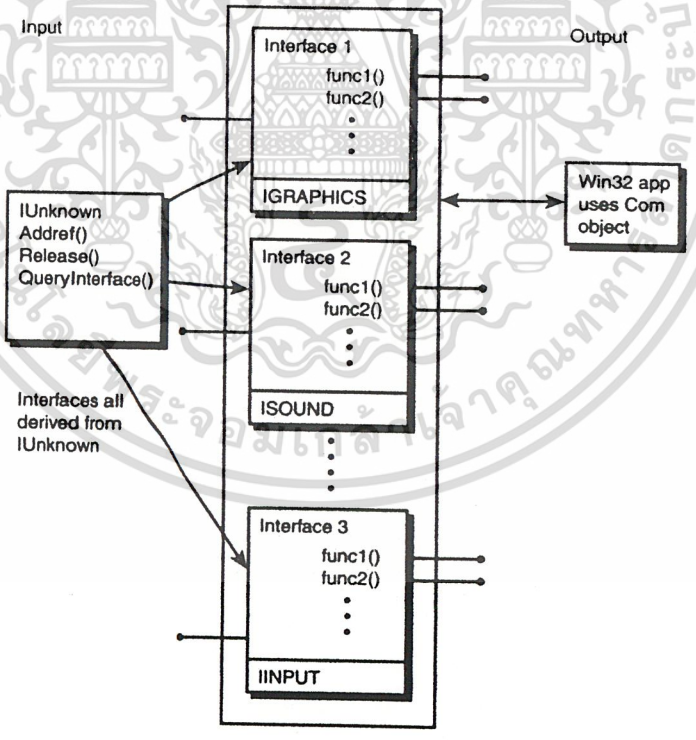
ทุกๆ COM อ็อบเจ็กต์นั้นยังจะต้องถูกสืบทอดมาจาก COM อินเตอร์เฟซมาตรฐาน ซึ่งเรียกว่าไออันโนน (IUnknown)โดยต้นแบบ (parent) อินเตอร์เฟซจะมีอยู่ด้วยกัน 3 วิธี คือ AddRef, Delete และ QueryInterface ซึ่ง 2 วิธี แรกจะจัดการอ็อบเจ็กต์ด้วยการอ้างอิงถึงจำนวน โดยการนับจำนวนของอินสแตนซ์ตั้งแต่เริ่มต้น ส่วนวิธีสุดท้ายจะจัดการเกี่ยวกับการสอบถามกับ COM อ็อบเจ็กต์ตัวอื่นๆ สำหรับอินเตอร์เฟซที่ต้องการและอินเตอร์เฟซที่มีอยู่ จากนั้นจะคืนค่าพอยเตอร์ที่ชี้ไปที่อินเตอร์เฟซนั้น

COM อ็อบเจ็กต์ที่ทำการร้องขอนั้น เรียกว่า COM ไคลเอนต์และอ็อบเจ็กต์ที่ตอบสนองการร้องขอนั้นคือ COM เซิร์ฟเวอร์ เมื่อไคลเอนต์ได้รับการเชื่อมต่อแล้วไคลเอนต์นั้นสามารถที่จะเรียกวิธีใดก็ได้ใน COM เซิร์ฟเวอร์นั้น โดยใช้การอ้างอิงของอินเตอร์เฟซพอยเตอร์ ส่วนการเชื่อมต่ออื่นๆ จะออกไปเพื่อทำให้งานของอ็อบเจ็กต์ที่เรียกเพื่อเข้ามาใช้งานง่ายขึ้น ในการเชื่อมต่อนี้จะมีวิธีที่ทำการแสดงรายชื่อของวิธีที่สนับสนุน เพื่อที่จะช่วยให้วิธีสามารถเรียกใช้งานได้ ก่อนที่จะรวมเข้าไว้ด้วยกัน ด้วยเทคนิคนี้จะอนุญาตให้สามารถสร้างชนิดของไลบรารีใดๆ และ แอพพลิเคชัน เช่น โปรแกรม Visual Basic ซึ่งมีข้อดีในการสร้างคุณสมบัติของ COM โปรแกรมที่มีความทนทานต่อความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 ภาพโดยรวมของ COM



รูปที่ 2-3 อินเตอร์เฟซของ COM อ็อบเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 COM และไดเร็กเอ็กซ์

ทุกๆ อินเตอร์เฟซของไดเร็กเอ็กซ์นั้นจะทำการสืบทอดมาจาก IUnknown และมีพื้นฐานในการพัฒนาจาก COM ด้วยเช่นกัน

การใช้ COM ของไมโครซอฟท์ทำให้ได้ชุดคอมโพเน็นต์ของไดเร็กเอ็กซ์ที่มีประสิทธิภาพเพิ่มขึ้น ซึ่งสามารถเข้ากันได้กับไดเร็กเอ็กซ์เวอร์ชั่นก่อนๆ ได้อย่างสมบูรณ์ ด้วยความอิสระ ไม่ขึ้นกับภาษาที่ใช้เขียนโปรแกรมของ COM ทำให้นักพัฒนาสามารถที่จะตัดสินใจเลือกภาษาที่ใช้เขียนโปรแกรมตามที่เหมาะสมกับนักพัฒนาเองมากที่สุด ซึ่งอาจเป็นภาษาที่ถนัด และยังทำให้ได้ประสิทธิภาพตามที่ต้องการ นอกจากนั้นนักพัฒนายังสามารถที่จะเลือกภาษาอะไรก็ตามที่ COM สนับสนุน และสามารถพัฒนาสภาพแวดล้อมได้ตามที่นักพัฒนาต้องการ ด้วยโปรแกรมมิ่งโมเดลของ COM โปรแกรมเมอร์จะแน่ใจได้ว่าแม้ว่าผู้ใช้จะอัพเกรดหรือเปลี่ยนติดตั้งฮาร์ดแวร์ใหม่ หรือจะเปลี่ยนไดเร็กเอ็กซ์เวอร์ชั่นใหม่ก็ตาม ซอฟต์แวร์ก็ยังคงจะสามารถใช้งานได้ตามปกติ โดยไม่จำเป็นต้องทำการติดตั้งใหม่

ในความเป็นจริงไดรเวอร์ของอุปกรณ์บน Windows ทุกตัวจะรวมกันเป็น WDM (Windows Driver Model) ซึ่งโครงสร้างภายในก็เป็น COM คอมโพเน็นต์ ดังนั้นจึงมีการโต้ตอบที่ดีกว่าระหว่างไดรเวอร์โมเดลกับหน้าที่ของไดรเวอร์ ในสถาปัตยกรรมของไดเร็กเอ็กซ์นั้นทำให้แน่ใจได้ว่าการพัฒนาสำหรับนักพัฒนาเกมจะมีความสะดวกมากขึ้น และลดเวลาในการทดสอบลงด้วย

2.1.4 HAL (Hardware Abstraction Layer)

HAL เป็นส่วนที่อยู่ล่างสุดของไดเร็กเอ็กซ์ ซึ่งประกอบไปด้วยส่วนควบคุมของฮาร์ดแวร์ที่ถูกทำโดยผู้ผลิตฮาร์ดแวร์ ที่จะควบคุมฮาร์ดแวร์โดยตรง โดยชั้นนี้จะให้ประสิทธิภาพอย่างมาก เพราะว่าสามารถที่จะติดต่อกับฮาร์ดแวร์ได้โดยตรง ในการทำงานจริงจะไม่สามารถติดต่อกับ HAL ได้เอง แต่ไดเร็กเอ็กซ์จะทำการจัดการให้โดยอัตโนมัติ

2.1.5 HEL (Hardware Abstraction Layer)

HEL จะอยู่บนชั้น HAL โดยทั่วไปไดเร็กเอ็กซ์จะถูกออกแบบให้สามารถใช้ข้อดีของฮาร์ดแวร์ต่างๆ ได้ แต่ถึงอย่างไรไดเร็กเอ็กซ์ก็ไม่สามารถที่จะทำงานได้กับอุปกรณ์ทุกชนิด ตัวอย่างเช่น การเขียนโค้ดแสดงผลกราฟิก สมมติว่าฮาร์ดแวร์ที่กำลังทำงานอยู่สนับสนุนการหมุน (Rotate) และปรับขนาด (Scale) ภาพบิตแมพ ซึ่งการเรียกใช้งานไดเร็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็กซ์ เพื่อที่จะปรับขนาดและหมุนภาพบิตแมพได้นั้น จะต้องมีฮาร์ดแวร์ที่สนับสนุนการปรับขนาดและหมุนภาพ ซึ่งถ้าฮาร์ดแวร์สนับสนุนการทำงาน ก็จะทำงานได้เต็มความสามารถ และใช้งานฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ แต่ถ้าฮาร์ดแวร์ไม่สนับสนุนการปรับขนาดและหมุนภาพ การทำงานของ HEL จะเข้ามาทำงานแทนที่ โดย HEL จะทำการจำลองหน้าที่การทำงานของ HAL ด้วยอัลกอริทึมทางซอฟต์แวร์ โดยเราจะไม่ทราบถึงความแตกต่างเลย แต่ถึงอย่างไรก็ตามโค้ดที่ได้ทำงานนั้น ก็จะทำงานได้ช้าลง เพราะว่าเป็นการจำลองการทำงาน จึงไม่สามารถที่จะทำงานได้เหมือนอย่างฮาร์ดแวร์ร้อยเปอร์เซ็นต์

ไดเร็กเอ็กซ์ไดรเวอร์นี้จะรวมเข้าด้วยกันกับ ไดเร็กเอ็กซ์เอพีไอ (API) ผ่านลำดับของบัพเฟอร์อ็อบเจ็กต์ ผลลัพธ์ส่วนมากจะถูกสร้างโดยคอมโพเน็นต์ ไดเร็กเอ็กซ์ API จะถูกเขียนขึ้นเพื่อตอบสนองบัพเฟอร์อ็อบเจ็กต์ โดยบัพเฟอร์นี้จะถูกจัดการ และถูกทำให้มีประสิทธิภาพขึ้นตามฮาร์ดแวร์ที่มีอยู่ โดย 2 เลเยอร์ของสถาปัตยกรรมของไดเร็กเอ็กซ์และแปลงไปยังเอาท์พุตดีไวซ์ที่เหมาะสมอย่างเป็นระเบียบ ในระดับของภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้

คุณสมบัติของการจำลองการทำงานของฮาร์ดแวร์ ก็คือชุดของฟังก์ชันที่ทำการจำลองคำสั่งทาง 3 มิติ โดยใช้ซอฟต์แวร์ ถ้าฟังก์ชันที่ต้องการโดยเกมนั้น ไม่สามารถที่จะเข้าถึงฮาร์ดแวร์ได้โดยผ่าน HAL ไดเร็กเอ็กซ์นั้นจะสลับการทำงานมายัง HEL เพื่อที่จะจำลองการทำงานโดยใช้ซอฟต์แวร์แทน ด้วยคุณสมบัตินี้จะช่วยให้นักพัฒนาสามารถที่จะทดสอบหรือตรวจสอบคุณสมบัติทาง 3 มิติในโหมดของซอฟแวร์ได้ด้วยตนเอง

สิ่งเหล่านี้จะทำให้ไดเร็กเอ็กซ์มีความยืดหยุ่นในการใช้งานอย่างมาก ซึ่งจะช่วยสนับสนุนการติดตั้งใช้งานฮาร์ดแวร์ใดๆ ไม่ว่าจะเป็นรุ่นเก่าหรือใหม่ ได้อย่างหลากหลาย เช่น ถ้าเรามีแอพพลิเคชันที่ต้องการคุณสมบัติใหม่ๆ ที่อยู่ในฮาร์ดแวร์รุ่นใหม่ ก็ยังจะสามารถที่จะทำงานในฮาร์ดแวร์รุ่นเก่าได้ โดยผ่านชั้น HEL ของไดเร็กเอ็กซ์เพื่อที่จะจำลองความสามารถใหม่ที่ต้องการ ดังนั้น ถ้าเรายิ่งติดตั้งไดเร็กเอ็กซ์รุ่นใหม่เท่าใด จะช่วยในการทำงานของฮาร์ดแวร์ในคอมพิวเตอร์ ดังนั้นเกมที่ถูกเขียนมาในรูปแบบหนึ่ง จะได้รับข้อดีของคุณสมบัติใหม่ของ ฮาร์ดแวร์ที่ติดตั้งอยู่ในเครื่อง โดยการรวมเข้าด้วยกันกับไดเร็กเอ็กซ์

ตัวอย่าง เช่น ถ้ามีไดเร็กเอ็กซ์ที่เวอร์ชั่นใหม่กว่า ที่ออกแบบมาให้สามารถทำงานกับคอมโพเน็นต์ได้อย่างมีประสิทธิภาพโดยผ่านเทคโนโลยีแอมแอมเอ็กซ์ (MMX) ดังนั้นซอฟต์แวร์ทั้งหมดที่ถูกเขียนขึ้นมาก่อนที่ MMX จะออกจะสามารถใช้งาน MMX ได้ แม้ว่าแนวความคิดของ MMX ในสมัยนั้น จะยังไม่เกิดขึ้นในตอนที่ซอฟต์แวร์นั้นถูกเขียนขึ้น ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางกลับกันซอฟต์แวร์ที่ถูกเขียนขึ้น ในตอนหลังที่มี MMX แล้ว ก็ยังสามารถทำงานได้ในโปรเซสเซอร์รุ่นเก่าก่อนที่มี MMX ได้ โดย HEL ไดรเวอร์นั้นจะสามารถจำลองหน้าที่การทำงานของ MMX ให้มากที่สุดเท่าที่จะเป็นไปได้ โดยผ่านซอฟต์แวร์โดยอนุญาตให้ผู้ใช้ได้ใช้ความสามารถนั้นด้วย

ดังนั้นเหตุผลหลักของข้อดีเหล่านี้ เป็นเพราะว่าการออกแบบสถาปัตยกรรมให้เป็น COM ซึ่งสามารถช่วยให้คอมโพเน็นต์อ็อบเจ็กต์สามารถติดต่อสื่อสารกันได้

2.1.6 ส่วนประกอบของ direct X

Direct X มีส่วนประกอบที่สำคัญได้แก่

ไดเร็กซ์ดรอว์ (Direct Draw)

Direct Draw ถือว่าเป็นส่วนประกอบหลัก DirectX ทางด้านการแสดงผลเลยก็ว่าได้ เพราะไม่ว่าจะเป็น Direct3D หรือ Direct show ก็ต้องใช้งานร่วมกับ Direct Draw จึงจะแสดงผลหารทำงานได้

ถ้าเปรียบ Direct Draw เป็นอุปกรณ์ในเครื่องพีซีแล้ว ก็คงจะเป็นเสมือนจอภาพ (monitor)นั้นเอง เพราะสามารถควบคุมการปรับความละเอียดของหน้าจอโดยผ่าน Direct Draw ได้ และ Direct Draw ยังเป็นตัวกลางในการจัดการทุกสิ่งทุกอย่างที่เกี่ยวข้องกับการแสดงผลด้วย ไม่ว่าจะเป็นการจัดการแบบ 2 มิติ หรือแบบ 3 มิติ ก็ต้องกระทำผ่าน Direct Draw

ไดเร็กซ์ทรีดี (Direct 3D)

Direct 3D เป็นส่วนประกอบเสริมของ Direct Draw ใช้ในการประมวลผลแบบ 3 มิติ ซึ่งจะมีรูปแบบการทำงานที่ซับซ้อนขึ้นไปอีก

ไดเร็กซ์เซวน์ (Direct Sound)

Direct Sound เป็นส่วนหนึ่งของ Direct X ซึ่งมีไว้จัดการกับเสียงเอฟเฟ็กต์โดยเฉพาะ ไม่ว่าจะเป็นการกำหนดความดังของเสียง กำหนดตำแหน่งของเสียงทางลำโพงซ้ายขวา รวมทั้งยังสามารถเล่นเสียงหลายเสียงพร้อมกันได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดเร็กมิวสิก (Direct Music)

Direct Music เป็นส่วนหนึ่งของ Direct X ซึ่งมีไว้จัดการกับเสียงเพลง เช่นเพลงจากไฟล์ MID, แทร็กในแผ่นซีดี และ อื่นๆ อีกมากมาย

ในการใช้งาน Direct Music จะมีความยุ่งยากมากกว่า Direct Sound เพราะว่าจะมีส่วนประกอบย่อยๆ มากกว่า ประกอบด้วย Direct Music, Music Performance, Music Leader และ Music Segment

ไดเร็กอินพุท (Direct Input)

Direct Input เป็นตัวจัดการเรื่องของการรับข้อมูลมาจากอุปกรณ์โดยตรง เช่น คีย์บอร์ด เมาส์ จอยสติ๊ก เป็นต้น

ไดเร็กพเล (Direct Play)

Direct Play เป็นตัวจัดการด้านการสื่อสาร เช่น อินเตอร์เน็ท แลน โมเด็ม พอร์ตอนุกรม (Serial Port) สำหรับการสร้างเกมแบบเล่นด้วยกันหลายคน (MultiPlayer)

ไดเร็กโชว์ (Direct Show)

Direct Show เป็นตัวจัดการด้านระบบมัลติมีเดีย เช่น AVI , MP3, MPG เป็นต้น

2.2 Visual C/C++

เนื่องจากในการสร้างเกมนั้น Direct X นับว่าเป็นเครื่องมือที่มีประสิทธิภาพสูง และ เป็นที่ยอมรับใช้กันอย่างกว้างขวาง แต่ในการใช้งาน Direct X ให้ได้ประสิทธิภาพสูงสุดนั้นต้องใช้กับ Visual C/C++ เพราะ Direct X ได้มีการออกแบบเพื่อให้มาใช้งานกับ Visual C/C++ โดยเฉพาะ

Visual C/C++ ได้รับการพัฒนาขึ้นมาจาก Micrcosoft C/C++ ให้เป็น IDE ที่ทำงานบนระบบปฏิบัติการวินโดวส์ได้อย่างเต็มที่ รองรับการพัฒนาโปรแกรมบนวินโดวส์โดยมี MFC (Microsoft Development Kit) เป็นไลบรารีที่จะช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมบนวินโดวส์

2.3 แนวคิดพื้นฐานของเกม

เมื่อพูดถึงแนวคิดพื้นฐานของเกม ก็คือการกล่าวถึงการทำงานของเกมนั่นเอง ซึ่งหลักการทำงานของเกมนั้นก็คือการตอบโต้แบบกราฟฟิกอย่างฉับไวกับผู้เล่น หรือ ผู้ใช้งาน ทั้งยังมีระบบความคิดสังเคราะห์เป็นของตนเองในการคิดวางแผนโจมตีหรือตั้งรับตามลักษณะการรุกของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ความคิดสังเคราะห์นี้เป็นอังกอริทึมทางคณิตศาสตร์ที่โปรแกรมเมอร์ใส่เข้าไป ซึ่งนอกจากผู้สร้างเกมจะเขียนโปรแกรมเป็นแล้วจะเห็นว่าต้องวางแผน และ สร้างความคิดให้กับตัวละครในเกมได้ด้วย อาจจะเป็นอังกอริทึมแบบง่ายๆ เช่นผู้เล่นยิงปืนมาทางด้านขวาก็ให้หลบไปทางด้านซ้าย เป็นต้น นอกจากนี้เกมในปัจจุบันจะเห็นว่ามีกราฟฟิกที่สวยงามมาก ดังนั้นก่อนการลงมือเขียนโปรแกรม เราต้องวางแผนว่ามีองค์ประกอบใดอยู่ในโปรแกรมของเราบ้าง จากนั้นก็ให้ออกแบบกราฟฟิกขององค์ประกอบนั้นๆ ออกมาก่อนแล้วจึงนำไปบรรจุในโปรแกรมของเรา

พื้นฐานการทำงานของเกม แบ่งออกเป็น 5 ส่วน คือ

- เนื้อหาเกม (Actual game content) คือเนื้อเรื่องในฉากที่ดำเนินอยู่
- การเลื่อนฉาก (Scrolling engine) ใช้เลื่อนฉากเมื่อตัวละครเคลื่อนไปเคลื่อนมาในฉาก
- การทำงานกับสคริบต์ (Scripting engine) สร้างเหตุการณ์ภายในเกมด้วยคำสั่งแบบข้อความโดยนำข้อความไปเทียบในตารางที่เก็บคำสั่งไว้จากนั้นก็สร้างเหตุการณ์ตามคำสั่ง และพารามิเตอร์ที่ส่งตามสคริบต์
- การต่อสู้ (Battle engine) เป็นระบบที่ควบคุมการต่อสู้ของผู้เล่นกับศัตรูว่ามีการหักคะแนนของผู้เล่นอย่างไร สามารถชนะกันได้อย่างไร เป็นต้น
- เมนูของเกม (Menu system) ที่ควบคุมลักษณะการแสดงผลกราฟฟิก การเล่นเสียงดนตรี เป็นต้น

2.4 Computer Graphic และ Image Processing

ในการสร้างเกมนั้นส่วนสำคัญส่วนหนึ่งที่ขาดไม่ได้ก็คือการจัดการเกี่ยวกับกราฟฟิกต่างๆ ซึ่งจะให้การทำงานที่ได้มีประสิทธิภาพ จะมีการนำความรู้ของวิชา Computer Graphic และ Image Processing มาใช้ด้วยเพราะฉะนั้นจึงต้องศึกษาบางส่วนเพื่อนำมาใช้งานควบคู่กับการใช้งาน Direct X

2.4.1หลักการปรับภาพความสว่างของภาพ (Brightness)

แม่สีของแสงแต่ละสี ซึ่งได้แก่ แดง เขียว และน้ำเงินนั้นจะมีค่าตั้งแต่ 0x00 ถึง 0xff ซึ่งจะเห็นว่า 0x00 นั้นเป็นสีดำ ส่วน 0xff จะเป็นสีของแม่สีที่สว่างที่สุด เช่น ถ้าเป็นค่าของสีแดงก็เป็นแดงสด ดังนั้นการปรับ ความมืด ความสว่างของภาพก็ไม่มีอะไรที่ซับซ้อน เราเพียงแค่เพิ่มหรือลดค่าสีของแม่สีแต่ละสีเท่านั้นเอง ถ้าเพิ่มค่าก็ทำให้สว่างขึ้น ถ้าลดค่าก็ทำให้มืดลง แต่การเพิ่มหรือลดค่าสีของแม่สีนี้จะต้องเพิ่มหรือลดให้เท่าๆ กัน เพราะเราต้องทำให้โทนสีของพิกเซลเป็นโทนเดิมด้วย เช่น ถ้าพิกเซลเป็นสีม่วง เมื่อเราเพิ่มค่าแม่สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่รวมกันเป็นสีม่วงแล้ว จะต้องทำให้พิกเซลสีม่วงเป็นสีม่วงอ่อน (สีจะสว่างขึ้น) นั่นคือต้องเพิ่มค่าแม่สีเท่าๆ กัน ถ้าหากเพิ่มไม่เท่ากันแล้ว จะทำให้สีเปลี่ยนไป

แต่มีข้อแม้บางอย่าง นั่นคือถ้าเราเพิ่มหรือลดค่าของแม่สีใดไปจนสุด เช่น ลดไปจนถึง 0x00 หรือเพิ่มไปจนถึง 0xff ค่าสีของแม่สีนั้นก็จะหยุดอยู่แค่นั้น แต่แม่สีอื่นที่ค่ายังไม่ถึงก็ยังคงสามารถเพิ่มหรือลดไปพร้อมๆ กันได้อีก เมื่อเป็นเช่นนี้จะเห็นว่า ในที่สุดแล้วหากลดค่าสีไปเรื่อยๆ ก็จะทำให้สีของพิกเซลเป็นสีดำ (0x000000) ซึ่งมืดที่สุด หรือหากเพิ่มค่าสีไปเรื่อยๆ ก็จะทำให้พิกเซลเป็นสีขาว (0xffffff) หรือสว่างที่สุด ดังนั้น เมื่อเราทำกับทุกๆ พิกเซลแล้ว ก็จะทำให้ภาพมืดลงหรือสว่างขึ้นได้

2.4.2 หลักการปรับความเข้มของสี (Contrast)

ทำได้โดยการลองปรับปุ่ม Contrast ของจอภาพคอมพิวเตอร์ จะเห็นว่า ถ้าลด Contrast ลง จะทำให้ภาพที่ปรากฏค่อยๆ กลายเป็นสีเทา แต่ถ้าปรับให้มากขึ้นสีของภาพก็จะเด่นขึ้นมา โดยหลักการง่ายๆ แล้ว Contrast จะมีค่ากลางค่าหนึ่งเข้ามาเกี่ยวข้อง คือว่าถ้าปรับ Contrast ให้สูงขึ้น จะหมายถึงการปรับค่าของแม่สีให้ห่างจากค่ากลางนี้ โดยที่ถ้าค่าของแม่สีเดิมมากกว่าค่ากลางก็ให้เพิ่มค่าของแม่สีให้มากขึ้น แต่ถ้าค่าของแม่สีน้อยกว่าค่ากลางก็ให้ลดค่าของแม่สีให้น้อยลง ส่วนการปรับ Contrast ให้ต่ำลงจะหมายถึงการปรับค่าสีของแม่สีให้เข้าใกล้ค่ากลางมากที่สุด นั่นคือถ้าค่าของแม่สีเดิมมีค่ามากกว่าค่ากลางก็จะปรับลดค่าของแม่สีนั้นลง แต่ถ้าค่าเดิมของแม่สีมีค่าน้อยกว่าค่ากลางก็ให้ปรับค่าของแม่สีนั้นให้สูงขึ้น โดยค่าของแม่สีที่ปรับขึ้นหรือลงนี้ เมื่อปรับแล้วจะต้องมีค่าอยู่ในช่วง 0x00 ถึง 0xff

ดังนั้นในการปรับ Contrast ขึ้นก็ไม่ได้หมายความว่าต้องเพิ่มค่าของแม่สีเสมอไป และการปรับ Contrast ลง ก็ไม่ได้หมายความว่าต้องลดค่าของแม่สีเสมอไป โดยมีสูตรการคำนวณดังนี้

ค่าใหม่ของแม่สี = ((ค่าเดิมของแม่สี – ค่ากลาง) * Contrast) + ค่ากลาง

โดยที่ค่าของ Contrast จะเป็นเลขจำนวนจริง มีค่าตั้งแต่ 0.00 ขึ้นไป จะเห็นว่า ถ้าค่าของ Contrast ต่ำที่สุด คือเป็น 0.00 จะทำให้ค่าแม่สีที่ได้เป็นค่ากลาง ซึ่งค่ากลางดังกล่าวจะเป็นค่าใดก็ได้ แต่จะต้องอยู่ในช่วง 0x00 ถึง 0xff ซึ่งโดยปกติแล้วการกำหนดค่ากลางจะขึ้นอยู่กับความต้องการ ว่าถ้าปรับ Contrast จนต่ำสุดแล้วจะให้ภาพเป็นสีเทาในโทนสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบไหน ถ้าต้องการให้เป็นสีเทาแบบปกติก็ระบุค่ากลางเป็น 0x7f หรือ 127 ซึ่งเป็นค่าที่อยู่กึ่งกลางระหว่าง 0x00 กับ 0xff ถ้าจะให้เป็นสีเทาอ่อนก็กำหนดให้มันสูงกว่า 0x7f เป็นต้น ดังนั้น อาจจะเรียกค่ากลางดังกล่าวได้ว่าค่าของสีเทา

2.4.3 การทำภาพพื้นผิว 2 มิติ ให้เป็น 3 มิติ (Bump Mapping)

Bump Mapping เป็นเทคนิคที่ทำให้ภาพพื้นผิว 2 มิติ ดูมีมิติ มีส่วนลึกและนูน แล้วก็มีแสงเงามืดและสว่าง ซึ่งจะเปลี่ยนไปตามตำแหน่งของแหล่งกำเนิดแสง เทคนิคนี้มักนิยมใช้กันในโปรแกรมสร้างภาพ 3 มิติ แล้วก็เกมทั้งหลาย

ที่จริงแล้ว Bump Mapping สามารถทำได้หลายวิธี บางวิธีก็สามารถทำได้โดยอาศัยการ์ดเร่งความเร็ว 3 มิติ เพื่อให้การสร้างภาพออกมาเร็วยิ่งขึ้น แต่โดยพื้นฐานที่ง่ายที่สุดอีกวิธีหนึ่งนั้น จะใช้วิธี Emboss Bump Mapping โดยเราจะต้องเขียนโปรแกรมเพื่อคำนวณด้วยตัวเอง เรียกว่าเป็น Software Rendering โดยมีขั้นตอนดังนี้

1) Texture Map เป็นภาพสีที่แสดงถึงพื้นผิว 2 มิติธรรมดาๆ ที่ต้องเตรียมไว้

2) Bump Map เป็นภาพขาวดำ (Grayscale) ที่แสดงถึงความสูงต่ำของ Texture Map โดยส่วนที่เป็นสีขาวจะหมายความว่าส่วนนั้นจะนูนขึ้นมา และส่วนที่เป็นสีดำจะเป็นส่วนที่ยุบลงไป ซึ่งส่วนที่นูนและยุบจะมีหลายระดับ ขึ้นอยู่กับค่าของสีขาวดำ ถ้าขาวมากก็จะนูนมาก จริงๆ แล้วภาพ Bump Map นี้สามารถเขียนโปรแกรม Convert ภาพ Textre Map ให้เป็นขาวดำได้โดยอัตโนมัติ โดยใช้วิธีการแปลงภาพสีให้เป็นภาพขาวดำแบบ Grayscale แต่ความลึกและนูนอาจไม่เป็นไปตามที่เราต้องการ ดังนั้นสร้างเอาเองจะดีกว่า วิธีที่ใช้สร้างภาพนี้จะใช้ Photoshop แปลงภาพ Textre Map ให้เป็นขาวดำ แล้วแต่งสีใหม่ โดยให้ส่วนที่เป็นข้อความมีสีขาว เพื่อที่มันจะได้ดูนูนขึ้นมาเด่นกว่าเดิม ซึ่งเวลาเขียนโปรแกรมเพื่อดึงค่าของสีจากภาพ Grayscale นี้ คุณสามารถดึงค่าของแม่สีใดมาใช้ก็ได้ เพราะแต่ละแม่สีจะมีค่าเท่ากัน เช่น สีเทาปานกลางจะมีค่า RGB เป็น 0x7f7f7f ซึ่งไม่ว่าจะดึงแม่สีใดก็จะได้ 0x7f

3) Light เป็นอาร์เรย์ของค่าตัวเลขในระดับต่างๆ ที่เรียงกันเพื่อแทนระดับความมืดและความสว่างของแสงหรือ Brightness ซึ่งก็ขึ้นอยู่กับคุณว่า จะให้แสงออกมาในรูปแบบไหน เช่น Spot Light หรือแสงจากดวงไฟ เป็นต้น แต่อย่าลืมว่าต้องมีส่วนมืดและสว่างสำหรับขนาดของแสงก็กำหนดได้ตามต้องการ แสงที่ใช้นี้จะเป็นแสงจากดวงไฟ โดยวางตำแหน่งของดวงไฟ หรือแหล่งกำเนิดแสนงไว้ตรงกลาง แล้วแสงจะกระจายออกไปในแนววงกลม มีขนาดเส้นผ่านศูนย์กลาง 256 พิกเซล คือสร้างอาร์เรย์ของแสงขนาด 256 คูณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

256 แล้วก็คำนวณค่าความมืดและสว่างโดยใช้วิธี Radial Gradient การไล่สีในแนววงกลม ซึ่งจะไล่สีในแนววงกลมจากสีดำที่อยู่รอบนอกไปเป็นสีขาวที่อยู่ด้านใน ซึ่งผลลัพธ์จากการไล่สีจะได้ค่า 0 ถึง 255 แต่เพื่อไม่ให้ภาพผลลัพธ์ Bump Mapping ดูสว่างจนเกินไป ก็สามารถทำให้มันเป็นค่าที่มีทั้งบวกและลบ โดยการลดค่าตามตัวแปรคงที่ BRIGHTNESS ซึ่งมีค่าเป็น –50 ซึ่งก็จะทำให้อาร์เรย์ของแสงมีค่าอยู่ในช่วง –50 ถึง 205 โดยที่ค่าติดลบจะหมายถึงลด Brightness ค่าบวกก็คือเพิ่ม Brightness

4) Light Map เป็นภาพขาวดำ (Grayscale) ที่ถูกสร้างขึ้นโดยการคำนวณจากค่าของสีเทาในภาพ Bump Map กับอาร์เรย์ของแสง โดยการวางตำแหน่งของแหล่งกำเนิดแสงไว้ในตำแหน่งที่ต้องการ ซึ่งถ้าตำแหน่งของแหล่งกำเนิดแสงเปลี่ยนไป ภาพ Light Map ก็จะเปลี่ยนไป ซึ่งภาพ Light Map นี้ จะแสดงถึงความลึกและความนูนของพื้นผิว รวมทั้งความมืดและความสว่างด้วย

5) ผลลัพธ์จากการทำ Bump Mapping นำภาพ Textre Map มาเพิ่มและลด Brightness ตามค่าที่ได้จากภาพ Light Map ทีละพิกเซล (อาจกล่าวได้ว่าภาพ Light Map ก็คือภาพที่แสดงถึงค่าของ Brightness) โดยเอาค่านี้ไปบวกหรือลบค่าของแม่สีในทุกๆ พิกเซลตามเรื่องหลักการปรับ Brightness หรือความสว่างของภาพ ก็จะได้ภาพ Bump Mapping ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันวิทยบริการ พระจอมเกล้าลาดกระบัง

2.5 Artificial intelligence [3]

Artificial Intelligence (AI) ตามความหมายแล้ว หมายถึง การสร้างสรรค์สิ่งต่าง ๆ ซึ่งทำให้ส่วนหนึ่งหรือหลาย ๆ ส่วนของ software หรือ hardware ได้เกิดการคิดขึ้น ซึ่งเป็นกระบวนการทางด้านข้อมูล (Information) ให้คล้ายกับมนุษย์ซึ่งสามารถคิดได้ โดยการคิดนั้นจะต้องคิดอย่างมีเหตุผล กระทำอย่างมีเหตุผล โดยการหาคำตอบของปัญหานั้นทำได้หลายวิธี ซึ่งการหาคำตอบที่ง่ายและเข้าใจได้ง่ายคือการหาคำตอบแบบ Tree decision

วิธีการค้นหาคำตอบเพื่อแก้ไขปัญหานั้นมีหลายวิธีซึ่งแต่ละวิธีจะมีข้อดี ข้อเสียต่างกัน โดยพิจารณาจากพารามิเตอร์ 4 ตัวคือ

