

เกมรูปแบบไทยสำหรับเด็ก

THAI GAMES FOR KIDS

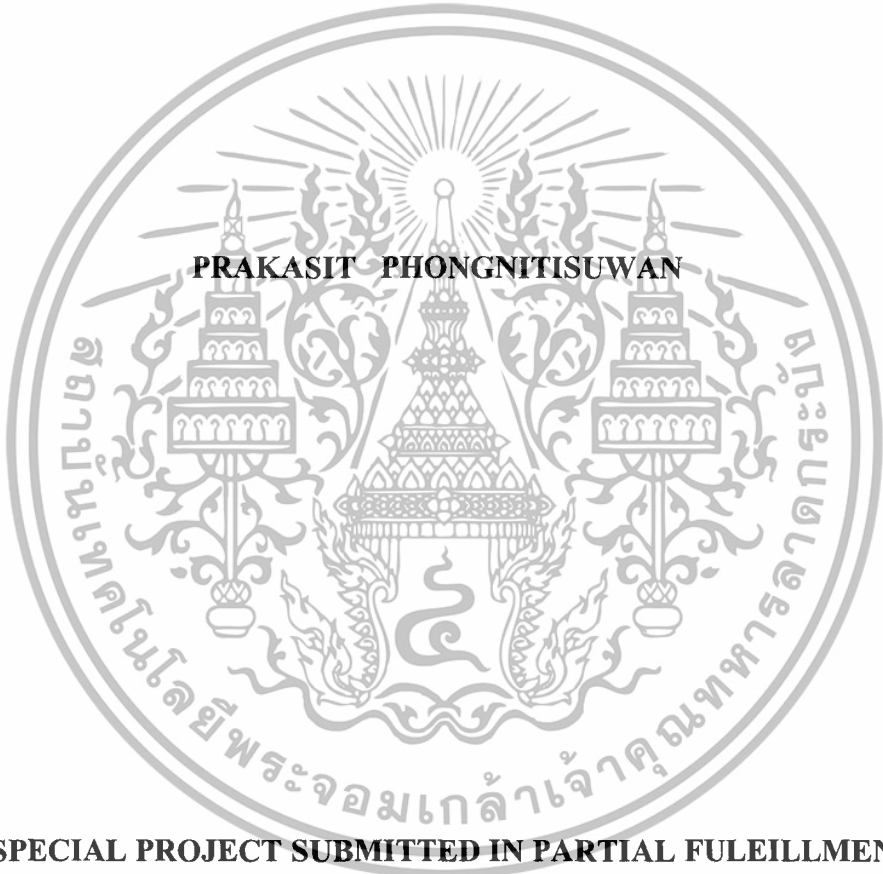


ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

11950638

THAI GAMES FOR KIDS

PRAKASIT PHONGNITISUWAN



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULEILLMENT
OF THEREQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECNOLOGY LADKRABANG
ACADEMIC YEAR 2007**

แต่ท่านอาจารย์ที่ปรึกษาที่มอบความเมตตากรุณาให้แก่ลูกศิษย์คนนี้
ประกาศิต



หัวข้อปัญหาพิเศษ เกมรูปแบบไทยสำหรับเด็ก
THAI GAMES FOR KIDS
ชื่อนักศึกษา นายประกาศิต พงศ์นิติสุวรรณ 47050782
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขา วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา ผศ.ดร.กรกช ประชุมรักษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2550

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.นันทิกา เบญจเทพานันท์	  
ประธานกรรมการ	
ผศ.ศิริลักษณ์ อนันต์สถิตย์สิน	
กรรมการ	
ผศ.ดร.กรกช ประชุมรักษ์	 
กรรมการและอาจารย์ที่ปรึกษา	



(รองศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	เกมรูปแบบไทยสำหรับเด็ก
ชื่อนักศึกษา	นายประกาศิต พงศ์นิธิสุวรรณ 47050782
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผศ.ดร.กรกช ประชุมรักษ์

บทคัดย่อ

ในปัจจุบันเกมคอมพิวเตอร์ได้รับความนิยมเป็นจำนวนมาก เกมที่มีอยู่นั้นมีมากมายหลากหลายรูปแบบแต่ในปัจจุบันพบว่าเกมที่ทำออกเป็นจำนวนมาก มีเกมที่เหมาะสมสำหรับเด็กเป็นจำนวนน้อย และยิ่งเกมที่จะเป็นรูปแบบไทยๆยังมีจำนวนน้อย ดังนั้นจึงมีแนวคิดที่จะสร้างเกมและพัฒนาเกมออกมาในรูปแบบไทยๆ และมีเนื้อหาของเกมที่เหมาะสมกับเด็กโดยสอดคล้องเอกลักษณ์ต่างๆของความเป็นไทย เช่น ตัวละครเป็นเด็กไทย แต่งกายในลักษณะของคนสมัยก่อน ที่สื่อออกมา ไม่ว่าจะเป็นทรงผม เสื้อผ้า ฉากที่ใช้ในการสร้างเกมก็จะเป็นวิถีชาวบ้านสมัยก่อน รูปทรงของบ้านเรือนต่างๆ แต่ยังคงรูปแบบของเกมที่ดีดูและน่าสนใจเอาไว้

ดังนั้นปัญหาพิเศษนี้จึงเสนอเกมสำหรับเด็กที่ทำออกมาโดยเน้นลักษณะความเป็นไทยๆ โดยแต่ละเกมที่ออกแบบมาจะมีลักษณะที่แตกต่างกัน โดยเกมจะถูกออกแบบมาทั้งในลักษณะที่คล้ายเกมกีฬา เน้นความสนุกสนาน และเกมที่ใช้ทักษะทางความคิดในการเล่นมีการเพิ่มระดับของความยากง่ายของเกมขึ้นไปเรื่อยๆซึ่งเกมที่ออกแบบมาทั้งหมดนี้ได้ถูกสร้างขึ้นมาโดยโปรแกรม Game Maker

Title	THAI GAMES FOR KIDS
Student	Mr. Prakasit Phongnitisuwan 47050782
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Sciences
Programmer	Computer Science
Academic Year	2007
Special Project Advisor	Assistant Professor Dr.Korakot Prachumrak

ABTRACT

Game computer becomes the most popular game that everyone is play in this time. So many types of games are not appropriate for the children in consequence becomes an idea of this special project to develop game in Thai culture pattern such as Thai kids character dress up in Thai casual wear, Thai hairstyle, back screen is present to Thai livelihood. However remain representing in interesting way.

So this special project is focus to present game in the Thai culture pattern and has different kind in each game by teaches the thinking skill, that has a level up if you finish each level. All of these games are created by 'Game Maker' program.

กิตติกรรมประกาศ

ในการทำโครงการปัญหาพิเศษเรื่อง สามารถทำสำเร็จลุล่วงไปได้ด้วยดีนั้น ทางผู้จัดทำต้องขอขอบพระคุณ ผศ.ดร.กรกช ประชุมรักษ์ อาจารย์ผู้รับผิดชอบโครงการปัญหาพิเศษฉบับนี้ ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาในด้านต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ได้ประศาสตร์วิชาความรู้ ทั้งทางด้านทฤษฎีและภาคปฏิบัติแก่ทางคณะผู้จัดทำจนกระทั่งปัญหาพิเศษนี้สัมฤทธิ์ผลด้วยดีทุกประการ

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ที่คอยให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์

นอกจากนี้ทางผู้จัดทำต้องขอขอบพระคุณ เพื่อนๆและรุ่นพี่ทุกคน ที่ให้ความช่วยเหลือในด้านต่างๆที่เกี่ยวข้องกับปัญหาพิเศษไว้ ณ ที่นี้



สารบัญ

หน้า

บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญภาพ	vii

บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของปัญหาพิเศษ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนในการดำเนินการ.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 โปรแกรม Game Maker.....	3
2.1.1 ความเป็นมา.....	3
2.1.2 หลักการออกแบบเกมเบื้องต้น(Basic Game Design Theory)	3
2.1.2.1 Interactive Design.....	3
2.1.2.2 Graphic Design.....	4
2.1.2.3 Game Play Design.....	4
2.1.2.4 Storytelling.....	8
2.2 เครื่องมือภายในโปรแกรม Game Maker.....	9
2.2.1 คำสั่งเมนูบาร์.....	9
2.2.1.1 File.....	9
2.2.1.2 Edit.....	15
2.2.1.3 Add.....	16
2.2.1.4 Script.....	25
2.2.1.5 Run.....	26
2.2.1.6 Window.....	31

สารบัญ (ต่อ)

หน้า

2.2.1.7 Help.....	31
2.2.2 Resource.....	32
2.2.2.1 Sprites Resource.....	32
2.2.2.2 Sounds Resource.....	44
2.2.2.3 Background Resource.....	47
2.2.2.4 Paths Resource	49
2.2.2.5 Script Resource.....	52
2.2.2.6 Fonts Resource	53
2.2.2.7 Time Lines Resource.....	55
2.2.2.8 Object Resource.....	57
2.2.2.9 Room Resource.....	73
บทที่ 3 ขั้นตอนการดำเนินการ.....	75
3.1 ขั้นตอนการวิเคราะห์และการออกแบบ.....	76
3.1.1 การออกแบบระบบเกมและการกำหนดโครงสร้างของเกม.....	76
3.1.1.1 เกมกินเมือง.....	76
3.1.1.2 เกมโยนตะกร้อลงห่วง.....	76
3.1.1.3 เกมเป็คน้อยมหัศจรรย์.....	76
3.1.2 การออกแบบรูปภาพและกราฟิก.....	77
3.1.3 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น.....	77
3.1.4 การออกแบบการจัดเก็บข้อมูล.....	77
3.1.5 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้.....	77
3.2 ขั้นตอนการสร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม.....	77
3.2.1 เกมกินเมือง.....	78
3.2.1.1 รูปภาพที่ใช้ในการสร้างหน้าจอหลัก.....	78
3.2.1.2 รูปภาพที่ใช้ในหน้าจอการเลือกระดับการเล่น.....	82
3.2.1.3 รูปภาพหน้าจอแสดงวิธีการเล่นเกม.....	83
3.2.1.4 รูปภาพของเกม.....	85
3.2.2 เกมโยนตะกร้อลงห่วง.....	88
3.2.2.1 รูปภาพที่ใช้ในการสร้างหน้าจอหลัก.....	88

สารบัญ (ต่อ)

	หน้า
3.2.2.2 รูปภาพของเกม.....	90
3.2.3 เกมเป็ดน้อยมหัศจรรย์.....	92
3.2.3.1รูปภาพที่ใช้ในการสร้างหน้าจอหลัก.....	92
3.2.3.2 รูปภาพของเกม.....	92
3.3 ขั้นตอนการสร้างเกม.....	94
3.3.1 เกมกินเมือง.....	94
3.3.1.1 หน้าจอหลัก.....	94
3.3.1.2 หน้าจอแสดงวิธีการเล่น.....	96
3.3.1.3 หน้าจอการเลือกระดับเกม.....	98
3.3.2 เกมโยนตะกร้อลงห่วง.....	102
3.3.3 เกมเป็ดน้อยมหัศจรรย์.....	104
3.4 ขั้นตอนการทดสอบเกม.....	106
บทที่ 4 ขั้นตอนการใช้งานเกม.....	107
4.1 เครื่องมือที่ใช้ในการทดลอง.....	107
4.1.1 ขั้นตอนการใช้งานโปรแกรม.....	107
4.1.1.1 การลงโปรแกรม.....	107
4.1.1.2 คู่มือการเล่นเกม.....	111
4.2 การออกแบบการทดลองเกม.....	116
4.2.1 ลักษณะการกดเมาส์.....	116
4.2.2 ลักษณะการกดคีย์บอร์ด.....	116
4.3 ผลการทดลอง.....	116
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	117
5.1 ขั้นตอนการออกแบบ Story Board.....	117
5.2 ขั้นตอนการทำเกม.....	117
5.3 ขั้นตอนการสร้างตัวละคร.....	117
5.4 ข้อเสนอแนะและแนวทางในการพัฒนา.....	118
รายการอ้างอิง.....	119

สารบัญภาพ

ภาพที่	หน้า
2.1 รูปหน้าจอหลักในโปรแกรม Game Maker.....	9
2.2 รูปเมนู File.....	9
2.3 รูปหน้าจอคำสั่ง General.....	11
2.4 รูปหน้าจอคำสั่ง Forms.....	12
2.5 รูปหน้าจอคำสั่ง Scripts and Code.....	12
2.6 รูปหน้าจอคำสั่ง Colors.....	13
2.7 รูปหน้าจอคำสั่ง Image Editor.....	14
2.8 รูปหน้าจอคำสั่ง Sound editor.....	14
2.9 รูปเมนู Edit.....	15
2.10 รูปเมนู Add.....	16
2.11 รูปหน้าจอคำสั่ง Graphics.....	17
2.12 รูปหน้าจอคำสั่ง Resolution.....	18
2.13 รูปหน้าจอคำสั่ง other.....	19
2.14 รูปหน้าจอคำสั่ง loading.....	20
2.15 รูปหน้าจอคำสั่ง Constants.....	21
2.16 รูปหน้าจอคำสั่ง Include.....	22
2.17 รูปหน้าจอคำสั่ง Error.....	23
2.18 รูปหน้าจอคำสั่ง Info.....	24
2.19 รูปเมนู Script.....	25
2.20 รูปหน้าจอคำสั่ง Search in Scripts.....	26
2.21 รูปเมนู Run หลัก.....	26
2.22 รูปหน้าจอคำสั่ง Run in Debug mode.....	27
2.23 รูปบริเวณไอคอน.....	28
2.24 รูปเมนู Run.....	29
2.25 รูปเมนู Watch.....	29
2.26 รูปเมนู Tools.....	29
2.27 รูปหน้าจอคำสั่ง Global Variables.....	30
2.28 รูปเมนู Window.....	31

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.29 รูปเมนู Help.....	31
2.30 รูปเมนู Resource ต่างๆ.....	32
2.31 รูปหน้าจอคำสั่ง Sprites Resource.....	32
2.32 รูปหน้าจอคำสั่ง Sprite Properties ส่วนหน้า.....	33
2.33 รูปหน้าจอคำสั่ง Sprite Properties ส่วนหลัง.....	34
2.34 รูปหน้าจอคำสั่ง Sprite Editor.....	35
2.35 รูปหน้าจอคำสั่ง loading a Sprite.....	36
2.36 รูปหน้าจอคำสั่ง Sprite Editor แบบแสดง Preview.....	37
2.37 รูปเมนู Edit ใน Sprite Editor.....	38
2.38 รูปเมนู Transform ใน Sprite Editor.....	39
2.39 รูปเมนู Images ใน Sprite Editor.....	40
2.40 รูปเมนู Animation ใน Sprite Editor.....	41
2.41 รูปบริเวณส่วนแสดงภาพ.....	42
2.42 รูปบริเวณส่วนแสดงภาพย่อย.....	43
2.43 รูปหน้าจอคำสั่ง Sounds Resource.....	44
2.44 รูปหน้าจอคำสั่ง Sound Properties.....	45
2.45 รูปหน้าจอคำสั่ง Kind.....	45
2.46 รูปหน้าจอคำสั่ง Effects.....	46
2.47 รูปหน้าจอคำสั่ง Volume.....	46
2.48 รูปหน้าจอหลักคำสั่ง Background Properties.....	47
2.49 รูปหน้าจอคำสั่ง Background Properties ด้านซ้าย.....	48
2.50 รูปหน้าจอหลักคำสั่ง Path Properties.....	49
2.51 รูปหน้าจอคำสั่ง Path Properties ด้านซ้าย.....	50
2.52 รูปหน้าจอคำสั่ง Path Properties ด้านขวา.....	51
2.53 รูปหน้าจอหลักคำสั่ง Script Resource.....	52
2.54 รูปหน้าจอฟังก์ชันของ Script Resource.....	53
2.55 รูปหน้าจอคำสั่ง Font Properties.....	54
2.56 รูปหน้าจอหลักคำสั่ง Time Lines Resource.....	55

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.57 รูปหน้าจอคำสั่ง Actions.....	56
2.58 รูปหน้าจอหลักคำสั่ง Time Lines Resource.....	57
2.59 รูปหน้าจอหลักคำสั่ง Object Properties.....	57
2.60 รูปแสดงตัวอย่างการ Mask.....	58
2.61 รูปหน้าจอคำสั่ง Events.....	59
2.62 รูปปุ่มการเลือก Events.....	59
2.63 รูปหน้าจอคำสั่ง Action Library	61
2.64 รูปหน้าจอคำสั่ง Start moving in a direction.....	62
2.65 รูปหน้าจอคำสั่ง Room Properties.....	73
3.1 รูปฉากแสดงการไหลตเข้าสู่เกมกินเมือง.....	78
3.2 รูปฉากแอนิเมชันก่อนเข้าสู่หน้าจอหลักเกมกินเมือง.....	79
3.3 รูปฉากหลังของหน้าจอหลักเกมกินเมือง.....	80
3.4 รูปปุ่มต่างๆของหน้าจอหลักเกมกินเมือง.....	80
3.5 รูปตัวละครหน้าจอหลักเกมกินเมือง.....	81
3.6 รูปปุ่มออฟชั่นหน้าหลักเกมกินเมือง.....	81
3.7 รูปฉากหลังของหน้าจอการเลือกระดับเกมกินเมือง.....	82
3.8 รูปปุ่มต่างๆของหน้าการเลือกระดับเกมกินเมือง.....	82
3.9 รูปตัวละครหน้าจอการเลือกระดับเกมกินเมือง.....	83
3.10 รูปฉากหลังของหน้าจอแสดงวิธีการเล่นเกมกินเมือง.....	83
3.11 รูปวิธีการเล่นเกมกินเมือง.....	84
3.12 รูปปุ่มๆต่างๆของหน้าจอวิธีการเล่นเกมกินเมือง.....	84
3.13 รูปฉากหลังของเกมกินเมือง.....	85
3.14 รูปปุ่มปรับเสียงเกมกินเมือง.....	85
3.15 รูปปุ่มปรับเพลงเกมกินเมือง.....	86
3.16 รูปตัวนับคะแนนเกมกินเมือง.....	86
3.17 รูปตัวนับจำนวนรอบเกมกินเมือง.....	86
3.18 รูปปุ่มการออกจากเกมกินเมือง.....	86
3.19 รูปตัวเมืองของเกมกินเมือง.....	87

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.20 รูปฉากแสดงการโหลดเข้าเกมโยนตะกร้อลงห่วง.....	88
3.21 รูปฉากหลังของหน้าจอหลักเกมโยนตะกร้อลงห่วง.....	88
3.22 รูปตัวละครของฉากหลังในหน้าจอหลักเกมโยนตะกร้อลงห่วง.....	89
3.23 รูปตัวละครของฉากเกมโยนตะกร้อลงห่วง.....	90
3.24 รูปตะกร้อ.....	91
3.25 รูปห่วงลักษณะต่างๆ.....	91
3.26 รูปฉากแสดงการโหลดเข้าเกมเป็ดน้อยมหัศจรรย์.....	92
3.27 รูปฉากหลังของเกมเป็ดน้อยมหัศจรรย์.....	92
3.28 รูปตัวละครของฉากเกมเป็ดน้อยมหัศจรรย์.....	93
3.29 รูปแถบพลังและคะแนนของฉากเกมเป็ดน้อยมหัศจรรย์.....	93
3.30 รูปหน้าจอหลักเกมกินเมือง.....	94
3.31 แผนภาพการทำงานในหน้าจอหลักเกมกินเมือง.....	95
3.32 รูปหน้าจอวิธีการเล่นเกมกินเมือง.....	96
3.33 แผนภาพการทำงานในหน้าจอวิธีการเล่นเกมกินเมือง.....	97
3.34 รูปหน้าจอการเลือกระดับเกมกินเมือง.....	98
3.35 แผนภาพการทำงานในหน้าจอการเลือกระดับเกมกินเมือง.....	99
3.36 รูปหน้าจอการเล่นเกมกินเมือง.....	100
3.37 แผนภาพการทำงานในหน้าจอการเล่นเกมกินเมือง.....	101
3.38 รูปหน้าจอการเล่นเกมโยนตะกร้อลงห่วง.....	102
3.39 แผนภาพการทำงานในหน้าจอการเล่นเกมโยนตะกร้อลงห่วง.....	103
3.40 รูปหน้าจอการเล่นเกมเป็ดน้อยมหัศจรรย์.....	104
3.41 แผนภาพการทำงานในหน้าจอการเล่นเกมเป็ดน้อยมหัศจรรย์.....	105
4.1 รูปหน้าจอเริ่มต้นการติดตั้ง (กดปุ่ม Next)	108
4.2 รูปหน้าจอ Information (กดปุ่ม Next)	108
4.3 รูปหน้าจอ License (เลือก I agree แล้วกดปุ่ม Next)	109
4.4 รูปหน้าจอ Directory (เลือกตำแหน่งที่จะติดตั้งแล้วกดปุ่ม Next)	109
4.5 รูปหน้าจอ Confirmation (ยืนยันการติดตั้งแล้วกดปุ่ม Start)	110
4.6 รูปหน้าจอ Installing (รอการติดตั้งสักครู่)	110

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.7 รูปหน้าจอ End (การติดตั้งเสร็จเรียบร้อยแล้วกดปุ่ม Exit)	111
4.8 แสดงหน้าจอเริ่มต้นของเกมกินเมือง.....	111
4.9 แสดงหน้าจอวิธีการเล่นเกมกินเมือง.....	112
4.10 แสดงหน้าจอการเลือกระดับของเกมกินเมือง.....	113
4.11 แสดงหน้าจอการเล่นเกมกินเมือง.....	113
4.12 แสดงหน้าจอการเล่นเกมโยนตะกร้อลงห่วง.....	114
4.13 แสดงหน้าจอการเล่นเกมเปิดน้อยมหัศจรรย์.....	115



บทที่ 1

บทนำ (Introduction)

1.1 ความสำคัญและที่มาของปัญหาพิเศษ

ในปัจจุบันเกมคอมพิวเตอร์ได้รับความนิยมเป็นจำนวนมาก เกมที่มีอยู่นั้นมีมากมายหลากหลายรูปแบบแต่ในปัจจุบันพบว่าเกมที่ทำออกเป็นจำนวนมาก มีเกมที่เหมาะสมสำหรับเด็กเป็นจำนวนน้อย และยิ่งเกมที่จะเป็นรูปแบบไทยๆยังมีจำนวนน้อย ดังนั้นจึงมีแนวคิดที่จะสร้างเกมและพัฒนาเกมออกมาในรูปแบบไทยๆ และมีเนื้อหาของเกมที่เหมาะสมกับเด็กโดยสอดคล้องเอกลักษณ์ต่างๆของความเป็นไทย เช่น ตัวละครเป็นเด็กไทย แต่งกายในลักษณะของคนสมัยก่อน ที่สื่อออกมา ไม่ว่าจะเป็นทรงผม เสื้อผ้า ฉากที่ใช้ในการสร้างเกมก็จะเป็นวิถีชาวบ้านสมัยก่อน รูปทรงของบ้านเรือนต่างๆ แต่ยังคงรูปแบบของเกมที่ดีดูและน่าสนใจเอาไว้

ดังนั้นการศึกษาหาความรู้ในเรื่องเหล่านี้จะมีประโยชน์โดยตรงสำหรับเด็ก ในการเล่นเกมเพื่อใช้เวลาว่างให้เกิดประโยชน์ และมีคุณสมบัติจริงๆ โดยไม่มีโทษต่อเด็ก และเป็นผลประโยชน์ทางอ้อมสำหรับผู้ปกครองที่ไม่ต้องกลัวว่าผู้เล่นที่เป็นบุตรหลานจะติดเกมที่ไม่มีประโยชน์ เพราะเกมนี้จะช่วยสอนเกี่ยวกับทักษะทางด้านตรรกะและพัฒนาทางด้านความคิดให้กับผู้เล่น แล้วยังสอดคล้องค่านิยม ศิลปะ และเอกลักษณ์ความเป็นไทย ที่แฝงมาในรูปแบบของเกมที่ได้ออกแบบมาโดยเน้นความเป็นไทย

1.2 วัตถุประสงค์

- 1). เพื่อเพิ่มความรู้เกี่ยวกับทักษะทางด้านตรรกะและพัฒนาทางด้านความคิดให้กับผู้เล่น
- 2). เพื่อสอดคล้องเอกลักษณ์ต่างๆของความเป็นไทย
- 3). สามารถนำความรู้ไปประยุกต์ใช้ในชีวิตจริงได้
- 4). เพื่อให้ผู้เล่นรู้จักการใช้เวลาว่างให้เกิดประโยชน์
- 5). เพื่อให้ผู้เล่นเกิดความเพลิดเพลินและเกิดความคิดสร้างสรรค์

1.3 ขอบเขตของปัญหาพิเศษ

- 1). ใช้ Game Maker เป็น Library สำคัญในการศึกษาวิธีการพัฒนาเกม
- 2). การสร้างโปรแกรมเกมนั้นจะพัฒนาบนเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป(PC-based) ที่มีระบบปฏิบัติการ Windows และเป็นแบบ Stand-alone

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1). ได้รู้จักการพัฒนาเกมอย่างมีระบบ
- 2). รู้จักออกแบบเกมเพื่อให้สามารถนำไปใช้กับชีวิตจริงได้
- 3). มีความรู้เรื่องเกี่ยวกับคอมพิวเตอร์มากขึ้น
- 4). ให้ความรู้เพิ่มเติมกับบุคคลที่มาเล่นเกม
- 5). เกิดการอนุรักษ์ศิลปปะและเอกรักษ์ความเป็นไทย

1.5 ขั้นตอนในการดำเนินงาน

- 1). ออกแบบเกมและคิด Story Board
- 2). ศึกษา Game maker
- 3). สร้างเกมตามทีออกแบบไว้
- 4). ทดสอบเกมที่สร้างขึ้น
- 5). ปรับปรุงข้อผิดพลาดและแก้ไข



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 โปรแกรม Game Maker

2.1.1 ความเป็นมา

โปรแกรม Game Maker เป็นโปรแกรมที่ถูกออกแบบมา เพื่อให้ทุกคนสามารถสร้างสรรค์งานเกมได้อย่างไร้ขีดจำกัด และสามารถสร้างเกมเป็นของตัวเองได้ โดยโปรแกรมถูกออกแบบมาให้มีอินเตอร์เฟซที่เข้าใจได้ง่ายและน่าใช้ เมื่อคุ้นเคยกับโปรแกรมนี้อาจก้าวสู่การเป็น Advance User คุณยังสามารถศึกษา GML (Game Maker Language) เพื่อเขียนสคริปต์และสามารถเรียกใช้ฟังก์ชันต่างๆได้อย่างหลากหลาย คุณจึงสามารถที่จะสร้างเกมแบบใดก็ได้ได้อย่างไร้ขีดจำกัด รวมถึงการใช้งานDLLซึ่งเสมือนปลั๊กอินทำโปรแกรม Game Maker มีประสิทธิภาพเพิ่มขึ้น

ซึ่งตัวโปรแกรมถูกออกแบบมาโดยมุ่งเน้นการสร้างเกม 2 มิติเป็นหลักแต่อย่างไรก็ตาม ตัวโปรแกรมก็มีฟังก์ชันที่จะสามารถสร้างเกม 3 มิติได้แต่ยังมีข้อจำกัดอยู่

2.1.2 หลักการออกแบบเกมเบื้องต้น(Basic Game Design Theory)

หลักการออกแบบเกมให้มีความสนุกนั้น ไม่มีกฎเกณฑ์ตายตัว จะขอยกตัวอย่างปัจจัยที่เกี่ยวข้องกับความรู้สึกรักของผู้เล่น โดยตรงดังต่อไปนี้

2.1.2.1 Interactive Design

การออกแบบการควบคุมในเกม ไม่ว่าจะเป็นการควบคุมตัวละคร ควบคุมมุมมอง ฯลฯ ถ้าเกมปราศจากอินเตอร์แอคทีฟแล้ว ก็จะกลายเป็นแอนิเมชันที่เราสามารถดูได้เท่านั้น

1.) Interactive Task คือ อินเตอร์เฟซหรือแถบควบคุมที่อยู่บริเวณหน้าจอเวลาเล่นเกมมีความสำคัญมากสำหรับนักเล่นเกมมือใหม่ ที่ยังไม่เคยเล่นเกมดังกล่าวมาก่อนเมื่อมีการออกแบบลำดับการจัดวางที่ดีแถบควบคุมนี้ จะเป็นตัวนำทางให้ผู้เล่นควบคุมเกมได้อย่างราบรื่น

2.) Interactive Device คือ การควบคุมจากภายนอก เช่น คีย์บอร์ด จอยสติ๊ก เมาส์ หน้าจอทัชสกรีน หรือจะเป็นอุปกรณ์เสริมอื่นๆที่เกมต้องการ เช่น กล้อง ไมค์ ฯลฯ ตามแต่ลักษณะของเกมที่ถูกออกแบบมาสิ่งต่างๆที่กล่าวมานั้นจะเป็นตัวควบคุมเกมให้ดำเนินไปอย่างราบรื่น

ในการออกแบบนั้นเราจะเลือกใช้การอินเตอร์แอคทีฟแบบใดก็ขึ้นอยู่กับลักษณะการควบคุมเป็นหลัก เกมส่วนใหญ่จะมีทั้ง 2 ประเภท จะต้องทำให้ผู้เล่นสะดวกในการควบคุมมากที่สุดเท่าที่จะเป็นไปได้ ออกแบบการควบคุมให้น้อยที่สุดแต่มีประสิทธิภาพในการควบคุมมากที่สุด เป็นสิ่งที่นักออกแบบพึงกระทำ เกมต่างๆบางเกมที่ใช้เวลาสร้างไม่ต่ำกว่าปีครึ่ง ใช้เงินทุนไปไม่ต่ำกว่าล้านเหรียญสหรัฐ ยังขายไม่

ออกเพราะผู้เล่นสับสนกับการบังคับ ถึงเกมจะมีกราฟิกที่สวยงามเพียงใดก็หมดสนุกได้เพราะความยุ่งยากในการบังคับ

ถ้าคุณกำลังคิดจะออกแบบเกมมาซักเกมเพื่อขายมัน จงคิดอยู่เสมอว่าผู้บริโภคหรือนักเล่นเกมค่อนข้างจะมีความอ่อนไหวมากต่อเกมที่เลือกเล่น ถ้ามีส่วนหนึ่งส่วนใดในเกมที่เขาไม่พอใจ เขาสามารถที่จะเลิกเล่นเกมของคุณได้ทันทีและหาเกมใหม่ที่ดีกว่ามาเล่น เนื่องจากในปัจจุบันมีเกมให้เลือกเล่นมากมายนัก ไม่มีเหตุผลอะไรที่เค้าจะต้องมาซื้อเกมของคุณ เท่านั้นไม่พอ เค้ายังจะบอกเพื่อนๆของเค้าปากต่อปากว่าเกมของคุณนั้นไม่ใช่อะไร โดยที่เค้าจะไม่สนใจว่าเกมของคุณทำมานานหลายปี หรือใช้เงินทุนมหาศาล!! สุดท้ายคือไม่มีใครเล่นเกมของคุณ

2.1.2.2 Graphic Design

นักเล่นเกมส่วนใหญ่ใช้เวลาในการตัดสินใจว่าเค้าจะเลือกเล่นเกมนี้หรือไม่ ในครั้งแรกที่เค้าเห็นเกมใหม่และเกมที่มีกราฟิกสวยงามอลังการ มันทำให้เค้าเกิดความรู้สึกประทับใจแรกพบ รู้สึกอยากเล่น และทำให้เค้าหลงใหลไปกับมันในช่วงแรก

ถ้าจะเปรียบเทียบให้เห็นภาพก็คือ ผู้ชายเมื่อเห็นผู้หญิงสองคนเดินผ่านมา คนแรกหน้าตาธรรมดา ส่วนคนที่2 เป็นคนสวย หุ่นดี บุคลิกดี แน่นอนว่าผู้ชายจะต้องหลงใหลไปกับผู้หญิงคนที่2ที่หน้าดูชมมากกว่า ความรู้สึกเหล่านี้เป็นไปตามสัญชาตญาณพื้นฐานของมนุษย์ที่นักออกแบบเกมไม่สามารถมองข้ามได้ เมื่อพูดถึงเรื่องของเกม เกมที่มีกราฟิกที่สวยงามนั้นมีชัยไปกว่าครึ่งแล้ว และเป็นสิ่งดึงดูดผู้เล่นได้เป็นอย่างดี องค์ประกอบของกราฟิกที่ดีนั้นมีหลายอย่าง ส่วนใหญ่มาจากหลักการออกแบบพื้นฐาน เช่น ทฤษฎีสี จังหวะ การจัดวาง ความกลมกลืน ฯลฯ รวมถึงลักษณะมุมมองที่เหมาะสมกับเกมที่เราก่อแบบ ถ้าเกมเป็น 3 มิติ กราฟิกต่างๆในเกม ซึ่งความสวยงามจะอยู่ที่ความประณีตในการสร้างโมเดล 3 มิติ การทำพื้นผิวให้กับโมเดล สำหรับเกมที่มีเนื้อเรื่อง เช่น เกม RPG การออกแบบตัวละครเป็นสิ่งที่ต้องพิถีพิถันเป็นอย่างมาก