1 Completeness หมายถึง การค้นหาคำตอบนั้นจะให้ความแน่ใจว่าจะได้คำตอบ

2 Time Complexity หมายถึง เวลาที่ใช้ในการค้นหาคำตอบ

3 Space complexity หมายถึง จำนวนหน่วยความจำที่ใช้สำหรับการค้นหาคำตอบ

4 Optimality หมายถึง ความเหมาะสมในการหาคำตอบที่ต้องการผลลัพธ์ที่ดีที่สุด

2.5.1 Breadth First Search

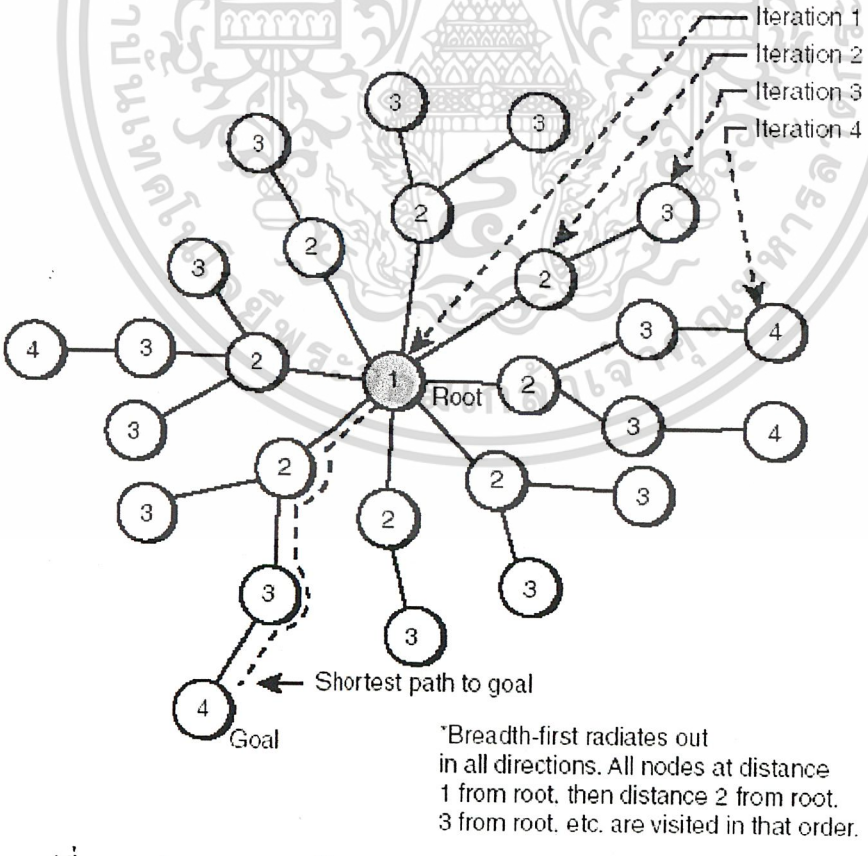
Breadth First Search เป็นการค้นหาคำตอบในทุกทิศทางในเวลาเดียวกัน ซึ่งลักษณะการค้นหาคำตอบจะเป็นคือ เมื่อโหนดเริ่มต้น (Root Node) ทำการแตกโหนด (Node) เป็นโหนดลูก (Child Node) แล้วทุก ๆ โหนดลูกจะทำการแตกโหนดลูกต่อไปเพื่อทำการค้นหาคำตอบต่อไป

```
void BreadthFirstSearch(NODE r)
{
NODE s; // used to scan
QUEUE q; // this is a first in first out structure FIFO
// empty the queue
EmptyQ(q);
// visit the node
Visit(r);
Mark(r);
// insert the node r into the queue
InsertQ(r, q);
// while queue isn't empty loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50274

```
while (q is not empty)
{
NODE n = RemoveQ(q);
for (each unmarked NODE s adjacent to n)
{
// visit the node
Visit(s);
Mark(s);
// insert the node s into the queue
InsertQ(s, q);
} // end for
} // end while
} // end BreadthFirstSearch
```



รูปที่ 2 .4 รูปแสดงการแตกโหนดของการคำตอบแบบ Breadth first search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี สามารถหาคำตอบได้อย่างแน่นอน (Complete)
เหมาะสมกับการค้นหาคำตอบที่การผลลัพธ์ที่ดีที่สุด (Optimal)
ข้อเสีย ใช้เวลามากในการค้นหาคำตอบ (more time)
ใช้เนื้อที่ memory มากการค้นหาคำตอบ (more Space)

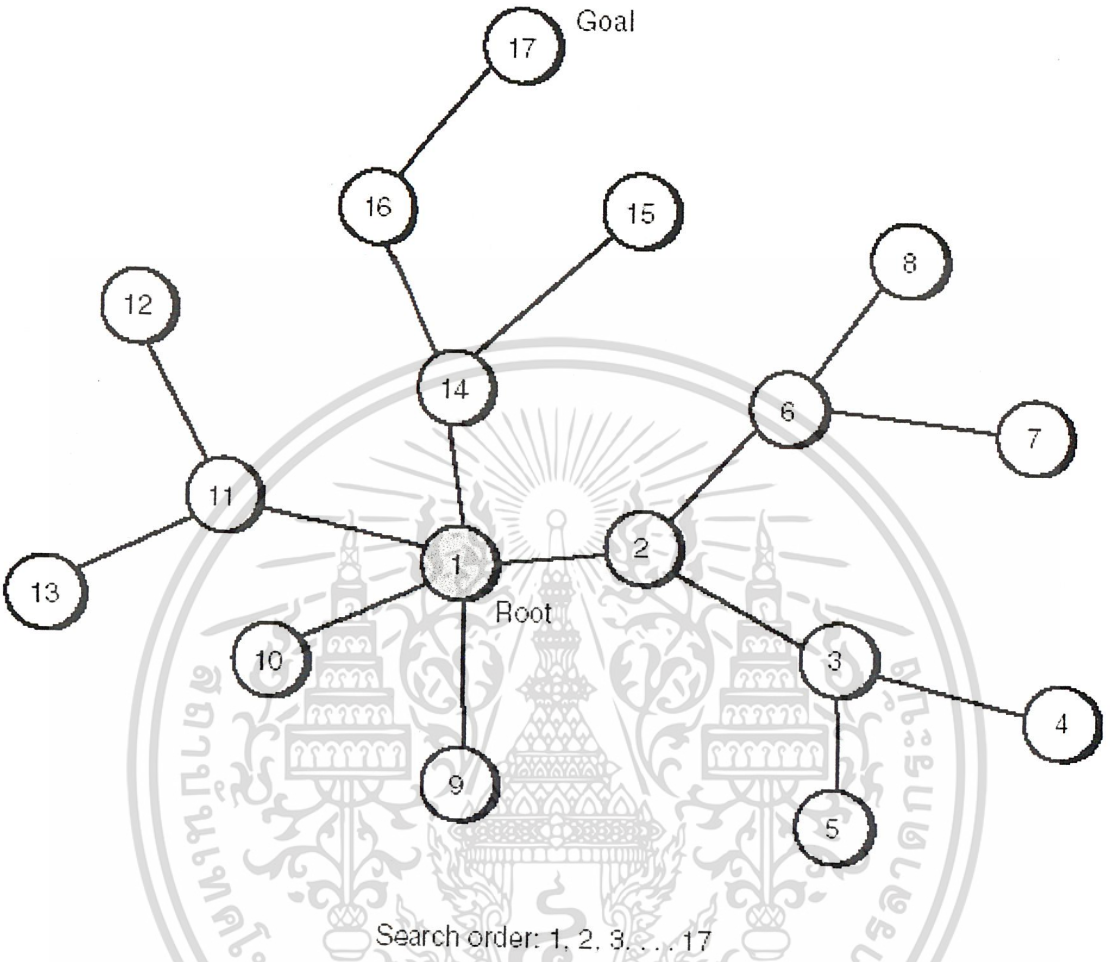
2.5.2 Depth First Search

Depth First Search มีลักษณะตรงกันข้ามกับ Breath First Search คือจะทำการค้นหาคำตอบในทิศทางเดียวไปจะกว่าจะไม่สามารถหาคำตอบได้ในทิศทางการค้นหาคำตอบหรือ เมื่อพบคำตอบ ในกรณีเมื่อไม่พบคำตอบจะทำการค้นหาคำตอบในทิศทางถัดไป

ซึ่งการหาคำตอบแบบ Depth First Search นั้นอาจจะกำหนด Limit ในการค้นหาคำตอบ เช่น ถ้าทำการ ค้นหาคำตอบไปในทิศทางนั้น 100 โหนดแล้วไม่พบคำตอบให้ทำการเปลี่ยนทิศทางการค้นหาคำตอบ

```
void DepthFirstSearch(NODE r)
{
NODE s; // used to scan
// visit and mark the root node
visit(r);
mark(r);
// now scan along from root all the nodes adjacent
while (there is an unvisited vertex s adjacent to r)
{
DepthFirstSearch(s);
} // end while
} // end DepthFirstSearch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 รูปแสดงการแตกโหนดของการคำตอบแบบ Depth first search

ข้อดี ใช้เนื้อที่ memory การค้นหาคำตอบน้อยกว่าแบบ Breadth First search

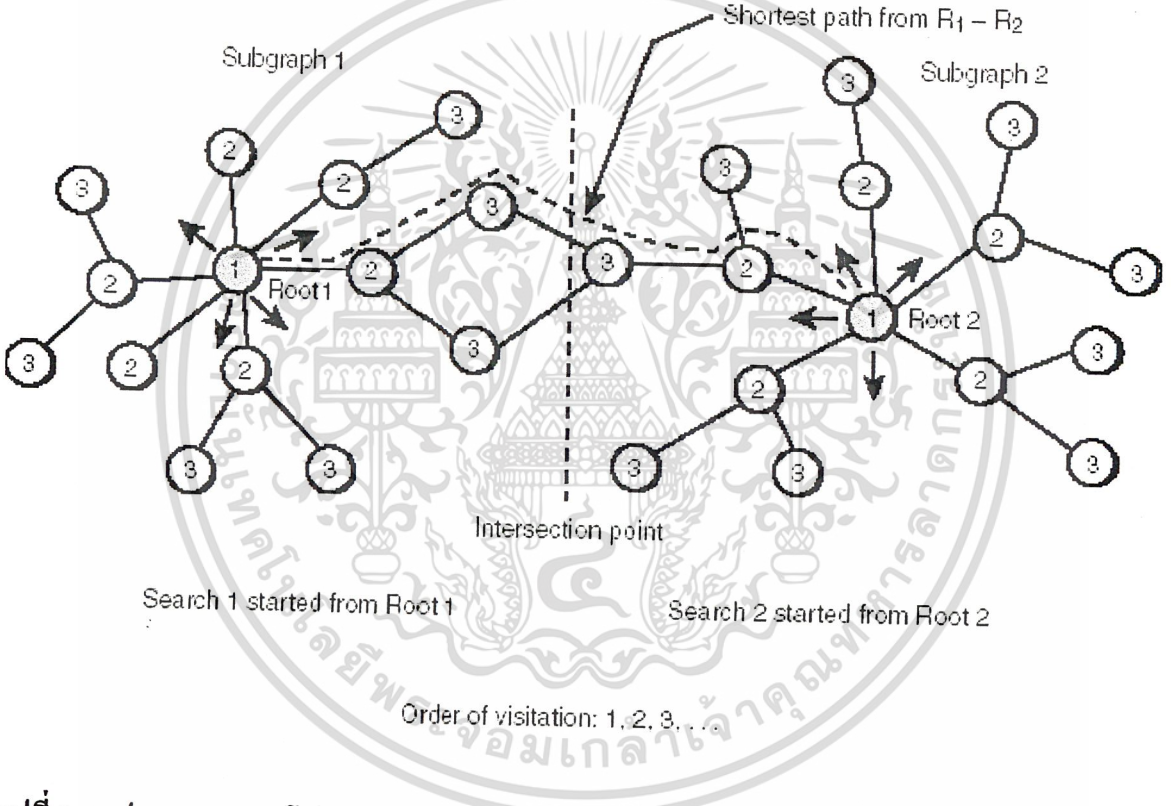
ข้อเสีย ไม่สามารถหาคำตอบได้ในกรณีที่ทิศทางในการหาคำตอบนั้นไม่มีที่ขอบเขตในการหาคำตอบ (No limited space) เมื่อไม่สามารถหาคำตอบได้จึงไม่ complete และเนื่องจากไม่สามารถหาคำตอบได้จึงไม่ Optimal

ใช้เวลามากในการค้นหาคำตอบ (more time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Bidirectional Breadth First-Search

Bidirectional Breadth มีรูปแบบการค้นหาคำตอบคล้ายกับ Beradth First-Search ซึ่งจุดเริ่มต้นของการค้นหา นั้นจะแตกต่างกันคือ Bidirectional Beradth First-Search จุดเริ่มต้นของการ Search นั้นจะเริ่มจาก จุดเริ่มต้น และ จุดปลายซึ่งเป็นจุดปลายทาง และเมื่อโหนดที่แตกออกนั้นชนกัน (Overlap) กันก็จะสามารถหาเส้นทางที่สั้นที่สุดในการหาคำตอบได้



รูปที่ 2.6 รูปแสดงการแตกโหนดของการคำตอบแบบ Bidirectional Breadth First-Search

ข้อดี สามารถหาคำตอบได้อย่างแน่นอน (Complete)
เหมาะสมกับการค้นหาคำตอบที่การผลลัพธ์ที่ดีที่สุด (Optimal)

ข้อเสีย ใช้เวลามากในการค้นหาคำตอบ (more time)
เนื้อที่ memory มากการค้นหาคำตอบ (more Space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 A* algorithm

A* algorithm เป็นการค้นหาหาคำตอบแบบมีทิศทางกล่าวคือ จะมีตัวช่วยในการตัดสินใจว่าควรจะ ค้นหาคำตอบไปทางใดเพื่อให้ได้คำตอบตามที่ต้องการ ซึ่งเป็น function ประกอบด้วย

$$f(n) = g(n) + h(n)$$

โดยมีค่าพารามิเตอร์ 3 ตัวคือ f(n) , g(n) และ h(n) ซึ่งแต่ละตัวมีความหมายดังต่อไปนี้

f(n) = ค่าในการตัดสินใจ เช่น กำหนดให้ search ไปทางทิศทางที่มีค่าน้อยที่สุด

g(n) = path cost ในการคิดคำนวณค่าตัดสินใจโดยเป็นค่าระยะทางจริงในการเดินทาง

h(n) = evaluation function ซึ่งเป็น function ที่สร้างขึ้นมาคำนวณค่าในการตัดสินใจ ซึ่งโดยทั่วไปจะเป็นระยะทางจริงจากจุดหนึ่งไปยังอีกจุดหนึ่ง

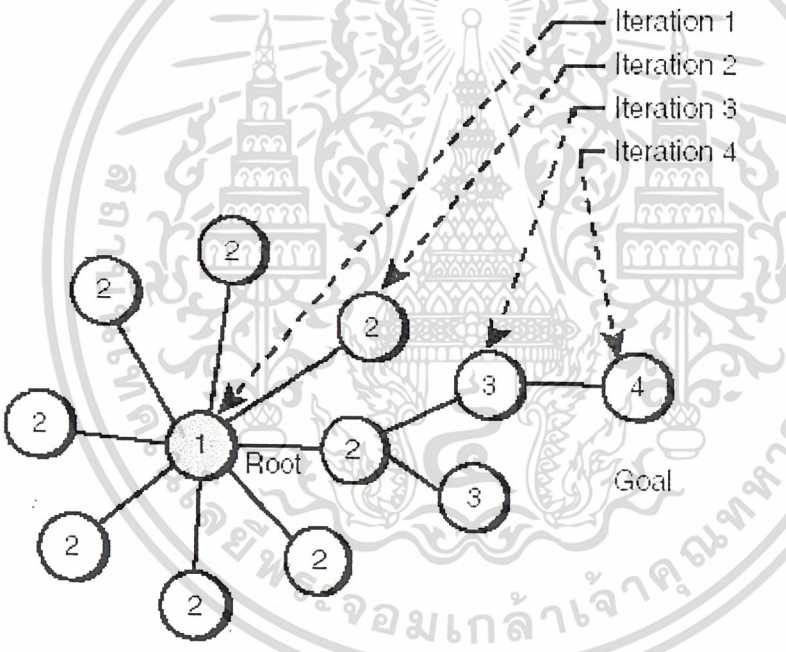
การหาคำตอบว่าจะไปยังโหนดใดนั้นจะตัดสินโดยดูจากค่า f(n) ที่ได้ เพื่อที่จะทำให้สามารถค้นหาจุดปลายทาง (Goal) ได้จะต้องมีรายละเอียด (list) ในการเก็บข้อมูลของโหนดที่ยังไม่ได้แตกโหนด (Open Node) และโหนดที่ได้แตกโหนด (Closed Node) โดย Open Node จะหมายถึงทุกๆโหนดที่ยังไม่ได้ทำการแตกโหนดนั้นแต่ไม่รวมถึงโหนดที่ไม่สามารถทำการแตกโหนดได้ ส่วน Closed Node นั้นคือโหนดที่ได้ทำการแตกโหนดแล้ว โดยแต่ละโหนดจะต้องเก็บค่า f(n) เพื่อใช้ในการพิจารณาในการตัดสินใจว่าจะไปยังทิศทางใด

```
void AStarSearch(NODE r)
{
NODE Bestnode;
NODE s;          // used to scan
// insert  root  node in openlist
OpenList(r);
// while Openlist is not empty
while (Openlist is not empty)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//selcet Bestnode is in Openlist from f(n)
Bestnode = GetBestnodeFormOpenlist();
CloseList(BestNode);

for (each unmarked NODE s adjacent to Bestnode)
        OpenList(s);
} // end while
} // end AstarSearch
```



รูปที่ 2.7 รูปแสดงการแตกโหนดของการคำตอบA* Algorithm

ข้อดี สามารถหาคำตอบได้อย่างแน่นอน (Complete)
เหมาะสมกับการค้นหาคำตอบที่การผลลัพธ์ที่ดีที่สุด (Optimal)
ใช้เวลาน้อยกว่าในการค้นหาคำตอบ แบบอื่นข้างต้น (less time)
เนื้อที่ memory น้อยการค้นหาคำตอบ (less Space)

ข้อเสีย การค้นหาคำตอบแบบ A* Algorithm จะขึ้นอยู่กับ heuristic Function (h(n))

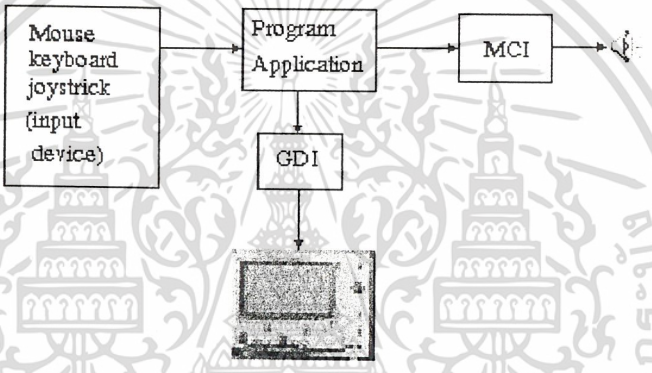
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าเมื่อเปรียบเทียบ A* Algorithm กับการค้นหาคำตอบโดยวิธีอื่นแล้วจะเห็นได้ว่าการค้นหาคำตอบโดยวิธีอื่นเป็นการค้นหาคำตอบแบบไม่มีทิศทางซึ่งเมื่อเปรียบเทียบ กับ A* Algorithm แล้วจะได้ว่า Completeness, Time Complexity, Space Complexity และ Optimal มีความเหมาะสมมากที่สุดในการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

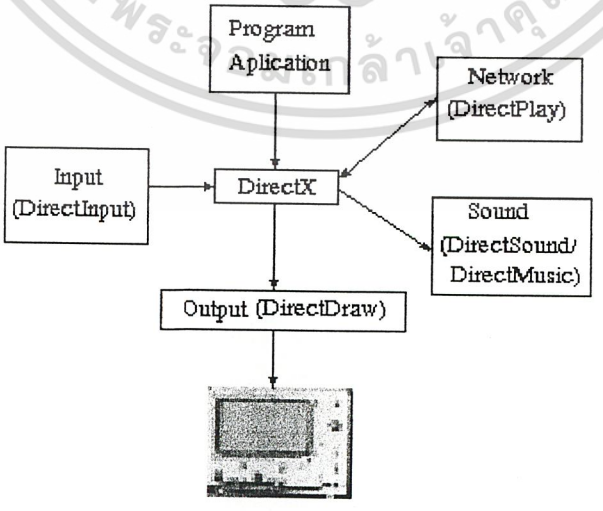
บทที่ 3
ทฤษฎี และการออกแบบ

วิธีเขียนเกมทั่วไปถ้าไม่ใช่ชุดคำสั่งของ Direct X จะใช้ชุดคำสั่งของ Windows คือ GDI (Graphic User Interface) และ ชุดคำสั่ง Media Control Interface (MCI) จาก Win32 API ในการเล่นไฟล์เสียง wav หรือการแสดงรูปออกทางหน้าต่าง ดังรูป



รูปที่ 3.1 แสดงรูปการทำงานของเกมคอมพิวเตอร์โดยใช้ GDI

การใช้ชุดคำสั่ง Direct X จะมีโครงสร้างในการเขียนดังนี้



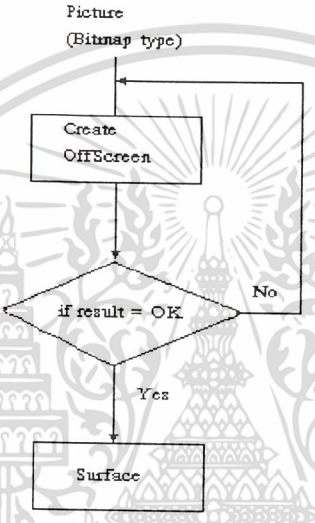
รูปที่ 3.2 แสดงรูปการทำงานของเกมคอมพิวเตอร์โดยใช้ Direct X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้น Direct X จะเป็นศูนย์กลางผ่านไปสู่ระบบ hardware แทนที่เราจะใช้ชุดคำสั่งด้าน hardware โดยตรงชุดคำสั่งใน Direct X จะอำนวยความสะดวกให้ในการแสดงผล และการรับส่งข้อมูล

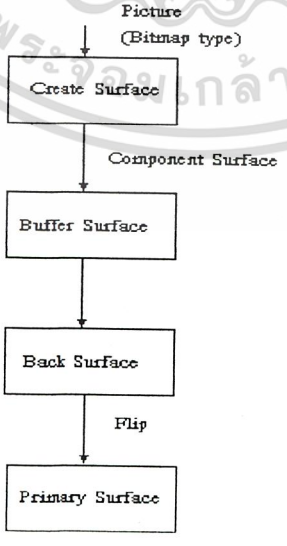
3.1 การสร้างพื้นผิว (Surface)

การสร้าง Surface เริ่มจากการนำรูป Bitmap นำมาสร้าง surface เพื่อะนำมาใช้ในการแสดงภาพ



รูปที่ 3.3 การสร้าง Surface ของเกม

3.2 การแสดงผลภาพ

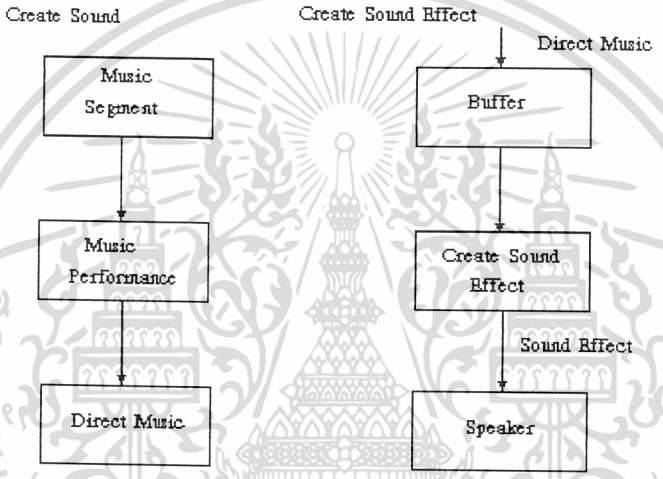


รูปที่ 3.4 การนำภาพมาแสดงผลที่หน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำภาพมาแสดงผลเริ่มจากสร้าง Surface เมื่อได้ Surface แล้วทำการวาดภาพลงใน Buffer Surface จากนั้นทำการนำภาพบางส่วนจากพื้นผิวสำรอง (Buffer Surface) มาเก็บไว้ในพื้นผิวที่จะทำการแสดงผล (Back Surface) เพื่อเตรียมแสดงผล จากนั้นทำการสลับ (Flip)

3.3 การสร้างเสียง



รูปที่ 3.5 การแสดงผลของเสียงประกอบ

การทำงานของ DirectMusic จะเริ่มโดย Music Performance จะเรียกข้อมูลจาก Music Segment มาแปล แล้วส่งไปให้ DirectMusic สร้างเสียงเพลงออกมา หลังจากนั้นถ้าต้องการนำเสียงที่ได้มาใส่ Effect เช่น ปรับความดัง ค่อย จะทำโดยนำข้อมูลเสียงมาเก็บไว้ใน Buffer จากนั้นทำการ Create Sound Effect เพื่อนำออกสู่ลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบในส่วนของตัวเกม

เกม Battle City Extreme จะมีแนวการเล่นแบบเกมกาษา ซึ่งจะเป็นเรื่องราวโดยจะมีเพื่อนหรือบุคคลภายในเกมจะบอกข้อมูลเพื่อให้สามารถผ่านด่านนั้นง่ายขึ้น ภายในเกมจะมีการซื้อของเพื่อนำมาแต่งรถถังเพื่อให้มีประสิทธิภาพมากขึ้นโดยราคานั้นจะขึ้นอยู่กับพลังโจมตีของอาวุธ พลังป้องกันและประเภทของอาวุธ และลักษณะการซื้อขายนั้นจะใช้เงินในการซื้อขายซึ่งเงินนั้นจะได้จากการเก็บของภายในฉากต่อสู้ ราคาการขายขึ้นอยู่กับว่าของชนิดนั้นหาง่ายหรือยาก การคำนวณค่าพลังโจมตีของอาวุธจะขึ้นอยู่กับระยะทาง และความแรงของอาวุธซึ่งถ้าโจมตีจากระยะทางที่ไกลความแม่นยำของอาวุธจะลดลง ในการออกแบบการควบคุมรถถังของ Computer นั้นจะขึ้นอยู่กับอาวุธที่ติดตั้งให้กับรถถังนั้นและ คำสั่งของรถถังที่ได้รับ

3.4.1 รูปแบบรถถัง

ในเกม Battle City Extreme นี้ รถถังแต่ละคันนั้นจะประกอบด้วยส่วนสำคัญอยู่ 5 ส่วนด้วยกันคือ ส่วนตัวรถ ส่วนอาวุธ ส่วนล้อ ส่วนเครื่องหลัง และ ส่วนของ คอมพิวเตอร์ ซึ่งผู้เล่นสามารถนำแต่ละส่วนมาประกอบกันเป็นรถถังในแบบของตนเองได้ แต่ละชิ้นส่วนนั้นจะมีหน้าที่ที่แตกต่างกัน การที่ผู้เล่นจะได้รถถังที่ดีนั้น ย่อมหมายถึงความสมดุลของแต่ละชิ้นส่วนที่นำมารวมกัน

3.4.1.1 ส่วนตัวรถ

แนวคิดของเกม

ในส่วนนี้จะเป็นส่วนของ Body ของรถ ซึ่งเป็นส่วนของคนขับ ในฉากต่อสู้หากรถถังส่วนนี้ถูกทำลายจะถือว่ารถถังถูกทำลาย เพราะฉะนั้น ในการปรับแต่งการเลือกส่วนตัวรถที่มีความทนทานสูงย่อมได้เปรียบกว่าตัวรถที่มีความทนทานน้อย แต่ว่าตัวรถที่มีความทนทานสูงก็มีข้อจำกัดในส่วนของน้ำหนักซึ่งจะเพิ่มตามไปด้วยจึงทำให้ในการปรับแต่งรถสามารถบรรทุกได้น้อยลง

ในส่วนของเกมจะแบ่งรูปแบบของตัวถังรถเป็น 2 แบบด้วยกัน คือ ตัวถังแบบเกราะหนัก และตัวถังแบบเกราะเบา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ

ตัวถังชนิดเกราะหนัก

ตัวถังชนิดนี้จะเป็นตัวถังที่มีความทนทานสูง แต่ก็มีน้ำหนักมาก

ชื่อ	พลัง	ป้องกัน	น้ำหนัก	ราคา
เกราะหนัก-1	56	2	30	150
เกราะหนัก-2	108	4	33	550
เกราะหนัก-3	216	8	36	2200
เกราะหนัก-4	432	16	39	8400
เกราะหนัก-5	864	32	42	32000

ตารางที่ 3.1 ตารางการกำหนดค่าพารามิเตอร์ของตัวถังรถประเภทเกราะหนัก

ตัวถังชนิดเกราะเบา

ตัวถังชนิดนี้จะมีความทนทานต่ำกว่าตัวถังชนิดแรก แต่จะมีน้ำหนักเบากว่า

ชื่อ	พลัง	ป้องกัน	น้ำหนัก	ราคา
เกราะเบา-1	42	1	26	130
เกราะเบา-2	84	2	29	500
เกราะเบา-3	168	4	32	2000
เกราะเบา-4	336	8	35	7600
เกราะเบา-5	672	16	38	28000

ตารางที่ 3.2 ตารางการกำหนดค่าพารามิเตอร์ของตัวถังรถประเภทเกราะเบา

ค่าพารามิเตอร์

พลัง คือ ขีดจำกัดของชิ้นส่วนนั้นๆ เมื่อถูกโจมตีพลังจะลดลง เมื่อชิ้นส่วนใดมีพลังเป็นศูนย์จะถือว่าชิ้นส่วนนั้นถูกทำลายและไม่สามารถใช้งานได้อีก

ป้องกัน คือ ค่าความทนทานต่อการโจมตีของตัวถังรถ

น้ำหนัก คือ ค่าที่แสดงน้ำหนักของตัวถังรถ

ราคา คือ มูลค่าของตัวถังรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.2 ส่วนอาวุธ

แนวคิดของเกม

ในเกมแบ่งรูปแบบของอาวุธเป็นรูปแบบต่างๆ 6 แบบด้วยกันดังนี้

มีด

รูปแบบของอาวุธ

- อาวุธชนิดนี้จะเป็นอาวุธที่ใช้ในการต่อสู้แบบประชิดตัวกับศัตรู ได้แก่ สว่าน ใบมีด เป็นต้น

รูปแบบการโจมตี

- การโจมตีจะเป็นลักษณะของการวิ่งเข้ากระแทก หรือ พุ่งชน ซึ่งมีระยะการโจมตีสั้นมาก กล่าวคือ อาวุธชนิดนี้จะโจมตีศัตรูได้ก็ต่อเมื่อเข้าประชิดตัวศัตรูได้เท่านั้น

ข้อดี

- มีความทนทานสูง
- มีพลังทำลายสูง
- มีความแม่นยำสูงมาก
- มีน้ำหนักเบา
- ใช้ได้ไม่จำกัด

ข้อเสีย

- มีระยะในการโจมตีที่ค่อนข้างใกล้ ซึ่งกว่าจะเข้าถึงตัวศัตรูได้อาจจะถูกโจมตีก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปืน

รูปแบบของอาวุธ

- อาวุธชนิดนี้จัดเป็นอาวุธพิสัยกลาง ซึ่งสามารถใช้โจมตีศัตรูที่อยู่ในระยะยิงซึ่งอยู่ห่างออกไป รูปแบบของปืนแต่ละรุ่นจะมีประสิทธิภาพแตกต่างกันออกไป ซึ่งนั้นก็หมายถึงราคาที่แตกต่างกันอีกด้วย

รูปแบบการโจมตี

- การโจมตีจะเป็นลักษณะของการใช้ยิงศัตรูที่อยู่ห่างออกไป การยิงจะเป็นการยิงทีละนัด เมื่อยิงแล้วจะทำการบรรจุกระสุนเตรียมยิงใหม่ ซึ่งจะทำโดยอัตโนมัติ แต่ต้องใช้เวลาในการเตรียมพร้อมในการยิงครั้งต่อไปนานกว่าอาวธแบบประชิดตัว เมื่อทำการบรรจุกระสุนเรียบร้อยแล้วจึงจะทำการยิงนัดต่อไปได้

ข้อดี

- มีระยะในการโจมตีที่ไกลกว่าอาวุธระยะประชิด ซึ่งสามารถโจมตีได้จากระยะไกลกว่า
- มีพลังทำลายสูง
- มีความแม่นยำสูง

ข้อเสีย

- มีกระสุนจำกัด
- ความต่อเนื่องของการโจมตีช้ากว่า อาวุธชนิดแรก
- มีความทนทานน้อยกว่า อาวุธชนิดแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปืนกล

รูปแบบของอาวุธ

- อาวุธชนิดนี้จัดเป็นอาวุธพิสัยกลาง ซึ่งมีพลังการทำลายไม่มากนัก สามารถใช้โจมตีศัตรูที่อยู่ในระยะยิง ซึ่งอยู่ห่างออกไป ซึ่งการยิงจะเป็นการยิงทีละหลายนัดในการยิง 1 ครั้ง

รูปแบบการโจมตี

- การโจมตีจะเป็นลักษณะของการใช้ยิงศัตรูที่อยู่ห่างออกไปไม่มากนัก การยิงจะเป็นการยิงทีละหลายๆ นัด ซึ่งลักษณะการทำลายจะเป็นการกระจายความเสียหายแก่แต่ละชิ้นส่วนของรถถังศัตรู

ข้อดี

- มีระยะในการโจมตีที่ไกลกว่าอาวุธระยะประชิด ซึ่งสามารถโจมตีได้จากระยะไกลกว่า
- ในการยิงหนึ่งครั้งสามารถยิงกระสุนได้ทีละหลายๆ นัด
- กระจายความเสียหายแก่รถถังศัตรู

ข้อเสีย

- มีกระสุนจำกัด
- มีความทนทานน้อยกว่า อาวุธชนิดแรก
- มีพลังทำลายต่ำ
- มีความแม่นยำต่ำ