2.1.2.3 Game Play Design

คือลักษณะการเล่นหลักของเกมนั้นๆรวมถึงกฎข้อบังคับภายในเกม เช่นหมากรุกมีลักษณะการเล่นแบบผลัดกันเดิน (Turn-base) และยูนิตแต่ละตัวมีรูปแบบการเดินที่แตกต่างกัน ฝ่ายใดโดนกินขุนก่อนจะแพ้ Game Play เปรียบเสมือนหัวใจสำคัญของเกม และเป็นนิยามของคำว่า “เกม” ถ้ามีปัจจัยนี้อย่างเดียวก็ถือว่าเป็นเกมได้ และนักออกแบบเกมควรให้ความสำคัญกับปัจจัยนี้มากที่สุดเป็นอันดับแรก ยกตัวอย่างเกม CM หรือ Championship Manager เกมวางแผนที่คุณจะต้องรับหน้าที่เป็นผู้จัดการทีมฟุตบอล และพาทีมของคุณประสบความสำเร็จให้ได้ เป็นที่น่าสนใจที่ว่าเกมนี้ไม่ใช้กราฟิกเป็นจุดขาย แต่ใช้ Game Play เป็นจุดขายนักเล่นเกมที่หลงใหลในเกมนี้หลายคนประทับใจในแนวความคิดและลักษณะการเล่น จนทำให้เกมนี้ถูกสร้างภาคต่อขึ้นมาเรื่อยๆ มีหลายเกมที่สร้างขึ้นมาเลียนแบบ โดยเพิ่มกราฟิก 3 มิติ ให้เห็นการ

เล่นทั้งสนาม แต่ก็ไม่ได้ได้รับความนิยมเท่าที่ควร เนื่องจากนักออกแบบไม่เข้าใจว่าเกมที่มีลักษณะการเล่น เช่นนี้ ไม่จำเป็นต้องใช้กราฟิกให้เห็นเป็นตัวละครเล่น เพียงแต่มีกราฟิกบางอย่างที่จะทำให้ผู้เล่นสามารถวิเคราะห์สิ่งต่างๆ เพื่อนำมาปรับปรุงทีมของเขาได้ก็เพียงพอแล้ว

เกมจะสนุกหรือไม่ขึ้นอยู่กับการเล่นแบบ Game Play ก่อนข้างมากเกมที่จะอยู่ได้ยาวนานเป็นที่จดจำและเป็นอมตะนั้น ขึ้นอยู่กับการออกแบบลักษณะการเล่นที่สามารถดึงดูดความสนใจของผู้เล่นได้ตลอดเวลา ซึ่งถ้าเกมใดออกแบบ Game Play มาอย่างดีไม่ว่าจะเล่นกี่ร้อยกี่พันครั้งก็ยังเป็นเกมที่สนุกไม่น่าเบื่ออยู่เสมอ

นักเล่นเกมอยากที่จะเล่นเกมที่สนุกเท่านั้น นี่คือความเป็นจริง เกมโดยส่วนใหญ่จะต้องเน้น Game Play ควบคู่ไปกับปัจจัยอื่นๆ ยกเว้นเกมบางประเภทซึ่งจะต้องเน้น Game Play เป็นตัวขับเคลื่อนหลัก ซึ่งนักออกแบบจะต้องเข้าใจถึงจุดประสงค์ของเกมนั้นๆก่อนจึงจะสามารถสร้างความกลมกลืนกับปัจจัยต่างๆได้ เมื่อพูดถึง Game Play ที่ดี นั่นคือต้องทำให้เกมมีความสนุกสนานเพราะฉะนั้นทำอย่างไรนักออกแบบเกมจึงจะสามารถสร้างเกมที่ไม่น่าเบื่อได้ เรื่องนี้เป็นสิ่งที่ละเอียดอ่อนและอาจจะต้องอาศัยหลักจิตวิทยาพื้นฐานของมนุษย์ในการอธิบาย การที่คนเราจะเลือกเล่นเกมสักเกมหนึ่งขึ้นอยู่กับพื้นฐานหลายอย่างโดยส่วนตัว สำหรับในเชิงการค้า ถ้านักออกแบบต้องการที่จะสร้างเกมสักเกมหนึ่งเพื่อขายให้ได้มากที่สุด คงจะต้องสำรวจตลาดกลุ่มเป้าหมายและวิเคราะห์ให้ดีเสียก่อนที่จะเลือกลักษณะเกมที่จะสร้างขึ้นมาเกมแต่ละแนวก็มีหลักการออกแบบแตกต่างกันไป เช่น เกมแอคชั่น จะต้องออกแบบการควบคุมให้ดีและออกแบบฉากให้ดูน่าสนใจ เกม RPG จะต้องออกแบบเนื้อเรื่องให้ชวนติดตาม เป็นต้น ถ้าจะพูดว่าเกมในแต่ละแนวมีหลักการออกแบบอย่างไรบ้างคงเป็นเรื่องที่ยาวมาก แต่ในทุกๆแนวเกมก็มีสิ่งที่นักออกแบบควรคำนึงอยู่คล้ายๆกัน เรียกว่าเป็น Fun Factor หรือปัจจัยที่จะก่อให้เกิดความสนุกสนานในเกม ยกตัวอย่างเช่น

- **Concentration** เกมจะต้องดึงดูดและผู้เล่นสามารถอยู่กับเกมได้นาน

- 1) ควรจะมีสิ่งกระตุ้นอย่างหลากหลายภายในเกม เพื่อให้ผู้เล่นได้จดจ่อและเพลิดเพลินไปกับสิ่งเหล่านั้น เช่น มีไอเท็มในเกมอย่างหลากหลายให้เลือกใช้ ซึ่งไอเท็มแต่ละอย่างจะให้ผลที่ต่างกันออกไป หรือในแต่ละเกมก็อาจจะมีมุมมองที่หลากหลาย
- 2) สิ่งกระตุ้นภายในเกมจะต้องมีความน่าสนใจอย่างเพียงพอซึ่งจะแตกต่างกันไปตามลักษณะพฤติกรรมของผู้เล่น
- 3) เกมควรจะเป็นที่นำสนใจของผู้เล่นได้อย่างรวดเร็ว และรักษาระดับความน่าสนใจอันนี้ให้ได้ตลอดทั้งเกม
- 4) เกมแต่ละเกมจะมีเป้าหมายหลัก ผู้เล่นไม่ควรจะทำสิ่งที่รู้สึกว่าจะไม่ใช่สิ่งสำคัญและไม่ใช่ว่าเป้าหมายหลักของเกม ซึ่งจะทำให้เกมน่าเบื่อ เช่น การพูดคุยภายในเกมทีนานเกินไป และไม่ก่อให้เกิดประโยชน์ต่อผู้เล่น

- **Challenge** เกมควรมีการทำท่ายอย่างเพียงพอและเหมาะสมกับระดับทักษะของผู้เล่นตามกลุ่มเป้าหมาย

- 1) ความท้าทายในเกมจะต้องเหมาะสมกับระดับทักษะของผู้เล่น เช่น ถ้าเราออกแบบเกมที่กลุ่มเป้าหมายอยู่ในวัยประมาณ 6-10 ปี เราจะต้องทราบว่าเด็กในวัยนี้สามารถรับระดับความยากได้แค่ไหน ไม่ควรยากหรือง่ายจนเกินไป
- 2) เกมควรมีความท้าทายหลายระดับสำหรับผู้เล่นต่าง ๆ กัน เช่น เกมที่เล่นได้ทุกเพศทุกวัยอาจมีโหมดให้ปรับระดับความยากง่าย
- 3) ระดับความท้าทายควรเพิ่มขึ้นตามทักษะการเล่นที่พัฒนาขึ้นของผู้เล่นตลอดทั้งเกม เช่น เกมในฉากแรกจะง่ายก่อน เพื่อเปิดโอกาสให้ผู้เล่นให้พัฒนาการเล่นและจะยากขึ้นไปเรื่อยๆ ในฉากต่อไป ในทางตรงกันข้าม ถ้าเปิดเกมมาในฉากแรกแล้วยากเกินไปสำหรับผู้เล่น ผู้เล่นจะหมดกำลังใจและเลิกเล่นไปในที่สุด

- **Player Skill** เกมจะเอื้อให้เกิดการส่งเสริมการพัฒนาในด้านทักษะและความชำนาญของผู้เล่น

- 1) ผู้เล่นสามารถเริ่มเล่นเกมโดยไม่ต้องอ่านคู่มือซึ่งระบบ Tutorial เป็นสิ่งจำเป็นสำหรับเกมใหม่ๆ ทุกเกม และต้องทำให้ผู้เล่นรู้สึกเหมือนเล่นเกมจริงๆ อยู่ ไม่ใช่แค่การสอน การเรียนรู้จึงจะไม่น่าเบื่อ
- 2) การเรียนรู้ในการเล่นไม่ควรน่าเบื่อ ควรสอดแทรกเข้าไประหว่างการเล่นเกม อย่าลืมว่าไม่ควรให้ผู้เล่นอ่านอะไรที่ยาวเกินไป สิ่งที่เราควรทำคือ ทำอะไรที่เป็นตัวหนังสือทั้งหมดให้เป็นกราฟิกที่เข้าใจง่าย
- 3) เกมควรมี Help Mode ในขณะที่เล่น ซึ่งผู้เล่นไม่จำเป็นต้องออกจากเกม
- 4) เกมควรจะเพิ่มทักษะของผู้เล่นเมื่อถึงเวลาที่เหมาะสมตลอดทั้งเกม
- 5) ผู้เล่นควรได้รับรางวัลที่เหมาะสมกับสิ่งที่ทำและทักษะที่พัฒนาขึ้น เช่น การได้รับโบนัสที่เป็นคะแนนเพิ่ม หรือได้รับไอเท็มที่พิเศษกว่าอันอื่นๆ เป็นต้น

- **Control** ผู้เล่นควรรู้สึกได้ถึงความสามารถในการควบคุมสิ่งต่างๆ ในเกมได้เป็นอย่างดี

- 1) ผู้เล่นควรรู้สึกว่าสามารถควบคุมตัวละครหรือยูนิตได้เป็นอย่างดีไม่ติดขัด
- 2) ผู้เล่นควรรู้สึกว่าได้ควบคุม Interactive Task (บนหน้าจอ) และ Interactive Device (เช่น Gamepad, Mouse, Keyboard เป็นต้น) อย่างสะดวก
- 3) ผู้เล่นควรรู้สึกว่าสามารถควบคุมการเริ่มเกม การหยุดเกม และการเซฟเกมได้อย่างสะดวก

- 4) เมื่อมีความผิดพลาดร้ายแรงในเกม ผู้เล่นควรได้รับความช่วยเหลือเพื่อแก้ไขความผิดพลาด เช่น ผู้เล่นเพลอ์ทิงไอเท็มสำคัญไป เกมก็ควรมีทางแก้ไขให้ มิเช่นนั้นผู้เล่นอาจหมดกำลังใจและพาลเลิกเล่นไป
 - 5) ผู้เล่นควรรู้สึกว่าได้ควบคุมการกระทำของตัวเองและแผนการที่ใช้ ซึ่งเป็นอิสระในการเล่นในเกมในแบบที่ตนเองต้องการ (ลักษณะนี้มีความสำคัญมากในเกมประเภทวางแผน)
- **Clear Goals** เกมควรมีการเคลียร์จุดมุ่งหมายต่างๆที่ตั้งไว้ในเวลาที่เหมาะสม
 - 1) เป้าหมายของเกมในส่วนย่อยๆ ควรเคลียร์ได้ตั้งแต่แรก
 - 2) เป้าหมายระดับปานกลางควรเคลียร์ได้ในเวลาที่เหมาะสม
 - **Feedback** ผู้เล่นต้องได้รับ “ผลที่ได้รับจากการกระทำ (Feedback)” ที่เหมาะสมในเวลาที่เหมาะสม (ผลที่ตามมา เช่น เก็บเหรียญหรือดีศัตรูแล้วคะแนนจะเพิ่มขึ้น กินขนมแล้วเก่งขึ้น เก็บใบไม้แล้วบินได้ เป็นต้น) ทั้งนี้เมื่อผู้เล่นเล่นเกมได้ดีก็ย่อมได้รับผลตอบแทนที่ดีตามไปด้วย เป็นการสร้างขวัญกำลังใจให้กับผู้เล่นให้อยากที่จะเล่นเกมต่อไป
 - 1) ผู้เล่นควรได้รับ Feedback ที่มีค่าในเกมสูงขึ้นเมื่อเล่นเข้าใจลึกลงไปเรื่อยๆ
 - 2) ผู้เล่นควรได้รับ Feedback ทันทีเมื่อกระทำสิ่งใดๆในเกมที่ดี และควรแสดงให้เห็นได้อย่างชัดเจนอาจจะแสดงทางกราฟิกหรือทางเสียงก็ได้ ทั้งนี้เพื่อให้ผู้เล่นสะดวกในการรับรู้และเข้าใจได้ทันที
 - 3) ผู้เล่นควรรู้สถานะและคะแนนของตัวเองเสมอ ซึ่งสำคัญมากสำหรับเกมที่เล่นแบบไม่มีที่สิ้นสุด เช่น Tetris ถ้าไม่มีข้อมูลเหล่านี้ ผู้เล่นจะตั้งคำถามให้กับตัวเองว่าจะเล่นต่อไปทำไม และพาลเลิกเล่นทันที
 - **Social Interaction** เกมควรรองรับและสร้างโอกาสให้มีการเข้าสังคมซึ่งสำคัญมากในยุคสมัยปัจจุบัน และเราไม่สามารถปฏิเสธได้ว่ามนุษย์เป็นสัตว์สังคม ตัวอย่างเช่น เกมออนไลน์ซึ่งมีกลุ่มคนที่นิยมเล่นกันมาก เนื่องจากเกมเอื้อให้เกิดการพบปะกันเสมือนสังคมใหญ่ๆแห่งหนึ่ง เมื่อเกิดสังคมสิ่งต่างๆที่สามารถดึงดูดผู้เล่นก็จะตามมาเอง หรือจะเป็นเกมที่สามารถเล่นผ่าน LAN เชื่อมโยงกัน หรือเกมที่สามารถเล่นแข่งกันกับเพื่อนอีกคนได้ อย่างเช่นเกมคอนโซลทั่วไป เกมเหล่านี้ทำให้เกิดสังคมขึ้นและจะเป็นสิ่งดึงดูดให้คนหันมาเล่นเกมอื่นๆโดยเน้นความสนุกสนานเป็น หนุ่คณะ
 - 1) เกมควรรองรับการต่อสู้และการช่วยเหลือกันระหว่างผู้เล่น
 - 2) เกมควรช่วยให้เกิดสังคมระหว่างผู้เล่น เช่น การแชต

- 3) เกมควรช่วยให้เกิดสังคมทั้งในและนอกเกม สิ่งที่ทำให้เกิดสังคมภายนอกเกม เช่น การพูดคุยกันภายนอกผ่านฟอรัมหรือเว็บบอร์ดถึงเรื่องเทคนิคการพิชิตเกม เรื่องของเทคนิคการเล่น ฯลฯ

2.1.2.4 Storytelling

Storytelling คือส่วนของเนื้อเรื่องหรือเนื้อหาภายในเกม ซึ่งแนวเกมที่จะต้องเน้นในเรื่องนี้เป็นพิเศษก็คือ แนว RPG, Adventure แต่ไม่ว่าจะเป็นเกมแนวใด ถ้ามีการสอดแทรกเรื่องราวลงไปก็จะสามารถสร้างความสนุกสนานและความบันเทิงให้กับผู้เล่นได้เป็นอย่างดี ซึ่งการสร้าง Storytelling ที่ดีนั้น นักออกแบบเกมจะต้องสวมวิญญาณเป็นผู้กำกับภาพยนตร์โดยเฉพาะเกม RPG ที่เนื้อเรื่องนั้นจะต้องเข้มข้นชวนติดตาม ซึ่งท่านอาจจะหาความรู้เพิ่มเติมได้จากหลักการเขียนบทภาพยนตร์หรืออาศัยการอ่านนิยายบ่อยๆ เกมสมัยใหม่หลายเกมในปัจจุบันมีส่วนของเนื้อเรื่องไม่ว่าเกมนั้นจะเล็กหรือใหญ่ก็ตาม ซึ่งเป็นเรื่องที่ดีในการออกแบบ

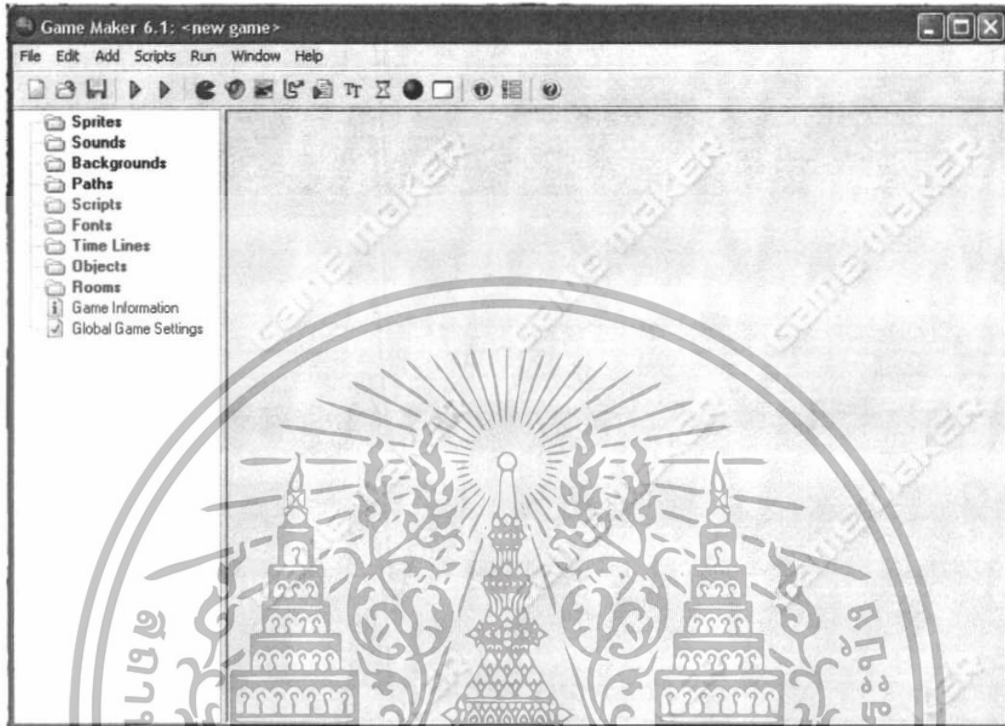
อย่างไรก็ดี ส่วนของ Storytelling นั้นเป็นส่วนที่เสริมเข้ามาเพื่อให้เกมมีรรถรสเพิ่มขึ้น นักออกแบบไม่ควรมุ่งเน้นในส่วนนี้มากเกินไปจนลืมเรื่องของ Game Play เพราะถ้าเกมปราศจากการเล่นที่ดีแล้วคงเรียกว่าเป็น “เกมที่ดี” ไม่ได้

ปัจจัยที่กล่าวมาทั้งหมดเป็นเพียงส่วนหนึ่งเท่านั้น โดยแต่ละหัวข้อสามารถที่จะแยกย่อยออกไปได้อย่างมากมาย นักออกแบบควรจะเรียนรู้และแสวงหาคำตอบด้วยประสบการณ์ของตัวเอง ต้องพยายามผนึกเอาปัจจัยต่างๆ ที่ได้กล่าวไปแล้ว มารวมกันในสัดส่วนที่เหมาะสมตามลักษณะเกมที่เราก่อแบบขึ้นมา ควรมีการสังเกตและวิเคราะห์เกมต่างๆ ว่ามีปัจจัยบ้างที่ทำให้เกมนั้นเป็นที่นิยมได้ รวมถึงเกมที่ไม่ได้รับความนิยมว่าเป็นเพราะสาเหตุใด เพื่อที่นักออกแบบเกมจะได้สร้างสรรค์สิ่งที่ดียิ่งขึ้นไป โดยดูเกมทั้งในอดีตและปัจจุบันเป็นบทเรียน

สุดท้าย เกมที่มีความสมดุลและลงตัวในทุกๆ ด้าน ไม่ว่าจะเป็นเรื่องของการเล่นหรือในแง่ของการตลาด จะเป็นเกมที่ประสบความสำเร็จได้อย่างแน่นอน

2.2 เครื่องมือภายในโปรแกรม Game Maker

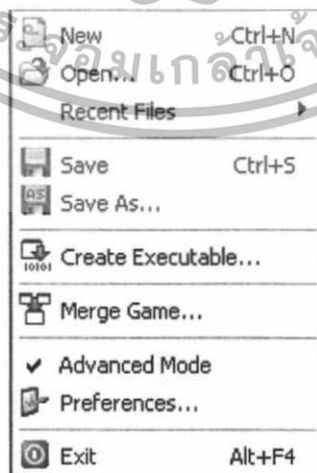
โปรแกรมนี้โดยภาพรวมจะมีคำสั่งที่อยู่ในแถบเมนูบาร์ด้านบน และคำสั่งบริเวณริชอร์ด(กรอบด้านซ้ายของโปรแกรมสำหรับเก็บริชอร์ดต่างๆที่จะใช้ในโปรแกรม)



รูปที่ 2.1 รูปหน้าจอหลักในโปรแกรม Game Maker

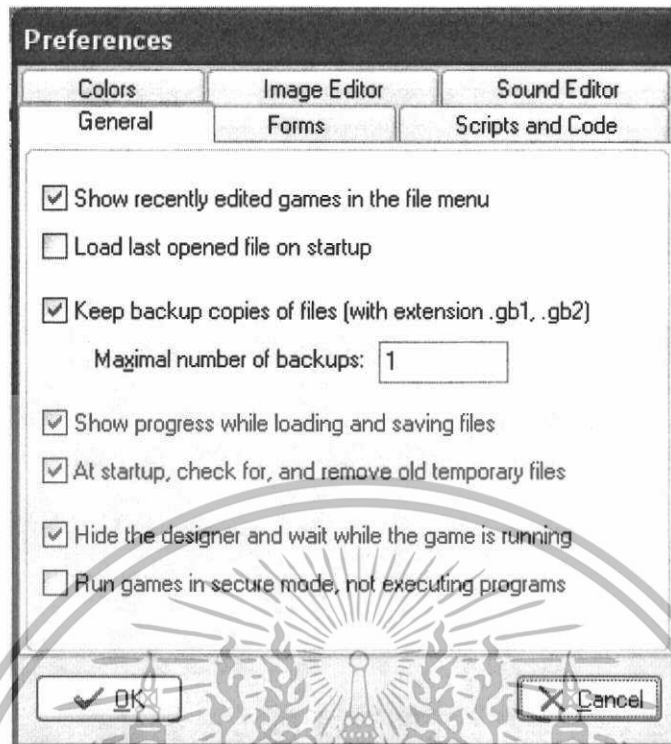
2.2.1 คำสั่งเมนูบาร์

2.2.1.1 File



รูปที่ 2.2 รูปเมนู File

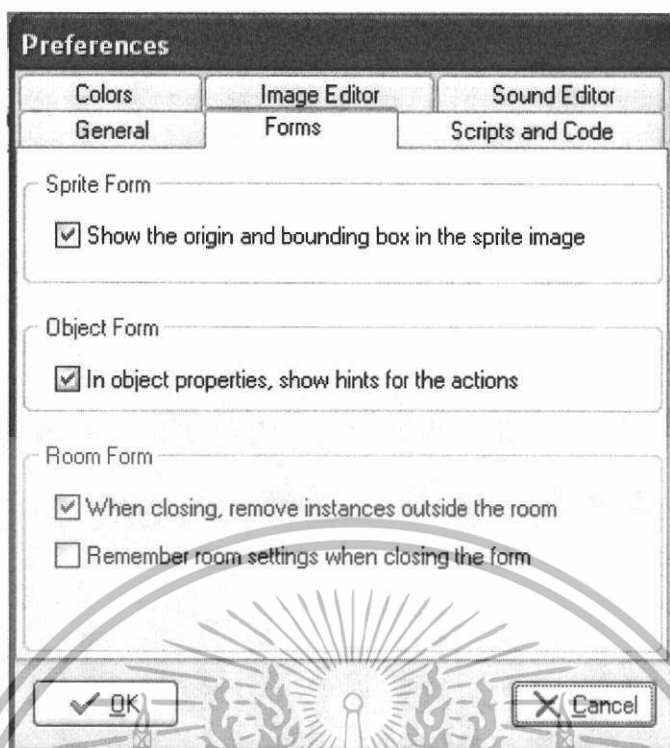
- 1) **New** ใช้เพื่อเริ่มทำโปรเจกต์ใหม่
- 2) **Open...** ใช้เปิดไฟล์โปรเจกต์เกมที่เราได้เซฟเอาไว้
- 3) **Recent File** ใช้เปิดไฟล์ที่เราเคยทำไว้ เรียงตามลำดับล่าสุด
- 4) **Save** เซฟโปรเจกต์ที่เรา กำลังทำอยู่ทับไฟล์เดิมที่ตั้งชื่อไว้แล้ว (ถ้ายังไม่ได้ตั้งชื่อ จะต้องตั้งชื่อโปรเจกต์ก่อน)
- 5) **Save as...** เซฟโปรเจกต์ที่เรา กำลังทำอยู่เป็นชื่อใหม่
- 6) **Create Executable...** ใช้สร้างไฟล์ EXE เมื่อเราสร้างเกมเสร็จเรียบร้อยแล้ว เพื่อนำไฟล์นี้ไปเปิดเล่นได้ทุกเครื่อง โดยที่เกมของเราไม่จำเป็นต้องใช้โปรแกรมใดๆ ในการช่วยเปิด
- 7) **Marge Game...** นำโปรเจกต์เกม (.GM6) อื่นๆ ที่ทำไว้แล้วมารวมกับโปรเจกต์ที่เรา กำลังทำอยู่
- 8) **Advance Mode** ให้เช็ถูกเพื่อแสดงโหมด Advance เครื่องมือต่างๆ จะถูกแสดงขึ้นมาให้เราใช้อย่างครบถ้วน ถ้าไม่ได้เช็ถูก เครื่องมือบางอย่างจะไม่สามารถแสดงได้ (แนะนำให้เช็ถูกไว้)
- 9) **Preferences...** ใช้สำหรับตั้งค่าต่างๆ เพื่อให้เราทำงานสะดวกยิ่งขึ้น ประกอบไปด้วย
 - **General**
 - 1) **Show recently edited game in the file menu** เช็ถูกเพื่อแสดงชื่อไฟล์โปรเจกต์ที่เราทำเมนู recent file
 - 2) **Load last opened file on startup** เช็ถูกเพื่อให้ตอนเปิดโปรแกรมจะโหลดโปรเจกต์เดิมที่เราทำไว้ครั้งที่แล้วขึ้นมาอัตโนมัติ
 - 3) **Keep backup copies of files** เช็ถูกเพื่อสร้างไฟล์แบ็คอัพขึ้นมาโดยอัตโนมัติ เมื่อเวลาโปรเจกต์เราเสียหาย สามารถเปิดไฟล์แบ็คอัพแทนได้ (สังเกตนามสกุล .gb1, .gb2 เป็นต้น) Maximal number of backups: สามารถกำหนดจำนวนไฟล์ที่เราจะแบ็คอัพไว้ได้(1-9)
 - 4) **Show progress while loading and saving files** เช็ถูกเพื่อแสดงสถานะในการโหลดข้อมูล ขณะกำลังโหลดข้อมูล ขณะกำลังโหลดและกำลังเซฟไฟล์
 - 5) **As startup, check for, and remove old temporary files** เมื่อเริ่มโปรแกรมให้ตรวจสอบและลบไฟล์ temp เดิมที่เก็บไว้
 - 6) **Hide the designer and wait while the game is running**
 - 7) **Ran game in secure mode, not executing programs** เมื่อเช็ถูกจะไม่สามารถรันไฟล์อื่นๆ ได้เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นอย่างไม่ได้ตั้งใจ



รูปที่ 2.3 รูปหน้าจอคำสั่ง General

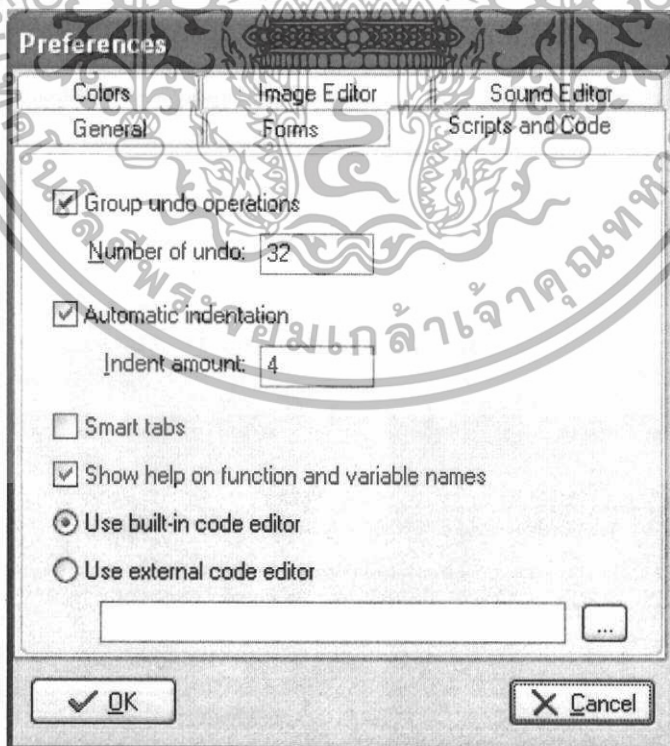
- Forms

- 1) **Show the origin and bounding box in the sprite image** เช็คลูกเพื่อแสดงตำแหน่งจุด origin หรือจุดกำเนิดของภาพ sprite (ภาพเคลื่อนไหวหรือไม่เคลื่อนไหวที่ใช้เป็นภาพกราฟิก) ในเกม
- 2) **In object properties, show hints for actions** เช็คลูกเพื่อแสดงข้อมูลต่างๆขณะเมาส์อยู่บนไอคอนที่ลากเข้ามาในช่อง action (จะกล่าวถึงในตอนต่อไป)
- 3) **When closing, remove instances outside the room** เช็คลูกเพื่อให้โปรแกรมถูกทำลาย instance (วัตถุที่เราใช้ภายในเกมอาจมีภาพ sprite หรือไม้ก็ได้) ที่อยู่ภายใน room (ส่วนหน้าของหน้าต่างที่เรากำหนดขึ้นมาใช้ในเกมนการรัน) ขณะเราปิด room นั้นๆ
- 4) **Remember room setting when closing the form** เช็คลูกเพื่อให้โปรแกรมจดจำการเช็คค่าต่างๆ เมื่อมีการปิด room นั้นๆ



รูปที่ 2.4 รูปหน้าจอคำสั่ง Forms

Scripts and Code

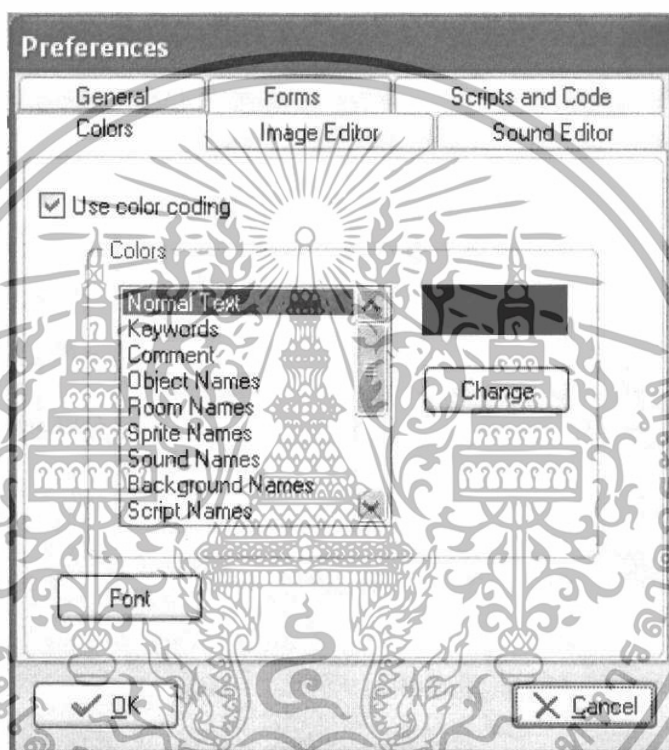


รูปที่ 2.5 รูปหน้าจอคำสั่ง Scripts and Code

สำหรับ Advance User สามารถแก้ไขสคริปต์ โดยเรียกโปรแกรมจากภายนอกได้

- Colors

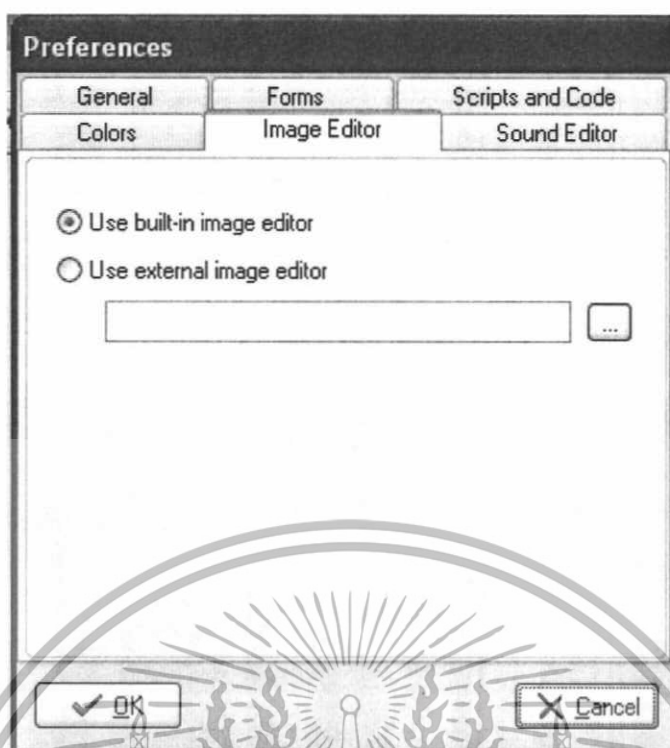
เป็นหน้าต่างค่าสีของตัวอักษรให้แตกต่างกันตามประเภทในการเขียนสคริปต์ เพื่อให้สะดวกในการมองเห็น