โล่

รูปแบบของอาวุธ

- อาวุธชนิดนี้ เป็นอาวุธที่ใช้ในการป้องกันตัว มีความทนทานต่อความเสียหายสูง

รูปแบบการโจมตี

- ใช้ในการป้องกันการโจมตีจากศัตรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี

- มีความทนทานต่อความเสียหายสูงมาก
- น้ำหนักเบา

ข้อเสีย

- ใช้โจมตีไม่ได้

เครื่องยิงลูกระเบิด

รูปแบบของอาวุธ

- อาวุธชนิดนี้จัดเป็นอาวุธพิสัยไกล ซึ่งมีพลังการทำลายล้างสูง และมีการทำลายเป็นวงกว้าง

รูปแบบการโจมตี

- การโจมตีจะเป็นลักษณะของการใช้ยิงศัตรูจากระยะไกล โดยที่ศัตรูไม่ทันตั้งตัว และนอกจากนี้ยังสามารถทำลายศัตรูได้ทีละหลายๆ ยูนิตอีกด้วย

ข้อดี

- มีระยะในการโจมตีที่ไกล ซึ่งสามารถโจมตรีได้จากระยะไกลๆ ได้
- มีพลังทำลายสูง
- มีรัศมีการทำลายล้างกว้าง

ข้อเสีย

- อาวุธ 1 ชิ้นบรรจุกระสุนได้น้อยมาก
- มีน้ำหนักมาก
- ความต่อเนื่องของการโจมตีช้า
- ในการยิงแต่ละครั้งทำได้ช้าเพราะต้องมีการจับเป้าหมายก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องยิงลูกจรวด

รูปแบบของอาวุธ

- อาวุธชนิดนี้จัดเป็นอาวุธพิสัยไกล ซึ่งมีพลังการทำลายล้างสูง และมีระบบการจับเป้าหมายอัตโนมัติ

รูปแบบการโจมตี

- การโจมตีจะเป็นลักษณะของการใช้ยิงศัตรูจากระยะไกล โดยต้องทำการจับเป้าหมายศัตรูก่อนแล้วจากนั้นจรวดจะติดตามเป้าหมายโดยอัตโนมัติ จึงทำให้สามารถยิงได้อย่างแม่นยำ

ข้อดี

- มีระยะในการโจมตีที่ไกล ซึ่งสามารถโจมตรีได้จากระยะไกลๆ ได้
- มีพลังทำลายสูง
- เป็นอาวุธระยะไกลที่สามารถยิงได้อย่างรวดเร็ว
- มีระบบการจับเป้าหมายอัตโนมัติ

ข้อเสีย

- อาวุธ 1 ชิ้นบรรจุกระสุนได้ไม่มากนัก
- มีน้ำหนักมาก

ในการเลือกใช้อาวุธแต่ละชนิดนั้น ควรที่จะเลือกส่วนของตัวรถที่เหมาะสมด้วย เพราะตัวรถแต่ละรุ่นจะมีการสนับสนุนอาวุธแต่ละชนิดต่างกัน ส่วนของตัวรถบางรุ่นอาจสนับสนุนอาวุธระยะประชิดได้ดีกว่า ส่วนของตัวรถบางรุ่นอาจสนับสนุนอาวุธระยะไกลได้ดีกว่า เป็นต้น

การออกแบบ

ในส่วนนี้จะเป็นส่วนของการกำหนดค่าพารามิเตอร์ที่เหมาะสมให้แก่อาวุธแต่ละชิ้นซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีด

ในเกมจะมีอาวุธประเภทมีดทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
มีด-1	25	20	2	95	1	-	1	70
มีด-2	50	40	4	95	1	-	4	280
มีด-3	100	80	8	95	1	-	7	1100
มีด-4	200	160	16	95	1	-	10	4500
มีด-5	400	320	32	95	1	-	13	18000

ตารางที่ 3.3 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทมีด

ปืน

ในเกมจะมีอาวุธประเภทปืนทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
ปืน-1	20	17	2	85	150	-	3	85
ปืน-2	40	34	4	85	150	-	6	350
ปืน-3	80	68	8	85	150	-	9	1500
ปืน-4	160	136	16	85	150	-	12	5500
ปืน-5	320	272	32	85	150	150	15	21000

ตารางที่ 3.4 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทปืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปืนกล

ในเกมจะมีอาวุธประเภทปืนกลทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
ปืนกล-1	18	6x4	2	75	120	600	2	80
ปืนกล-2	36	12x4	4	75	120	600	5	330
ปืนกล-3	72	32x3	8	75	120	450	8	1400
ปืนกล-4	144	48x4	16	75	120	600	11	5000
ปืนกล-5	288	128x3	32	75	120	450	14	19000

ตารางที่ 3.5 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทปืนกล

โล่

ในเกมจะมีอาวุธประเภทโล่ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
โล่-1	56	-	4	85	0	-	5	80
โล่-2	112	-	8	85	0	-	8	350
โล่-3	224	-	16	85	0	-	11	1600
โล่-4	448	-	32	85	0	-	14	5800
โล่-5	896	-	64	85	0	-	17	20000

ตารางที่ 3.6 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทโล่

ระเบิด

ในเกมจะมีอาวุธประเภทระเบิดทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
ระเบิด-1	18	25	1	60	200	5	7	90
ระเบิด-2	36	50	2	60	200	5	10	400
ระเบิด-3	72	100	4	60	200	5	13	1700
ระเบิด-4	144	200	8	60	200	5	16	7500
ระเบิด-5	288	400	16	60	200	4	19	26000

ตารางที่ 3.7 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทระเบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จรวด

ในเกมจะมีอาวุธประเภทจรวดทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	โจมตี	ป้องกัน	ความแม่นยำ	ระยะ	จำนวนกระสุน	น้ำหนัก	ราคา
จรวด-1	18	15x2	1	95	1	6	8	95
จรวด-2	36	30x2	2	95	1	6	11	450
จรวด-3	72	40x3	4	95	1	9	14	1800
จรวด-4	144	80x3	8	95	1	9	17	8000
จรวด-5	288	240x2	16	95	1	6	20	27000

ตารางที่ 3.8 ตารางการกำหนดค่าพารามิเตอร์ของอาวุธประเภทจรวด

ค่าพารามิเตอร์

พลัง คือ ขีดจำกัดของอาวุธ

โจมตี คือ ค่าพลังทำลายของอาวุธ

ป้องกัน คือ ค่าความทนทานต่อการโจมตีของอาวุธ

ความแม่นยำ คือ ค่าความน่าจะเป็นในการโจมตีถูกเป้าหมาย

ระยะ คือ ระยะโจมตีของอาวุธ

จำนวนกระสุน คือ จำนวนกระสุนที่บรรจุได้ในอาวุธนั้น

น้ำหนัก คือ ค่าที่แสดงน้ำหนักของอาวุธ

ราคา คือ มูลค่าของอาวุธ

3.4.1.3 ส่วนล้อรถถัง

แนวคิดของเกม

ส่วนนี้นับว่าเป็นส่วนสำคัญอีกส่วนหนึ่ง ซึ่งส่วนนี้จะเป็นส่วนที่ใช้ในการรองรับน้ำหนักรถถังทั้งคัน ส่วนล้อที่ดีจะสามารถรองรับน้ำหนักได้มาก ซึ่งทำให้ผู้เล่นสามารถปรับแต่งรถถังได้หลากหลายขึ้น เพราะในการปรับแต่งรถถังจะไม่สามารถบรรทุกน้ำหนักเกินได้ ทำให้บางครั้งผู้เล่นไม่สามารถติดอาวุธที่มีประสิทธิภาพดี แต่มีน้ำหนักมากได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากในเรื่องของการบรรทุกน้ำหนักแล้ว ส่วนของการเคลื่อนที่บนพื้นสถานะต่างๆ ก็สำคัญเช่นกันล้อแต่ละรุ่นจะวิ่งได้ดีในสถานะพื้นผิวที่แตกต่างกัน การปรับแต่งล้อให้เหมาะสมกับสภาพภูมิประเทศย่อมจะเป็นการได้เปรียบคู่ต่อสู้อีกทางหนึ่ง

รูปแบบล้อรถถังในเกมจะแบ่งได้เป็น 5 แบบด้วยกันได้แก่

ล้อยาง เป็นล้อที่ใช้วิ่งได้ดีในพื้นที่ราบ เช่น บนถนน เมือง ทุ่งหญ้า เป็นต้น เป็นล้อที่สามารถใช้ความเร็วได้สูง แต่มีข้อเสียที่ใช้ได้ไม่ดี ในพื้นที่ที่มีพื้นผิวขรุขระ

ล้อตีนตะขาบ เป็นล้อที่ใช้ได้ดีในพื้นที่ขรุขระ พื้นทราย เป็นต้น เป็นล้อที่ใช้ความเร็วได้ในระดับปานกลาง สามารถบรรทุกน้ำหนักได้สูง

ล้อแบบกึ่งผสม เป็นล้อแบบที่ผสมระหว่าง ล้อยาง กับ ล้อตีนตะขาบ เข้าด้วยกัน จึงทำให้ใช้ได้ดีแม้ในพื้นราบ หรือพื้นขรุขระ

ขากล เป็นล้อที่มีรูปแบบเป็นขากล ใช้ความเร็วได้ไม่มากนัก แต่ เคลื่อนที่ในพื้นที่ที่มีความขรุขระมากๆ ได้ดี

โฮเวอร์คราฟท์ เป็นลักษณะของล้อที่เป็นแบบโฮเวอร์คราฟท์ รองรับน้ำหนักได้น้อย แต่วิ่งได้ดีในทุกสภาพพื้นผิว ที่มีความชันต่ำ

การออกแบบ

ล้อยาง

ในเกมจะมีล้อประเภทนี้ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	ป้องกัน	การรับน้ำหนัก	A	B	C	D	E	F	การปีน	ราคา
ล้อยาง-1	24	1	43	10	3	-	3	1	-	1	80
ล้อยาง-2	48	2	52	10	3	-	3	1	-	1	300
ล้อยาง-3	96	4	61	10	3	-	3	1	-	1	1300
ล้อยาง-4	192	8	70	10	3	-	3	1	-	1	5200
ล้อยาง-5	384	16	79	10	3	-	3	1	-	1	20000

ตารางที่ 3.9 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทล้อยาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล้อตีนตะขาบ

ในเกมจะมีล้อประเภทนี้ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	ป้องกัน	การรับน้ำหนัก	A	B	C	D	E	F	การปีน	ราคา
ล้อตีนตะขาบ-1	36	2	52	7	6	5	5	6	-	2	100
ล้อตีนตะขาบ-2	72	4	67	7	6	5	5	6	-	2	420
ล้อตีนตะขาบ-3	144	8	82	7	6	5	5	6	-	2	1650
ล้อตีนตะขาบ-4	288	16	97	7	6	5	5	6	-	2	6200
ล้อตีนตะขาบ-5	576	32	112	7	6	5	5	6	-	2	24000

ตารางที่ 3.10 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทล้อตีนตะขาบ

ล้อกึ่งผสม

ในเกมจะมีล้อประเภทนี้ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	ป้องกัน	การรับน้ำหนัก	A	B	C	D	E	F	การปีน	ราคา
ล้อกึ่งผสม-1	28	2	49	8	5	3	3	5	-	2	95
ล้อกึ่งผสม-2	56	4	61	8	5	3	3	5	-	2	370
ล้อกึ่งผสม-3	112	8	73	8	5	3	3	5	-	2	1200
ล้อกึ่งผสม-4	224	16	85	8	5	3	3	5	-	2	4800
ล้อกึ่งผสม-5	448	32	97	8	5	3	3	5	-	2	22000

ตารางที่ 3.11 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทล้อกึ่งผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขากล

ในเกมจะมีล้อประเภทนี้ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	ป้องกัน	การรับน้ำหนัก	A	B	C	D	E	F	การปีน	ราคา
ขากล-1	40	2	60	6	6	6	5	5	-	4	120
ขากล-2	80	4	75	6	6	6	5	5	-	4	500
ขากล-3	160	8	90	6	6	6	5	5	-	4	2100
ขากล-4	320	16	105	6	6	6	5	5	-	4	6500
ขากล-5	640	32	120	6	6	6	5	5	-	4	26000

ตารางที่ 3.12 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทขากล

โฮเวอร์คราฟท์

ในเกมจะมีล้อประเภทนี้ทั้งหมด 5 แบบด้วยกัน ดังนี้

ชื่อ	พลัง	ป้องกัน	การรับน้ำหนัก	A	B	C	D	E	F	การปีน	ราคา
โฮเวอร์คราฟท์-1	20	1	38	11	7	2	10	10	10	0	95
โฮเวอร์คราฟท์-2	40	2	47	11	7	2	10	10	10	0	380
โฮเวอร์คราฟท์-3	80	4	56	11	7	2	10	10	10	0	1500
โฮเวอร์คราฟท์-4	160	8	65	11	7	2	10	10	10	0	6200
โฮเวอร์คราฟท์-5	320	16	74	11	7	2	10	10	10	0	22000

ตารางที่ 3.13 ตารางการกำหนดค่าพารามิเตอร์ของล้อรถประเภทโฮเวอร์คราฟท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพารามิเตอร์

พลัง คือ ขีดจำกัดของล้อนั้นๆ

ป้องกัน คือ ค่าความทนทานต่อการโจมตีของล้อรถ

การรับน้ำหนัก คือ ค่าที่แสดงถึงความสามารถในการรับน้ำหนักของล้อรถ

A คือ ความเร็วในการเคลื่อนที่บนพื้นราบ

B คือ ความเร็วในการเคลื่อนที่บนพื้นทราย

C คือ ความเร็วในการเคลื่อนที่บนพื้นวิบาก

D คือ ความเร็วในการเคลื่อนที่บนพื้นน้ำแข็ง

E คือ ความเร็วในการเคลื่อนที่บนน้ำตื้น

F คือ ความเร็วในการเคลื่อนที่บนน้ำลึก

การปีน คือ ความสามารถในการปีนขึ้นที่สูงของล้อชนิดนั้นๆ

ราคา คือ มูลค่าของล้อรถ

3.4.1.5 เครื่องหลัง

แนวคิดของเกม

เครื่องหลัง เป็นอุปกรณ์ที่ใช้ติดด้านหลังของรถถัง อุปกรณ์ชนิดนี้จะใช้ในการบรรทุกสิ่งของรวมถึงการเพิ่มความสามารถในส่วนของความเร็วและการบรรทุกน้ำหนักของรถถังเพิ่มขึ้น

การออกแบบ

ชื่อ	จำนวนบรรทุก	ความเร็ว +	การรับน้ำหนัก +	น้ำหนัก	ราคา
เครื่องหลัง -1	2	0	0	2	120
เครื่องหลัง -2	0	1	2	2	130

ตารางที่ 3.14 ตารางการกำหนดค่าพารามิเตอร์ของเครื่องหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพารามิเตอร์

จำนวนบรรทุก คือ ค่าความสามารถในการบรรทุกไอเท็มของรถถัง

ความเร็ว+ คือ ค่าความเร็วที่เพิ่มขึ้นที่ได้จากการติดเครื่องหลังชนิดนั้นๆ

การรับน้ำหนัก + คือ ค่าความสามารถในการรับน้ำหนักที่เพิ่มขึ้นที่ได้จากการติดเครื่องหลังชนิดนั้นๆ

น้ำหนัก คือ น้ำหนักของเครื่องหลัง

ราคา คือ มูลค่าของเครื่องหลัง

3.4.1.5 คอมพิวเตอร์

แนวคิดของเกม

คอมพิวเตอร์ เป็นอุปกรณ์ที่ใช้เชื่อมความสามารถพิเศษของคนขับรถถังเข้ากับรถถัง ทำให้สามารถใช้ความสามารถพิเศษที่ติดตั้งได้

การออกแบบ

ชื่อ	SUP Slot	STA Slot	FGT Slot	COM Slot	ราคา
Celetron 1.4 Thz	0	1	1	1	120
Tentium 2 Thz	2	8	8	8	100000

ตารางที่ 3.15 ตารางการกำหนดค่าพารามิเตอร์ของเครื่องคอมพิวเตอร์

ค่าพารามิเตอร์

SUP Slot คือ จำนวนช่องที่สามารถติดsupport ability ได้

STA Slot คือ จำนวนช่องที่สามารถติด status ability ได้

FGT Slot คือ จำนวนช่องที่สามารถติด fighting ability ได้

COM Slot คือ จำนวนช่องที่สามารถติด command ability ได้

ราคา คือ มูลค่าของเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของรถถัง

รถถังในเกม Battle City Extreme นั้นผู้เล่นสามารถปรับแต่งได้อย่างหลากหลาย แต่หากให้แบ่งเป็นชนิดแล้วจะสามารถแบ่งได้เป็น 3 ชนิดหลักๆ ด้วยกัน

- **รถถังติดอาวุธพิสัยใกล้** รถถังชนิดนี้เป็นรถถังที่ปรับแต่งเพื่อการโจมตีแบบประชิดตัว จึงมักจะมีความเร็ว และ ความ แข็งแกร่งสูง เพื่อที่จะเคลื่อนที่เข้าใกล้ตัวศัตรู และ โจมตี ศัตรูอย่างรวดเร็ว

- **รถถังติดอาวุธพิสัยกลาง** รถถังชนิดนี้เป็นรถถังที่ปรับแต่งเพื่อการโจมตีโดยใช้ปืนใหญ่ และ ปืนกล จึงมักจะมีความเร็วในระดับปานกลาง รูปแบบการโจมตี จะรักษาระยะไม่เข้าใกล้ศัตรูมากจนเกินไป และอาศัยจังหวะในการโจมตี

- **รถถังติดอาวุธพิสัยไกล** รถถังชนิดนี้เป็นรถถังที่ปรับแต่งเพื่อการโจมตีระยะไกล เนื่องจากอาวุธประเภทนี้มักมีน้ำหนักมาก รถถังประเภทนี้จึงมักจะมีความเร็วต่ำ การโจมตีจะอาศัยการโจมตีระยะไกลก่อนที่ศัตรูจะเข้าถึงตัว

3.4.2 การปรับแต่งรถถัง

การปรับแต่งรถถังนั้นสามารถทำได้หลากหลายรูปแบบ การที่จะปรับแต่งรถถังแบบใดนั้นขึ้นอยู่กับความถนัดของผู้เล่น ซึ่งการปรับแต่งรถถังจะมีหลักการดังนี้

- รถถังแต่ละคันจะประกอบอาวุธได้สูงสุด 4 ชิ้น แต่ผู้เล่นจะติดตั้งอาวุธกี่ชิ้นนั้นขึ้นอยู่กับความพอใจของผู้เล่น แต่มีข้อแม้ว่า น้ำหนักรวมของส่วนตัวรถ และ อาวุธทั้งหมดที่นำมาติดตั้งต้องไม่เกิน น้ำหนักที่ล้อรถถังที่ผู้เล่นเลือกใช้รองรับได้
- ส่วนของตัวรถ การปรับแต่งตัวรถนั้น รถถังแต่ละคันจะมีรูปแบบส่วนของตัวรถตามรูปแบบอาวุธที่ใช้ เช่น รถถังที่ติดตั้งอาวุธแบบประชิดตัวที่มีน้ำหนักเบา ก็สามารถติดตั้งส่วนของตัวรถที่แข็งแกร่งที่มีน้ำหนักมากได้โดยน้ำหนักรวมไม่เกินภาระของส่วนล้อที่จะรองรับได้ ส่วนรถถังที่มีการติดตั้งอาวุธระยะไกลซึ่งมีน้ำหนักมาก ก็จะติดตั้งได้เพียงส่วนของตัวรถที่มีน้ำหนักเบา ซึ่งมีความแข็งแกร่งของเกราะน้อยกว่า
- ส่วนของล้อรถถัง โดยทั่วไปแล้วการติดตั้งต้องคำนึงถึงสภาพภูมิประเทศที่จะนำไปใช้ และ รองรับน้ำหนักของล้อให้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของเครื่องหลัง จะมีหลักการในการติดตั้ง 2 แบบ คือเพื่อใช้ในการบรรทุกไอเท็ม หรือ การเพิ่มสถานะในการเคลื่อนที่และการรองรับน้ำหนัก
- ส่วนของคอมพิวเตอร์ หลักการคือจะเน้นที่จำนวนสล็อตว่ามีมากน้อยแค่ไหน และเป็นสล็อต ชนิดใด นอกจากนี้โครงสร้างของสถาปัตยกรรมของเครื่องคอมพิวเตอร์ก็เป็นส่วนสำคัญ
- ส่วนขอการบรรทุกไอเท็ม ในการบรรทุกไอเท็มนั้นผู้เล่นจะสามารถบรรทุกไอเท็มได้ตามจำนวนที่เครื่องหลังสามารถบรรจุได้
- ส่วนของสีรถ ผู้เล่นสามารถปรับเปลี่ยนสีของรถถังได้ทั้งหมด 8 สี

3.4.3 ไอเท็ม

สิ่งของสนับสนุนในการต่อสู้นั้นผู้เล่นจะสามารถนำบรรทุกไปได้สูงสุดไม่เกิน 8 ชิ้น แล้วแต่ความสามารถในการบรรทุกของรถถังแต่ละคัน ซึ่งสิ่งของเหล่านี้ผู้เล่นสามารถนำไปใช้ระหว่างต่อสู้ได้ รูปแบบสิ่งของจะแบ่งออกเป็นประเภทต่างๆ ได้ 5 ประเภทด้วยกัน

- **สิ่งของประเภทใช้ในการซ่อมแซม** สิ่งของประเภทนี้จะใช้ในการซ่อมแซมความเสียหายที่เกิดจากการต่อสู้ และการแก้สภาพผิดปกติของรถถัง

ชื่อ	การนำไปใช้	ราคา
Repair 1	ซ่อมแซมส่วนที่เสียหาย พลัง+100	10
Repair 2	ซ่อมแซมส่วนที่เสียหาย พลัง+500	80
Repair 3	ซ่อมแซมส่วนที่เสียหาย พลัง+2500	640
Repair All	ซ่อมแซมส่วนที่เสียหายทั้งหมดจนสมบูรณ์	2100
Rebirth 1	ซ่อมแซมส่วนที่ถูกทำลาย พลัง+100	20
Rebirth 2	ซ่อมแซมส่วนที่ถูกทำลาย พลัง+500	140
Rebirth 3	ซ่อมแซมส่วนที่ถูกทำลาย พลัง+2500	980
Rebirth 4	ซ่อมแซมส่วนที่ถูกทำลายทั้งหมดจนสมบูรณ์	3200

ตารางที่ 3.16 ตารางแสดงรายละเอียดสิ่งของประเภทซ่อมแซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **สิ่งของประเภทกับดัก** สิ่งของประเภทนี้จะใช้ในการวางกับดักศัตรู โดยการวางไว้ตามจุดต่างๆ ในฉาก จะทำงานเมื่อศัตรูเข้ามาอยู่ในขอบเขต

ชื่อ	การนำไปใช้	ราคา
Wheel Trap	ทำลายล้อของรถถังที่เหยียบกับดัก	50

ตารางที่ 3.17 ตารางแสดงรายละเอียดสิ่งของประเภทกับดัก

- **สิ่งของประเภทสนับสนุน** สิ่งของประเภทนี้จะใช้ในการสนับสนุนการต่อสู้ ไม่ว่าจะเป็นการโจมตี การหนี การป้องกันตัว

ชื่อ	การนำไปใช้	ราคา
Meat	ใช้เป็นเหยื่อล่อ มอนสเตอร์	5

ตารางที่ 3.18 ตารางแสดงรายละเอียดสิ่งของประเภทสนับสนุน

- **สิ่งของประเภทกระสุน** สิ่งของประเภทนี้จะใช้ในการเติมกระสุนแก่อาวุธชนิดต่างๆ

ชื่อ	การนำไปใช้	ราคา
Gun Ballet	ใช้เติมกระสุนให้อาวุธประเภทปืน	15
Machine Ballet	ใช้เติมกระสุนให้อาวุธประเภทปืนกล	15
Missile Ballet	ใช้เติมกระสุนให้อาวุธประเภทจรวด	20
Bomb Ballet	ใช้เติมกระสุนให้อาวุธประเภทระเบิด	20

ตารางที่ 3.19 ตารางแสดงรายละเอียดสิ่งของประเภทกระสุน

- **สิ่งของประเภทอื่นๆ** สิ่งของประเภทนี้จะไม่สามารถนำมาใช้ในการต่อสู้ได้ แต่สามารถที่จะนำไปแลกเป็นเงินเพื่อนำมาใช้ในการปรับแต่งรถถังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ	การนำไปใช้	ราคา
Branch	ไม่มี	5
Resin	ไม่มี	5
Trunk	ไม่มี	10
Root	ไม่มี	20
Coconut Oil	ไม่มี	10
Jelly	ไม่มี	15
Iron	ไม่มี	20
Rubber	ไม่มี	15

ตารางที่ 3.20 ตารางแสดงรายละเอียดสิ่งของประเภทอื่นๆ

3.4.4 คนขับ

ในรถถังแต่ละคันจะมีตัวละครที่เป็นคนขับ ซึ่งตัวละครนี้จะมีความสามารถแตกต่างกันขึ้นอยู่กับระดับของตัวละคร ซึ่งระดับของตัวละครนี้จะส่งผลถึงความสามารถในการควบคุมรถถัง การโจมตี ความแม่นยำที่แตกต่างกัน ตัวละครจะสามารถยกระดับความสามารถของตนเองได้ด้วยการเก็บค่าประสบการณ์ ตัวละครจะมีความสามารถใหม่ๆ เมื่อมีค่าประสบการณ์เพิ่มขึ้น

3.4.4.1 ค่าสถานะของตัวละคร

ตัวละครแต่ละตัวจะมีค่าสถานะไม่เหมือนกัน ซึ่งการเลือกรถถังให้เหมาะสมกับคนขับนั้นย่อมทำให้เกิดประสิทธิภาพในการรบที่ดี ค่าสถานะของตัวละครจะมีผลในการควบคุมรถถังดังนี้

- **พลังชีวิต** ค่านี้แสดงถึงค่าความอดทนของตัวละคร ในการถูกโจมตีแต่ละครั้ง ตัวละครมีโอกาสที่จะได้รับผลกระทบต่อการโจมตีนั้นๆ หากการโจมตีทำให้พลังชีวิตของตัวละครลดเป็น 0 ตัวละครนั้นจะถูกทำลาย
- **พลังทักษะ** ค่านี้แสดงถึงค่าความสามารถในการใช้ทักษะของตัวละคร โดยในการใช้ทักษะบางชนิดจะทำให้พลังทักษะลดลง
- **ความแข็งแกร่ง** ค่านี้แสดงถึงความแข็งแกร่งของตัวละคร ที่จะสามารถทนทานต่อการโจมตี และความสามารถในการฟื้นพลังของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การควบคุม** ค่านี้แสดงถึงความสามารถในการควบคุมรถถัง เมื่อค่านี้มีค่ามากคนขับจะสามารถลดการไถลของรถถังที่เกิดจากการโจมตีของอาวุธศัตรู ซึ่งการไถลนี้ถ้ามีมากอาจทำให้เกิดการเสียจังหวะในการโจมตีของผู้เล่น
- **การโจมตี** ค่านี้แสดงถึงความสามารถในการโจมตีของคนขับ ถ้ามีค่ามากการโจมตีจะมีประสิทธิภาพมากขึ้น
- **การตั้งรับ** ค่านี้แสดงถึงความสามารถในการตั้งรับของคนขับ ถ้ามีค่ามากการตั้งรับจะมีประสิทธิภาพมากขึ้น
- **การหลบหลีก** ค่านี้แสดงถึงความสามารถในการหลบหลีกของคนขับถ้าค่านี้มีค่ามากจะทำให้โอกาศการหลบหลีกการโจมตีของศัตรูได้สูง
- **การใช้อาวุธพิสัยใกล้** ค่านี้แสดงถึงความชำนาญในการใช้อาวุธระยะประชิดตัวของคนขับ ค่านี้ถ้ามีค่ามากคนขับจะสามารถดึงประสิทธิภาพของอาวุธชิ้นนั้นออกมาได้มาก
- **การใช้อาวุธพิสัยกลาง** ค่านี้แสดงถึงความชำนาญในการใช้อาวุธพิสัยกลางของคนขับ ได้แก่ปืนใหญ่ และ ปืนกล ค่านี้ถ้ามีค่ามากคนขับจะสามารถดึงประสิทธิภาพของอาวุธชิ้นนั้นออกมาได้มาก
- **การใช้อาวุธพิสัยไกล** ค่านี้แสดงถึงความชำนาญในการใช้อาวุธพิสัยไกลของคนขับ ได้แก่เครื่องยิงลูกระเบิด และ เครื่องยิงลูกจรวด ค่านี้ถ้ามีค่ามากคนขับจะสามารถดึงประสิทธิภาพของอาวุธชิ้นนั้นออกมาได้มาก
- **พลังเวทย์** ค่านี้แสดงถึงความสามารถในการใช้เวทย์มนต์ ถ้าค่านี้มีค่ามากจะทำให้ตัวละครสามารถใช้ทักษะประเภทเวทย์มนต์ได้มีประสิทธิภาพมากยิ่งขึ้น
- **ทักษะช่างเครื่อง** ค่านี้แสดงถึงความสามารถในการซ่อมแซมรถถังของคนขับ ซึ่งจะใช้ในการซ่อมแซมส่วนที่เสียหาย ถ้ามีค่ามากจะซ่อมได้เร็วขึ้น
- **โชค** ค่านี้แสดงถึงโชคของคนขับ ในกรณีที่มีการต้องเสี่ยงโชคเข้ามาเกี่ยวข้อง ถ้ามีค่านี้มากๆ จะทำให้มีโอกาศสำเร็จสูงขึ้น

3.4.4.2 ความสามารถพิเศษ

เมื่อตัวคนขับมีระดับมากขึ้น คนขับจะมีความสามารถพิเศษเพิ่มขึ้น ซึ่งความสามารถพิเศษนี้จะใช้ได้ก็ต่อเมื่อผู้เล่นจะต้องนำความสามารถพิเศษเหล่านี้ติดตั้งกับระบบคอมพิวเตอร์ของรถถัง

ความสามารถพิเศษในเกมนี้จะแบ่งเป็น 4 แบบด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสามารถพิเศษในการควบคุม (Command Ability)

ความสามารถพิเศษนี้เมื่อนำมาติดตั้งแล้วผู้เล่นนอกจากจะใช้คำสั่งทั่วไปในการเคลื่อนที่ และ โจมตีแล้ว ผู้เล่นจะสามารถใช้คำสั่งแบบใหม่ที่นำมาติดตั้ง ใช้ในการควบคุมรถถังได้

ชื่อ	การใช้งาน
Repair Type	ทำให้สามารถใช้ไอเท็มประเภทซ่อมแซมได้

ตารางที่ 3.21 ตารางแสดงรายละเอียด Command Ability

- ความสามารถพิเศษเสริมการต่อสู้ (Fighting Ability)

ความสามารถพิเศษนี้เมื่อนำมาติดตั้งแล้วในขณะต่อสู้ความสามารถชนิดนี้จะสนับสนุนการต่อสู้ ซึ่งความสามารถชนิดนี้ใช้อัตโนมัติในขณะต่อสู้

ชื่อ	การใช้งาน
Fire	ทำให้สามารถใช้เวทย์มนต์ไฟช่วยในการโจมตีได้
Ice	ทำให้สามารถใช้เวทย์มนต์น้ำแข็งช่วยในการโจมตีได้
Thunder	ทำให้สามารถใช้เวทย์มนต์สายฟ้าช่วยในการโจมตีได้
Water	ทำให้สามารถใช้เวทย์มนต์น้ำช่วยในการโจมตีได้
Wind	ทำให้สามารถใช้เวทย์มนต์ลมช่วยในการโจมตีได้
Earth	ทำให้สามารถใช้เวทย์มนต์ดินช่วยในการโจมตีได้
Holy	ทำให้สามารถใช้เวทย์มนต์แสงช่วยในการโจมตีได้
Dark	ทำให้สามารถใช้เวทย์มนต์ความมืดช่วยในการโจมตีได้

ตารางที่ 3.22 ตารางแสดงรายละเอียด Fighting Ability

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสามารถพิเศษเสริมสถานะ (Status Ability)

ความสามารถพิเศษนี้เมื่อนำมาติดตั้งแล้วจะเพิ่มค่าสถานะต่างๆ ให้กับรถถัง ทำให้คนขับมีค่าสถานะที่เพิ่มขึ้น แต่เมื่อนำความสามารถนี้ออกค่าสถานะของคนขับก็จะกลับเป็นค่าเดิม

ชื่อ	การใช้งาน
HP Plus	เพิ่มพลังชีวิตของตัวละคร 10-50 %
AP Plus	เพิ่มพลังทักษะของตัวละคร 10-50 %
Str Plus	เพิ่มพลังความแข็งแกร่งของตัวละคร 10-50 %
ConPlus	เพิ่มการควบคุมของตัวละคร 10-50 %
Luck Plus	เพิ่มโชคของตัวละคร 10-50 %
Int Plus	เพิ่มพลังเวทย์ของตัวละคร 10-50 %
Eng Plus	เพิ่มทักษะช่างเครื่องของตัวละคร 10-50 %
Sw Plus	เพิ่มความชำนาญในการใช้อาวุธประเภทมีดของตัวละคร 10-50 %
Mw Plus	เพิ่มความชำนาญในการใช้อาวุธประเภทปืนและปืนกลของตัวละคร 10-50 %
Lw Plus	เพิ่มความชำนาญในการใช้อาวุธประเภทจรวดและระเบิดของตัวละคร 10-50 %
Map	ทำให้สามารถมองเห็นแผนที่ของฉาก

ตารางที่ 3.23 ตารางแสดงรายละเอียด Status Ability

- ความสามารถพิเศษสนับสนุน (Support Ability)

ความสามารถพิเศษนี้เมื่อนำมาติดตั้งแล้วจะสนับสนุนความสามารถพิเศษอื่นให้มีความสามารถเพิ่มมากขึ้น แต่ความสามารถชนิดนี้จะให้ผลได้ก็ต่อเมื่อมีเส้นเชื่อมกับความสามารถพิเศษที่สนับสนุนเท่านั้น

ชื่อ	การใช้งาน
Expand	ทำให้สามารถใช้ไอเท็มประเภทซ่อมแซมได้

ตารางที่ 3.24 ตารางแสดงรายละเอียด Support Ability

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 รูปแบบฉาก

ในส่วนฉากของเกมนั้นแต่ละฉากจะมีรูปแบบที่ต่างกันออกไป ซึ่งส่วนหลักของฉากจะประกอบด้วย 2 ส่วนด้วยกันคือส่วนของพื้นดิน และส่วนของวัตถุ