รูปที่ 2.6 รูปหน้าต่างคำสั่ง Colors

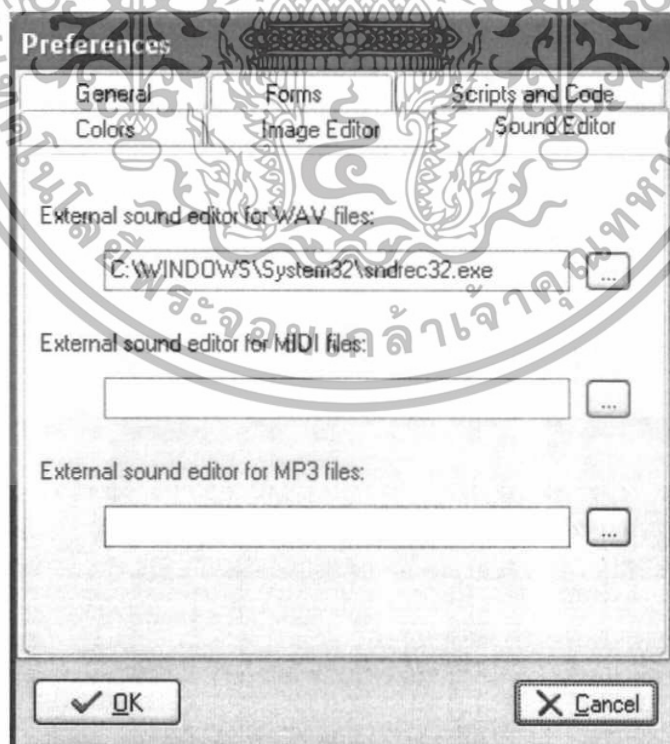
- Image Editor

ภายในโปรแกรม Game Maker นี้มีตัว Editor เพื่อสามารถดัดแปลงแก้ไขรูปภาพที่นำเข้ามาได้ แต่ถ้าต้องการเรียกโปรแกรมกราฟิกอื่นๆ เช่น Photoshop ฯลฯ จากภายนอก ก็สามารถทำได้โดยเลือก Use external image editor แล้วจึงเรียกพาไปที่โปรแกรมที่ต้องการ



รูปที่ 2.7 รูปหน้าจอคำสั่ง Image Editor

Sound editor



รูปที่ 2.8 รูปหน้าจอคำสั่ง Sound editor

ภายในโปรแกรม Game maker นั้นไม่มีโปรแกรมที่จะสามารถที่จะแก้ไขเสียงได้จึงต้องเรียกใช้โปรแกรมภายนอกเท่านั้น ไฟล์เสียงจะแบ่งออกเป็นสามประเภท wav, midi, mp3 สามารถเลือกพาไปทีโปรแกรมที่เราต้องการได้

2.2.1.2 Edit



รูปที่ 2.9 รูปเมนู Edit

- 1) **Add (Resource)** ใช้สำหรับเพิ่มรีซอร์สต่างๆ เช่น add sprite จะเป็นการเพิ่มรูปภาพที่จะใช้ในเกมเข้ามาในตัวโปรแกรม และถูกจัดเก็บไว้ที่โฟลเดอร์ sprite, add sounds จะเป็นการเพิ่มรายการเสียงที่จะใช้ในเกมเข้ามาอยู่ในโฟลเดอร์ Sounds เป็นต้น วิธีการเพิ่มรีซอร์สนี้สามารถคลิกขวาที่โฟลเดอร์นั้นๆ และเลือก Add... ได้ทันที สะดวกกว่าวิธีที่เลือกจากแถบ Edit มาก
- 2) **Duplicate** เมื่อมีรีซอร์สอยู่ภายในโฟลเดอร์แล้ว เราสามารถทำสำเนาได้ โดยเลือกคำสั่งนี้ หรือคลิกขวาที่รีซอร์ส แล้วเลือก Duplicate จะสะดวกมากกว่าเลือกจากแถบ Edit
- 3) **Add Group** ใช้สำหรับสร้างโฟลเดอร์กลุ่มขึ้นมาใหม่ สามารถตั้งชื่อได้ตามต้องการช่วยให้รีซอร์สของเราถูกจัดเก็บเป็นหมวดหมู่มากยิ่งขึ้น สะดวกต่อการทำงาน สามารถคลิกขวาที่รีซอร์สเพื่อเลือกคำสั่งได้
- 4) **Delete** ใช้ลบรีซอร์สที่เราไม่ต้องการทิ้งไป สามารถคลิกขวาที่รีซอร์สนั้นเพื่อเลือกคำสั่งได้
- 5) **Rename** ใช้เปลี่ยนชื่อรีซอร์สเป็นชื่อใหม่ สามารถคลิกขวาที่รีซอร์สนั้นเพื่อเลือกคำสั่งได้

- 6) **Properties...** เปิดหน้าต่างคุณสมบัติขึ้นมาเพื่อจัดการใส่ค่าต่างๆลงในรีซอร์สนั้นๆ สามารถคลิกขวาที่รีซอร์สนั้นเพื่อเลือกคำสั่งได้
- 7) **Find Resource...** ให้ค้นหาทรัพยากรในกรณีที่มีเป็นจำนวนมากโดยพิมพ์ชื่อรีซอร์สลงไป แล้วคลิกปุ่ม OK
- 8) **Expand Resource Tree** ใช้สำหรับเปิดโฟลเดอร์แสดงให้เห็นรีซอร์ส ทั้งหมดที่มีอยู่
- 9) **Collapse Resource Tree** ใช้ปิดโฟลเดอร์ทั้งหมด
- 10) **Show object information** แสดงข้อมูลต่างๆของทุกอ็อบเจกต์ ซึ่งอ็อบเจกต์ก็คือวัตถุที่ Add เข้ามาเพื่อนำไปใช้ในเกม ข้อมูลต่างๆ ในขณะนั้น เช่น อ็อบเจกต์ นั้นๆใช้ภาพ sprite อะไรที่มี event ไบบ้าง ซึ่งจะกล่าวถึงในลำดับต่อไป

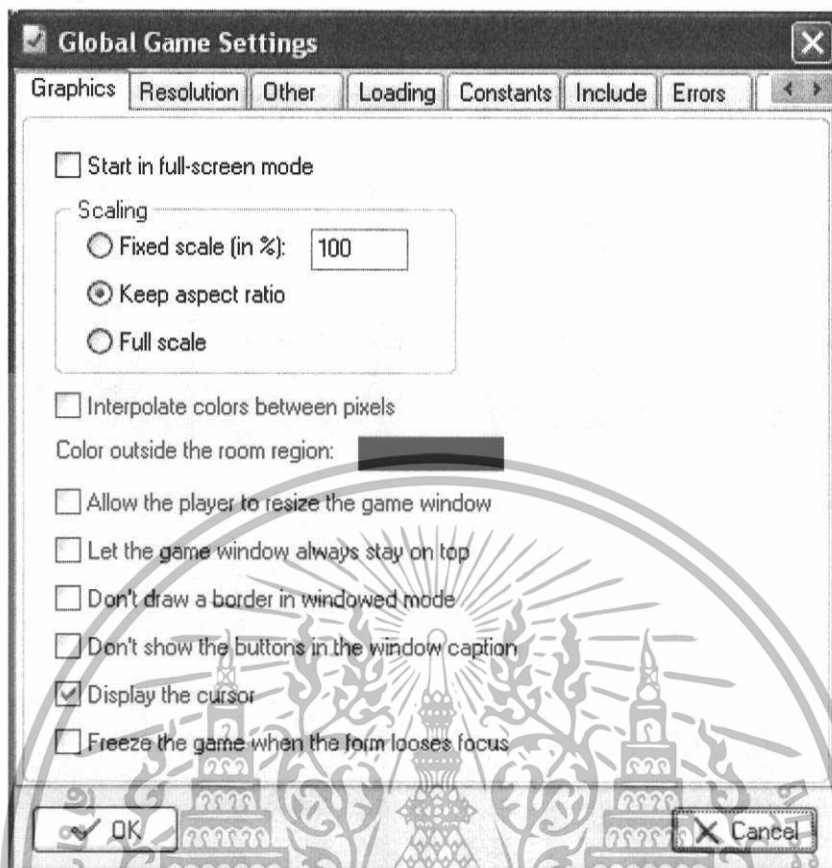
2.2.1.3 Add



รูปที่ 2.10 เมนู Add

- 1) **Add...** (รีซอร์ส 9 อย่าง) ใช้สำหรับเพิ่มรีซอร์ส ดังกล่าวเข้ามาในโปรแกรม สามารถคลิกขวาที่โฟลเดอร์ของรีซอร์สนั้นเพิ่มได้เช่นกัน
- 2) **Change Game information** ใช้แก้ไขหน้าต่างข้อมูลเกมของเรา สามารถเขียนวิีเล่นเกมโดยการกดปุ่ม F1 (ในกรณีที่ตั้งค่าไว้)
- 3) **Change Global Game Setting** ใช้สำหรับตั้งค่าต่างๆของเกมเราให้เป็นไปตามที่เราต้องการ ประกอบไปด้วยแท็บต่างๆ ดังนี้

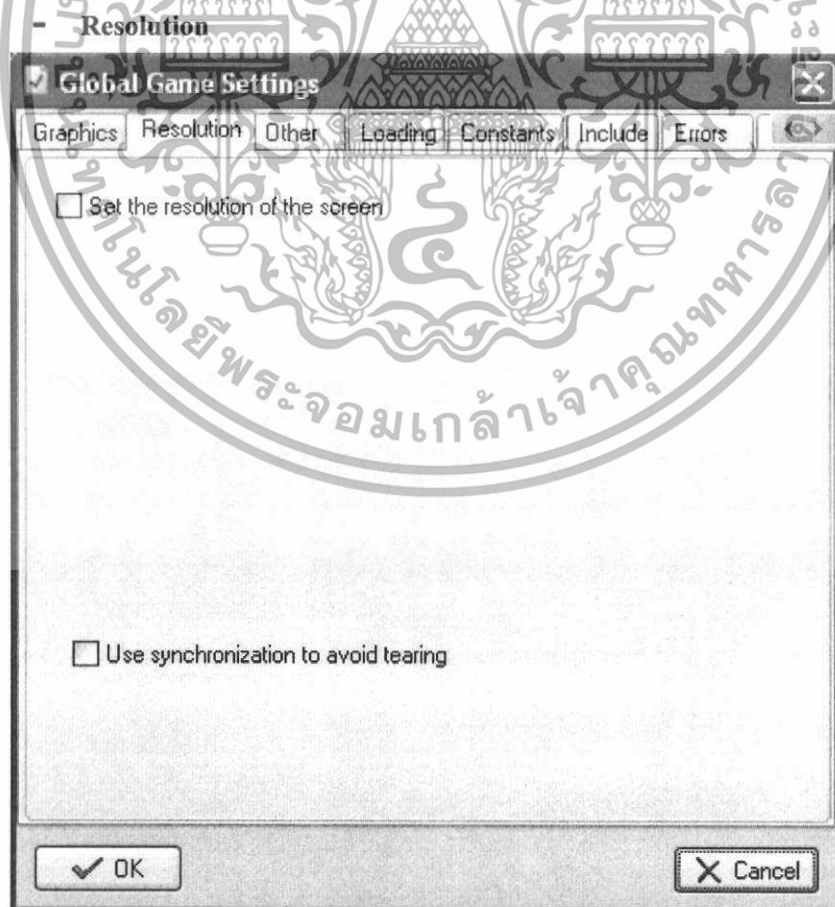
- Graphics



รูปที่ 2.11 รูปหน้าจอคำสั่ง Graphics

- 1) **Start in full-screen mode** เมื่อใช้คลิก เกมจะรันขึ้นมาแบบเต็มจอ
- 2) **Scaling**
 - **Fix scale (in %)** ใช้กำหนดขนาดตายตัว 100% คือถ้าปรกติที่เราตั้งค่าไว้ ถ้ามีการเปลี่ยนขนาดหน้าจอโดยผู้เล่น ขนาดของเกมก็จะไม่ขยายหรือลดลงไปตามขนาดของกรอบหน้าจอ
 - **Keep aspect ratio** เมื่อมีการย่อหรือขยายหน้าจอของเกมจะรักษาสัดส่วนเดิมที่เราได้ตั้งไว้อยู่ตลอดเวลา ทำให้สัดส่วนของรูปภาพในเกมไม่เพี้ยนไปจากเดิม แต่จะเปลี่ยนแปลงเพียงแค่นั่น
 - **Full scale** เมื่อมีการย่อหรือขยายหน้าจอของเกม จะขยายหรือย่อตามขนาดของหน้าจอทุกประการโดยไม่สนใจสัดส่วนเดิมที่เราตั้งไว้
- 3) **Interpolate colors between pixels** ขณะที่รันเกม จะทำให้ภาพ sprite, background, titles ดู smooth ขึ้นเมื่อภาพนั้นมีการหมุนหรือขยาย แต่ภาพจะเบลอล็กน้อย

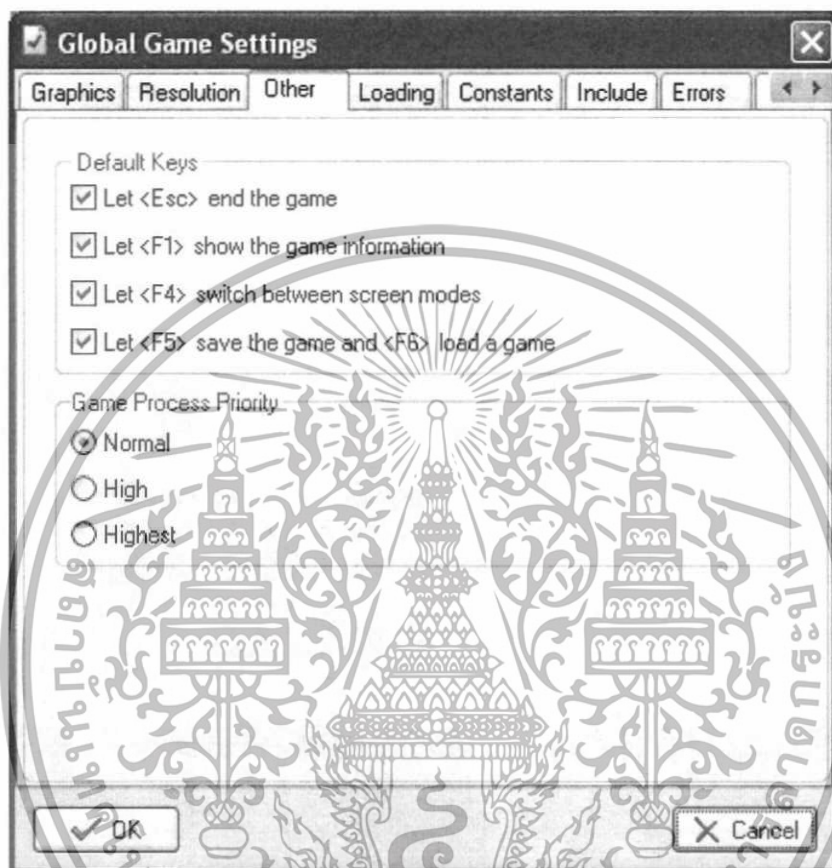
- 4) **Colors outside the room region** เมื่อมีการขยายขนาดของหน้าจอให้ใหญ่กว่าสัดส่วนของขนาดเกม เราจะมองเห็นพื้นหลังของหน้าจอ ซึ่งสามารถเลือกสีพื้นหลังได้จากตัวเลือกนี้
- 5) **Allow the player to resize the game window** อนุญาตให้ผู้เล่นเกมสามารถเปลี่ยนขนาดของหน้าจอได้
- 6) **Let the game window always stay on top** ทำให้อาจของเกมน้อยด้านบนสุดเหนือหน้าจออื่นทุกอันตลอดเวลา
- 7) **Don't draw a border in windowed mode** เมื่อเช็คลูก หน้าต่างของเกมจะไม่แสดงขอบหน้าจอ ผู้เล่นจะไม่สามารถปรับเปลี่ยนโดยการดึงที่ขอบหน้าจอได้
- 8) **Don't show the buttons in the window caption** หน้าจอของเกมจะไม่แสดงปุ่มที่ขอบของหน้าจอ ซึ่งปกติจะมีอยู่ด้วยกัน 3 ปุ่ม คือ ย่อลงมาเก็บ ย่อ/ขยายขนาด ปิดโปรแกรม รวมถึงไม่แสดงไอคอนที่ขอบหน้าจอด้วย
- 9) **Display the cursor** เมื่อเช็คลูก จะแสดงเคอร์เซอร์เมื่อรันเกม
- 10) **Freeze the game when the from loses focus** เมื่อผู้เล่นเลือกไปที่โปรแกรมอื่นๆขณะที่รันเกมอยู่ เกมจะหยุดรอผู้เล่นจนกว่าผู้เล่นจะเลือกหน้าจอเกมเหมือนเดิม



รูปที่ 2.12 รูปหน้าจอคำสั่ง Resolution

สามารถกำหนด Color Depth ของเกม สามารถปรับความละเอียดของหน้าจอที่เหมาะสมกับเกม และตั้งความถี่ในการรีเฟรชภาพต่อวินาทีได้