- ส่วนประกอบของฉาก

- **ส่วนของพื้นดิน** จะแบ่งลักษณะเป็น 6 แบบด้วยกัน

1 . พื้นราบ พื้นลักษณะนี้จะเป็นในลักษณะของพื้นเรียบทั่วไป ได้แก่ บนถนน พื้นที่เมือง ทุ่งหญ้า พื้นลักษณะนี้รถถังแต่ละคันมากจะใช้ความเร็วได้เต็มที่ ไม่ว่าจะใช้ล้อรถถังแบบใด

2 . พื้นที่ขรุขระ พื้นลักษณะนี้จะเป็นในลักษณะของพื้นที่ในป่า ทะเลทราย โคลน ซึ่งพื้นลักษณะนี้จะใช้ความเร็วได้ไม่มากนัก แต่หากเป็นล้อรถถังใช้ได้ดีในพื้นขรุขระแล้วจะใช้ความเร็วได้เต็มที่

3 . พื้นที่ขรุขระมาก พื้นลักษณะนี้จะเป็นในลักษณะของพื้นที่ในบริเวรภูเขาที่มีเศษหินมากมาย ซึ่งพื้นลักษณะนี้รถถังส่วนใหญ่จะวิ่งไม่ค่อยได้ จะวิ่งได้ก็จะเป็นพวกที่ปรับแต่งแบบใช้ขากลซึ้งปีนได้ดี

4 . พื้นที่น้ำตื้น พื้นลักษณะนี้จะเป็นในลักษณะของพื้นที่ริมชายฝั่งแม่น้ำ หรือ ทะเล ซึ่งพื้นลักษณะนี้รถถังหนักจะวิ่งไม่ค่อยได้ดีนัก

5. พื้นที่น้ำลึก พื้นลักษณะนี้จะเป็นในลักษณะของพื้นที่ในแม่น้ำ หรือ ทะเล ซึ่งพื้นลักษณะนี้รถถังจะวิ่งไม่ได้ เว้นแต่รถถังประเภทลอยตัวได้ หรือ ปรับแต่งล้อเป็นแบบโฮเวอร์คราฟท์

6. พื้นที่น้ำแข็ง พื้นลักษณะนี้จะเป็นในลักษณะของพื้นที่ที่เป็นน้ำแข็ง ซึ่งพื้นลักษณะนี้จะเคลื่อนที่ได้ค่อนข้องยาก

- **ส่วนของวัตถุ**

1 . ต้นไม้ ต้นไม้ในเกมจะเป็นของรูปแบบตามสภาพภูมิประเทศ ซึ่งจะประกอบด้วย ต้นไม้ขนาดใหญ่ที่มักพบในป่าดิบชื้น ต้นไม้ทรงสูง ที่มักพบตามเนินเขา ต้นมะพร้าวที่มักพบตามเกาะ หรือชายทะเล และต้นไม้ปลูกที่มักพบในเมือง

2. สิ่งก่อสร้าง ประกอบด้วย ตึก บ้าน รั้ว ซึ่งจะประกอบด้วยหลายๆ รูปแบบ ตามลักษณะของพื้นที่

3 . หินขนาดใหญ่ ซึ่งมักพบในพื้นที่ที่เป็นภูเขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉากในเกมจะมีทั้งหมด 10 ฉากประกอบ ด้วย ฉากเมือง ฉากภูเขา ฉากทะเลทราย ฉากป่าดงดิบ และ ฉากที่มีการผสมรูปแบบต่างๆ เข้าด้วยกัน

3.5 การสร้างภาพ และ Effect ของภาพ

3.5.1 การสร้างพื้นผิว

การสร้างพื้นผิวเป็นส่วนของการสร้างพื้นดินของเกม ซึ่งจะประกอบด้วยพื้นผิวลักษณะต่างๆ รวมถึงการกำหนดสถานะของพื้นผิวและความสูงของพื้นผิว

ในการสร้างพื้นผิวนั้น ขั้นแรกจะมีการสร้างบัฟเฟอร์ขนาดใหญ่เพื่อใช้เก็บพื้นผิวทั้งหมดภายในฉาก จากนั้นจะนำส่วนของพื้นผิวมาแปะเข้าไปในบัฟเฟอร์ ในส่วนของพื้นผิวนี้จะแบ่งออกเป็น 3 ส่วนด้วยกัน

- **ส่วนของพื้น** ส่วนนี้จะเป็นการนำ Texture ที่ได้จากการออกแบบมาแปะในบัฟเฟอร์ที่สร้างไว้ ในส่วนนี้ Texture ที่นำมาประกอบในฉากจะมีการกำหนดสถานะของพื้นไว้ว่าเป็นพื้นชนิดใด และนอกจากนี้ในการแปะรูปก็จะมีการกำหนดความสูงให้แก่พื้นผิว

รูปที่ 3.6 รูปตัวอย่าง Texture ที่นำมาใช้ในการสร้างพื้นผิว

ในการสร้างพื้นผิวขนาดใหญ่มากๆ นั้นจะสามารถสร้างได้โดยการนำเอา Texture จำนวนมากมารวมกัน และเพื่อจะให้ได้ขนาดที่ต้องการก็จะใช้ในส่วนของการ Clipping เข้ามาช่วย

- **ส่วนของวัตถุบนพื้นดิน** ในส่วนนี้ก็เป็นอีกส่วนประกอบหนึ่งของพื้นดิน จะเป็นลักษณะของการนำรูปภาพที่ได้ออกแบบไว้มาวางบนพื้นดิน สิ่งของที่นำมาวางนี้จะไม่มีการกำหนดสถานะหรือความสูงแต่อย่างใด แต่สิ่งของเหล่านี้จะใช้เพื่อเพิ่มความสวยงามให้แก่ฉาก

- **ส่วนของพื้น 3 มิติ** ในส่วนนี้จะมีลักษณะเช่นเดียวกันกับส่วนของพื้นแบบแรก แต่ Texture ที่ได้ จะนำมาประมวลผลแปลงภาพให้เป็น 3 มิติ ตามทฤษฎีในบทที่ 2 ซึ่งจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ภาพดูนูนขึ้นหรือลึกลงไป ซึ่งจะทำให้พื้นดินของเกมมีความสมจริงยิ่งขึ้น แต่สำหรับในส่วนของเกม ในการสร้างพื้น 3 มิตินี้จะใช้การคำนวณโดยโปรแกรม จึงทำให้เสียเวลาในการสร้าง เพื่อให้ใช้เวลาในการสร้างให้น้อยที่สุดจึงต้องมีการกำหนดโครงสร้างและลำดับการสร้างให้เหมาะสม

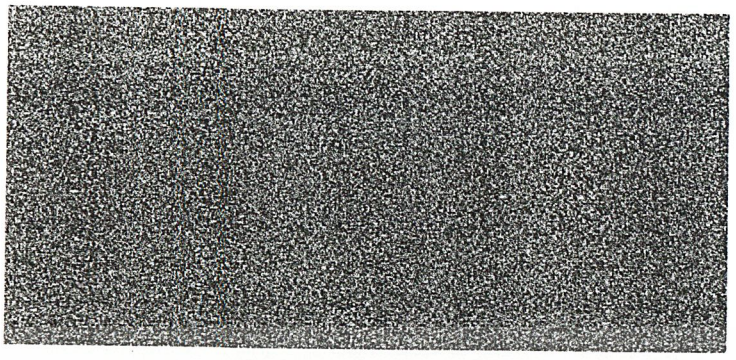
ในส่วนนี้จะเป็นการสร้างภาพ 2 มิติให้มีมุมมองเหมือนภาพ 3 มิติ โดยมีขั้นตอนดังนี้

ขั้นที่ 1 ทำการสร้างโมเดลของแสงที่จะมีความมืดความสว่างทำให้พื้นผิวดูเป็นมิติ โดยรูปที่สร้างจะมีลักษณะเป็นรูป Grayscale โมเดลแสงที่ได้จากการเตรียม จะใช้โทนแสงที่ไม่สูงแตกต่างกันมากนักเพราะ โทนแสงที่สูงแตกต่างกันโกนไปจะทำให้รูปวัตถุที่ได้มีความแตกต่างของโทนสีมากเกินไปจนผิดปกติ

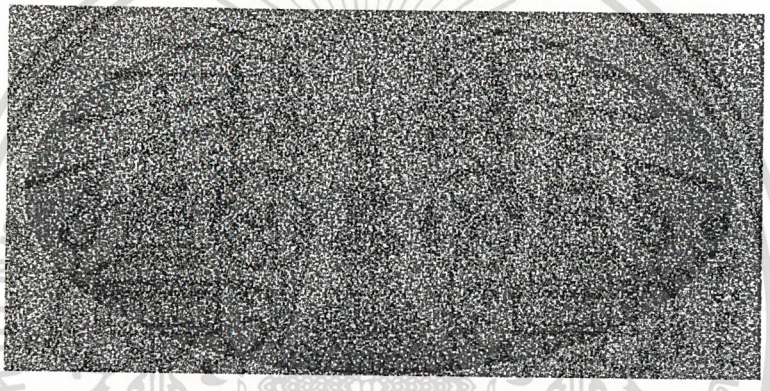
รูปที่ 3.7 รูปตัวอย่างModelของแสงที่นำมาใช้ในการสร้างพื้นผิว3 มิติ

ขั้นตอนที่ 2 นำโมเดลของแสงมาเฉลี่ยค่ากับพื้นผิวที่ต้องการให้เกิดมุมมอง 3 มิติ ก็จะทำให้ได้ภาพที่ดูนูนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 รูปพื้นผิวก่อนที่จะนำมาทำการเฉลี่ยค่าแสงกับโมเดลแสง

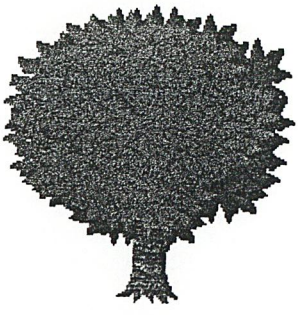


รูปที่ 3.9 รูปพื้นผิวหลังจากนำมาทำการเฉลี่ยค่าแสงกับโมเดลแสงแล้ว

3.5.2 การสร้างวัตถุ

การสร้างวัตถุประกอบด้วยขั้นตอนดังนี้

ขั้นที่ 1 การออกแบบรูปทรง ในส่วนนี้จะเป็นการออกแบบรูปแบบของวัตถุแล้วใช้โปรแกรมประเภทวาดรูปในการสร้าง

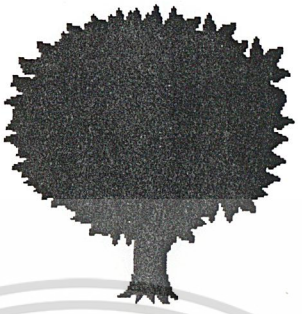


รูปที่ 3.10 รูปแสดงตัวอย่างการออกแบบวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 การสร้าง เงา

- ทำการแปลงสีของวัตถุให้เป็นโทนสีเดียว จะได้ภาพดังรูปข้างล่าง



รูปที่ 3.11 รูปแสดงการแปลงวัตถุให้เป็นโทนสีเดียว

- ทำการกลับภาพ และ ทำการ โย้ภาพไป 45 องศา

รูปที่ 3.12 รูปแสดงการโย้ภาพ

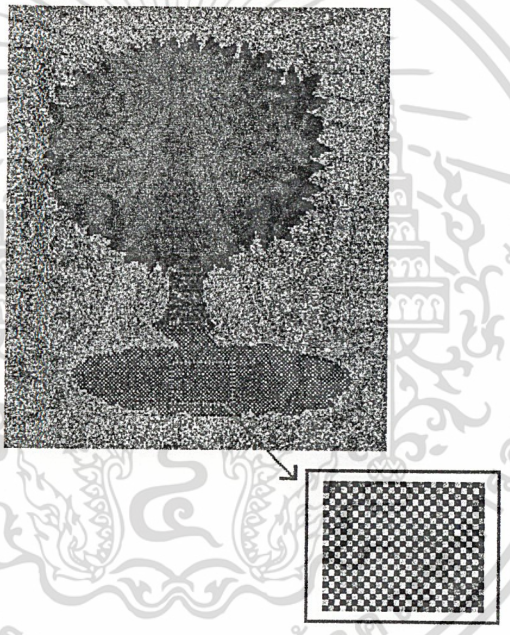
- ต่อไปนำภาพที่ 3.34 และ 3.36 มารวมกัน

รูปที่ 3.13 รูปแสดงการรวมวัตถุและเงา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 การแปะภาพเพื่อนำมาแสดงผลในเกม

ในส่วนของการแสดงผลเราจะนำรูปทั้งหมดไปแปะในบัฟเฟอร์เลยไม่ได้ เพราะส่วนที่เป็นเงาจะมืดเกินไปจนไม่เห็นพื้นหลัง การที่เราจะทำให้เห็นพื้นหลังและเงาของวัตถุได้ทั้ง 2 อย่างพร้อมกันนั้น จะต้องทำการเฉลี่ยค่าทั้ง 2 รูปเข้าด้วยกัน แต่ข้อจำกัดในเรื่องของความเร็วถ้าทำการเฉลี่ยค่าของรูปปริมาณมากจะทำให้โปรแกรมทำงานได้ไม่ทัน เพราะฉะนั้นจะใช้เทคนิคของการสลับจุดเล็กในส่วนของเงาที่เป็นสีดำ กับจุดที่ทำให้ใสไปถึงพื้นดิน ซึ่งเทคนิคนี้แม้จะให้คุณภาพของรูปที่ลดลงแต่ก็สามารถทำงานได้เร็วกว่าแบบแรกมาก



รูปที่ 3.14 รูปแสดงการใช้เทคนิคสลับจุด

3.5.3 การสร้างภาพรถถัง

การสร้างภาพรถถังนั้นจะสร้างโดยการนำภาพของชิ้นส่วนต่างๆ ของรถถังมาแปะรวมกัน โดยชิ้นส่วนแต่ละชิ้นส่วนจะแบ่งเป็นโซนสีต่างๆ เมื่อทำการโหลดภาพจากไฟล์ภาพที่ได้จะมีลักษณะของโทนสีง่ายๆ จากนั้น จะทำการนำ Texture ไปแปะยังส่วนต่างๆ ของรถถังเพื่อจะทำให้เกิดภาพที่มีมุมมองเหมือน 3 มิติ

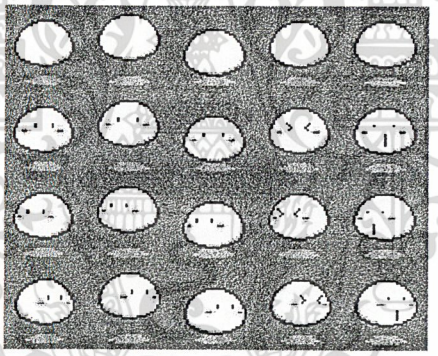
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.15 รูปภาพแสดงการใส่ Texture ลงในชิ้นส่วนของรถถัง

จากรูป เป็นการใส่ Texture ในชิ้นส่วนที่เป็นอาวุธของรถถัง จากขั้นตอนนี้จะทำเหมือนกันในทุกชิ้นส่วนของรถถัง หลังจากนั้นจะได้ชิ้นส่วนที่ใส่ Texture แล้ว เมื่อนำมาแปะรวมกันจะได้ภาพรถถังที่สมบูรณ์

3.5.4 การสร้างภาพมอนสเตอร์

การสร้างภาพมอนสเตอร์จะสร้างโดยการวาดรูปมอนสเตอร์ในอิริยาบถต่างๆ ไว้แล้วทำการเขียนโปรแกรมเพื่อทำการดึงภาพมาแสดงผลให้มอนสเตอร์สามารถแสดงท่าทางตามต้องการ



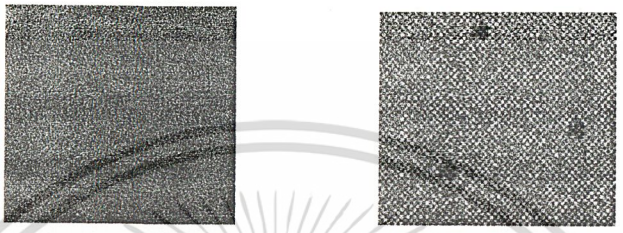
รูปที่ 3.16 รูปแสดงการออกแบบมอนสเตอร์

3.5.5 การสร้างผิวน้ำ

การสร้างผิวน้ำสามารถทำได้โดยขั้นแรก ต้องสร้าง Texture ของส่วนที่เป็นผิวน้ำขึ้นมา หากนำ Texture ของพื้นน้ำไปแปะลงในฉากเลย ภาพที่ได้จะมีลักษณะของน้ำที่นิ่งซึ่งจะทำให้ภาพที่ได้ขาดความสมจริง เพื่อสร้างความสมจริงให้กับพื้นน้ำมากขึ้นสามารถทำได้โดยการทำให้ผิวน้ำดูเหมือนเคลื่อนไหว โดยในการแปะ Texture นั้นจะต้องทำการแปะโดยการเลื่อนตำแหน่งของการแปะ Texture ไปเรื่อยๆ ตามแนวทิศทางการไหลของน้ำ เมื่อทำเช่นนี้แล้วจะทำให้ภาพที่ได้มีความสมจริงยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างความน่าสนใจให้กับพื้นผิวน้ำมากขึ้นสามารถทำได้โดยการนำค่าสีของ Texture ของพื้นน้ำ มาเฉลี่ยกับค่าสีของส่วนที่เป็นพื้นดินที่อยู่ใต้ผิวน้ำ จะทำให้พื้นผิวน้ำที่ได้มีลักษณะใส และเพิ่มความสมจริงให้กับผิวน้ำมากขึ้น



รูปที่ 3.17 รูปแสดงการเปรียบเทียบของผิวน้ำที่ทำให้ใสโดยการเฉลี่ยค่าสี

3.5.6 การคำนวณการเคลื่อนที่ของวัตถุ

การคำนวณการเคลื่อนที่ของวัตถุ คือการหาตำแหน่งที่จะทำการแปะภาพในแต่ละครั้ง ซึ่งจะทำให้ภาพที่ได้เหมือนมีการเคลื่อนที่ สำหรับในส่วนของตัวเกม จะใช้การเคลื่อนที่ทั้งหมด 4 แบบด้วยกัน ได้แก่ การเคลื่อนที่แบบตรง การเคลื่อนที่แบบโค้ง การเคลื่อนที่แบบวงกลม และการเคลื่อนที่แบบคลื่น

- การเคลื่อนที่แบบตรง สามารถทำได้โดยการคำนวณหาตำแหน่งจากสมการของเส้นตรง $Ax + By + C = 0$
- การเคลื่อนที่แบบโค้ง สามารถทำได้โดยการคำนวณหาตำแหน่งจากสมการ $y^2 + Dx + Ey + F = 0$
- การเคลื่อนที่แบบวงกลม สามารถทำได้โดยการคำนวณหาตำแหน่งจากสมการ $x^2 + y^2 + Ax + By + C = 0$
- การเคลื่อนที่แบบคลื่น สามารถทำได้โดยการคำนวณหาตำแหน่งจากสมการของ sine

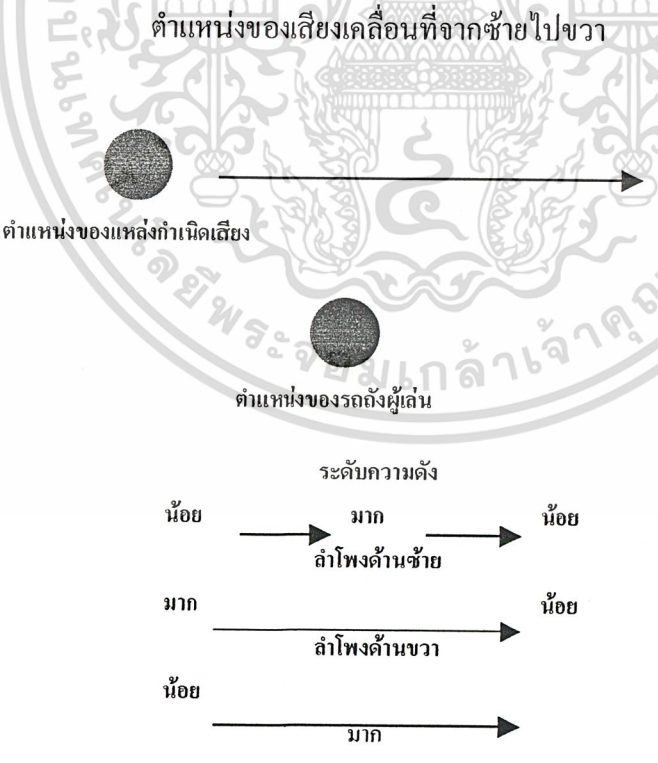
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบระบบเสียงในเกม

ระบบเสียงในเกมจะแบ่งออกเป็น 2 ส่วนด้วยกัน คือส่วนของเสียงดนตรี และส่วนของเสียงเอฟเฟ็คท์

ส่วนของเสียงดนตรี จะทำการโหลดขึ้นเมื่อเข้าสู่ฉาก และจะทำการเปลี่ยนแปลงเสียงดนตรีไปตามเหตุการณ์ที่เกิดขึ้น

ส่วนของเสียงเอฟเฟ็คท์ ในส่วนนี้จะมีลูกเล่นมากกว่า คือจะมีลูกเล่นในส่วนของการแพนเสียงและการปรับระดับความดังของเสียง โดยการแพนเสียงและการปรับระดับความดังของเสียงนี้จะยึดจากตำแหน่งของรถถังผู้เล่น และแหล่งกำเนิดเสียง โดยเมื่อแหล่งกำเนิดเสียงอยู่ทางด้านซ้ายมือของรถถังผู้เล่น เสียงที่เกิดขึ้นก็จะเกิดทางลำโพงด้านซ้าย และเมื่อแหล่งกำเนิดเสียงเคลื่อนที่ผ่านผู้เล่นไปทางด้านขวา ก็จะทำให้เกิดการแพนเสียงจากลำโพงด้านซ้ายไปทางลำโพงด้านขวา ส่วนระดับความดังของเสียงก็จะเพิ่มมากขึ้นเมื่อแหล่งกำเนิดเสียงเคลื่อนที่เข้าใกล้ตำแหน่งของรถถังผู้เล่นมากขึ้น และระดับความดังของเสียงจะลดลงเมื่อตำแหน่งของแหล่งกำเนิดเสียงเคลื่อนที่ไกลออกไปจากตำแหน่งรถถังผู้เล่น



รูปที่ 3.18 รูปแสดงการแพนและปรับระดับของเสียงเอฟเฟ็คท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 หลักการทำงานของ Map Engine

BCX-Map Engine เป็นเครื่องมือที่ใช้ในการสร้างฉากในเกม Battle City Extreme เนื่องจากการสร้างเกมแต่ละเกมนั้นต้องใช้เวลาในการสร้างเป็นเวลานาน โดยเฉพาะในส่วนที่เป็นกราฟฟิกและข้อมูลในแต่ละฉาก ซึ่งเป็นงานที่ค่อนข้างซ้ำซากจำเจ และไม่สามารถทำได้ดีนักโดยการออกแบบผ่านทางการเขียนโปรแกรมโดยตรง ย่อมจะเป็นการดีกว่าถ้าหากเราสามารถออกแบบผ่านทางการเชื่อมต่อในรูปแบบกราฟฟิก แล้วทำการแปลงภาพกราฟฟิกที่ได้มาเป็นส่วนของโปรแกรมโดยอัตโนมัติ

เพราะฉะนั้นผู้ที่สร้างเกมส่วนมากต่างได้มีการนำเครื่องมือ (Game Engine) มาใช้ช่วยในการสร้างเกม ซึ่งจะทำให้สามารถสร้างเกมได้รวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น สำหรับใน Battle City Extreme นั้นก็เช่นเดียวกันกับเกมอื่นๆ คือมีปัญหาในการออกแบบฉาก เพื่อแก้ไขปัญหานี้ Battle City Extreme จึงได้มีการนำ Game Engine เข้ามาช่วยในการออกแบบฉากแต่ละฉาก เพื่อให้การทำงานมีประสิทธิภาพมากขึ้น

Game Engine ที่มีในปัจจุบันก็มีอยู่หลายตัวด้วยกัน แต่การที่จะใช้Game Engine ที่ทำงานเข้ากับ Battle City Extreme 100 เปอร์เซ็นต์นั้นคงเป็นไปได้ยาก เพราะจะมีปัญหาในส่วนของรูปแบบข้อมูลที่ไม่เหมือนกัน ถ้าหากจะให้การทำงานเป็นไปได้ดีที่สุดจะต้องเขียน Game Engine ขึ้นใช้เอง เนื่องจากทีมงานมีความเข้าใจเกี่ยวกับระบบของตัวเกมนี้ดีอยู่แล้ว ย่อมไม่ใช่เรื่องยากมากนักที่จะพัฒนา Game Engine ขึ้นใช้เอง ในการพัฒนา Game Engine ขึ้นใช้นั้น หากเราเลือกเฉพาะส่วนที่จำเป็นจริงๆ จะทำให้ใช้เวลาไม่มากนักในการพัฒนา ซึ่งก็นับว่าเป็นความคุ้มค่าในการพัฒนา Engine ขึ้นใช้เอง ซึ่งในส่วนของตัว Battle City Extreme นั้นจะใช้ Engine ช่วยในส่วนของการสร้างฉากเท่านั้น ซึ่ง Engine ตัวนี้ทางทีมงานได้ใช้ชื่อว่า “BCX-Map Engine ”

การใช้งาน BCX-Map Engine

- กำหนดค่าเริ่มต้นของฉากแต่ละฉาก
- กำหนดจำนวนและตำแหน่งวัตถุต่างๆ ในฉาก โดยการลากแปะ
- กำหนดจำนวนและตำแหน่งรถถังในฉาก โดยการลากแปะ
- กำหนดจำนวนและตำแหน่งมอนสเตอร์ในฉาก
- สร้างฉากหลังของแต่ละฉากโดยการนำพื้นผิวรูปแบบต่างๆ ไปวางในฉากหลัง
- กำหนดสถานะและความสูงของพื้นผิวในฉาก
- สร้างส่วนประกอบอื่นๆ ที่เพิ่มความสวยงามให้แก่ฉาก
- สร้างเอฟเฟ็คท์ภายในฉาก
- แปลงข้อมูลเป็นไฟล์ .MAP (เป็นรูปแบบไฟล์ที่ใช้เป็นฉาก ใน Battle City Extreme)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Save / Load ข้อมูลในการสร้างฉาก

3.8โครงสร้างของไฟล์ .MAP

ไฟล์ .MAP นี้เป็นไฟล์แผนที่ของฉากแต่ละฉากในเกม ซึ่งไฟล์ชนิดนี้จะสร้างจากโปรแกรม BCX-Map Engine ซึ่งเป็นโปรแกรมที่ทีมงานสร้างขึ้นเพื่อใช้สร้างฉากในเกมโดยเฉพาะ โครงสร้างของ ไฟล์ .MAP ประกอบด้วย

- **Version** เป็นข้อมูลชนิด integer จะเก็บค่าเวอร์ชันของ Map Engine
- **Width** เป็นข้อมูลชนิด integer จะเก็บค่าความกว้างของฉาก
- **Hight** เป็นข้อมูลชนิด integer จะเก็บค่าความสูงของฉาก
- **MainFloor** เป็นข้อมูลชนิด integer จะเก็บค่า ID ของพื้นหลักของฉาก
- **FloorStatus** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [width/40][hight/40] ใช้เก็บข้อมูลสถานะของพื้นดิน
- **FloorHight** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [width/40][hight/40] ใช้เก็บข้อมูลความสูงของพื้นดิน
- **Floor3D_Count** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [36][4][8] ใช้เก็บจำนวนของพื้นที่เป็น 3 มิติทั้งหมดในฉากไว้
- **Floor3D_Buffer** เป็นข้อมูลชนิดอาร์เรย์ชนิด POINT ขนาด [36][4][8] ใช้เก็บตำแหน่งของพื้นที่เป็น 3 มิติ
- **FloorCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของการนำ Texture ไปแปะที่พื้นของฉาก
- **FloorType** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่าชนิดของ Texture
- **FloorShape** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่ารูปแบบของ Texture
- **FloorSize** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่าขนาดของ Texture
- **FloorPoint** เป็นอาร์เรย์ชนิด POINT ที่มีขนาด [FloorCount] ใช้เก็บค่าตำแหน่งของ Texture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **FloorRect** เป็นอาร์เรย์ชนิด RECT ที่มีขนาด [FloorCount] ใช้เก็บค่าขนาดของ Surface ที่เกิดจากการ copy และ paste
- **ObjectContent** เป็นข้อมูลชนิด integer ใช้เก็บค่าจำนวนของรูปจำลองต้นแบบของวัตถุ
- **ModelObjectType** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ที่มีขนาด [ObjectContent] ใช้เก็บค่าชนิดของรูปจำลองต้นแบบของวัตถุ
- **ObjectCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของวัตถุทั้งหมดภายในฉาก
- **ObjectType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด [ObjectCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของวัตถุ
- **ObjectPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [ObjectCount] ใช้เก็บค่าตำแหน่งของวัตถุ
- **TankContent** เป็นข้อมูลชนิด integer ใช้เก็บค่าจำนวนของรูปจำลองต้นแบบของรถถัง
- **TankBody** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนตัวถังรถของรูปจำลองต้นแบบของรถถัง
- **TankWheel** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนล้อรถของรูปจำลองต้นแบบของรถถัง
- **TankWeapon** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent][4] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนอาวุธรถของรูปจำลองต้นแบบของรถถัง
- **TankBackpack** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนเครื่องหลังของรูปจำลองต้นแบบของรถถัง
- **TankComputer** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนคอมพิวเตอร์ของรูปจำลองต้นแบบของรถถัง
- **TankTexture** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของTextureของรูปจำลองต้นแบบของรถถัง
- **TankPirot** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขคนขับของรูปจำลองต้นแบบของรถถัง
- **TankCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของรถถังทั้งหมดภายในฉาก
- **TankType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด[TankCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของรถถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **TankPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [TankCount] ใช้เก็บค่าตำแหน่งของรถถัง
- **Music** เป็นข้อมูลชนิด integer ที่ใช้เก็บหมายเลขเสียงดนตรีของฉาก
- **PlayerTankPosition** เป็นข้อมูลชนิด POINT ที่ใช้เก็บตำแหน่งเริ่มต้นของรถถังผู้เล่น
- **MonsterCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของมอนสเตอร์ทั้งหมดภายในฉาก
- **MonsterType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด [MonsterCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของมอนสเตอร์
- **MonsterPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [MonsterCount] ใช้เก็บ ค่าตำแหน่งของมอนสเตอร์

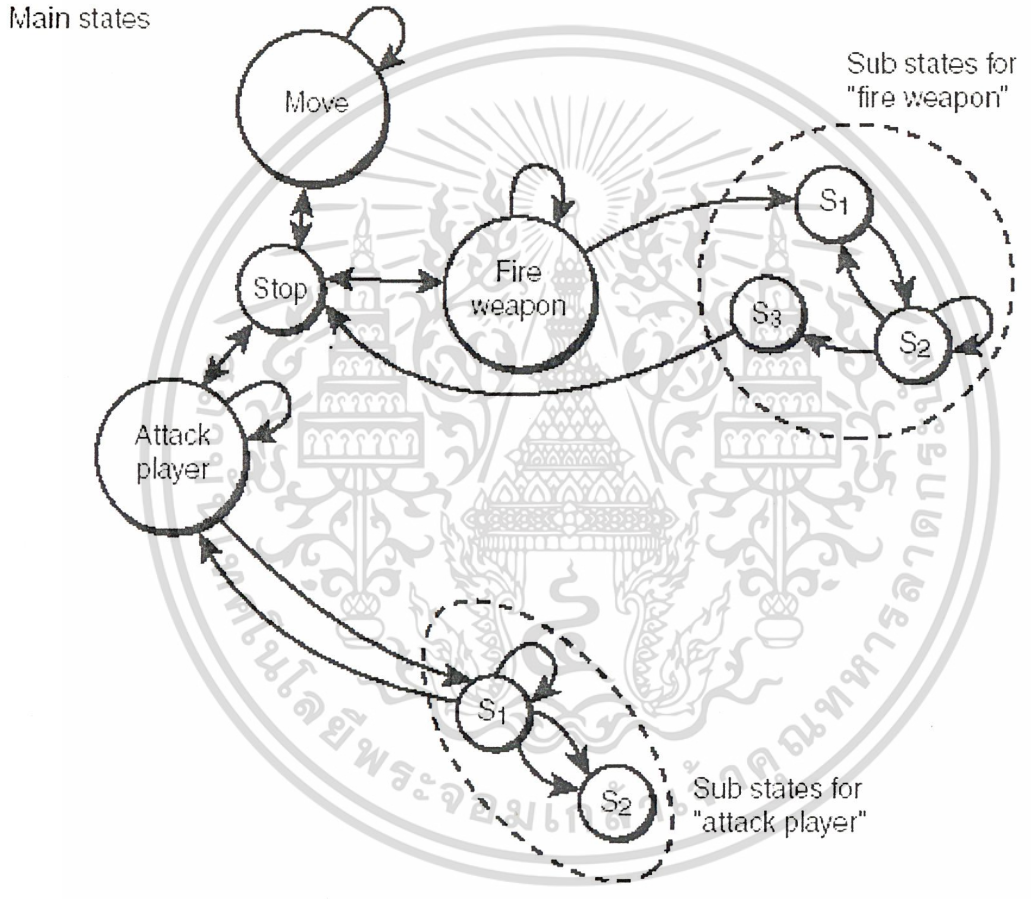
หลักการทำงานของ MapEngine จะมีหลักการทำงานเหมือนกับโปรแกรมวาดรูปทั่วไป แต่การทำงานของ MapEngine แทนที่จะใช้การวาดรูปจะใช้การนำรูปไปแปะแทน โดยจากรูปที่นำไปแปะโปรแกรม MapEngine จะทำการแมปข้อมูลที่ได้จากการแปะรูปไปเป็นข้อมูลที่ใช้ในเกม โดยจะจัดการเก็บลำดับการทำงานภายใน MapEngine เป็นแบบ History แล้วจะทำการแปลงข้อมูลจาก History ที่เก็บไว้ไปเป็นข้อมูลของไฟล์ .MAP ตามโครงสร้างที่ระบุไว้ข้างต้น

ส่วนสำหรับการนำไปใช้ ในส่วนของตัวเกมจะมีส่วนที่ทำการอ่านข้อมูลจาก ไฟล์ .MAP แล้วแปลงเป็นข้อมูลที่ใช้ในเกม จากนั้นจะนำข้อมูลที่ได้มาทำการสร้างฉากในแต่ละฉากต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การออกแบบระบบควบคุม (Computer Control)

ในการเล่นเกมนั้นสิ่งที่จะทำให้เกมนั้นสนุกได้ขึ้นอยู่กับรายละเอียดของเกมรวมถึงการโต้ตอบจาก Computer ทำให้เกมนั้นดูท้าทาย ในการที่จะทำให้ Computer คิดอย่างมีเหตุผล มีการออกแบบรายละเอียดทั้งหมดก่อน ในกรณี เกมรถถังได้แก่ Status Tank , weapon



รูปที่ 3.19 แสดงระบบโดยรวมของระบบ AI

ในการออกแบบจะแบ่ง AI ออกเป็น 2 ส่วนใหญ่คือ

1 การหาเส้นทาง (Pathfinding)

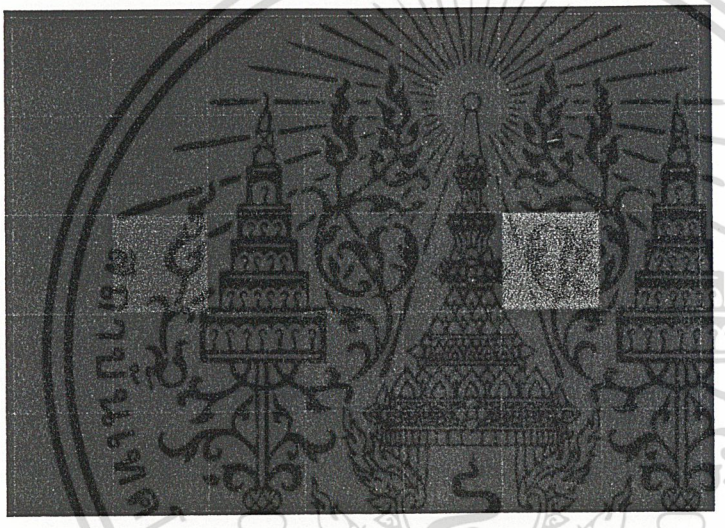
2 การโจมตี (Attack)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.1 การหาเส้นทาง (Pathfinding)

การหาเส้นทาง (Pathfinding) คือการหาเส้นทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง ซึ่งการหาเส้นทางนั้นมีหลายวิธี เช่น Breadth First Search, Depth Fist Search, Bidirectional Breadth - First Search , A* algorithm เป็นต้นแต่ที่เราเลือกนำมาใช้คือ A* algorithm

สมมุติว่าเราต้องการเดินจากจุด A ไปยังอีกจุด B และสมมุติให้ระหว่างจุดมีสิ่งกีดขวางขวางอยู่ แสดงตามภาพข่างล่างนี้ โดยที่สีเขียวเป็นจุดเริ่มต้น (Start A) และ สีแดงเป็นจุด x ปลายทาง (Goal B) และสีน้ำเงินเป็นพื้นที่ที่ไม่สามารถเดินได้



รูปที่ 3.20 แสดงสถานะของพื้นที่

สิ่งแรกที่ต้องทำคือการแบ่งพื้นที่ออกเป็นช่องสี่เหลี่ยมเพื่อง่ายต่อการค้นหาพื้นที่และเส้นทางเดิน ต่อมาจะต้องแปลงพื้นที่ที่แบ่งออกเป็นสี่เหลี่ยมแล้วเป็น Array 2 มิติ โดย Array แต่ Array แทนช่อง สี่เหลี่ยมและแต่ละค่าของ Array จะต้องมีสนะบอกว่าสามารถทำการเดินได้หรือไม่ เมื่อทำการหาเส้นทางได้แล้วในการเดินช่องหนึ่งไปยังอีกช่องหนึ่งจะต้องเดินจากจุดกึ่งกลางของสี่เหลี่ยมช่องหนึ่งไปยังสี่เหลี่ยมอีกช่องหนึ่ง จุดกึ่งกลางเหล่านั้น เรียกว่า โหนด (Node)

โหนดในการหาเส้นทางโดยทั่วไปจะหมายถึง จุดกึ่งกลางของรูปร่างที่ได้ทำการแบ่งพื้นที่ไม่ว่าจะแบ่งพื้นที่ออกเป็น สี่เหลี่ยม สามเหลี่ยม หรือ หกเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

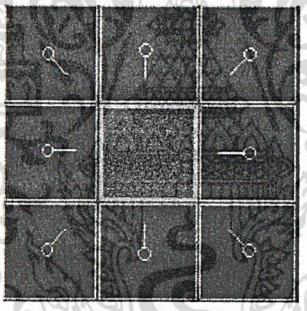
3.9.1.1 ขั้นตอนการค้นหาเส้นทาง

การค้นหาเส้นทางจะเริ่มจากจุด Starting A โดยทำการ ค้นหาพื้นที่สี่เหลี่ยมที่ติดกับจุด Starting A (สี่เหลี่ยมที่อยู่ติดกับสี่เหลี่ยมที่สนใจเรียกว่า Adjacent Square) และทำการเลือกโหนดซึ่งอาจจะเป็นโหนดมีระยะห่างจากจุด Goal มากที่สุด และทำต่อไปเรื่อย ๆ จนกว่าจะค้นพบ Goal ขั้นตอนในการหาคำตอบทำได้โดย

1 เริ่มแรกนำจุด A เป็นโหนด A และเก็บเข้าไปยัง "Open list " โดยที่ Open list เป็น list ซึ่งเก็บโหนด ซึ่งอาจจะเป็น เส้นทางเดิน ไปยังจุด Goal และเป็นโหนด ที่ยังไม่ได้ทำการแตกโหนด

2 ค้นหาทุก ๆ โหนดข้างเคียงของโหนด A ที่สามารถเดินได้ และเก็บค่าลงเข้าไปใน Open list โดย Openlist จะต้องเก็บค่าโหนดแล้วยังต้องเก็บค่าว่าโหนดเหล่านั้นเป็นโหนดลูกของโหนดใด ซึ่งในกรณีเริ่มนั้น โหนดเหล่านี้เป็นโหนดลูกของโหนด A

3 นำโหนด A ออกจาก Openlist และทำการเก็บ ค่า " Closed list" ซึ่งเมื่อเก็บค่าลง Closed list เราจะไม่นำมาพิจารณาในการแตกโหนดอีก



รูปที่ 3.21 แสดงการแตกโหนดในการหาเส้นทาง

จากรูปกำหนดให้

- สี่เหลี่ยมสีเขียวแสดงถึงจุด Start Point
- สี่เหลี่ยมที่มีขอบสีฟ้าหมายถึงได้ถูกเก็บค่าลงใน Closed list
- สี่เหลี่ยมที่มีขอบสีฟ้าหมายถึงได้ถูกเก็บค่าลงใน Open list โดยแต่ละช่องจะมีสัญลักษณ์ว่ามาจากสี่เหลี่ยมรูปใด

หลังจากนั้นเมื่อเสร็จขั้นตอนที่ 3 แล้วให้เลือกโหนดข้างเคียงจาก Open list แล้วทำขั้นตอนที่ 1 – 3 โดยแทนโหนด A เป็นโหนด ที่เลือกมา ทำไปจนกว่าจะพบจุด Goal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.1.2 การคำนวณค่าการตัดสินใจ

ในการค้นหาคำตอบแบบ A* algorithm จะมี function ในการประมาณค่าซึ่งประกอบด้วย

$$f(n) = g(n) + h(n)$$

โดยการหาเส้นทางนั้นค่า g(n) และ h(n) จะกำหนดดังนี้ คือ

g(n) = ระยะทางสะสม (Pathcost) ตั้งแต่โหนดเริ่มต้นไปยังโหนดอ้างอิง โดยในกรณีแตกโหนดที่มุมค่า cost จะเท่ากับ 2 โดยกรณีอื่น ๆ มีค่าเท่ากับ 1

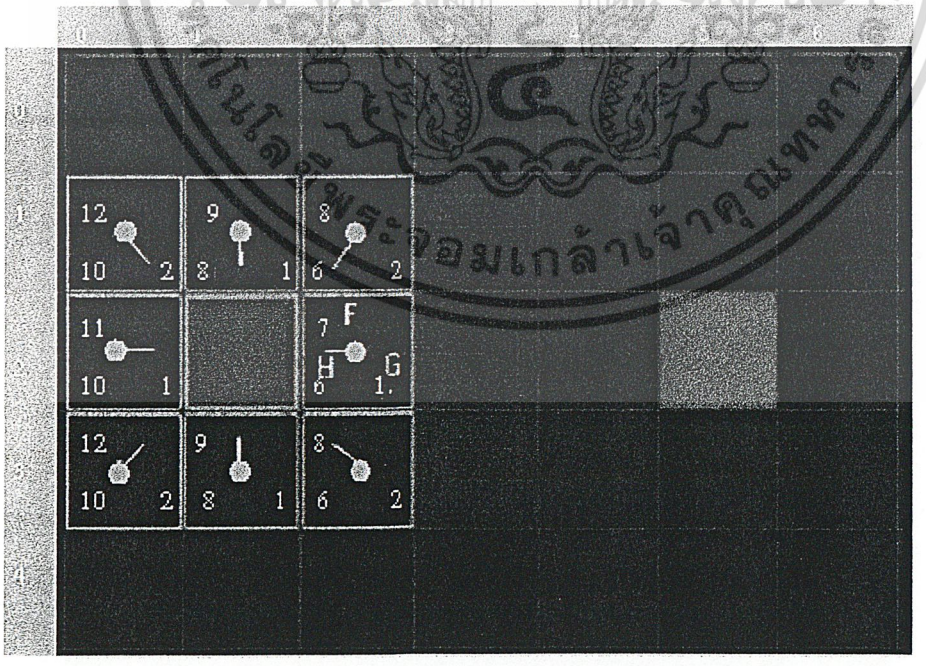
h(n) = (A + B) + | A- B |

โดยที่

A = ระยะห่างตามแนวแกน X ระหว่างจุดอ้างอึง และ จุด Goal

B = ระยะห่างตามแนวแกน Y ระหว่างจุดอ้างอึง และ จุด Goal

เส้นทางการเดินนั้นจะถูกสร้างโดยการเลือกโหนดจาก Open list ที่มีค่าของ f(n) น้อยที่สุดทุกครั้งที่มีการเลือกโหนดจาก Open list ซึ่งเมื่อทำการคำนวณหาค่า f(n) แล้วจะได้ค่าดังรูป



รูปที่ 3.22 แสดงค่า f(n) h(n) และ g(n) ในแต่ละโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาค่า f(n) ในแต่ละโหนด

Node (0,1)

g(n) = 2 ,h(n) = (| 5-0 | + | 2-1|) + || 5-0 | - | 2-1|| = 10

f(n) = 2+10 = 12

Node (1,1)

g(n) = 1 ,h(n) = (| 5-1 | + | 2-1|) + || 5-1 | - | 2-1|| = 8

f(n) = 1+8 = 9

Node (2,1)

g(n) = 2 ,h(n) = (| 5-2 | + | 2-1|) + || 5-2 | - | 2-1|| = 6

f(n) = 2+6 = 8

Node (0,2)

g(n) = 1 ,h(n) = (| 5-0 | + | 2-2|) + || 5-0 | - | 2-2|| = 10

f(n) = 1+10 = 11

Node (2,2)

g(n) = 1 ,h(n) = (| 3-0 | + | 2-2|) + || 3-0 | - | 2-2|| = 6

f(n) = 1+6 = 7

Node (0,3)

g(n) = 2 ,h(n) = (| 5-0 | + | 2-3|) + || 5-0 | - | 2-3|| = 10

f(n) = 1+10 = 12

Node (1,3)

g(n) = 1 ,h(n) = (| 5-1 | + | 2-3|) + || 5-1 | - | 2-3|| = 8

f(n) = 1+8 = 9

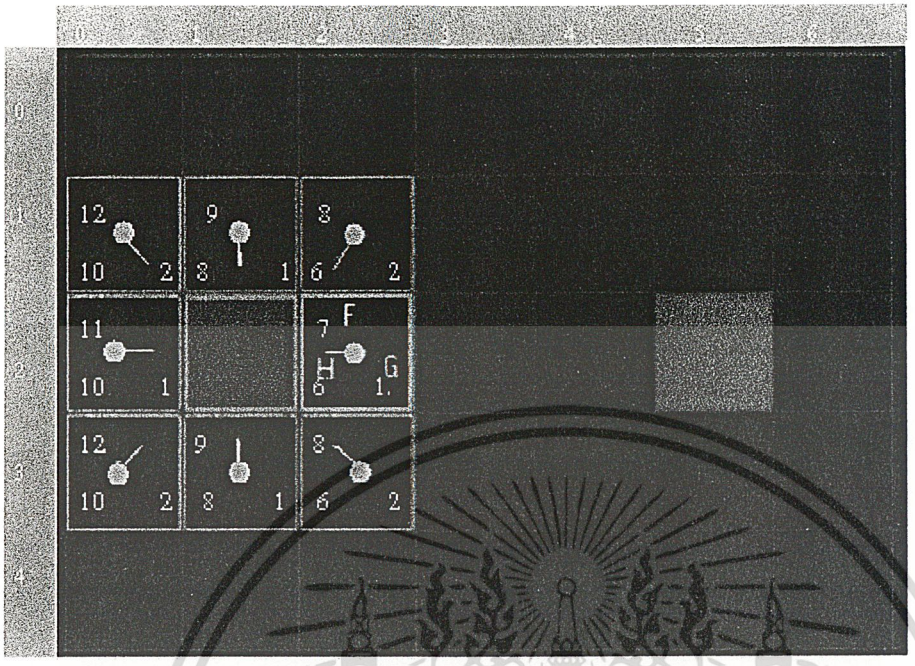
Node (2,3)

g(n) = 2 ,h(n) = (| 5-2 | +| 2-3|) + || 5-2 | -| 2-3|| = 6

f(n) = 2+6 = 8

คำนวณหาค่า f(n) ของแต่ละโหนดแล้วนำมาเปรียบเทียบจะได้ว่าค่าที่น้อยที่สุด ได้แก่ Node (2,2) โดยการเปรียบเทียบนั้นจะรวมไปถึงค่า f(n) ซึ่งอยู่ใน Open list ด้วย โดยที่โหนดที่เป็นโหนดแม่จะถูกนำออกจาก Open List และถูกนำมาเก็บไว้ใน Closed List

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



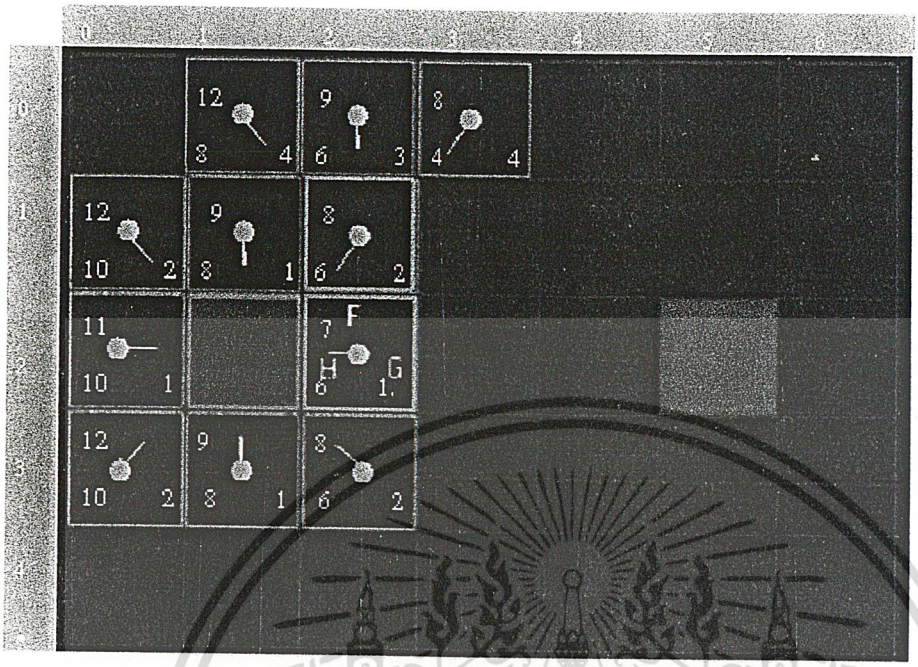
รูปที่ 3.23 แสดงการเลือกโหนด

เมื่อได้เลือกโหนดซึ่งได้จาก f(n) แล้วนำโหนดที่เลือกนำมาหาโหนดข้างเคียงแต่ในกรณีดังตัวอย่าง เราจะได้ว่าเมื่อเลือก Node(2,2) ดังกล่าวแล้วไม่สามารถหาโหนดข้างเคียงได้เนื่องจากสิ่งกีดขวาง ทำให้ไม่สามารถแตกโหนดได้

เมื่อทำการหาโหนดข้างเคียงแล้วจะต้องทำการเลือกโหนดที่จะทำการแตกโหนดต่อไปจากการหาพิจารณาค่า f(n) จะได้ว่า Node(2,1) และ Node(2,3) มีค่าน้อยที่สุด คือ 8

โหนดต่อไปที่จะทำการแตกโหนดจะเป็น Node(2,1) หรือ Node(2,3) ตามตัวอย่างนี้เราจะเลือก Node(2,1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงกระบวนการแตกโหนด

เมื่อเลือก Node(2,1) แล้วจะสามารถหาโหนดข้างเคียงได้ 3 โหนด คือ

Node(0,1) , Node(0,2) และ Node(0,3)

หาค่า f(n) ของแต่ละโหนด

Node(0,1)

g(n) = 2+2 = 4 ,h(n) = (| 5-1 | +| 2-0|) + || 5-1 | - | 2-0|| = 8

f(n) = 4+8 = 12

Node(0,2)

g(n) = 2 +1 = 3 ,h(n) = (| 5-2 | +| 2-0|) + || 5-2 | - | 2-0|| = 6

f(n) = 3+6 = 9

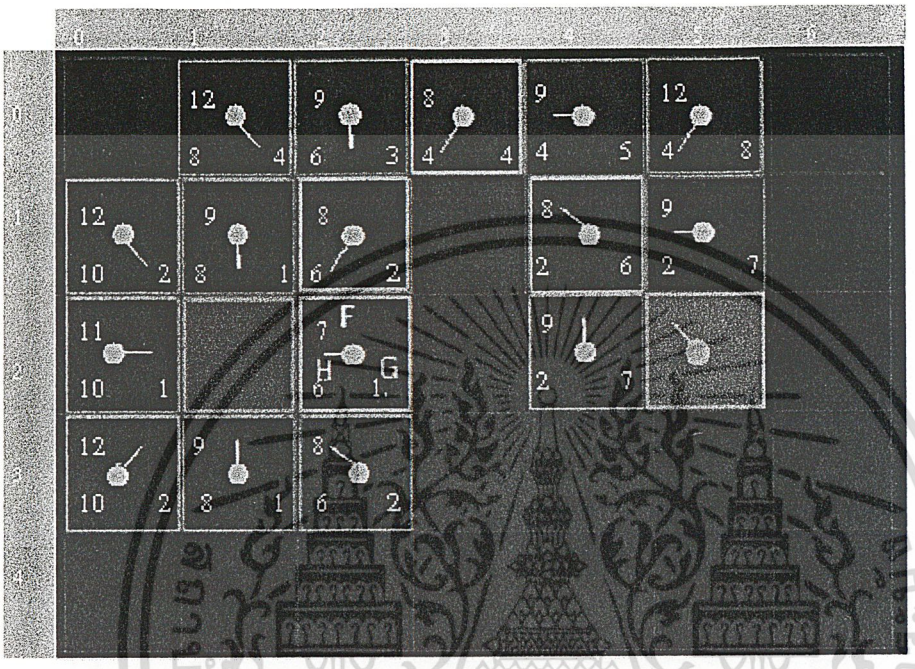
Node(0,3)

g(n) = 2+2 = 4 ,h(n) = (| 5-3 | + |2-0|) + || 5-2 | - |2-0|| = 9

f(n) = 4+4 = 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

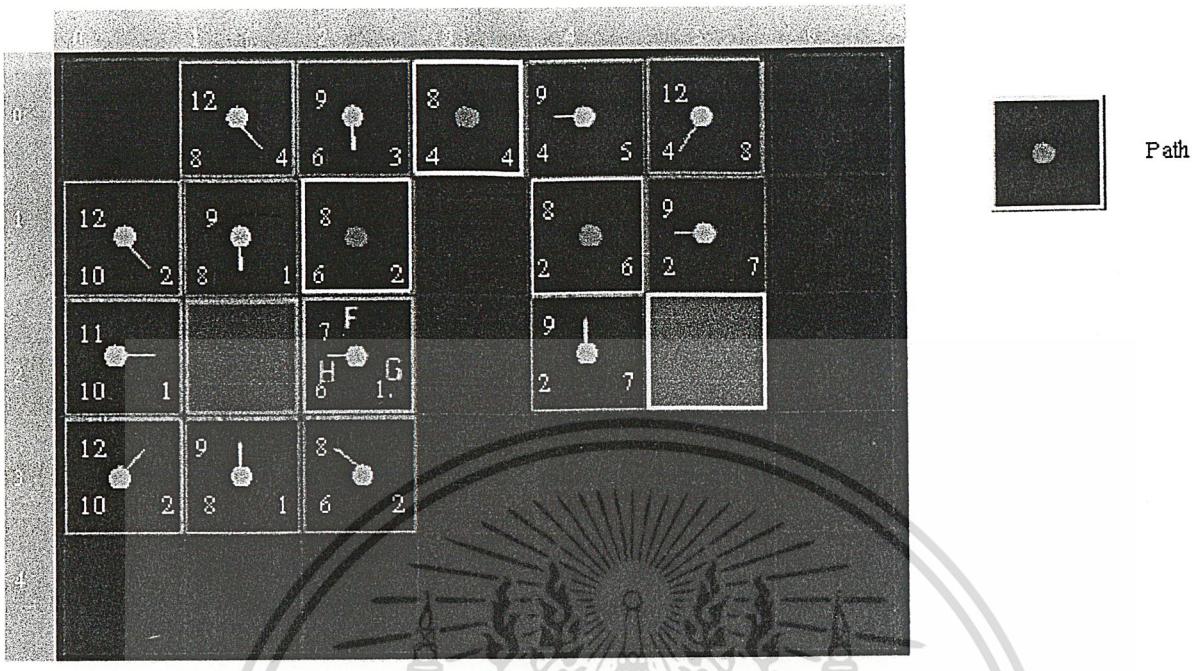
เมื่อพิจารณาโหนดข้างเคียง และโหนดใน Open list จะได้โหนดที่มีค่า f(n) ที่สุดคือ Node (0,3) และ Node (2,3) ซึ่งโหนดนี้เองจะเป็นโหนดแม่เพื่อใช้ในการหาเส้นทางต่อไป ผลลัพธ์สุดท้ายจะได้ดังรูป



รูปที่ 3.25 แสดงกระบวนการแตกโหนด

เมื่อค้นหาเส้นทางถึงจุด Goal แล้วในการสร้างเส้นทางเราจะพิจารณาจากโหนดใน Closed list ซึ่งก็คือ Node (1,2), Node (1,3), Node (2,1), Node (2,2), Node (2,3), Node (3,0) และ Node (4,1) โดยพิจารณาว่าจุด Goal นั้นเป็นโหนดลูกของโหนดใด เมื่อได้โหนดที่เป็นโหนดแม่แล้วจะต้องนำไปพิจารณาว่าโหนดแม่ที่ได้นั้นเป็นโหนดลูกของโหนดใดทำไปจนกว่าจะได้โหนดที่เป็นโหนดเริ่มต้น ดังนั้นเราสามารถสร้างเส้นทางได้คือ เริ่มต้น Node(1,2) -> Node(2,1) -> Node(1,2) -> Node(3,0)-> Node(4,1) -> จุด Goal Node(5,2) แสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

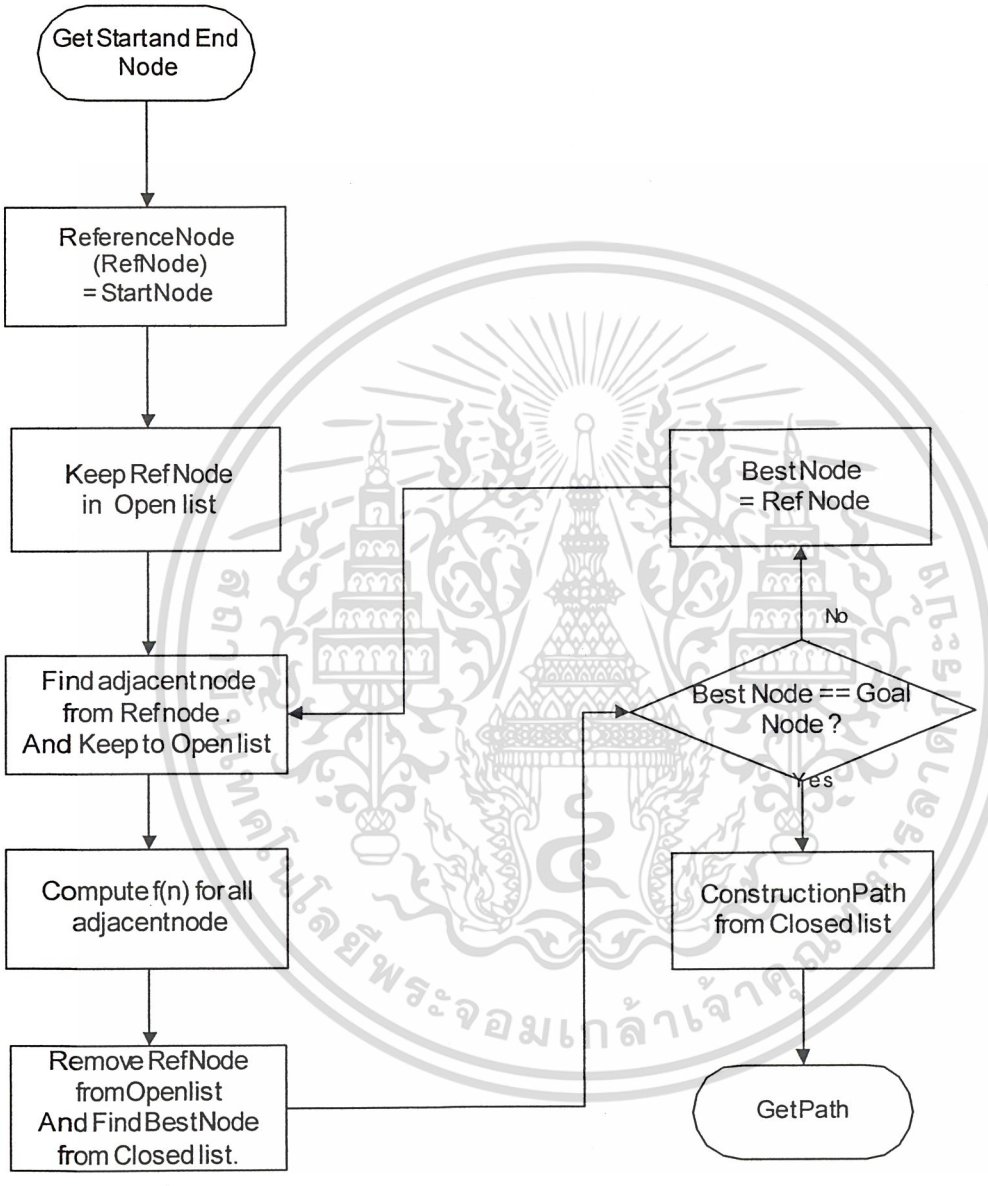


รูปที่ 3.26 แสดงเส้นทางจากจุดเริ่มต้นไปยังเป้าหมาย

จะเห็นค่า f(n) จะเป็นค่าที่ใช้ในการตัดสินใจเลือกโหนดที่จะเป็นเส้นทางเดิน เมื่อเปลี่ยน heuristic Function (h(n)) จะทำให้เส้นทางเดินเปลี่ยนไป ดังนั้นการเลือกควรจะเลือก h(n) ที่เหมาะสมกับปัญหา เพื่อให้ได้เกิดผลลัพธ์ที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow chart แสดงกระบวนแตกโหนดเพื่อทำการหาเส้นทาง



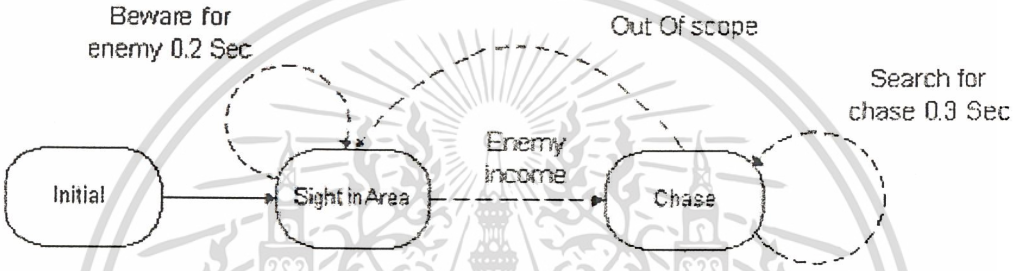
รูปที่ 3.27 Flow chart แสดงกระบวนแตกโหนดเพื่อทำการหาเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.2 การโจมตี (Attack)

การโจมตีถือได้ว่าเป็นส่วนหนึ่งของ AI ก็ว่าได้เพราะเนื่องจาก จะต้องทำให้ Computer พิจารณาว่าควรจะเข้าโจมตีเมื่อไร ซึ่งการเข้าโจมตีนั้นก็ควรจะมีเหตุผลตัวอย่าง เช่น เมื่อเห็นตัวผู้เล่นรถถังจะเข้ามาโจมตี หรือในกรณีรถถังได้ติดอาวุธที่สามารถยิงไกล ได้ก็จะใช้อาวุธยิงไกลเข้าโจมตีตัวผู้เล่นก่อน ดังนั้นเพื่อให้รถถังโจมตีอย่างมีเหตุผลจึงจำเป็นต้องออกแบบในส่วนการโจมตี

ซึ่งสามารถเขียน State diagram ของการโจมตีมีดังต่อไปนี้



รูปที่ 3.28 State diagram ของการโจมตี

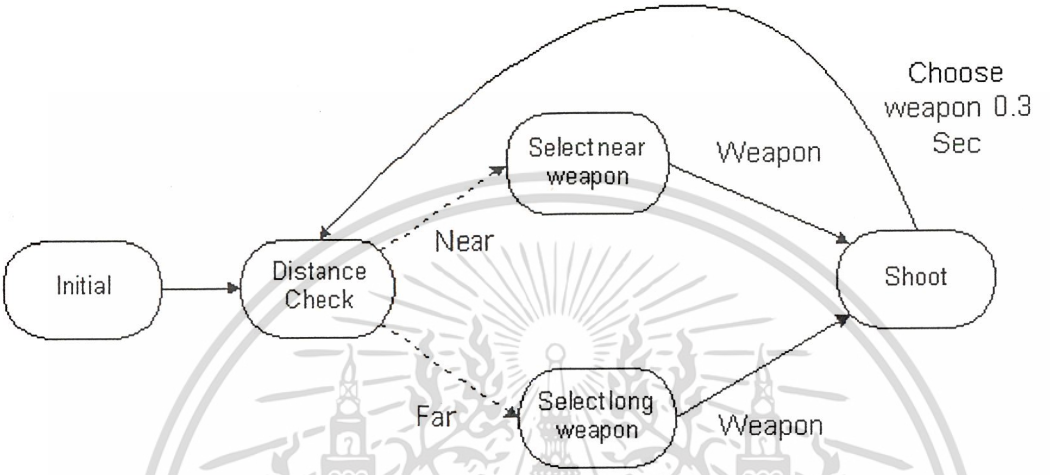
จาก State Diagram ข้างต้นอธิบายได้ดังนี้

Initial State จะเป็น State ของการเริ่มต้นโดยจะประกอบด้วยหน้าที่ของตัวรถถังเองเช่น มีหน้าที่ยืนอยู่กับที่ หน้าที่ในการเดิน random ภายในพื้นที่ที่กำหนด เมื่อดำเนินเกมทุก 0.2 วินาทีจะมีการตรวจว่ามีศัตรูเข้ามาภายในระยะมองเห็นภายในพื้นที่ที่กำหนดไว้หรือไม่ ในกรณีที่ศัตรูเข้ามาในระยะมองเห็น รถถังก็จะทำการเข้าโจมตีตัวผู้เล่นโดยจะทำการค้นหาว่าจะทำการโจมตีอย่างไรทุก ๆ 0.3 วินาที ในกรณีที่ศัตรูออกจากระยะการมองเห็นรถถังจะตัดสินใจว่าจะทำอย่างไรโดยดูจากคำสั่งของตัวรถถังเองและ พื้นที่ที่กำหนดโดยถ้าออกจากพื้นที่ที่กำหนดแล้วรถถังจะทำการกลับเข้ามายัง พื้นที่ที่กำหนดเพื่อทำคำสั่งของตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเลือกใช้อาวุธ

การออกแบบการเลือกใช้อาวุธ จะต้องเลือกใช้ให้เหมาะกับสถานะการ โดยการออกแบบแสดงได้โดย State Diagram ดังนี้



รูปที่ 3.29 State diagram ของการเลือกใช้อาวุธ

จาก State Diagram แสดงถึงการเลือกใช้อาวุธทำการโจมตี โดย Initial State จะอยู่ในสถานะพร้อมโจมตีหรืออยู่ในระยะมองเห็น ซึ่งเมื่อเข้าทำการโจมตีการเลือกใช้อาวุธจะขึ้นอยู่กับระยะทางของ รถถังกับศัตรู ถ้าอยู่ในระยะที่สามารถยิงไกลได้จะทำการเลือกอาวุธยิงไกลเพื่อทำการโจมตี ในทำนองเดียวกันเมื่อศัตรูเข้าใกล้ระยะยิงใกล้ก็จะทำการเลือกอาวุธยิงใกล้เช่นกัน

เมื่อเลือกอาวุธสำหรับการ โจมตีตามแต่ละชนิดแล้ว ยังต้องเลือกด้วยว่าจะโจมตีด้วยอาวุธใดซึ่งการโจมตีนั้นขึ้นอยู่กับตัวแปล 3 ตัวคือ