- Other

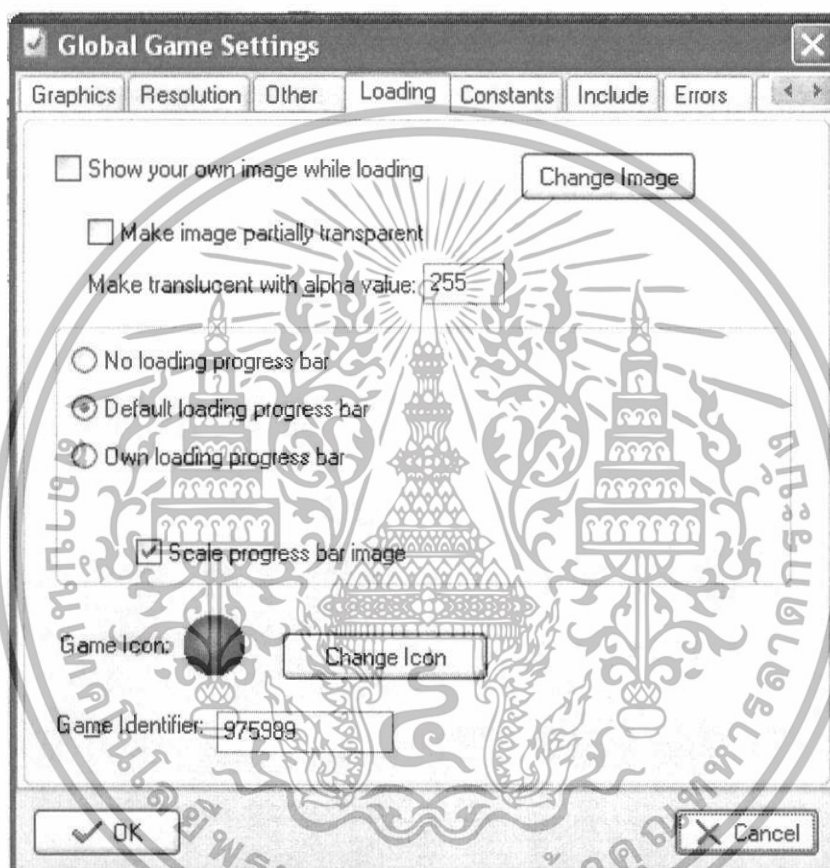


รูปที่ 2.13 รูปหน้าจอคำสั่ง other

- 1) Let <Esc> end the game เมื่อเช็คถูก เวลาเล่นเกมกดปุ่ม Esc จะออกจากโปรแกรม
- 2) Let <F1> show the game information เมื่อเช็คถูก เวลาเล่นเกมกดปุ่ม F1 จะเป็นการแสดงข้อมูลของเกมที่เราเขียนไว้
- 3) Let <F4> switch between screen mode เมื่อเช็คถูก เวลาเล่นเกม กดปุ่ม F4 จะเป็นการสลับไปเป็นแบบเต็มจอ
- 4) Let <F5> save the game and <F6> load a game เมื่อเช็คถูก เวลาเล่นเกม กดปุ่ม F5 จะเป็นการเซฟเกมในขณะนั้น กดปุ่ม F6 จะเป็นการโหลดเกมกลับไปเล่นตรงจุดที่เราได้เซฟไว้ในครั้งที่แล้ว

- 5) **Game process priority** เป็นการกำหนดระดับความสำคัญในการแบ่งส่วนของการประมวลผลของระบบวินโดวส์ จะทำให้เกมของเรามีในระดับใด มีให้เลือกสามระดับคือ ธรรมดา สูง และสูงที่สุด เกมของเราจะรับการประมวลผลก่อน ซึ่งจะทำให้เกมของเราประมวลผลได้เร็วขึ้น

- **Loading**



รูปที่ 2.14 รูปหน้าจอคำสั่ง loading

คุณสามารถเปลี่ยนภาพตอนโหลดเกมได้ สามารถกำหนดค่าความโปร่งใสของภาพได้ สามารถเปลี่ยนรูปบาร์ที่ใช้บอกสถานะขณะโหลดและสามารถเปลี่ยนไอคอนได้ 32x32 เท่านั้น

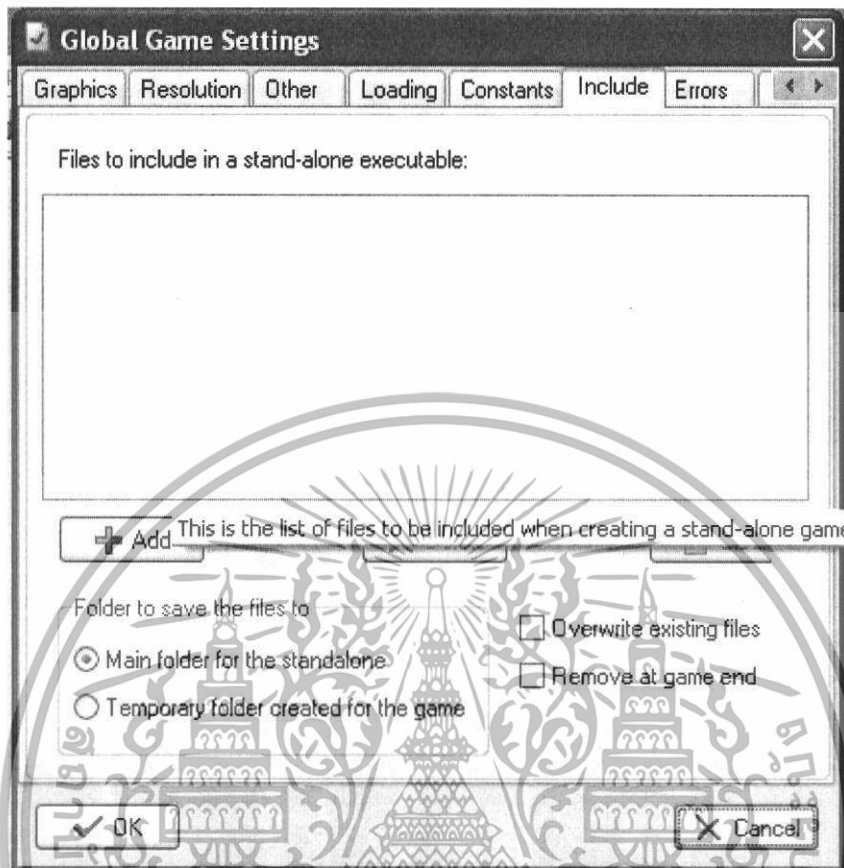
- Constants



รูปที่ 2.15 รูปหน้าจอคำสั่ง Constants

สามารถกำหนดค่าคงที่ภายในเกมเพิ่มเติมได้

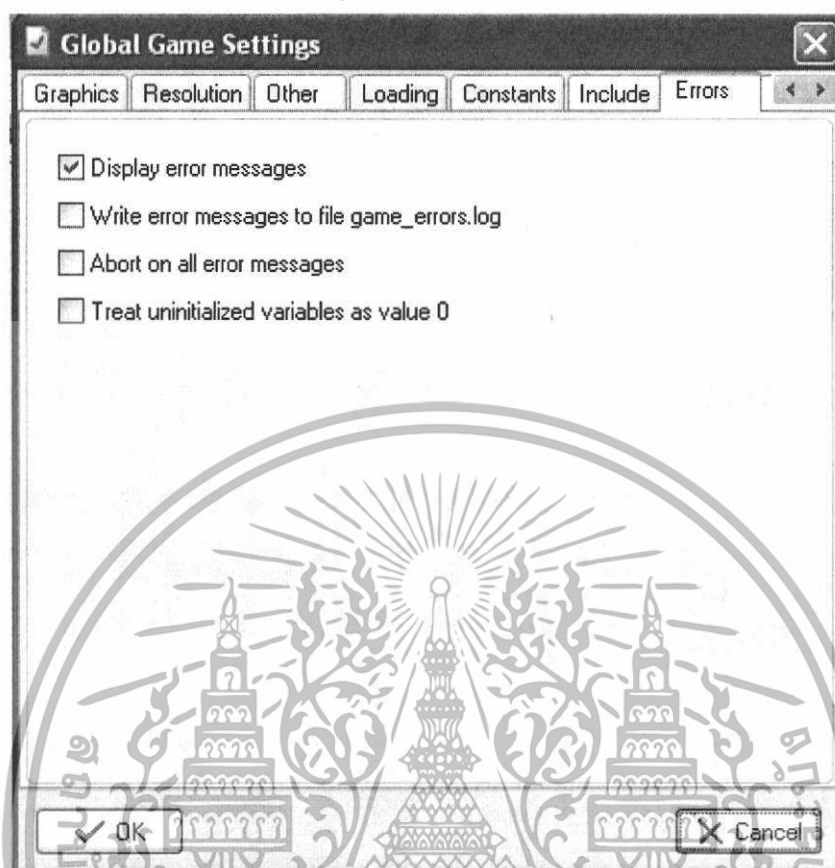
- Include



รูปที่ 2.16 รูปหน้าจอคำสั่ง Include

สามารถใส่ไฟล์อื่นๆ เพิ่มเติมลงไปได้ เช่น ไฟล์วิดีโอ เมื่อสร้างเป็น EXE จะมีเพียงแค่ไฟล์เดียว และเมื่อรันไฟล์ โปรแกรมจะเซฟไฟล์ทั้งหมดไปที่ Main folder for the standalone คือโฟลเดอร์ของไฟล์ที่รันเกมนั้น หรือ Temporary folder created for the game คือสร้าง temp โฟลเดอร์ขึ้นมาแล้วเซฟไฟล์ต่างๆไว้ที่นั่น ถ้าต้องการให้ลบไฟล์ทั้งหมดทิ้งหลังจากปิดเกม ให้เลือก Remove at game end ถ้าในโฟลเดอร์เป้าหมายนั้นมีไฟล์ชื่อนั้นอยู่แล้วโดยปกติโปรแกรมจะไม่เขียนทับลงไป แต่ถ้าอยากให้เซฟทับลงไปก็ให้เลือก Overwrite existing files

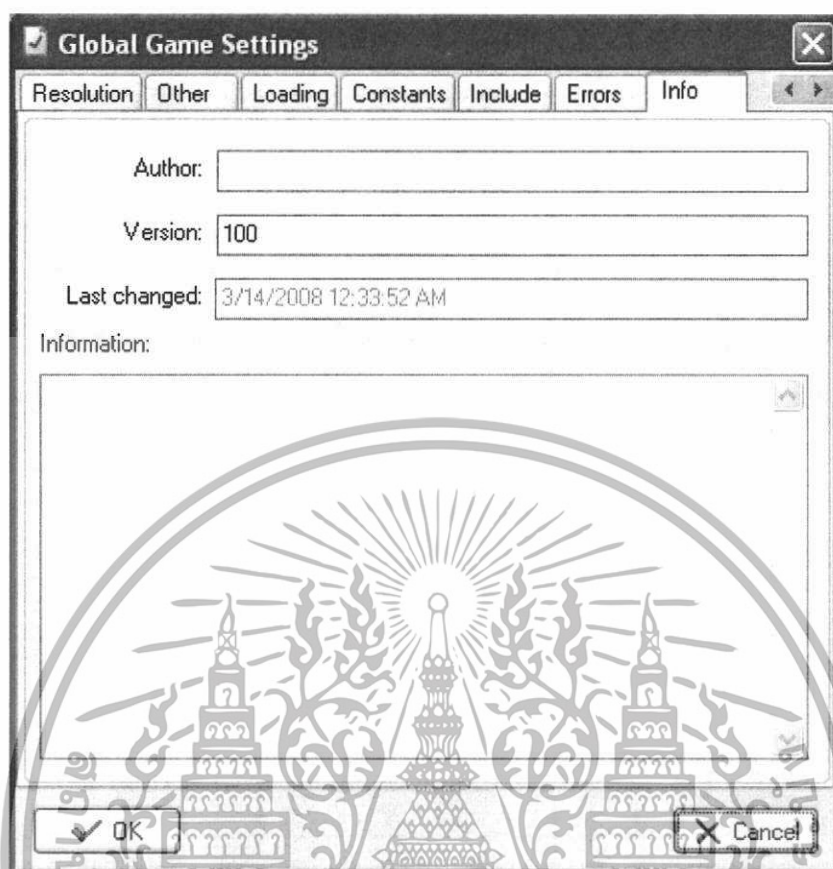
- Error



รูปที่ 2.17 รูปหน้าจอคำสั่ง Error

- 1) **Display error messages** ขณะรันเกม เมื่อโปรแกรมตรวจพบข้อผิดพลาด จะแสดงกล่องข้อความขึ้นมา พร้อมแสดงข้อมูลความผิดพลาดนั้นๆ
- 2) **Write error messages to file game_errors.log** สามารถนำข้อมูลข้อผิดพลาดเก็บไว้ในไฟล์ชื่อ game_errors ซึ่งจะถูกเซฟให้อยู่ในโฟลเดอร์เดียวกับไฟล์ที่เรารันเกม
- 3) **Abort on all error messages** เมื่อเจอข้อผิดพลาด ให้ออกจากเกมเสมอ
- 4) **Treat uninitialized variables as value 0** เมื่อโปรแกรมเจอตัวแปรที่ไม่รู้จัก ให้เซตค่าเป็น 0 โดยอัตโนมัติ

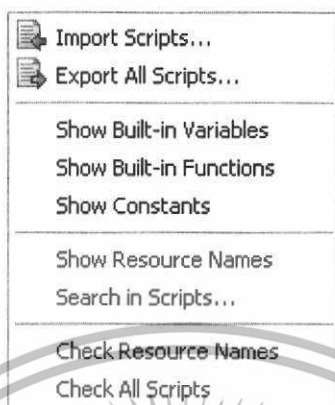
- Info



รูปที่ 2.18 รูปหน้าจอคำสั่ง Info

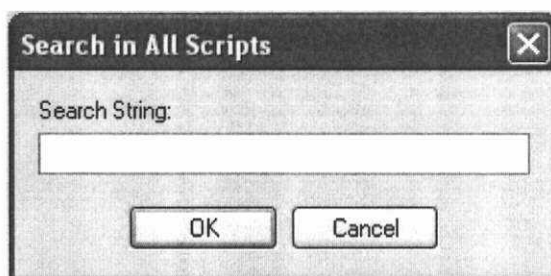
ใช้สำหรับใส่ชื่อผู้สร้าง ข้อมูลเวอร์ชันของเกม และข้อมูลที่จำเป็นต่างๆซึ่งมีความสำคัญในกรณีที่เกมของเราพัฒนาไปหลายเวอร์ชัน

2.2.1.4 Script



รูปที่ 2.19 เมนู Script

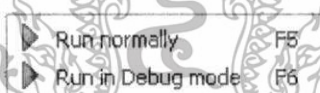
- 1) **Import Scripts...** นำสคริปต์เข้ามาจากภายนอกเพื่อใช้ในเกม (สำหรับผู้ที่ศึกษาเรื่องการเขียนสคริปต์ด้วยภาษา GML แล้ว)
- 2) **Export All Scripts...** นำสคริปต์นั้นที่ออกไปภายนอก เพื่อเรียกเข้าไปใช้ในโปรเจกต์อื่นๆ (สำหรับผู้ที่ศึกษาเรื่องการเขียนสคริปต์ด้วยภาษา GML แล้ว)
- 3) **Show Build – In Variables** แสดงตัวแปรทั้งหมดที่โปรแกรมรู้จักอยู่แล้ว (ห้ามตั้งชื่อตัวแปรใดๆซ้ำกับชื่อเหล่านี้ เนื่องจากเป็นคำสงวน)
- 4) **Show Build – In Functions** แสดงฟังก์ชันทั้งหมดที่โปรแกรมรู้จักอยู่แล้ว (ห้ามตั้งชื่อตัวแปรใดๆซ้ำกับชื่อเหล่านี้ เนื่องจากเป็นคำสงวน)
- 5) **Show Constants** แสดงค่าคงที่ทั้งหมดที่โปรแกรมรู้จักอยู่แล้ว (ห้ามตั้งชื่อตัวแปรใดๆซ้ำกับชื่อเหล่านี้ เนื่องจากเป็นคำสงวน)
- 6) **Show Resource Names** แสดงรายชื่อของรีซอร์สทั้งหมดที่เราสร้างขึ้นและสามารถเลือกเพื่อแก้ไขได้
- 7) **Search In Scripts...** ค้นหาที่ต้องการหาในสคริปต์ทั้งหมด และสคริปต์ที่มีคำนั้นจะถูกแสดงขึ้นมาพร้อมกับบอกตำแหน่ง สามารถเลือกเพื่อแก้ไขได้



รูปที่ 2.20 รูปหน้าจอคำสั่ง Search in Scripts

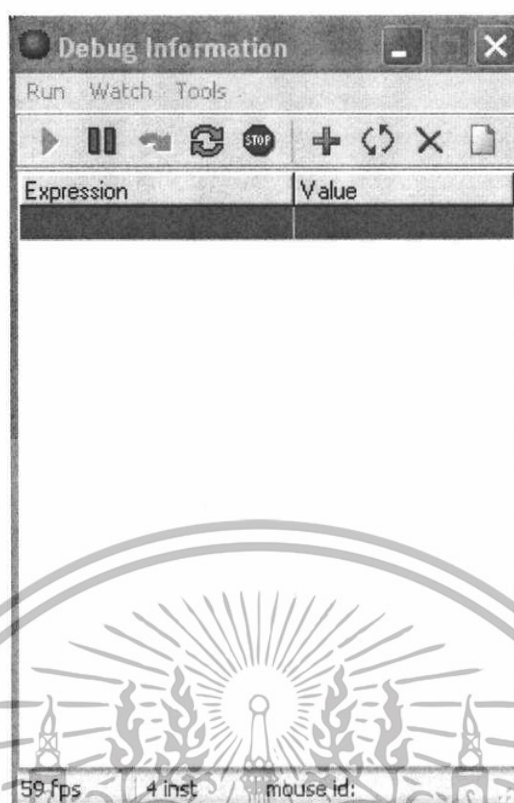
- 8) **Check Resource Names** ตรวจสอบชื่อของรีซอร์สทั้งหมดว่าชื่อใดไปตรงกับชื่อที่สงวนเอาไว้ เพราะถ้ามีชื่อใดตรงโปรแกรมจะฟ้อง error ทันทีที่เจอ (คำที่สงวนไม่สามารถจะใช้ตั้งชื่อตัวแปร ได้คือ keyword เช่น if, then, else, repeat, do, until, while, switch, case, al, other, self เป็นต้น หรือชื่อของตัวแปร ฟังก์ชัน คำคงที่ที่เป็น build-in ทุกตัว)
- 9) **Check All Script** ตรวจสอบสคริปต์ทั้งหมดที่มีอยู่ว่าสคริปต์ใดเขียนผิดไปจากโครงสร้างภาษา GML เมื่อโปรแกรมตรวจสอบเจอ จะปรากฏชื่อของสคริปต์นั้นๆ และตำแหน่งที่มีการเขียนผิดพลาด

2.2.1.5 Run



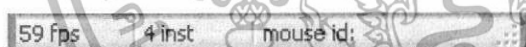
รูปที่ 2.21 รูปเมนู Run หลัก

- 1) **Run normally** ใช้รันเกมแบบธรรมดา
- 2) **Run in Debug mode** ใช้รันเกมแบบสามารถตรวจสอบค่าตัวแปรที่เราตั้งขึ้นมาได้อย่างอิสระ เมื่อถูกรันจะสังเกตเห็นหน้าต่าง Debug Information สามารถตรวจสอบได้ดังต่อไปนี้



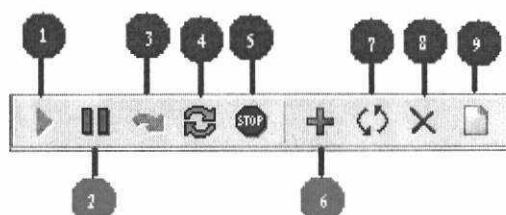
รูปที่ 2.22 รูปหน้าจอคำสั่ง Run in Debug mode

• บริเวณกรอบด้านล่าง



- fps แสดงอัตราความเร็วเป็นเฟรมต่อวินาที หรือ frame rate ปกติเราสามารถตั้งค่าได้ที่ room setting ถ้าค่าตกลงไปจะทำให้เกมกระตุกหรือช้าลง ซึ่งเกิดจากหลายสาเหตุ เช่น มีวัตถุในเกมมากเกินไป เพราะฉะนั้นวัตถุใดที่ไม่จำเป็นต้องใช้ต้องทำลายทิ้งให้หมด เช่น กระสุนที่ยิงออกนอกจอไปแล้วจำเป็นต้องทำลายทิ้งให้หมด โดยจำนวนวัตถุในเกมทั้งหมดจะสังเกตได้ในหัวข้อต่อไป
- inst วัตถุทั้งหมดที่มีภายในเกมหรือเรียกว่า instance ตรงนี้จะแสดงจำนวนทั้งหมดที่มีในเกม ถ้ามากเกินไปอาจทำให้เกมช้าลง
- mouse id เมื่อนำเมาส์ไปวางบน instance ใดๆ แล้วปรากฏค่า id ของ instance นั้นๆ ขึ้นมา instance ทุกตัวในเกมจะมีค่า id ไม่ซ้ำกันเลย แม้จะใช้ออบเจกต์เหมือนกันก็ตาม

- บริเวณไอคอน



รูปที่ 2.23 รูปบริเวณไอคอน

- 1) ปุ่ม **Run** ทำให้เกมดำเนินต่อไป
- 2) ปุ่ม **Pause** ทำให้เกมหยุดชั่วขณะ
- 3) ปุ่ม **Take a single step** ขณะที่เราหยุดเกม สามารถใช้ปุ่มนี้ทำให้ภาพเล่นทีละเฟรมได้
- 4) ปุ่ม **Restart** เริ่มรันเกมใหม่
- 5) ปุ่ม **End** จบเกม
- 6) ปุ่ม **Add a expression to watch** สามารถใส่ชื่อตัวแปรและค่าคงที่ที่เราต้องการดูค่า เมื่อใส่ชื่ออย่างถูกต้อง ชื่อจะปรากฏในกรอบด้านล่างพร้อมกับค่าของตัวแปรนั้นๆ เมื่อเราอยากตรวจสอบค่าตัวแปรที่เป็น local จะต้องเขียนอ้างอิงให้ถูกต้อง
- 7) ปุ่ม **Change the select watch the expression** ใช้เลือกแก้ไขชื่อตัวแปรที่เราใส่ลงไป ในกรอบด้านล่าง
- 8) ปุ่ม **Delete the select watch the expression** ใช้ลบตัวแปรเฉพาะตัวที่เราเลือกในกรอบด้านล่าง
- 9) ปุ่ม **Clear all watch the expression** ใช้ลบตัวแปรทั้งหมดที่เราใส่ลงไป ในกรอบด้านล่าง

● เมนูบาร์

Run Watch Tools

- Run เหมือนที่ได้กล่าวมาข้างต้นแล้ว



รูปที่ 2.24 รูปเมนู Run

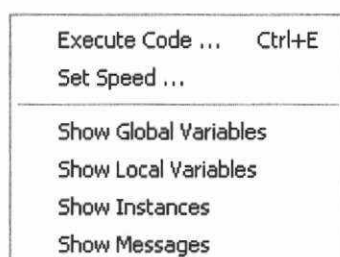
- Watch ส่วนใหญ่เหมือนที่ได้กล่าวมาข้างต้นแล้ว



รูปที่ 2.25 รูปเมนู Watch

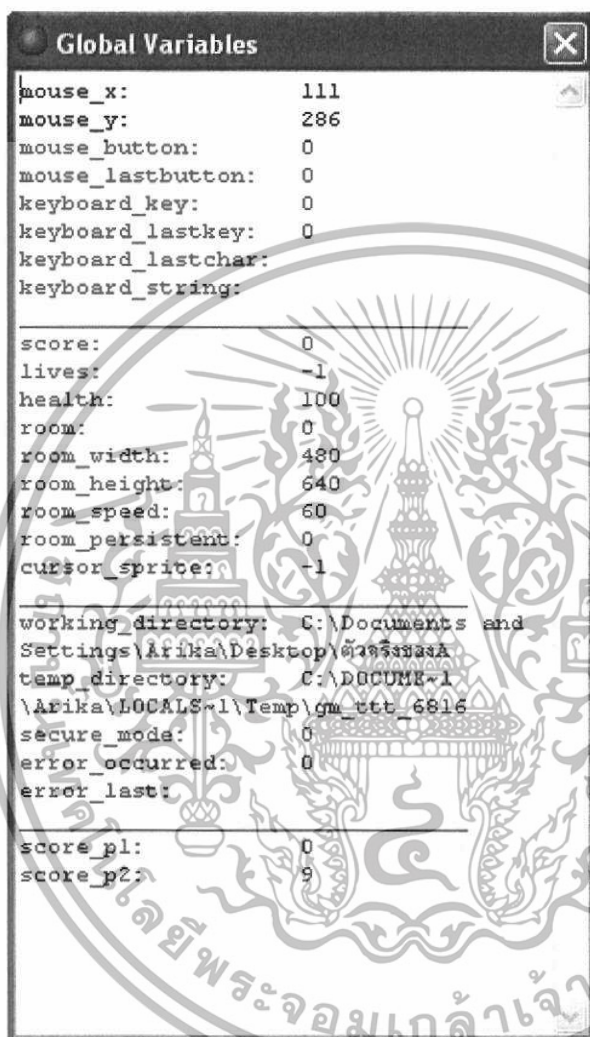
- 1) Load... ใช้โหลดไฟล์ text ที่เราเก็บชื่อตัวแปรเอาไว้
- 2) Save... ใช้เซฟชื่อตัวแปรทั้งหมดในกรอบด้านล่างเก็บไว้ไฟล์ text

- Tools



รูปที่ 2.26 รูปเมนู Tools

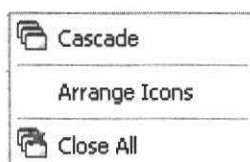
- 1) **Execute Code...** สามารถเขียนโค้ดลงไปเพื่อตรวจสอบได้
- 2) **Set Speed...** สามารถปรับเปลี่ยนความเร็วของเกมได้ หน่วยจะเป็น fps (เฟรมต่อวินาที)
- 3) **Show Global Variables** ใช้แสดงค่าคงที่หรือตัวแปรทั้งหมดที่เป็นตัวแปรแบบ global



รูปที่ 2.27 รูปหน้าจอคำสั่ง Global Variables

- 4) **Show Local Variables** ใช้แสดงค่าทุกค่าของ instance หรือออบเจกต์โดยสามารถใส่ชื่อของออบเจกต์หรือ instance_id
- 5) **Show Instances** ใช้แสดงวัตถุทุกชิ้นในเกม
- 6) **Show Messages** ใช้ฟังก์ชัน show_debug_message (string) สอดแทรกภายในโค้ดต่างๆ ที่เราเขียนขึ้นเพื่อตรวจสอบการรันที่ถูกต้อง

2.2.1.6 Window



รูปที่ 2.28 รูปเมนู Window

ใช้จัดรูปแบบของหน้าต่างให้เป็นระเบียบเรียบร้อยหรือปิดหน้าต่างทั้งหมด

2.2.1.7 Help



รูปที่ 2.29 รูปเมนู Help

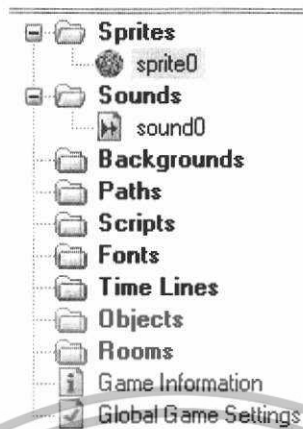
สามารถดู Help เพิ่มเติมได้ สามารถลงทะเบียนเพื่อขอใช้แบบเวอร์ชันเต็ม มีการเชื่อมโยงที่เว็บไซต์หลักของโปรแกรม Game Maker และสุดท้ายสามารถเวอร์ชันที่เรากำลังใช้อยู่ของ Game Maker ได้

ไอคอนด้านล่างคำสั่งที่เมนูบาร์



วิธีการใช้ไอคอนทั้งหมดอยู่ในเมนูบาร์ซึ่งได้อธิบายไปแล้ว การแสดงในรูปแบบไอคอนทำให้ผู้ใช้งานโปรแกรมเข้าใจง่ายในภาพที่แสดงบนไอคอน และสามารถใช้งานคำสั่งต่างๆ ได้อย่างสะดวก รวดเร็วยิ่งขึ้น

2.2.2 Resource

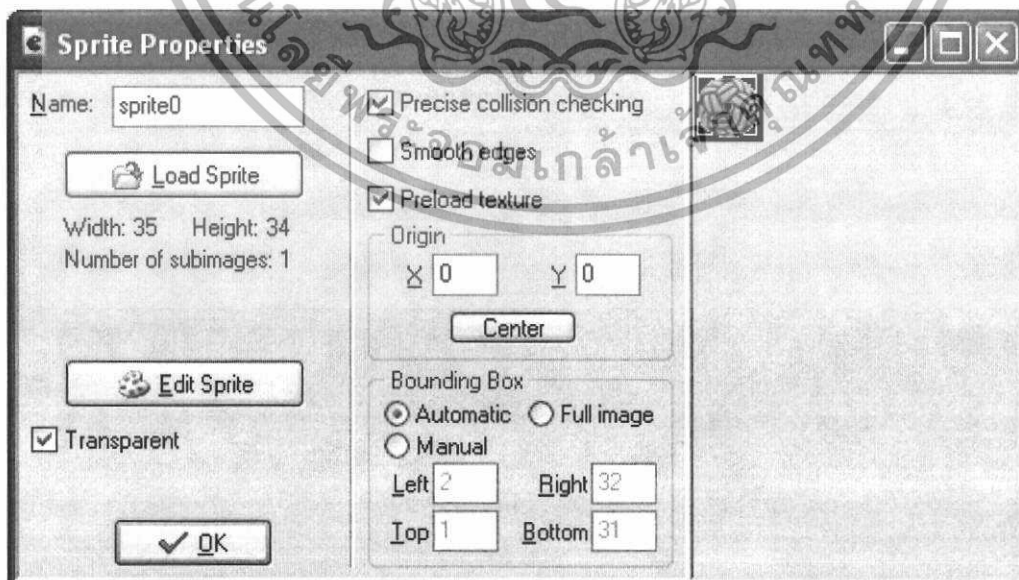


รูปที่ 2.30 รูปเมนู Resource ต่างๆ

ทางด้านซ้ายของจอ บริเวณนี้ค่อนข้างมีความสำคัญมาก เพราะจะเป็นพื้นที่ที่บรรจुरีซอร์สทั้งหมดของเกมไว้ โดยแบ่งแยกเป็นหมวดหมู่อย่างชัดเจนดังนี้

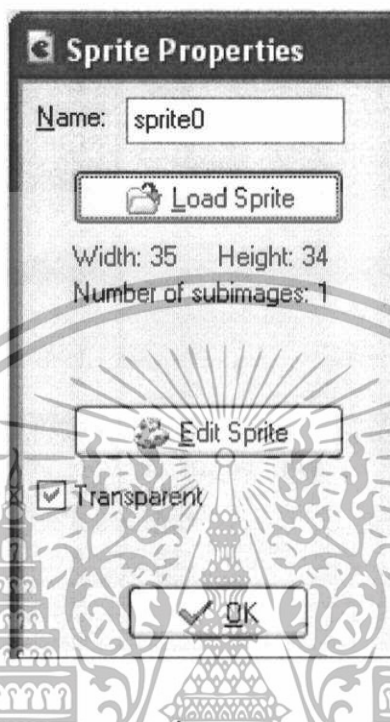
2.2.2.1 Sprites Resource

ใช้สำหรับนำภาพจากภายนอกเข้ามาเพื่อใช้ในเกมส์ ซึ่งภาพนั้นอาจเป็นภาพเคลื่อนไหวหรือไม่เคลื่อนไหวก็ได้ โดยจะเรียกภาพนี้ว่า “Sprite” เมื่อเราคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add Sprite จะปรากฏหน้าต่าง Sprite Properties ขึ้นมาดังรูปด้านล่าง เพื่อให้เราได้คุณลักษณะต่างๆลงไปด้วย



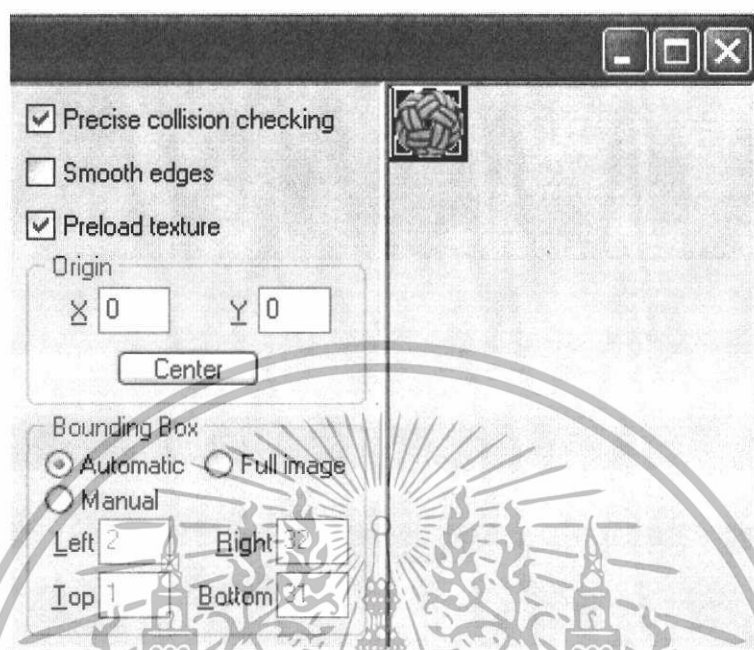
รูปที่ 2.31 รูปหน้าต่างคำสั่ง Sprites Resource