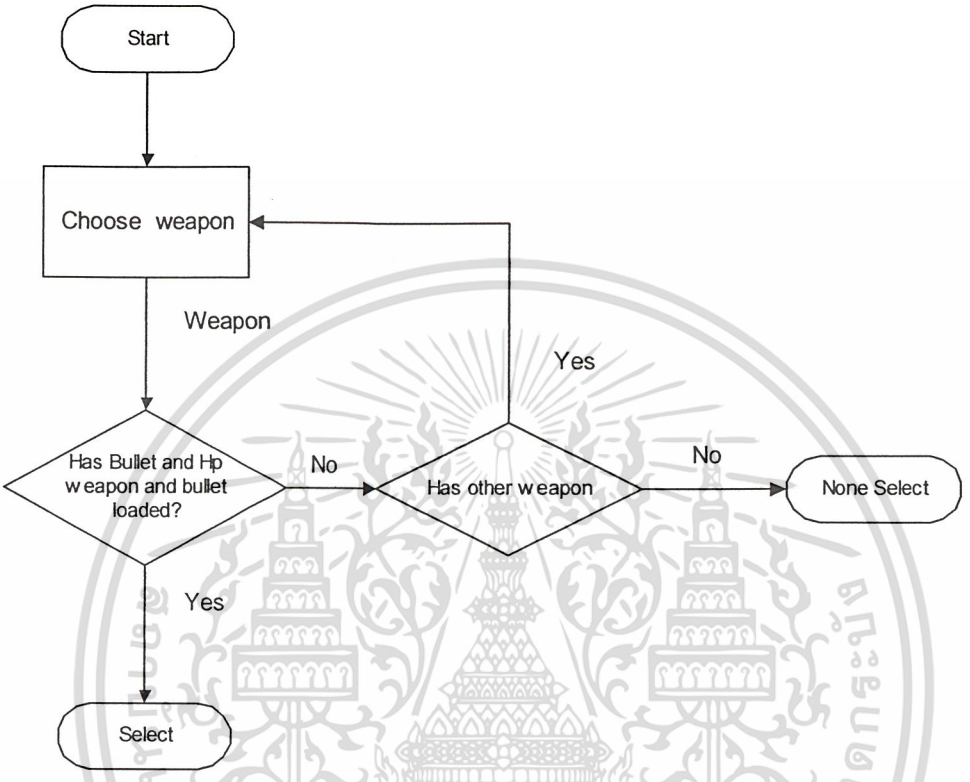
1 กระสุนปืน ในกรณีกระสุนหมดจะไม่ทำการยิงปืนชนิดนั้น

2 พลังของอาวุธ ถ้าเป็น 0 จะหมายถึงอาวุธนั้นเสียหายจะไม่สามารถใช้โจมตีได้

3 การบรรจุกระสุน คือความพร้อมในการยิง กล่าวคือเมื่อทำการโจมตีไปครั้งหนึ่งแล้วจะมีเวลาในการบรรจุกระสุน เมื่อเวลาที่ว่านี้ผ่านไปอาวุธจะสามารถโจมตีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก 3 ตัวแปลข้างต้นเราสามารถเขียน Flowchart ได้ดังนี้



รูปที่ 3.30 Flowchart แสดงกระบวนการเลือกใช้อาวุธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ตัวอย่างผลการดำเนินงาน และ ชิ้นงาน

4 .1รูปแบบของเกม Battle City Extreme

Battle City Extreme เป็นเกมแนวผจญภัยแบบ 2 มิติ ซึ่งจะมีลักษณะของการดำเนินเกมไปตามเนื้อเรื่อง

เนื้อหาของเกม Battle City Extreme จะกล่าวถึงสงครามรถถังที่รบพุ่งกันของฝ่ายต่างๆ อย่างรุนแรง ตัวเอกในการดำเนินเรื่อง จะเป็นสมาชิกในหน่วยเหยี่ยว ซึ่งเป็นหน่วยรบพิเศษ ที่มีความชำนาญสูงในการรบด้วยรถถัง และการลอบจู่โจม

การดำเนินไปของเกมจะเป็นไปตามเนื้อเรื่องซึ่งจะเกิดเหตุการณ์ต่างๆ ขึ้น แล้วจะมีภารกิจที่ต้องทำในแต่ละเหตุการณ์แตกต่างกันไปตามเนื้อเรื่อง

4.2 ฉากเปิดเกม

ฉากเปิดเกมจะเป็นฉากเริ่มต้นเมื่อเข้าสู่เกม จะมีรูปแบบดังรูปที่ 4.1

รูปที่ 4.1 รูปแสดงฉากเปิดเกม

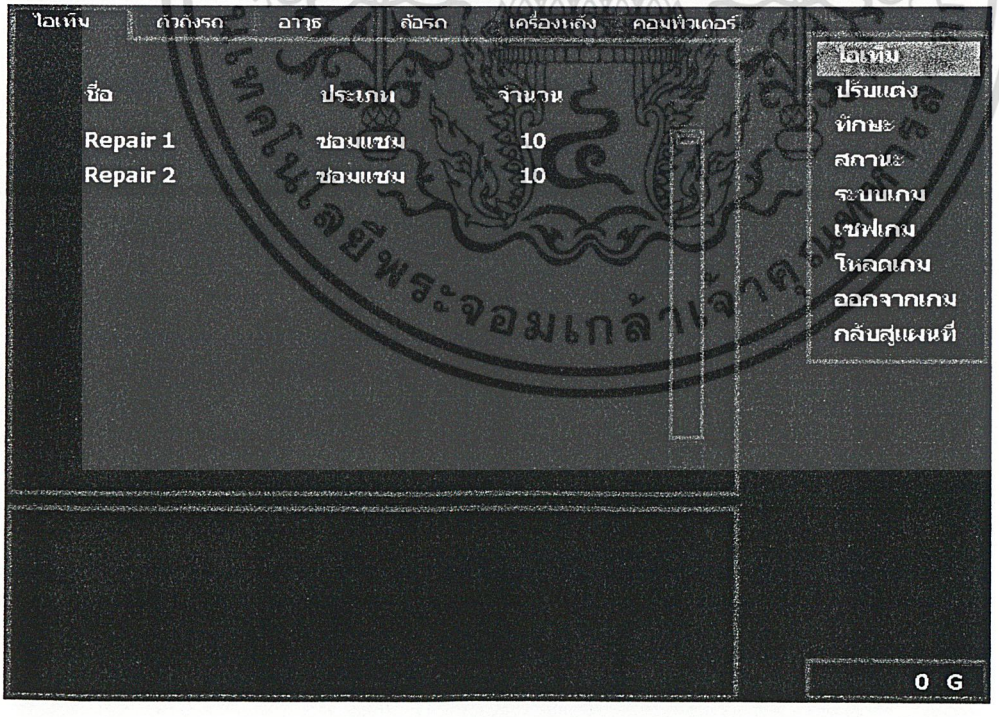
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของฉากเปิดเกมนี้ จะมีเมนูให้เลือกทั้งหมด 4 เมนูด้วยกัน

- เมนูเริ่มเกมใหม่ ในส่วนนี้จะเป็นการเข้าสู่เกมครั้งแรก โดยเรื่องราวในเกมจะเริ่มต้นตั้งแต่แรก
- เมนูโหลดเกม ในส่วนนี้จะเป็นการเข้าสู่เกม โดยการโหลดจากเซฟของเกมที่ได้เซฟไว้
- เมนูระบบเกม ในส่วนนี้จะเป็นการปรับระบบของเกม โดยจะมีส่วนของการควบคุมและส่วนของระบบการซูมภาพ ในส่วนของการควบคุมจะเป็นการเลือกใช้อุปกรณ์ควบคุมว่าจะใช้คีย์บอร์ด หรือ จอยสติ๊ก ส่วนในส่วนของการซูมภาพจะเป็นการปรับระบบการซูมภาพของเกมว่าจะซูมหรือไม่ซูม
- เมนูออกจากเกม ในส่วนนี้จะเป็นเมนูที่ใช้เลือกเมื่อผู้เล่นต้องการออกจากโปรแกรม

4.3 เมนูหลักของเกม

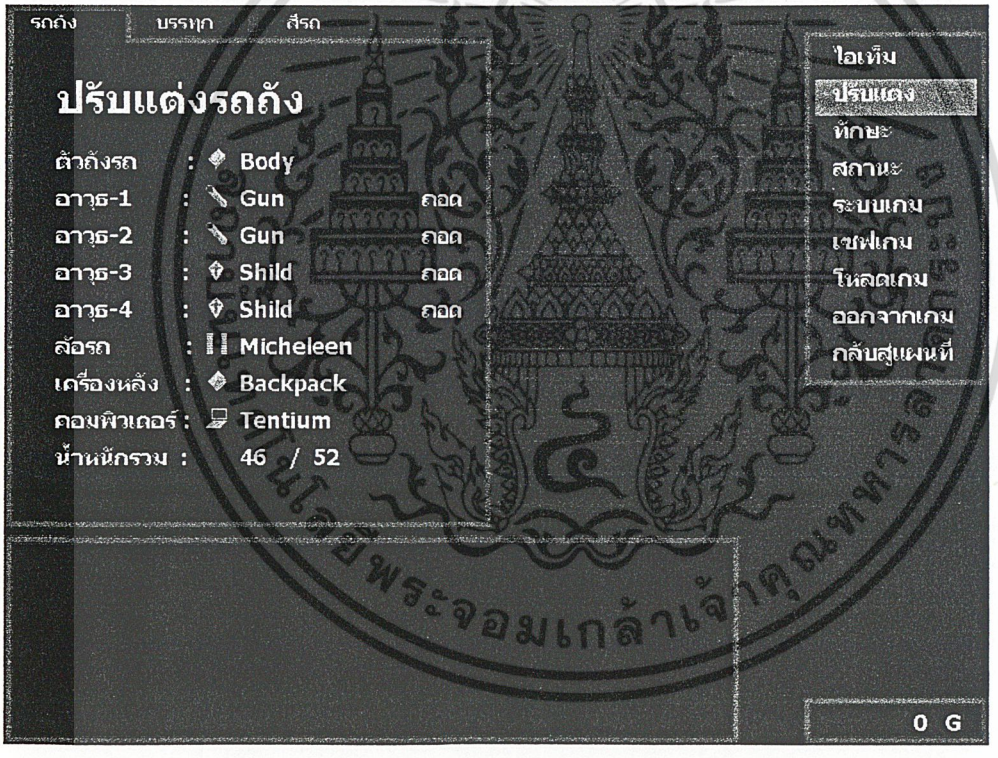
ในส่วนของเมนูหลักของเกมนี้ จะเป็นส่วนที่ผู้เล่นใช้ในการปรับแต่งรวมทั้งควบคุมระบบต่างๆ ของเกม เมื่อผู้เล่นอยู่ในขณะเล่นเกมส่วนประกอบของเมนูหลักของเกมจะประกอบด้วยเมนูย่อยทั้งหมด 9 เมนู ซึ่งมีส่วนประกอบดังนี้



รูปที่ 4.2 รูปแสดงเมนูไอเท็มในเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

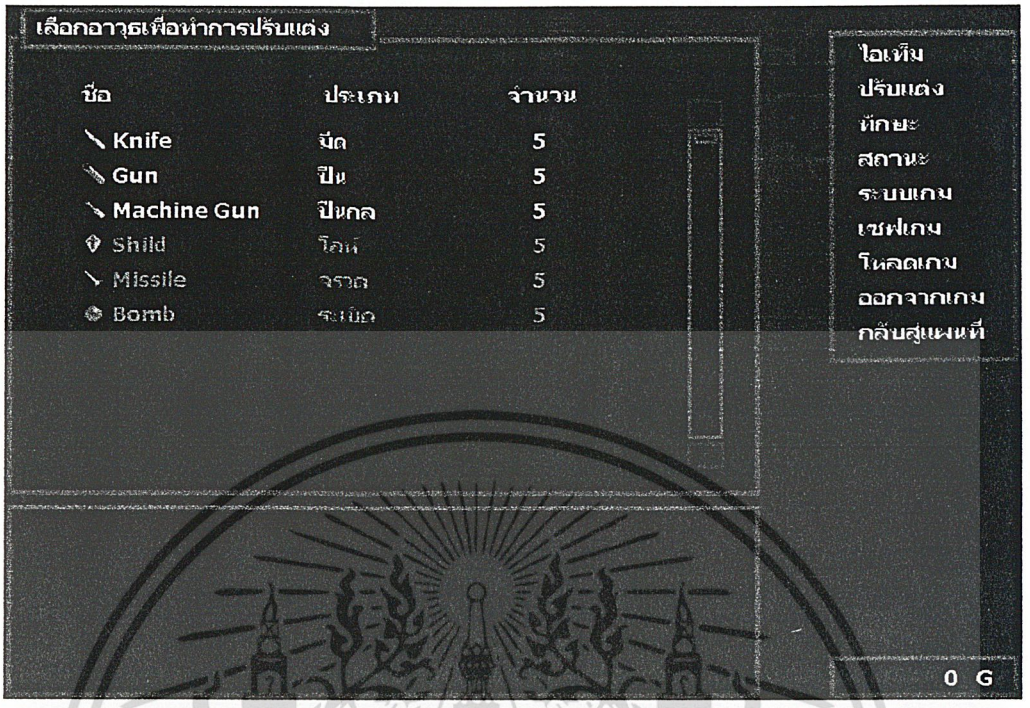
- เมนูไอเท็ม ในเมนูส่วนนี้จะแสดงไอเท็มทั้งหมดของผู้เล่นว่าประกอบด้วยไอเท็มอะไรบ้าง ในส่วนของเมนูนี้จะแบ่งเป็นเมนูย่อยทั้งหมด 6 ส่วน คือ ส่วนที่แสดงไอเท็ม ส่วนที่แสดงตัวถังรถ ส่วนที่แสดงอาวุธ ส่วนที่แสดงล้อรถ ส่วนที่แสดงเครื่องหลัง และ ส่วนที่แสดงคอมพิวเตอร์
- เมนูปรับแต่งรถถัง ในส่วนนี้จะเป็นส่วนที่เกี่ยวกับการปรับแต่งรถถัง ซึ่งประกอบด้วย 3 ส่วนด้วยกัน ส่วนแรกจะเป็นส่วนที่เกี่ยวกับการปรับแต่งชิ้นส่วนของรถถังอันได้แก่ ตัวถังรถ อาวุธ ล้อรถ เครื่องหลัง และคอมพิวเตอร์ ส่วนที่สองส่วนนี้จะเป็นส่วนที่ใช้ในการปรับแต่งการบรรทุกสิ่งของ ซึ่งจะเป็นการกำหนดสิ่งของที่จะนำไปด้วยในฉากต่อสู้ และส่วนสุดท้ายจะเป็นส่วนของการปรับแต่งสีรถ



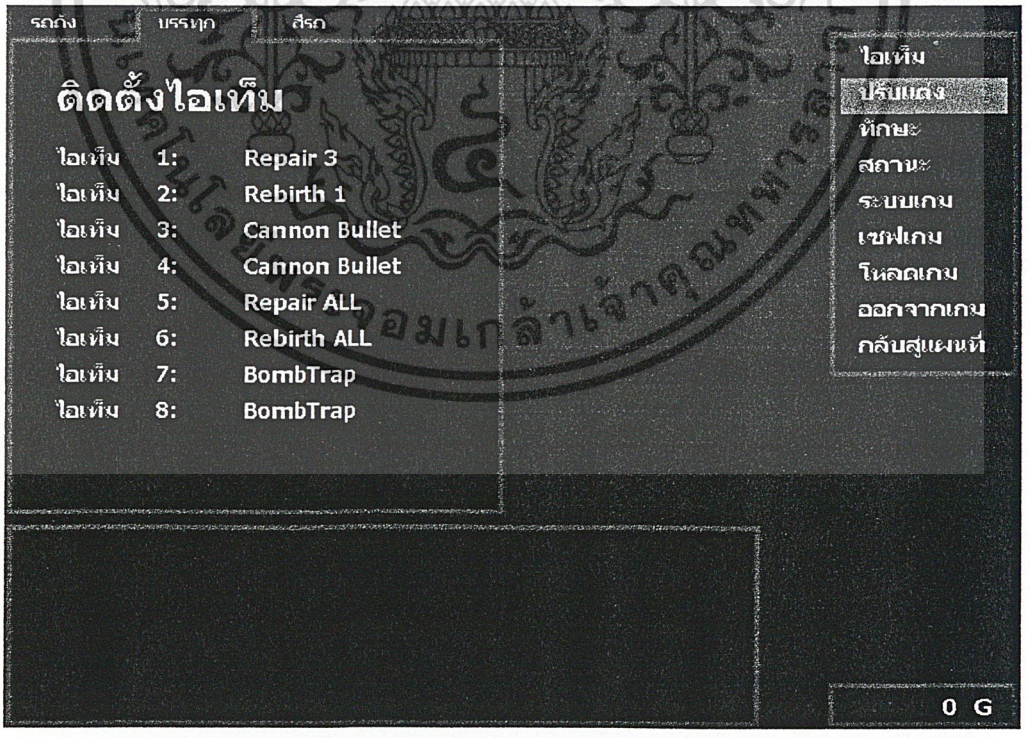
รูปที่ 4.3 รูปแสดงเมนูการปรับแต่งชิ้นส่วนของรถถัง

ในส่วนของเมนูการปรับแต่งชิ้นส่วนของรถถัง ผู้เล่นสามารถเปลี่ยนชิ้นส่วนของรถถังโดยการนำเมาส์ไปคลิกที่ชื่อของชิ้นส่วนนั้นๆ และจะมีเมนูแสดงชิ้นส่วนชนิดนั้นที่มีขึ้นมา ซึ่งเมนูนี้จะแสดงรายชื่อชิ้นส่วนทั้งหมดที่เรามี ชื่อที่เป็นสีขาวจะเป็นชิ้นส่วนที่เราสามารถนำมาประกอบกับรถถังของเราได้ ส่วนชื่อที่เป็นสีเทา ชิ้นส่วนนี้จะไม่สามารถนำมาประกอบกับรถถังของเราได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



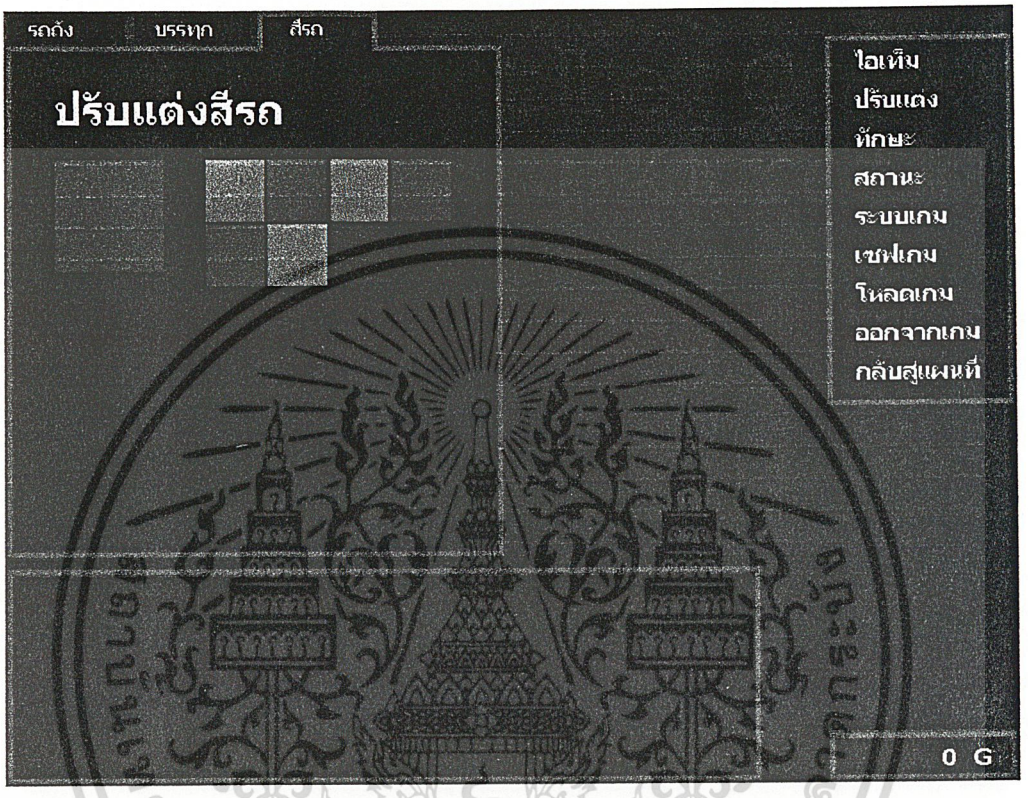
รูปที่ 4.4 รูปแสดงเมนูแสดงชิ้นส่วนที่จะนำมาติดตั้ง



รูปที่ 4.5 รูปแสดงการปรับแต่งการบรรทุกสิ่งของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของเมนูบรรทุกสิ่งของนี้จะแสดงจำนวนช่องที่สามารถบรรทุกไอเท็มได้โดยจำนวนช่องที่บรรทุกไอเท็มได้จะขึ้นอยู่กับจำนวนการบรรทุกของเครื่องหลัง

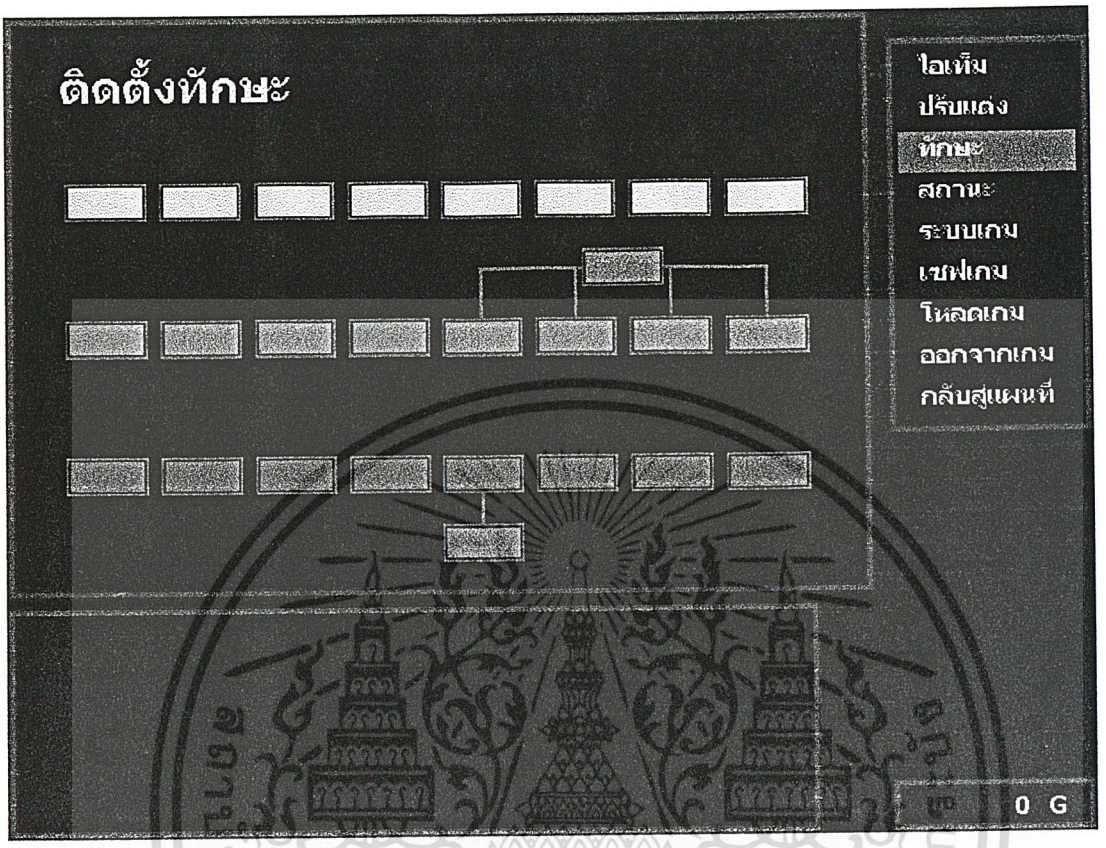


รูปที่ 4.6 รูปแสดงเมนูการปรับแต่งสีของรถถัง

ในส่วนของเมนูปรับแต่งสีรถ ช่องสี่เหลี่ยมขนาดใหญ่ทางด้านซ้ายมือจะแสดงถึงสีที่ผู้เล่นใช้อยู่ขณะนี้ และช่องสี่เหลี่ยมขนาดเล็กทางด้านขวามือจะแสดงถึงสีที่ผู้เล่นสามารถเลือกเพื่อทำการเปลี่ยนสีรถถังของตนได้

- เมนูทักษะ ในเมนูส่วนนี้ผู้เล่นจะใช้ในการปรับแต่งทักษะของตัวละคร ซึ่งทักษะของตัวละครที่จะสามารถนำไปใช้ในเกมได้นั้น จะต้องทำการเชื่อมต่อกับคอมพิวเตอร์ของรถถังก่อน โดยการเชื่อมต่อระหว่างคอมพิวเตอร์กับทักษะของตัวละครสามารถทำได้โดยการนำทักษะของตัวละครไปติดตั้งในสล็อตของเครื่องคอมพิวเตอร์ ซึ่งสล็อตของเครื่องคอมพิวเตอร์นี้จะมีกี่สล็อตขึ้นอยู่กับเครื่องคอมพิวเตอร์แต่ละรุ่น โดยการติดตั้งทักษะของตัวละครในแต่ละสล็อตจะทำได้ก็ต่อเมื่อสล็อตที่จะทำการติดตั้งรองรับประเภทของทักษะชนิดนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 รูปแสดงสล็อตของเครื่องคอมพิวเตอร์

ในส่วนของสล็อตของเครื่องคอมพิวเตอร์นี้ จะแบ่งประเภทของสล็อตเป็น 4 ประเภท คือ

สีเหลือง เป็นCommand Slot เป็นสล็อตที่ใช้ติดตั้งทักษะประเภทเสริมการควบคุม (Command Ability)

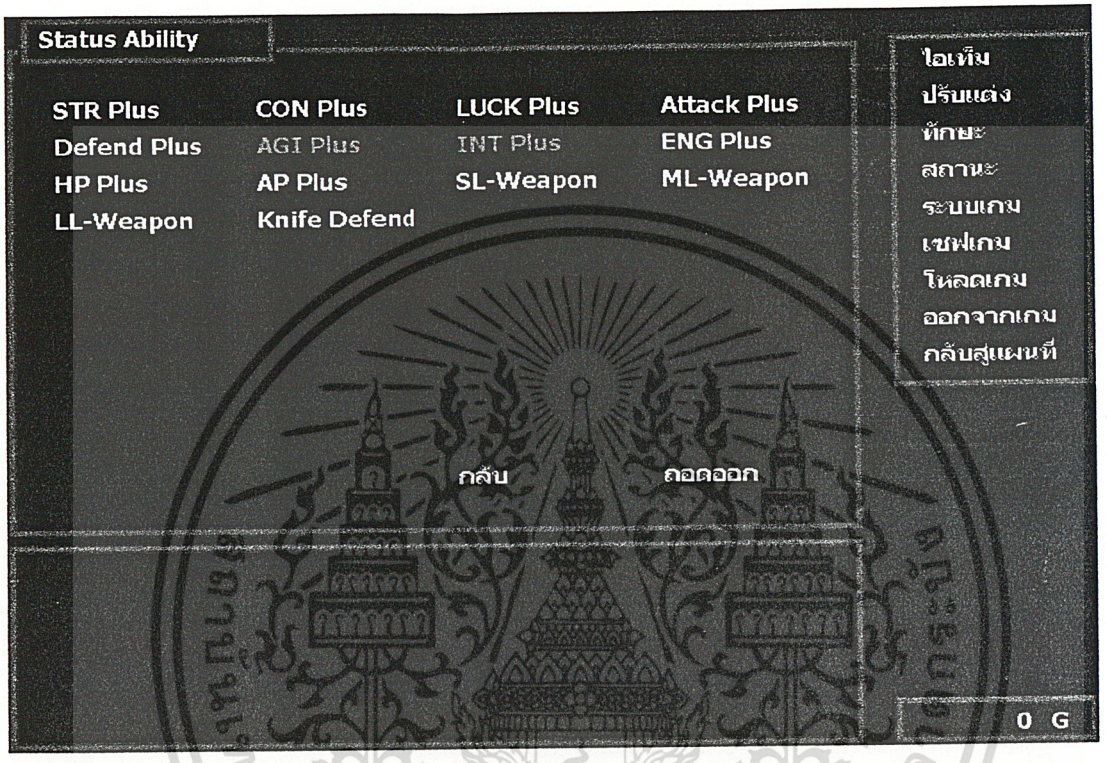
สีเขียว เป็น Fighting Slot เป็นสล็อตที่ใช้ติดตั้งทักษะประเภทเสริมการต่อสู้ (Fighting Ability)

สีชมพู เป็น Status Slot เป็นสล็อตที่ใช้ติดตั้งทักษะประเภทเสริมสถานะ (Status Ability)

สีฟ้า เป็น Support Slot เป็นสล็อตที่ใช้ติดตั้งทักษะประเภทเสริมประสิทธิภาพของ ability ชนิดต่างๆ (Support Ability) สล็อตประเภทนี้จะเสริมความสามารถให้กับสล็อตอื่นที่มีความสัมพันธ์ที่เชื่อมต่อกันเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการติดตั้งทักษะแต่ละชนิดนั้นสามารถทำได้โดยการนำเมาส์ไปคลิกที่ตำแหน่งสล็อตที่ต้องการติดตั้งทักษะนั้นๆ โดยเมื่อนำเมาส์ไปคลิกแล้วจะแสดงเมนู ability ที่ตัวละครตัวนั้นมีเพื่อทำการเลือกนำมาติดตั้งในสล็อตนั้น

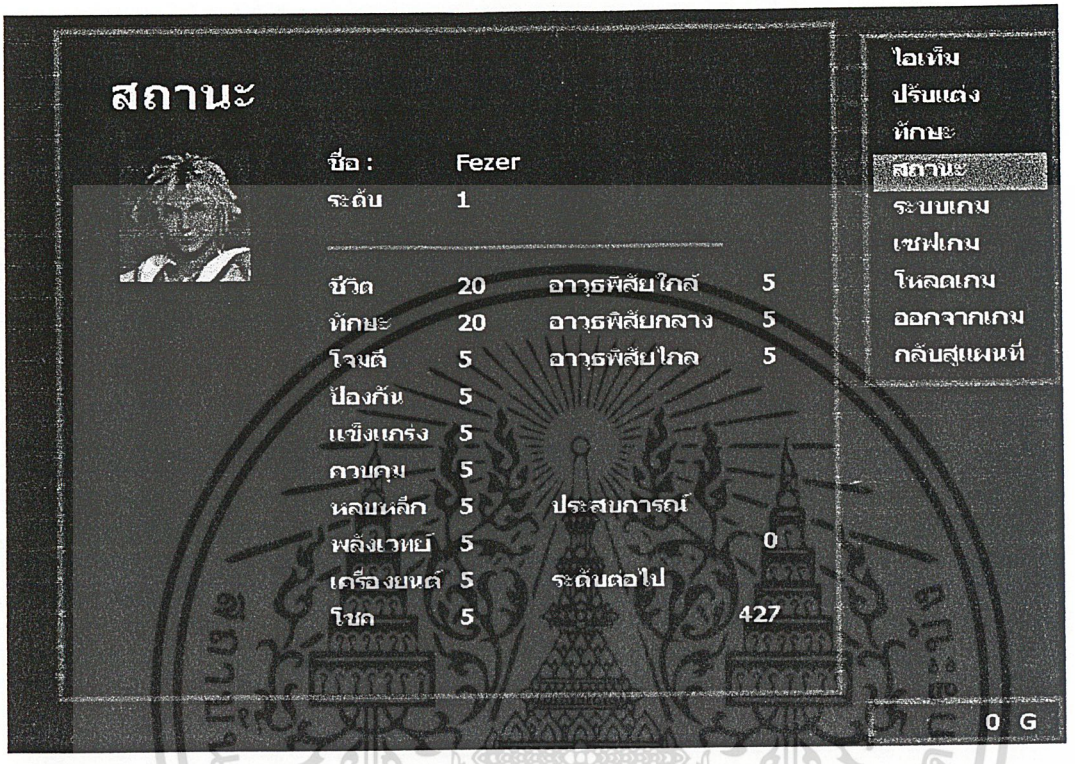


รูปที่ 4.8 รูปแสดงทักษะที่มีเพื่อทำการติดตั้ง

ในส่วนของการเลือกทักษะเพื่อนำไปติดตั้งในสล็อตนั้นจะสามารถเลือกได้เมื่อชื่อของทักษะนั้นมีสีขาว ส่วนชื่อที่เป็นสีเทาเป็นทักษะที่ได้นำไปติดตั้งแล้วจะนำไปติดตั้งอีกไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมนูแสดงสถานะ เมนูนี้จะแสดงค่าสถานะของตัวละครดังรูปด้านล่าง



รูปที่ 4.9 รูปแสดงเมนูสถานะของตัวละคร

ค่าสถานะต่างๆ ประกอบด้วย

ชื่อ	แสดงถึงชื่อของตัวละครที่เป็นคนขับรถถัง
ระดับ	แสดงถึงระดับความสามารถของตัวละครที่เป็นคนขับรถถัง
ชีวิต	แสดงถึงพลังชีวิตของตัวละคร
ทักษะ	แสดงถึงพลังในการใช้ทักษะของตัวละคร
โจมตี	แสดงถึงความสามารถในการโจมตีของตัวละคร
ป้องกัน	แสดงถึงความสามารถในการป้องกันของตัวละคร
แข็งแกร่ง	แสดงถึงความแข็งแกร่งของตัวละคร และความสามารถในการฟื้นตัว
ควบคุม	แสดงถึงความสามารถในการควบคุมรถถังของตัวละคร
หลบหลีก	แสดงถึงความสามารถในการหลบหลีกของตัวละคร
พลังเวทย์	แสดงถึงพลังเวทย์ของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องยนต์ แสดงถึงทักษะในด้านเครื่องยนต์ของตัวละคร

โชค แสดงถึงโชคของตัวละคร

อาวุธพิสัยใกล้ แสดงถึงความชำนาญในการใช้อาวุธประเภทมีด

อาวุธพิสัยกลาง แสดงถึงความชำนาญในการใช้อาวุธประเภทปืนและปืนกล

อาวุธพิสัยไกล แสดงถึงความชำนาญในการใช้อาวุธประเภทจรวดและระเบิด

ประสบการณ์ แสดงถึงค่าประสบการณ์ที่ตัวละครมี

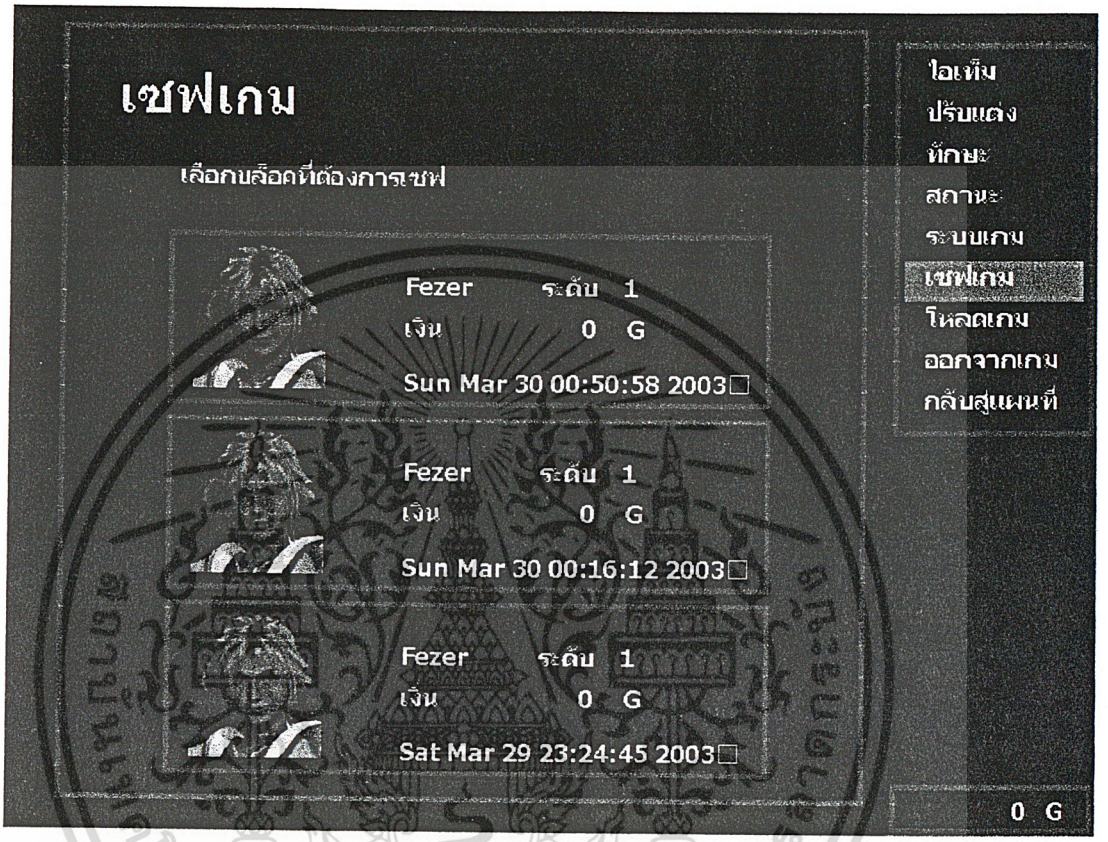
ระดับต่อไป แสดงถึงระดับค่าประสบการณ์ที่ตัวละครจะยกระดับไประดับต่อไป

- เมนูระบบเกม จะใช้ในการปรับแต่งการควบคุมและการซูมภาพของเกม

รูปที่ 4.10 รูปแสดงเมนูการปรับแต่งระบบเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมนู Save เกม ในส่วนของเมนูนี้จะเป็นการบันทึกการเล่นของผู้เล่นเพื่อทำการโหลดเกมมาเล่นในตำแหน่งที่ผู้เล่นเคยเล่นไว้โดยไม่ต้องเริ่มเกมใหม่แต่แรก



รูปที่ 4.11 รูปแสดงการเซฟเกม

ในส่วนของการเซฟเกมนี้ ในเกมจะมีสล็อตให้สำหรับทำการเซฟทั้งหมด 3 สล็อตด้วยกัน

- เมนูโหลดเกม ในส่วนนี้จะเป็นการโหลดตำแหน่งของเกมที่เคยเล่นไว้เมื่อทำการเซฟมาเริ่มเล่น
- เมนูออกจากเกม ในส่วนนี้จะเป็นเมนูที่ใช้เลือกเมื่อผู้เล่นต้องการออกจากเกม
- เมนูกลับสู่แผนที่ ในส่วนของเมนูนี้จะเป็นการออกจากเมนูหลักเพื่อกลับไปยังแผนที่เพื่อเริ่มเล่นเกมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ร้านขายของ

ร้านขายของนี้จะเป็นที่ๆ ผู้เล่นสามารถทำการแลกเปลี่ยนได้ โดยการนำสิ่งของที่หาได้จากการต่อสู้มาแลกเป็นเงิน หรือนำเงินที่มีอยู่มาซื้ออาวุธหรือไอเท็มต่างๆ เพื่อนำมาทำการปรับแต่งรถถังของตน

รูปที่ 4.12 รูปแสดงเมนูซื้อขายแลกเปลี่ยน

4.5 ฉากแผนที่

ฉากแผนที่เป็นฉากที่แสดงถึงแผนที่ของประเทศต่างๆ ทั้งหมดภายในโลก โดยจะมีการเชื่อมโยงระหว่างเมืองแต่ละเมืองเข้าด้วยกัน ซึ่งผู้เล่นสามารถเดินทางผ่านจากเมืองหนึ่งไปยังอีกเมืองหนึ่ง โดยผ่านเส้นทางที่กำหนดไว้

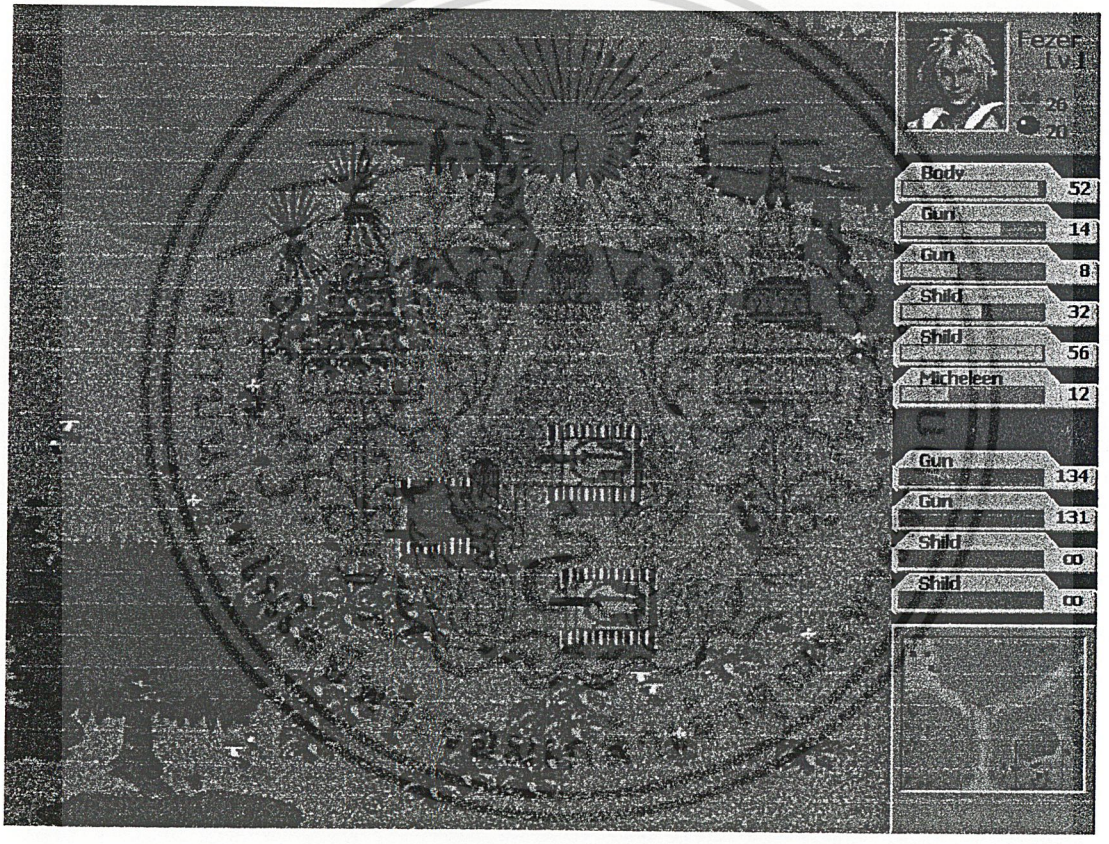
รูปที่ 4.13 รูปแสดงแผนที่โลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.13 จุดสีแดงจะเป็นจุดของฉากต่างๆ ที่ผู้เล่นสามารถเข้าไปได้ ส่วนจุดสีน้ำเงินจะเป็นจุดที่แสดงตำแหน่งของเมือง ซึ่งที่จุดนี้ผู้เล่นจะสามารถเข้าไปในเมืองเพื่อทำการซื้อขายแลกเปลี่ยนหรือว่าหาข้อมูลเพิ่มเติมจากคนในเมือง

4.6 ฉากในเกม

ฉากในเกมนี้จะเป็นฉากที่ผู้เล่นจะต้องดำเนินเกมไปตามเงื่อนไขของแต่ละฉาก โดยในฉากนี้ผู้เล่นจะต้องต่อสู้กับศัตรูที่เป็นรถถังและศัตรูที่เป็นมอนสเตอร์เพื่อให้บรรลุเงื่อนไขของแต่ละฉาก



รูปที่ 4.14 รูปแสดงตัวอย่างฉากของเกม

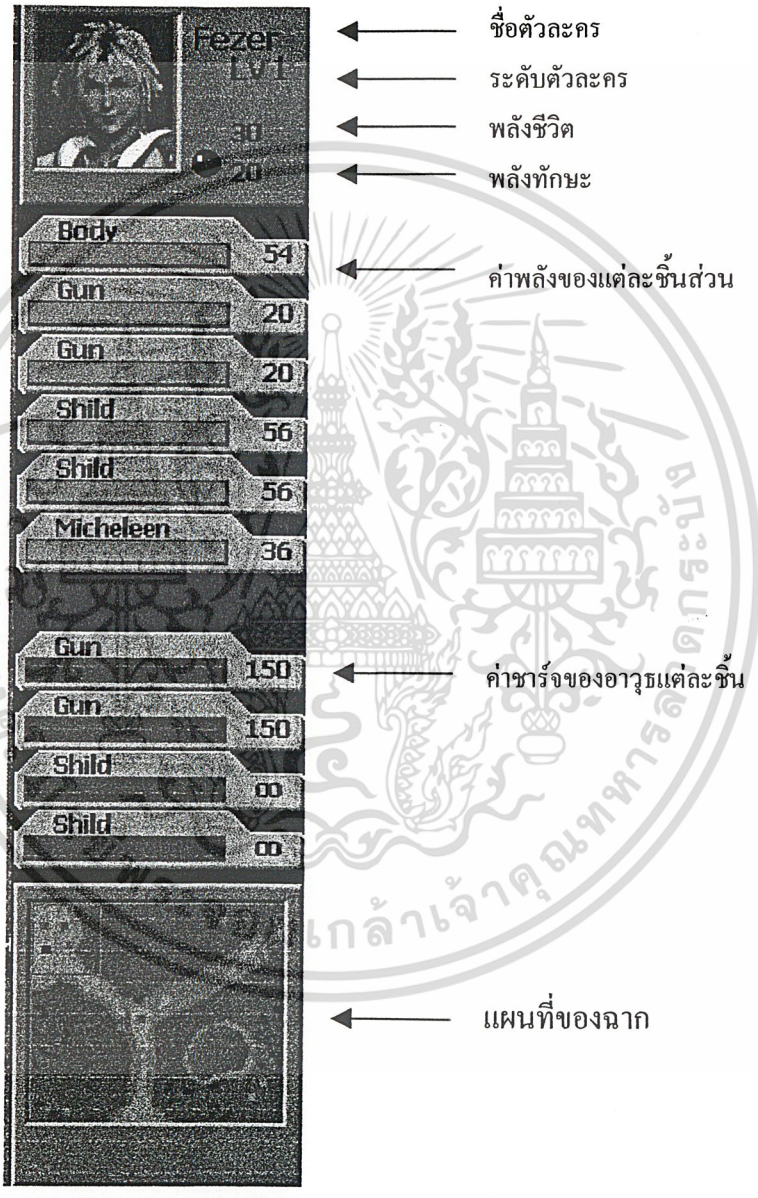
ในส่วนของฉากของเกมนี้จะประกอบด้วยส่วน 2 ส่วน คือ ส่วนที่เป็นฉากแสดงการดำเนินไปของเกม และส่วนที่แสดงถึงค่าพารามิเตอร์ต่างๆ ของรถถัง

ส่วนที่แสดงการดำเนินของฉาก จะเป็นส่วนที่แสดงภาพวิว ภาพวัตถุ ภาพรถถัง และภาพเหตุการณ์ต่างๆ ซึ่งในส่วนนี้จะเป็นส่วนที่ผู้เล่นสามารถมองเห็นการดำเนินไปของเกม

ส่วนที่ 2 ส่วนของค่าพารามิเตอร์ ในส่วนนี้จะแบ่งส่วนย่อยเป็น 3 ส่วน ส่วนบนสุดจะเป็นส่วนที่แสดงถึงค่าพารามิเตอร์ของคนขับรถถัง ส่วนกลางจะเป็นส่วนที่แสดงถึงสภาพของชิ้นส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

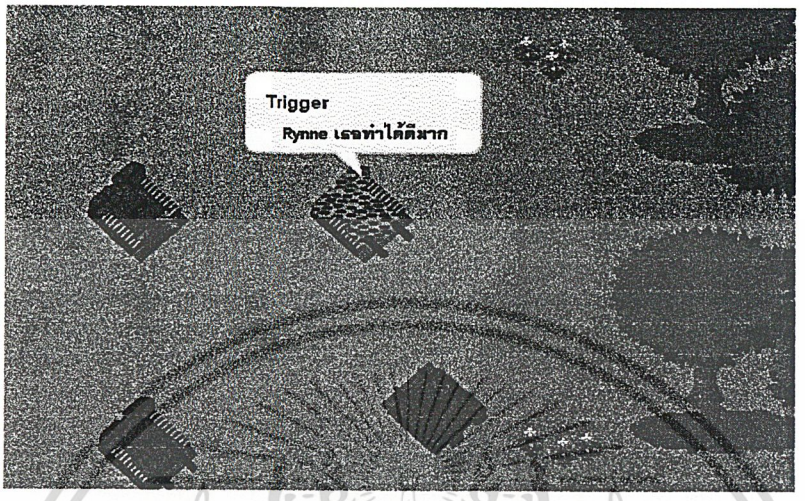
ต่างๆ ของรถถัง โดยในส่วนนี้จะมีลักษณะเป็นแท่งสี่เหลี่ยมสีเขียว และส่วนสุดท้ายจะเป็นส่วนที่แสดงถึงเวลาในการชาร์จอาวุธของรถถัง โดยผู้เล่นจะสามารถใช้อาวุธนั้นๆ ได้ก็ต่อเมื่อค่าที่ชาร์จชาร์จจนเต็ม ส่วนพิเศษเป็นส่วนของแผนที่ย่อย ส่วนนี้ปกติจะไม่มี จะมีก็ต่อเมื่อผู้เล่นติดตั้งทักษะที่ชื่อว่า MAP ซึ่งทักษะชนิดนี้จะทำให้ผู้เล่นสามารถมองเห็นลักษณะทั้งหมดของฉาก



รูปที่ 4.15 รูปแสดงการอธิบายค่าพารามิเตอร์ของฉาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในฉากการดำเนินเกมนี้ นอกจากผู้เล่นจะต้องทำภารกิจต่างๆ ให้สำเร็จแล้ว ในแต่ละฉากยังจะมีเหตุการณ์ต่างๆ แตกต่างกันไป ซึ่งรวมไปถึงบทสนทนาต่างๆ ในเกม



รูปที่ 4.16 รูปแสดงการสนทนาในเกม

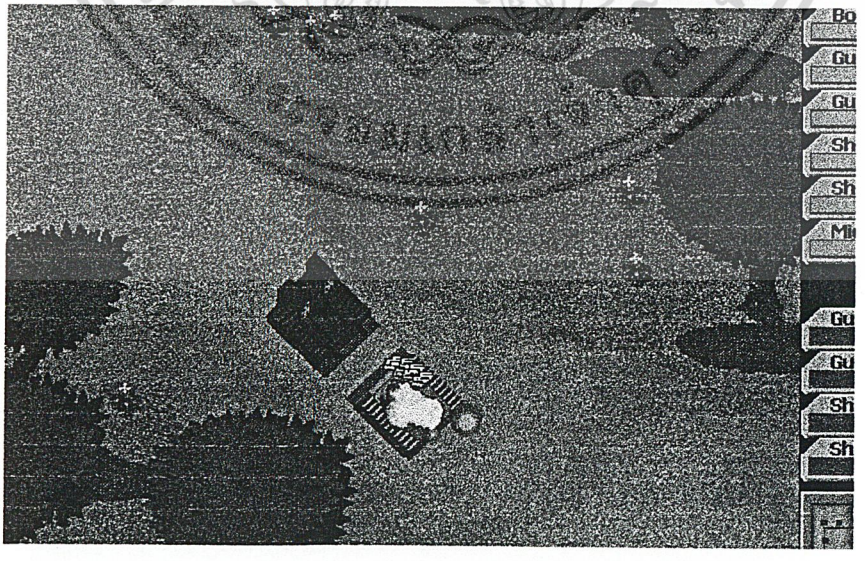
ในการต่อสู้ในเกมนั้น เมื่อเกิดการโจมตีขึ้นจะมีค่าตัวเลขแสดงถึงความเสียหายที่เกิดขึ้นกับตัวละครตัวนั้น โดยตัวเลขต่างๆ มีความหมายดังนี้

ตัวเลขสีขาว แสดงถึง ค่าความเสียหายที่ตัวรถถังหรือมอนสเตอร์นั้นได้รับ

ตัวเลขสีแดง แสดงถึง ค่าความเสียหายที่เกิดขึ้นกับตัวละครที่เป็นคนขับรถถัง

ตัวเลขสีเหลือง แสดงถึง ค่าความเสียหายที่เกิดจากการโจมตีที่เกิด critical

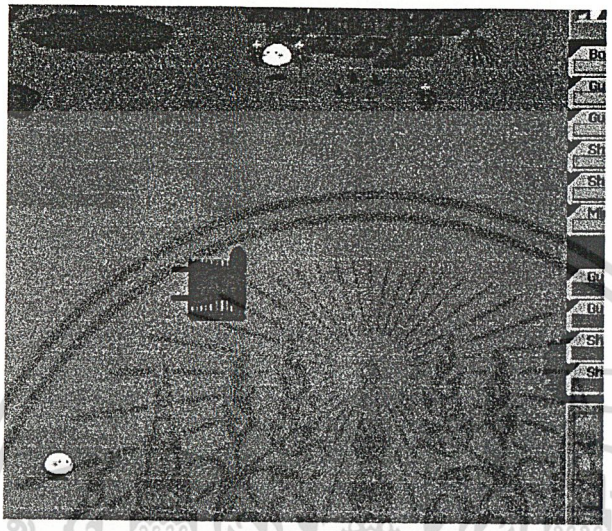
ตัวเลขสีเขียว แสดงถึง ค่าตัวเลขที่เกิดจากการรักษาหรือซ่อมแซมของตัวละครนั้นๆ



รูปที่ 4.17 รูปแสดงค่าตัวเลขที่แสดงถึงความเสียหายของรถถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอนสเตอร์ภายในฉากจะมีอยู่ในฉากบางฉากเท่านั้น ซึ่งมอนสเตอร์แต่ละตัวจะมีลักษณะนิสัยและรูปแบบที่แตกต่างกันออกไป มอนสเตอร์บางตัวจะไม่ทำร้ายผู้เล่นหากผู้เล่นไม่ทำการโจมตีมอนสเตอร์ตัวนั้น มอนสเตอร์บางตัวจะมีนิสัยดุร้าย เมื่อมองเห็นผู้เล่นก็จะเข้าโจมตี



รูปที่ 4.18 รูปแสดงมอนสเตอร์ในฉาก

ในฉากต่อสู้นี้ผู้เล่นนอกจากจะใช้การต่อสู้แบบทั่วไปแล้ว ยังสามารถใช้ทักษะต่างๆ ที่ผู้เล่นได้ทำการติดตั้งมาจากส่วนเมนูหลักนำมาใช้ได้ โดยลักษณะทักษะที่นำมาใช้จะแบ่งตามชนิดของทักษะแต่ละชนิดไป

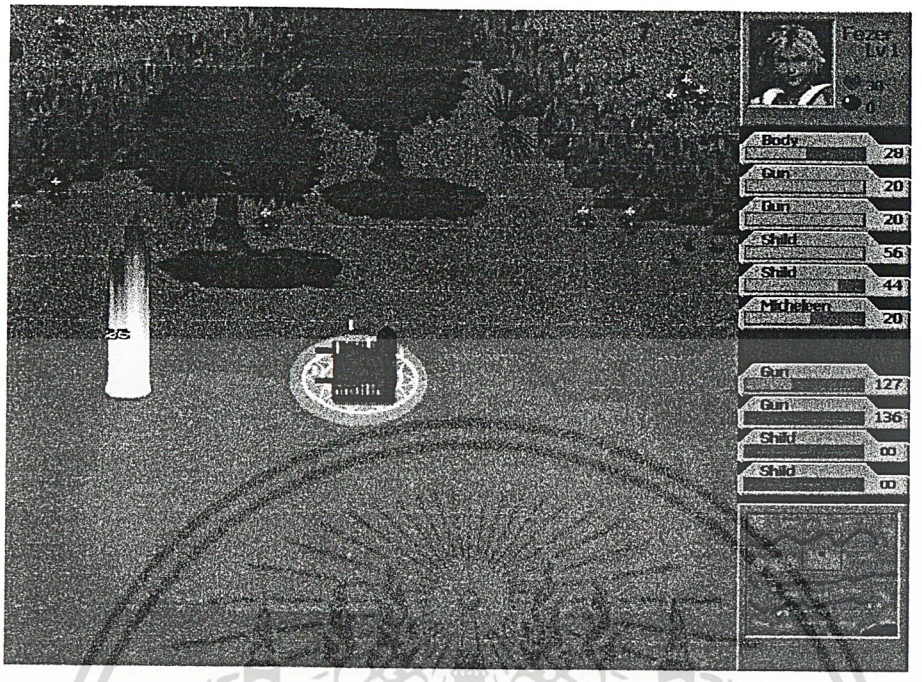
ทักษะประเภทเสริมการต่อสู้ (Fighting Ability) จะสามารถใช้ได้โดยการกดปุ่มควบคุมแล้วจะมีเมนู pop up เพื่อให้เลือกชนิดของทักษะที่ใช้ โดยการใช้ทักษะชนิดนี้ผู้เล่นจะต้องเสียค่าพลังทักษะ

ทักษะประเภทควบคุม (Command Ability) ทักษะชนิดนี้จะมีการใช้งาน 2 แบบด้วยกัน คือ แบบแรก เมื่อผู้เล่นติดตั้งจะสามารถทำให้ผู้เล่นใช้ไอเท็มได้ตามที่ทักษะนั้นระบุไว้ แบบที่สอง เมื่อผู้เล่นนำทักษะมาติดตั้งแล้ว ในการนำไปใช้ผู้เล่นจะต้องกดลำดับ Command ให้ถูกต้องตามประเภทของทักษะนั้นๆ

ทักษะประเภทเสริมสถานะ (Status Ability) ทักษะชนิดนี้จะไม่มีการนำไปใช้โดยตรงในฉากต่อสู้ แต่ทักษะชนิดนี้เมื่อผู้เล่นนำมาติดตั้งจะทำให้ค่าสถานะต่างๆ ของตัวละครเพิ่มขึ้นตามชนิดของทักษะนั้นๆ

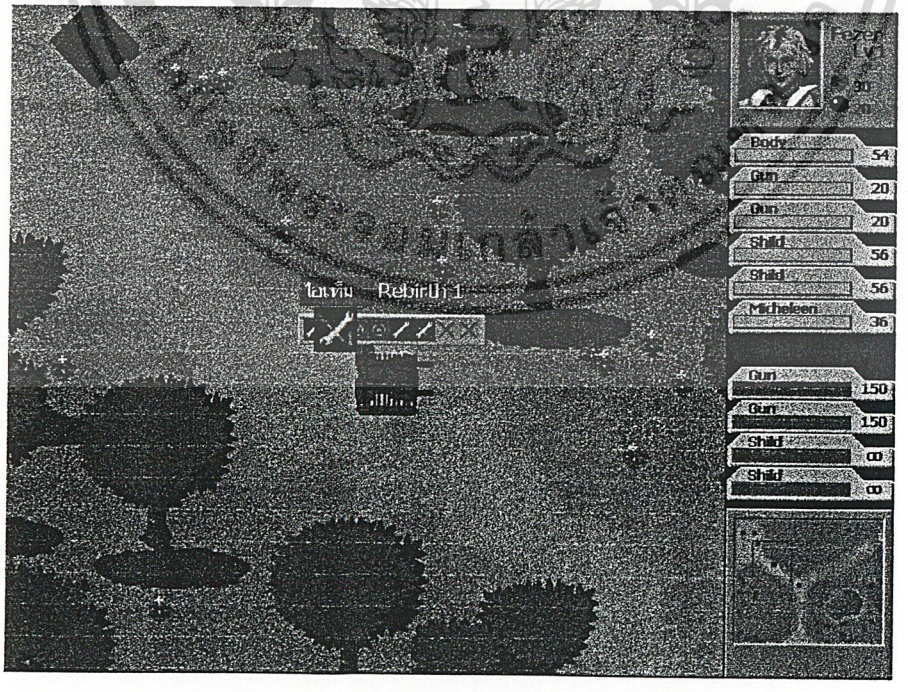
ทักษะประเภทสนับสนุน (Support Ability) ทักษะชนิดนี้จะใช้ในการเสริมความสามารถของทักษะชนิดอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 รูปแสดงตัวอย่างการใช้ทักษะเสริมในการต่อสู้

นอกจากการใช้ทักษะเสริมในการต่อสู้แล้ว ผู้เล่นยังสามารถใช้ไอเท็มเพื่อเสริมในการต่อสู้ได้อีกด้วย



รูปที่ 4.20 รูปแสดงการเลือกใช้ไอเท็มของผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 โปรแกรม BCX-MapEngine

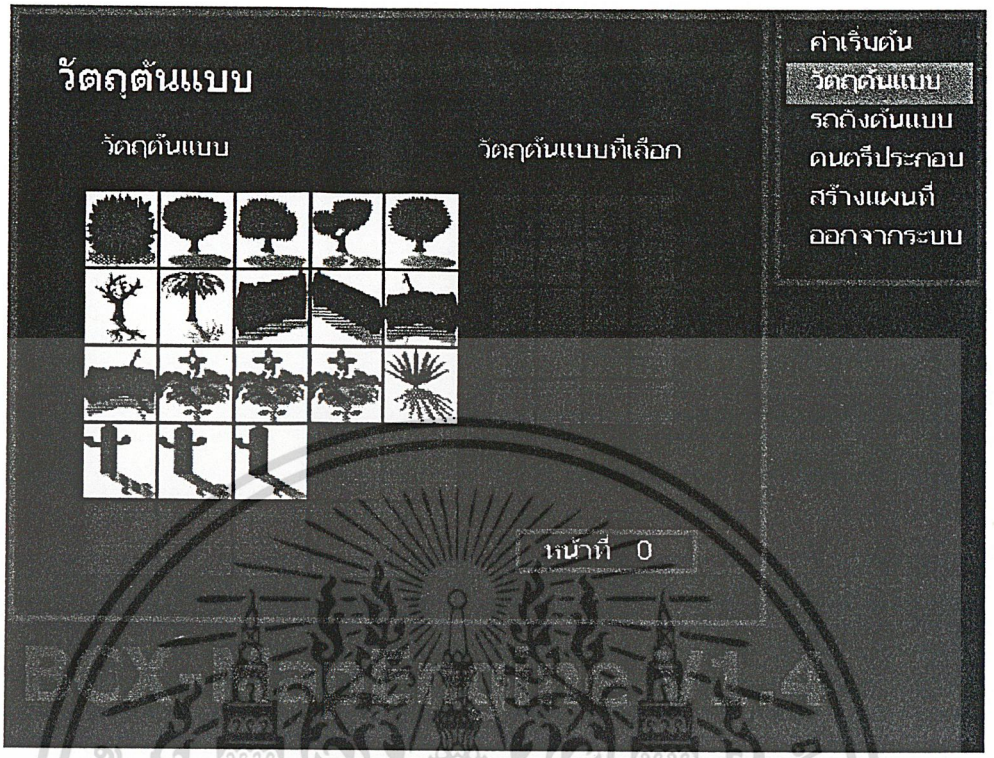
โปรแกรม BCX-MapEngine ที่ใช้ในการสร้างฉาก ปัจจุบันจะเป็นเวอร์ชัน 1.4 ซึ่งทางทีมงานได้พัฒนาให้มีความสามารถมากขึ้น

การใช้งานจะเริ่มโดยการกำหนดค่าเริ่มต้นของฉากแต่ละฉากซึ่งจะประกอบด้วย ขนาดของฉาก รูปแบบของพื้นผิวหลักของฉาก โดยขนาดของฉากจะมีให้เลือก 3 ขนาด คือ 1600x1200 พิกเซล, 2400x1800 พิกเซล และ 3200x 2400 พิกเซล พื้นผิวหลักของฉากจะมีให้เลือกทั้งหมด 5 แบบด้วยกัน คือ ทุ่งหญ้า, ทะเลทราย, โคลน, น้ำแข็ง และน้ำตื้น ในส่วนของเมนูค่าเริ่มต้นนี้ จะถูกพัฒนาขึ้นมาใช้ตั้งแต่เวอร์ชัน1

รูปที่ 4.21 รูปแสดงเมนูกำหนดค่าเริ่มต้นของฉาก

ส่วนต่อไปจะเป็นส่วนที่ผู้ออกแบบจะทำการเลือกรูปแบบของวัตถุต้นแบบที่จะโหลดนำมาใช้ในฉาก ในส่วนของการโหลดวัตถุต้นแบบนำมาใช้ในการสร้างฉากนี้ ผู้ออกแบบสามารถเลือกวัตถุต้นแบบได้สูงสุด 20 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 รูปแสดงเมนูเลือกวัตถุต้นแบบ

ในส่วนของการเลือกวัตถุต้นแบบนี้ ช่องสี่เหลี่ยมทางด้านซ้ายจะแสดงถึงรูปวัตถุต้นแบบทั้งหมดที่มีในเกม ส่วนช่องสี่เหลี่ยมทางด้านขวามือ จะแสดงถึงรูปของวัตถุต้นแบบที่ผู้ออกแบบจะนำไปใช้ในการสร้างฉาก

เมนูรถถังต้นแบบ ส่วนนี้จะใช้ในการเลือกรถถังต้นแบบ แต่สำหรับ BCX-MapEngine เวอร์ชัน 1.3 ขึ้นไป จะไม่มีการนำมาใช้งาน เพราะในส่วนของเมนูนี้รถถังที่โหลดจะมีรูปแบบที่ตายตัว แต่สำหรับในส่วนของตัวเกมนั้นต้องการที่จะให้มีการเปลี่ยนรูปแบบของรถถังไปตามเนื้อเรื่องที่เกิดขึ้น เพราะฉะนั้นเมนูส่วนนี้จึงไม่ได้มีการนำไปใช้งาน สำหรับเมนูส่วนนี้ ตั้งแต่เวอร์ชัน 1.3 ขึ้นไป จะถูกปิดไม่ให้ใช้งานได้

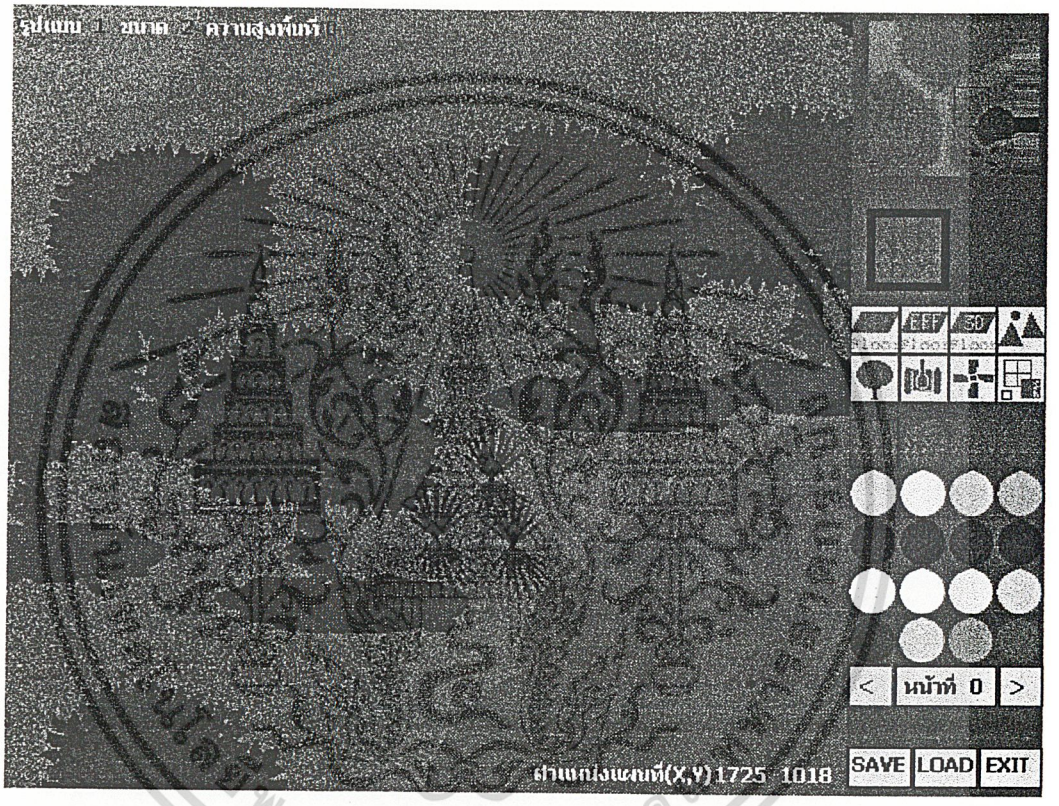
เมนูดนตรีประกอบ ส่วนนี้จะเป็นการเลือกดนตรีประกอบที่จะนำมาใช้ประกอบฉาก แต่สำหรับเวอร์ชัน 1.3 ขึ้นไป จะไม่มีการนำไปใช้ เพราะตัวเกมเวอร์ชันใหม่จะไม่ใช้เสียงประกอบที่ตายตัว แต่จะใช้เสียงประกอบที่เปลี่ยนแปลงไปตามเหตุการณ์ต่างๆ สำหรับเมนูในส่วนนี้จะยังใช้ในการฟังเสียงดนตรีประกอบได้เพื่อทำการเลือกเสียงดนตรีประกอบ แต่เสียงดนตรีประกอบที่เลือกจะไม่มีผลกับตัวเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูสร้างแผนที่ ในส่วนของเมนูนี้เมื่อผู้ออกแบบได้ทำการกำหนดค่าเริ่มต้นต่างๆ ของฉากเรียบร้อยแล้ว ก็จะเลือกเพื่อเข้าสู่การออกแบบฉาก

เมนูออกจากโปรแกรม สำหรับเมนูนี้จะใช้ในการออกจากโปรแกรม

สำหรับในส่วนของการออกแบบฉากนี้จะไม่กล่าวถึงการออกแบบที่ใช้เวอร์ชันที่เก่ากว่า 1.4 แต่จะอธิบายการทำงานของเวอร์ชัน 1.4 เท่านั้น



รูปที่ 4.23 รูปแสดงตัวอย่างการใช้งานโปรแกรม BCX-MapEngine V1.4

ในการใช้งานผู้ออกแบบจะสามารถนำพื้นผิวและวัตถุต่างๆ จากเมนูทางด้านขวามือมาทำการวางลงในฉากเพื่อให้ได้ฉากที่สมบูรณ์

ในส่วนของเมนูออกแบบจะประกอบด้วยส่วนต่างๆ ดังนี้

ส่วนของแผนที่ จะเป็นส่วนบนสุดของเมนูออกแบบ ซึ่งจะแสดงถึงมุมมองของฉากทั้งหมด ในส่วนนี้จะทำให้ผู้ออกแบบสามารถมองเห็นมุมมองที่กว้างมากขึ้น และนอกจากนี้ผู้ออกแบบยังสามารถทำการเลื่อนหน้าจอไปยังตำแหน่งต่างๆ ของฉากโดยการใช้เมาส์ดับเบิลคลิกที่ตำแหน่งต่างๆ ของแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแสดงรูปแบบ ในส่วนนี้จะเป็นส่วนที่แสดงถึงพื้นผิวหรือวัตถุที่ผู้ออกแบบได้ทำการเลือกจากส่วนของเมนูออกแบบ

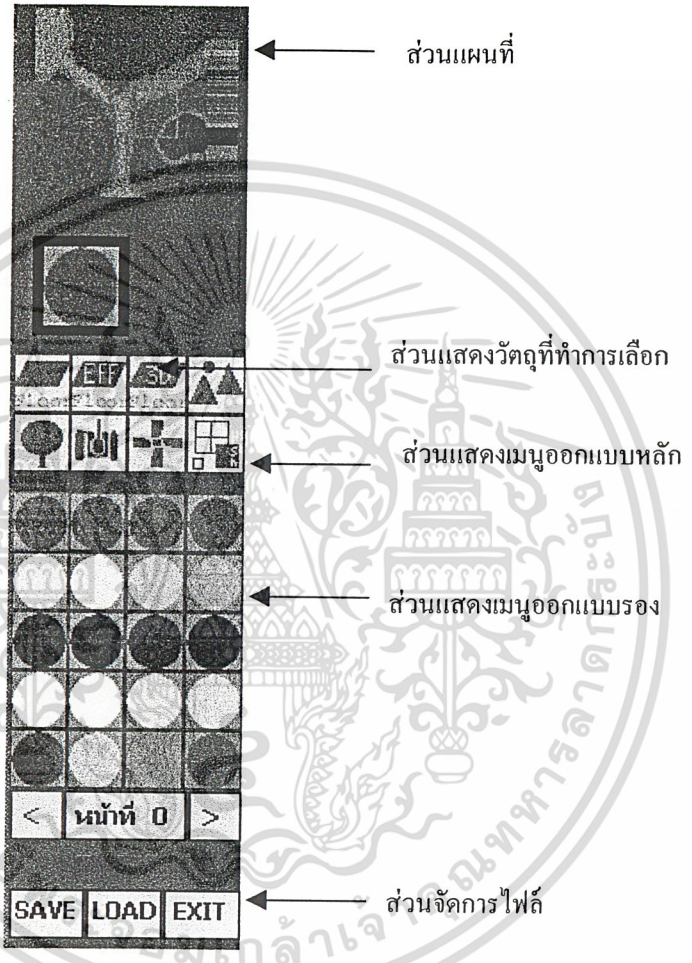
ส่วนเมนูออกแบบหลัก ในส่วนนี้จะเป็นเมนูหลักของการออกแบบ ซึ่งในส่วนของเมนูหลักนี้จะมีทั้งหมด 8 เมนูด้วยกัน

- เมนูพื้นผิว ในส่วนนี้จะเป็นเมนูที่แสดงถึงพื้นผิวต่างๆ ที่ผู้ออกแบบจะนำไปใช้ในการสร้างพื้นดิน
- เมนูส่วนประกอบพื้นผิว ในส่วนนี้จะเป็นเมนูที่แสดงถึงภาพประกอบต่างๆ ที่จะนำมาใช้ในการแปะลงในพื้นผิว เพื่อเพิ่มความสวยงามให้แก่พื้นผิวมากยิ่งขึ้น
- เมนูพื้น 3 มิติ ในส่วนนี้จะเป็นส่วนที่แสดงถึงรูปแบบพื้นที่ทำการสร้างให้มีมุมมองเป็น 3 มิติ เพื่อนำมาใช้ในการสร้างฉากให้มีรูปแบบสูงและต่ำ
- เมนูหน้าผา ในส่วนนี้จะใช้ในการสร้างส่วนที่เป็นหน้าผาภายในฉาก
- เมนูวัตถุ ในส่วนนี้จะเป็นส่วนที่แสดงถึงวัตถุต้นแบบทั้งหมดที่ผู้ออกแบบได้ทำการเลือกจากเมนูกำหนดค่าเริ่มต้น
- เมนูรถถัง ในส่วนนี้จะเป็นส่วนที่แสดงถึงรูปแบบของรถถังชนิดต่างๆ
- เมนูเอฟเฟ็คท์ ในส่วนนี้จะแสดงถึงเอฟเฟ็คท์ที่จะนำมาใช้ในฉาก
- เมนูอื่นๆ ในส่วนนี้จะเป็นการใช้คำสั่งเพื่อปรับเปลี่ยนรูปแบบของฉากและเครื่องมือต่างๆ ได้แก่
  - ส่วนกำหนดความสูง ในส่วนนี้จะใช้กำหนดความสูงของพื้นผิว
  - ส่วนกำหนดสถานะ ในส่วนนี้จะใช้ในการกำหนดสถานะของพื้นผิว
  - ส่วนคัดลอก ในส่วนนี้จะเป็นส่วนที่ใช้ในการคัดลอกพื้นผิวจากบริเวณหนึ่ง เพื่อนำไปวางที่บริเวณหนึ่ง โดยรูปแบบการคัดลอกจะมีลักษณะเป็นพื้นที่สี่เหลี่ยม
  - ส่วนคัดลอก 2 ในส่วนนี้จะเป็นส่วนที่ใช้ในการคัดลอกพื้นผิวจากบริเวณหนึ่ง เพื่อนำไปวางที่บริเวณหนึ่ง โดยรูปแบบการคัดลอกจะมีลักษณะเป็นพื้นที่วงกลม
  - ส่วนเคลื่อนย้าย ในส่วนนี้จะใช้ในการเคลื่อนย้ายตำแหน่งของรถถังที่ได้มีการวางไปยังแผนที่แล้ว
  - ส่วนกำหนดตำแหน่งมอนสเตอร์ ในส่วนนี้จะใช้กำหนดตำแหน่งของมอนสเตอร์ภายในฉาก

ส่วนเมนูออกแบบรอง ในส่วนนี้จะแสดงถึงเมนูย่อยต่างๆ จากส่วนของเมนูออกแบบหลัก ซึ่งจะมีรูปแบบแตกต่างกันไปตามส่วนของเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนจัดการไฟล์ ในส่วนนี้จะเป็นส่วนจัดการกับไฟล์ .MAP ซึ่งประกอบไปด้วยคำสั่งโหลดไฟล์ , เซฟไฟล์ และส่วนออกจากโปรแกรม



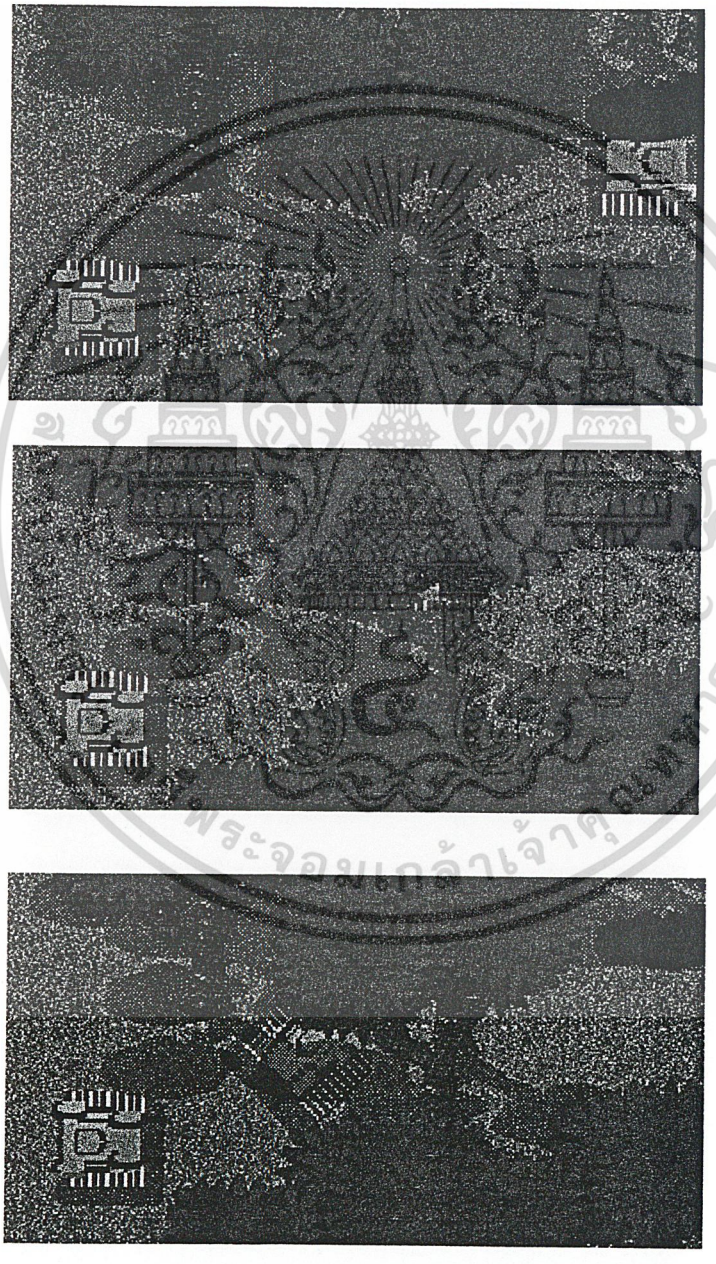
4.24 รูปแสดงเมนูในส่วนออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การค้นหาเส้นทาง

เมื่อนำ A* Algorithm นำมาใช้ในการหาเส้นทางแล้วผลที่ได้คือรถถังสามารถค้นหาเส้นทางเพื่อไปถึงจุดหมายที่ต้องการได้

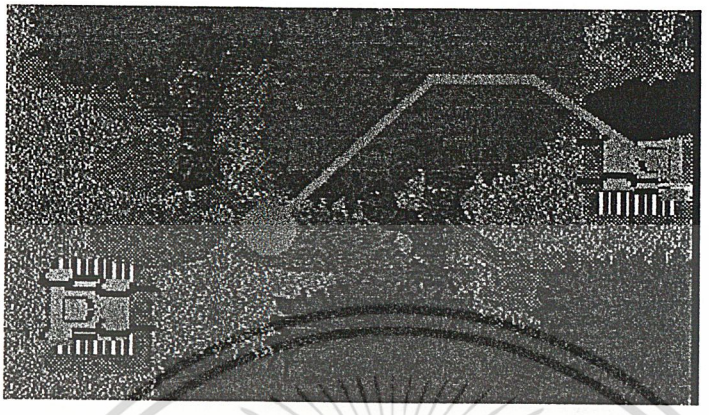
กรณีการเข้าโจมตีตัวผู้เล่นโดยมีสิ่งกีดขวางระหว่างทาง



รูปที่ 4.25 แสดง การเดินเข้าโจมตีศัตรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่าเส้นทางการเดินของรถถังเพื่อเข้ามาโจมตีศัตรูจะเป็นดังต่อไปนี้



รูปที่ 4.26 แสดง การเส้นทางในการเดินหาเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

จากตารางการดำเนินงานที่วางไว้ การดำเนินงานเป็นไปตามแผนที่วางไว้ได้ยังไม่ดีมากนัก เนื่องจากปัญหาในการทำงาน หรือบางทีก็มีแนวคิดใหม่ๆ ที่เพิ่มเข้าไปในตัวเกม จึงทำให้งานบางชิ้นใช้เวลาในการทำงานมากเกินไป เช่น ในการวาดรูปภาพเพื่อนำมาประกอบฉากภายในเกม แต่งานบางชิ้นก็สามารถทำได้เร็วกว่าที่กำหนด ถึงแม้ว่าเวลาโดยรวมอาจจะคลาดเคลื่อนไปบ้างเล็กน้อย แต่ก็ถือว่าการดำเนินงานเป็นไปได้ด้วยดี

จากตัวเกมที่สร้างสำเร็จนี้ สามารถสร้างได้ครบตามวัตถุประสงค์ที่ตั้งไว้แต่แรก นอกจากนี้แล้วยังเพิ่มส่วนประกอบอื่นๆ ซึ่งทำให้เกมมีความสมบูรณ์มากยิ่งขึ้นอีกด้วย โดยตัวเกมที่สร้างนี้สามารถนำไปติดตั้ง บนระบบปฏิบัติการและ ประสิทธิภาพดังนี้

ระบบปฏิบัติการ	Windows 98 / ME / XP / 2000 / NT
ประสิทธิภาพเครื่องที่แนะนำ	
หน่วยประมวลผล	Duron 750 MHz หรือสูงกว่า
หน่วยความจำ	64 Mb หรือสูงกว่า
การ์ดแสดงผล	Standard VGA 32 Mb หรือสูงกว่า
การ์ดเสียง	สนับสนุน DirectSound
เนื้อที่ฮาร์ดดิส	50 MB

5.2 ปัญหาในการดำเนินงาน

- เนื่องจาก Direct X ไม่สามารถ Run แบบ Step by step ได้ จึงต้อง Run โปรแกรมทั้งหมดเพื่อดูผล ซึ่งถ้าหากการทำงานผิดพลาด เครื่องคอมพิวเตอร์อาจจะหยุดทำงานไปเลย ซึ่งส่วนนี้นับว่าทำให้เสียเวลาในการทำงานมาก
- การสร้างกราฟฟิกต้องใช้เวลา และ ทักษะอย่างมาก การสร้างเกมนั้นส่วนของกราฟฟิกนับว่าเป็นอีกส่วนที่สำคัญมาก เพราะฉะนั้นการทำงานส่วนนี้ต้องทำงานให้ได้ในระดับที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การหาเสียงประกอบที่เหมาะกับเกมนั้นค่อนข้างยาก เนื่องจากในส่วนของเสียงประกอบนี้ไม่สามารถที่จะสร้างเองได้
- ในส่วนของการประมวลผลการทำงานในส่วนของเกมนับว่าต้องใช้การประมวลผลที่ค่อนข้างสูง จนบางครั้งหากมีการประมวลผลมากเกินไป จะทำให้โปรแกรมเกิดการล่าช้า ซึ่งทำให้ต้องเสียเวลาในการแก้ไขเพื่อให้มีการประมวลผลที่เร็วขึ้น ในบางกรณีที่แก้ไขไม่ได้จึงต้องตัดการประมวลผลที่หนักเกินไปออกไป
- ปัญหาหน่วยความจำของการ์ดแสดงผลที่จำกัด ปัญหาส่วนนี้นับว่ามีผลเป็นอย่างมากในการสร้างภาพกราฟฟิก สำหรับปัญหาส่วนนี้ทีมงานจะต้องแก้ไขการจัดสรรหน่วยความจำหลายครั้งกว่าจะได้การจัดสรรที่ลงตัว ซึ่งในการแก้ไขแต่ละครั้งจะต้องแก้ไขส่วนที่เป็นโครงสร้างหลักของเกม จึงทำให้เสียเวลาค่อนข้างมาก สำหรับการแก้ไขปัญหานี้ ปัจจุบันอยู่ในระดับที่ใช้งานได้ แต่ยังไม่ดีที่สุด เพราะการจัดสรรหน่วยความจำของการ์ดแสดงผลนั้น สามารถจัดสรรให้มีปริมาณเพิ่มขึ้นได้ โดยการใช้งานร่วมกับหน่วยความจำหลัก ซึ่งในส่วนนี้จะต้องมีการเขียนโปรแกรมเพื่อจัดการหน่วยความจำ ซึ่งค่อนข้างซับซ้อนและใช้เวลามาก เพื่อจะให้ได้การจัดสรรหน่วยความจำที่ลงตัว
- ในส่วนการออกแบบควบคุมยังไม่สามารถกำหนดขนาดของสิ่งกีดขวางให้เหมาะสมกับขนาดของช่องที่ได้ทำการเก็บค่าสิ่งกีดขวาง ทำให้ในบางครั้งทำให้เส้นทางเดินที่หาได้ไม่ดีนัก
- เนื่องจากการสร้างเกมนั้นต้องใช้เวลาในการทำมากในด้านกราฟฟิก (Graphic) และด้านการออกแบบระบบตัวเกม จึงทำให้ต้องใช้เวลามาก และถ้าจะทำให้ดีมากขึ้นนั้นก็จะต้องให้เวลามากขึ้นไปด้วย

5.3 ข้อจำกัดของโครงงาน

- ตัวเกมไม่สามารถใช้ได้กับคอมพิวเตอร์ที่มีหน่วยประมวลผลต่ำ หรือ หน่วยความจำน้อย
- มีปัญหาเรื่องการแสดงรูปแบบตัวอักษรภาษาไทยบนเครื่องบางเครื่องที่มีรูปแบบภาษาไม่ตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางในการพัฒนา

การพัฒนาด้านกราฟฟิกจะพัฒนาเป็นเกม 2 มิติผสมผสาน 3 มิติได้โดยจะใช้ระบบของเกมแบบเดิมที่ได้ออกแบบไว้หรือจะทำการออกแบบระบบใหม่ขึ้นมาเพื่อให้มีความน่าสนใจ หรือพัฒนาเครื่องมือในการสร้างเกมเพื่อให้สะดวกในการสร้างมากขึ้นเพื่อความรวดเร็ว

การพัฒนาด้านระบบควบคุม ซึ่งเกมที่ได้สร้างขึ้นนี้ยังไม่ได้มีระบบการควบคุมที่สามารถเรียนรู้ได้ว่าควรจะปฏิบัติอย่างไรในสถานะการณ์เช่นนั้น โดยสามารถเรียนรู้และเปลี่ยนแปลงพฤติกรรมให้เหมาะสมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] มานพ พรเพ็นรวิชานนท์ , “ก้าวสู่ Game Developer มือโปร กับ Direct X”, บริษัท วิตตี้ กรุ๊ป พิมพ์ครั้งที่ 1 2544

[2] นิรุธ อำนวยศิลป์ ,“การเขียนเกมด้วย Visual C++ และ Direct X” พิมพ์ครั้งที่ 1 2545

[3] Andre Lamothe “Trick of the Windows Game Programming Gurus”, E-book 1999

[4] พีรภัทร์ สว่างเพียร, “เทคนิคการเขียนโปรแกรมและเกมด้วย Visual C++”,บริษัท ซีเอ็ดยูเคชั่น พิมพ์ครั้งที่ 1 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

A* Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

A* Algorithm

```
/*-| STRUCT |-------------------------------------------------------------------*\
| Name: NODE
| Desc: the Node,defines one tile for the AStar pathfinding algorithm
\*-------------------------------------------------------------------------------*/
struct NODE
{
        int  x,   // X pos of the tile
             y;   // Y pos of the tile
        long f;   // the cost to move from the start to the node and from the node to the goal(the
                  heuristic)
        int       Parent,// this identifies the parent from which the successors was created(NULL if it is
                  the first node)
                  ID;   // the id which the parent will refer to
};

// ** ASTAR CLASS
*********************************************************************************
/*-| CLASS |-----------------------------------------------------------*\
| Name: AStar
| Desc: the Pathfinding class
\*-------------------------------------------------------------------*/
class AStar
{
        void ConstructPath(NODE n);
        bool IsInOPEN(NODE n);
        bool IsInCLOSED(NODE n);
        void AddToOPEN(NODE n);
        void AddToCLOSED(NODE n);
        void RemoveFromOPEN(NODE n);
        void GenerateSuccs(NODE n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        void GenerateSucc(int x,int y,int Parent);
        NODE GetBestNode();
        NODE GetNodeFromCLOSED(int ID);
        int nOpen,nClosed,cOpen,cClosed,cID;
        NODE nStart,nGoal;
        NODE* OPEN;
        NODE* CLOSED;
public:
        int nPath;
        NODE* PATH;
        bool FindPath(POINT StartPos,POINT EndPos);
};Path;
//** MEMBER FUNCTIONS ***************************************************************
//////////////////////////////////////////////////////////////////////////////
// ASTAR ////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////

/* -| ASTAR FUNCTION |-------------------------------------------------------*\
| Name: ConstructPath
| Desc: constructs a path trough the .Parent value of the node that found the goal
\* -| ASTAR FUNCTION |-------------------------------------------------------*/
void AStar::ConstructPath(NODE n)
{
        PATH =(NODE*) malloc(300 *sizeof(NODE));
        nPath=0;
        NODE nTemp=n;
        int a=0;
        NODE *AltPath;
        while (nTemp.ID != -1)
        {
                PATH[nPath++]=nTemp;
                nTemp=GetNodeFromCLOSED(nTemp.Parent);
        }
        PATH[nPath++]=nTemp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AltPath=(NODE*) malloc(nPath*sizeof(NODE));

        for (int i=nPath-1;i>-1;i--)
        {
                AltPath[a++]=PATH[i];
        }

        free(PATH);
        PATH=AltPath;
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: IsInOPEN
| Desc: Verifies if a node is in the OPEN list
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
bool AStar::IsInOPEN(NODE n)
{
        if (OPEN==NULL) return false;

        int Len=(sizeof(*OPEN)*cOpen)/sizeof(NODE);
        for (int i=0;i<Len;i++)
        {
                if (OPEN[i].x == n.x && OPEN[i].y == n.y)
                        break;
        }
        if (i==Len) return false;
        return true;
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: IsInCLOSED
| Desc: Verifies if a node is in the CLOSED list
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
bool AStar::IsInCLOSED(NODE n)
{
        if (CLOSED==NULL) return false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int Len=(sizeof(*CLOSED)*cClosed)/sizeof(NODE);
        for (int i=0;i<Len;i++)
        {
                if (CLOSED[i].x == n.x && CLOSED[i].y == n.y)
                        break;
        }
        if (i==Len) return false;
        return true;
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: AddToOPEN
| Desc: Adds a node to the OPEN list
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
void AStar::AddToOPEN(NODE n)
{
        if (cOpen>nOpen)
        {
                nOpen+=100;
                OPEN=(NODE*) realloc(OPEN,sizeof(NODE)*nOpen);
        }
        OPEN[cOpen++]=n;
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: AddToClosed
| Desc: Adds a node to the CLOSED list
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
void AStar::AddToCLOSED(NODE n)
{
        if (cClosed>nClosed)
        {
                nClosed+=100;
                CLOSED=(NODE*) realloc(CLOSED,sizeof(NODE)*nClosed);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLOSED[cClosed++]=n;
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: RemoveFromOPEN
| Desc: Removes a node from the OPEN list
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
void AStar::RemoveFromOPEN(NODE n)
{
        if (OPEN==NULL) return;

        int Len=(sizeof(*OPEN)*cOpen)/sizeof(NODE);
        for (int i=0;i<Len;i++)
                if (OPEN[i].x == n.x && OPEN[i].y == n.y)
                        break;
        for (int j=i;j<cOpen;j++)
                OPEN[j]=OPEN[j+1];
        cOpen--;
}

/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: GenerateSucc
| Desc: Generates a node and adds it to OPEN
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
void AStar::GenerateSucc(int x,int y,int Parent)
{
        // First we check if the new successor is ot of the Map area or is a
        // forbidden tile.So if it is clear for movement,create the node and
        // if the node is not already created(is not in OPEN) and it is not
        // in CLOSED then create it
        if (x<0) return; if (y<0) return;
        if (x>Map.LenX) return;if (y>Map.LenY) return;
        if (Map.GetTile(x,y) != '>') return;

        NODE n;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        n.x=x; n.y=y; n.ID=cID++; n.Parent=Parent;
        //Assume h function = (abs(nGoal.x - n.x)+1) * (abs(nGoal.y - n.y)+1)
        n.f=(abs(nGoal.x - n.x)+1) * (abs(nGoal.y - n.y)+1);
        if (!IsInOPEN(n) && !IsInCLOSED(n))
                AddToOPEN(n);
}
/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: GenerateSuccs
| Desc: Generate successors for the node
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
void AStar::GenerateSuccs(NODE n)
{
        GenerateSucc(n.x-1,n.y-1,n.ID); // Upper Left
        GenerateSucc(n.x  ,n.y-1,n.ID); // Upper Center
        GenerateSucc(n.x+1,n.y-1,n.ID); // Upper Right
        GenerateSucc(n.x-1,n.y  ,n.ID); // Center Left
        GenerateSucc(n.x+1,n.y  ,n.ID); // Center Right
        GenerateSucc(n.x-1,n.y+1,n.ID); // Lower Left
        GenerateSucc(n.x  ,n.y+1,n.ID); // Lower Center
        GenerateSucc(n.x+1,n.y+1,n.ID); // Lower Right

        RemoveFromOPEN(n);
        AddToCLOSED(n);
}

/* -| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: GetNodeFromCLOSED
| Desc: Retrieves the node from the CLOSED list (for path construction)
\* -| ASTAR FUNCTION |-----------------------------------------------------------*/
NODE AStar::GetNodeFromCLOSED(int ID)
{
        for (int i=0;i<cClosed;i++)
                if (CLOSED[i].ID == ID)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                return CLOSED[i];
        return CLOSED[0]; // to silence the warning

}

/* –| ASTAR FUNCTION |-----------------------------------------------------------*\
| Name: GetBestNode
| Desc: Retrieves the node that is the closest to the goal (defined by the heuristic)
\* –| ASTAR FUNCTION |-----------------------------------------------------------*/
NODE AStar::GetBestNode()
{
        long f=OPEN[0].f;
        int ArrPos=0;
        int Len=(sizeof(*OPEN)*cOpen)/sizeof(NODE);
        NODE tNode;

        for (int i=1;i<Len;i++)
        {
                if (OPEN[i].f<=f)
                {
                        f=OPEN[i].f;
                        ArrPos=i;
                }
        }
        tNode.f         =OPEN[ArrPos].f;
        tNode.x         =OPEN[ArrPos].x;
        tNode.y         =OPEN[ArrPos].y;
        tNode.Parent=OPEN[ArrPos].Parent;
        tNode.ID        =OPEN[ArrPos].ID;
        return tNode;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* -| ASTAR FUNCTION |-------------------------------------------------------------------*\
| Name: FindPath
| Desc: the main function of AStar,this one finds the path,or not,and stores it in the PATH
\* -| ASTAR FUNCTION |-------------------------------------------------------------------*/
bool AStar::FindPath(POINT StartPos,POINT EndPos)
{
        // -- Memory allocation ------------------
        if (nPath>0 && nPath<300) free(PATH);
        OPEN  =(NODE*) malloc(1000*sizeof(NODE));
        CLOSED=(NODE*) malloc(1000*sizeof(NODE));
        nOpen  =1000;
        nClosed=1000;
        cOpen  =1;
        cClosed=0;
        cID=0;
        // - Search initialization --------------
        NODE n;
        n.f=n.Parent=n.ID=-1;n.x=StartPos.x;n.y=StartPos.y;
        nStart=n;
        n.f=n.Parent=n.ID=0;n.x= EndPos.x;n.y= EndPos.y;
        nGoal=n;
        OPEN[0]=nStart;
        // - Pathfinding ----------------------
        while (cOpen) // Do while OPEN is not empty,if OPEN is empty,it means that the path was not
found
        {
                n=GetBestNode();
                if (n.x==nGoal.x && n.y==nGoal.y)
                        break;

                GenerateSuccs(n);
        }
        // - Memory deallocation ----------------
        if (cOpen !=0) ConstructPath(n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        free(OPEN);
        free(CLOSED);

        if (cOpen==0)
                return false;
        else
                return true;
}
```

<u>หมายเหตุ</u> **Map คือ ตัวแปรเก็บค่าสถานะต่างๆ ในแผนที่**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โครงสร้างของไฟล์ .MAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โครงสร้างของไฟล์ .MAP

ไฟล์ .MAP นี้เป็นไฟล์แผนที่ของฉากแต่ละฉากในเกม ซึ่งไฟล์ชนิดนี้จะสร้างจากโปรแกรม BCX-Map Engine ซึ่งเป็นโปรแกรมที่ทีมงานสร้างขึ้นเพื่อใช้สร้างฉากในเกมโดยเฉพาะ โครงสร้างของ ไฟล์ .MAP ประกอบด้วย

- **Version** เป็นข้อมูลชนิด integer จะเก็บค่าเวอร์ชันของ Map Engine
- **Width** เป็นข้อมูลชนิด integer จะเก็บค่าความกว้างของฉาก
- **Hight** เป็นข้อมูลชนิด integer จะเก็บค่าความสูงของฉาก
- **MainFloor** เป็นข้อมูลชนิด integer จะเก็บค่า ID ของพื้นหลักของฉาก
- **FloorStatus** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [width/40][hight/40] ใช้เก็บข้อมูลสถานะของพื้นดิน
- **FloorHight** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [width/40][hight/40] ใช้เก็บข้อมูลความสูงของพื้นดิน
- **Floor3D_Count** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [36][4][8] ใช้เก็บจำนวนของพื้นที่เป็น 3 มิติทั้งหมดในฉากไว้
- **Floor3D_Buffer** เป็นข้อมูลชนิดอาร์เรย์ชนิด POINT ขนาด [36][4][8] ใช้เก็บตำแหน่งของพื้นที่เป็น 3 มิติ
- **FloorCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของการนำ Texture ไปแปะที่พื้นของฉาก
- **FloorType** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่าชนิดของ Texture
- **FloorShape** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่ารูปแบบของ Texture
- **FloorSize** เป็นอาร์เรย์ชนิด integer ที่มีขนาด [FloorCount] ใช้เก็บค่าขนาดของ Texture
- **FloorPoint** เป็นอาร์เรย์ชนิด POINT ที่มีขนาด [FloorCount] ใช้เก็บค่าตำแหน่งของ Texture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **FloorRect** เป็นอาร์เรย์ชนิด RECT ที่มีขนาด [FloorCount] ใช้เก็บค่าขนาดของ Surface ที่เกิดจากการ copy และ paste
- **ObjectContent** เป็นข้อมูลชนิด integer ใช้เก็บค่าจำนวนของรูปจำลองต้นแบบของวัตถุ
- **ModelObjectType** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ที่มีขนาด [ObjectContent] ใช้เก็บค่าชนิดของรูปจำลองต้นแบบของวัตถุ
- **ObjectCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของวัตถุทั้งหมดภายในฉาก
- **ObjectType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด [ObjectCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของวัตถุ
- **ObjectPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [ObjectCount] ใช้เก็บค่าตำแหน่งของวัตถุ
- **TankContent** เป็นข้อมูลชนิด integer ใช้เก็บค่าจำนวนของรูปจำลองต้นแบบของรถถัง
- **TankBody** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนตัวถังรถของรูปจำลองต้นแบบของรถถัง
- **TankWheel** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนล้อรถของรูปจำลองต้นแบบของรถถัง
- **TankWeapon** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent][4] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนอาวุธรถของรูปจำลองต้นแบบของรถถัง
- **TankBackpack** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนเครื่องหลังของรูปจำลองต้นแบบของรถถัง
- **TankComputer** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของส่วนคอมพิวเตอร์ของรูปจำลองต้นแบบของรถถัง
- **TankTexture** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขชนิดของTextureของรูปจำลองต้นแบบของรถถัง
- **TankPirot** เป็นข้อมูลชนิดอาร์เรย์ชนิด integer ขนาด [TankContent] ที่ใช้เก็บค่าหมายเลขคนขับของรูปจำลองต้นแบบของรถถัง
- **TankCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของรถถังทั้งหมดภายในฉาก
- **TankType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด[TankCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของรถถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **TankPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [TankCount] ใช้เก็บค่าตำแหน่งของรถถัง
- **Music** เป็นข้อมูลชนิด integer ที่ใช้เก็บหมายเลขเสียงดนตรีของฉาก
- **PlayerTankPosition** เป็นข้อมูลชนิด POINT ที่ใช้เก็บตำแหน่งเริ่มต้นของรถถังผู้เล่น
- **MonsterCount** เป็นข้อมูลชนิด integer ที่ใช้เก็บจำนวนของมอนสเตอร์ทั้งหมดภายในฉาก
- **MonsterType** เป็นข้อมูลชนิดอาร์เรย์ ชนิด integer ที่มีขนาด [MonsterCount] ใช้เก็บ ID ของรูปจำลองต้นแบบของมอนสเตอร์
- **MonsterPosition** เป็นข้อมูลชนิดอาร์เรย์ ชนิด POINT ที่มีขนาด [MonsterCount] ใช้เก็บ ค่าตำแหน่งของมอนสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้