- **Sprite Properties**



รูปที่ 2.32 รูปหน้าจอคำสั่ง Sprite Properties ส่วนหน้า

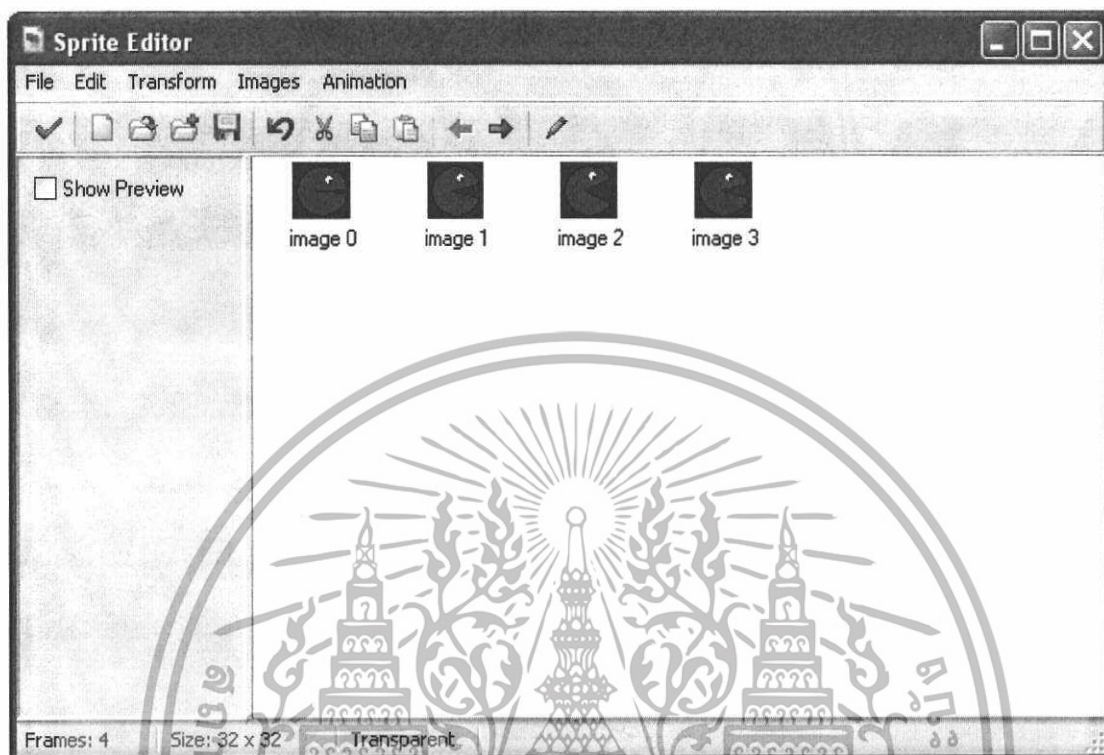
- 1) **Name** ตั้งชื่อให้กับรูปภาพ
- 2) **Load Sprite** ใช้โหลดภาพจากภายนอกเข้ามาเก็บไว้เพื่อใช้ในเกม สามารถโหลดภาพเข้ามาได้หลายนามสกุล
- 3) **Width** บอกขนาดความกว้างของภาพ มีหน่วยเป็น pixel
- 4) **Height** บอกขนาดความสูงของภาพ มีหน่วยเป็น pixel
- 5) **Number of subimages** บอกจำนวนภาพย่อยๆทั้งหมดในกรณีที่ภาพนั้นเป็นภาพเคลื่อนไหว
- 6) **Edit Sprite** ใช้สำหรับแก้ไขรูปภาพที่โปรแกรมมีมาให้สามารถทำได้หลายอย่าง รวมถึงการสร้างแอนิเมชันแบบง่ายๆ
- 7) **Transparent** เมื่อเช็คถูกจะทำให้ภาพไม่แสดงสีพื้นหลัง
- 8) **OK** ใช้ยืนยันการกำหนดค่าต่างๆ



รูปที่ 2.33 หน้าจอคำสั่ง Sprite Properties ส่วนหลัง

- 9) **Precise collision checking** กำหนดให้ภาพนี้สามารถตรวจสอบการชนกันระหว่างวัตถุได้
- 10) **Smooth edge** ทำให้ขอบของภาพดูเรียบขึ้น
- 11) **Preload texture** เมื่อเช็ถูกจะทำให้เมื่อตอนรันเกม ภาพทุกภาพที่เช็คคำสั่งนี้จะถูกโหลดเข้าไปเก็บไว้ในหน่วยความจำที่วีดิโอการ์ดเพื่อที่จะแสดงผลในเกม
- 12) **Origin** จุดกำเนิดของภาพ
- 13) **Center** ปรับจุดกำเนิดให้อยู่กลางภาพโดยอัตโนมัติ
- 14) **Bounding Box** ขอบที่จะบอกกรอบเขตในการตรวจสอบการชนกันของวัตถุ
- 15) **บริเวณพื้นด้านขวา** จะแสดงรูปภาพที่เรานำเข้ามาโดยจะแสดงจุด origin เป็นรูปกากบาทตามตำแหน่งที่เราตั้งค่าไว้

- **Sprite Editor** เมื่อคลิกที่ปุ่ม  **Edit Sprite** จะปรากฏหน้าต่างดังรูป



รูปที่ 2.34 รูปหน้าต่าง Sprite Editor

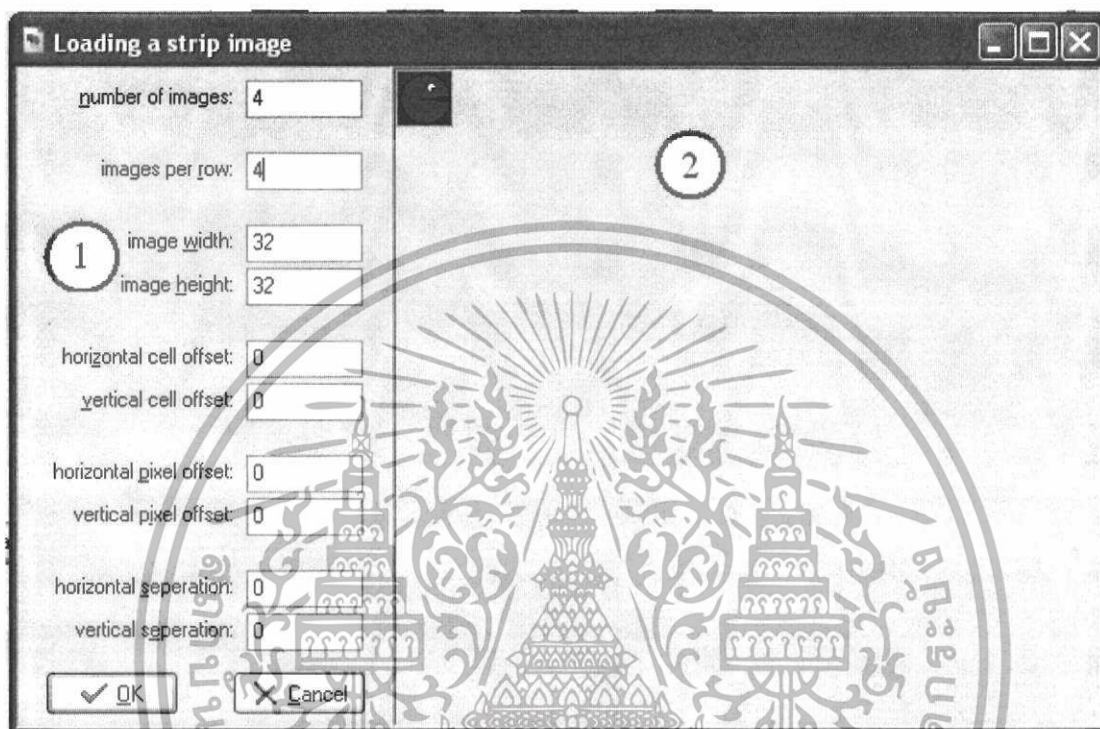
แถบเมนูบาร์ของ Sprite Editor

File Edit Transform Images Animation

- File

- 1) **New...** ใช้สร้างภาพใหม่ขึ้นมา
- 2) **Create from File...** ใช้โหลดภาพเข้ามาและจะทับภาพที่มีอยู่เดิมทั้งหมด
- 3) **Add from File...** เพิ่มเติมภาพจากที่มีอยู่แล้ว โดยนำไฟล์มาจากภายนอก เมื่อโหลดเข้ามาภาพจะไปอยู่ต่อจากภาพเดิม
- 4) **Save as Gif...** สามารถเซฟภาพไปเก็บไว้เป็นไฟล์ Gif ได้
- 5) **Save as Strip...** สามารถเซฟภาพออกไปในลักษณะภาพแบบ strip

- 6) **Create from strip...** นำไฟล์เข้ามาเพื่อให้โปรแกรมแบ่งภาพภายในเป็นชิ้นย่อยๆ และประกอบกันเป็นแอนิเมชันโดยอัตโนมัติ
- 7) **Add from strip** เพิ่มภาพ strip ต่อท้ายภาพที่มีอยู่เดิม
- 8) **Close saving changes** เซฟรูปภาพที่เราทำไว้เรียบร้อยแล้วและออกจากหน้าต่างนั้น



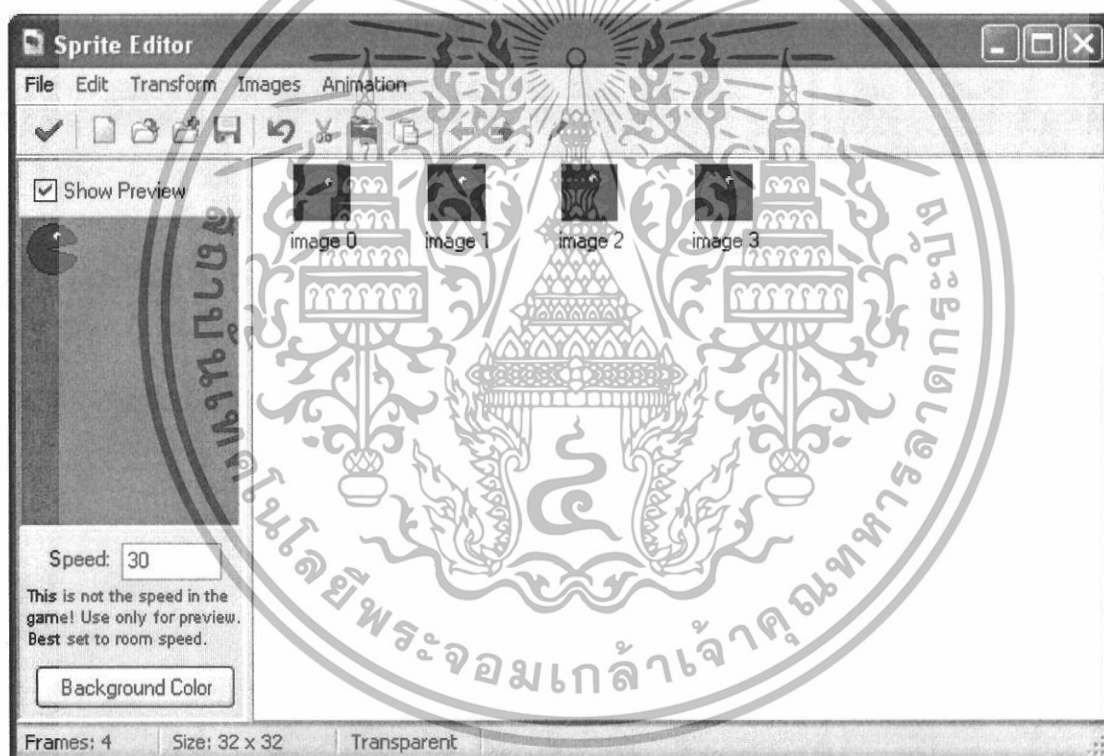
รูปที่ 2.35 รูปหน้าจอคำสั่ง loading a Sprite

- บริเวณหมายเลข 1
 - 1) **number of image** กำหนดรูปย่อยๆทั้งหมดว่ามีกี่รูป
 - 2) **image per row** กำหนดจำนวนรูปในแนวนอน จำนวนรูปที่เหลือจะถูกยกไปในแถวต่อไป
 - 3) **image width** กำหนดความกว้างในแต่ละภาพมีหน่วยเป็น pixel
 - 4) **image height** กำหนดความสูงในแต่ละภาพมีหน่วยเป็น pixel
 - 5) **horizontal cell offset** ใช้เลื่อนกรอบทั้งหมดไปในแนวนอนซึ่ง 1 หน่วยจะเท่ากับ ความกว้างของภาพที่เซตไว้
 - 6) **vertical cell offset** ใช้เลื่อนกรอบทั้งหมดไปในแนวนอนซึ่ง 1 หน่วยจะเท่ากับ ความสูงของภาพที่เซตไว้

- 7) **horizontal pixel offset** กำหนดให้กรอบแรกตำแหน่งซ้ายบนสุดในแนวนอนห่างจากตำแหน่ง (0,0) เท่าไร ตำแหน่ง (0,0) คือ จุดซ้ายบนสุดของรูปภาพใหญ่
- 8) **vertical pixel offset** กำหนดให้กรอบแรกตำแหน่งซ้ายบนสุดในแนวตั้งห่างจากตำแหน่ง (0,0) เท่าไร ตำแหน่ง (0,0) คือ จุดซ้ายบนสุดของรูปภาพใหญ่
- 9) **horizontal separation** กำหนดระยะห่างในแนวนอนระหว่างภาพย่อยๆแต่ละภาพ
- 10) **vertical separation** กำหนดระยะห่างในแนวตั้งระหว่างภาพย่อยๆแต่ละภาพ

- บริเวณหมายเลข 2

แสดงให้เห็นภาพ Strip ทั้งหมดและกรอบสี่เหลี่ยมในการแบ่งรูปต่างๆออกจากกัน เมื่อทำเสร็จแล้วจะได้ภาพแอนิเมชันประกอบไปด้วยภาพย่อยๆ 4 ภาพตามที่ต้องการ



รูปที่ 2.36 รูปหน้าจอคำสั่ง Sprite Editor แบบแสดง Preview

- File

- 1) **Add from strip** เพิ่มภาพ strip ต่อท้ายภาพเดิมที่มีอยู่แล้ว
- 2) **Close saving changes** เซฟรูปภาพที่เราทำไว้เรียบร้อยแล้วและออกจากหน้าต่างนี้

- Edit

 Undo	Ctrl+Z
 Cut	Ctrl+X
 Copy	Ctrl+C
 Paste	Ctrl+V
 Clear	Shift+Del
 Delete	Del
 Move Left	Ctrl+L
 Move Right	Ctrl+R
 Add empty	Shift+A
 Insert empty	Shift+I
 Edit...	Ctrl+E

รูปที่ 2.37 รูปแบบ Edit ใน Sprite Editor

- 1) **Undo** ย้อนกลับไปในสิ่งที่เรากระทำก่อนหน้านี้
- 2) **Cut** ลบภาพที่เลือกและทำสำเนาไปที่ clipboard
- 3) **Copy** ทำสำเนาภาพที่เลือกไปที่ clipboard
- 4) **Paste** วางภาพที่ทำสำเนาไว้ที่ clipboard
- 5) **Clear** ลบภาพที่เลือกให้เหลือแต่พื้นหลัง
- 6) **Delete** ลบภาพย่อยๆ ที่เลือกทิ้ง
- 7) **Move Left** สลับภาพที่เลือกกับภาพด้านซ้าย
- 8) **Move Right** สลับภาพที่เลือกกับภาพด้านขวา
- 9) **Add empty** เพิ่มภาพที่มีแต่พื้นหลังตรงที่เลือก
- 10) **Insert empty** แทรกภาพที่มีแต่พื้นหลังตรงที่เลือก
- 11) **Edit** แก้ไขภาพย่อยๆ

- Transform

	Mirror Horizontal	Ctrl+Alt+M
	Flip Vertical	Ctrl+Alt+F
	Shift...	Ctrl+Alt+H
	Rotate 90 clockwise	Ctrl+Alt+L
	Rotate 90 counter	Ctrl+Alt+O
	Rotate 180	Ctrl+Alt+T
	Rotate...	Ctrl+Alt+R
	Resize Canvas...	Ctrl+Alt+C
	Stretch...	Ctrl+Alt+E
	Scale...	Ctrl+Alt+S

รูปที่ 2.38 รูปเมนู Transform ใน Sprite Editor

- 1) **Mirror Horizontal** กลับข้างรูปภาพทั้งหมดในแนวนอน
- 2) **Flip Vertical** กลับข้างรูปภาพทั้งหมดในแนวตั้ง
- 3) **Shift...** ขยับภาพทั้งหมดไปในทิศทางที่ต้องการ
- 4) **Rotate 90 clockwise** หมุนภาพทั้งหมดไป 90 องศาแบบตามเข็มนาฬิกา
- 5) **Rotate 90 counter** หมุนภาพทั้งหมดไป 90 องศาแบบทวนเข็มนาฬิกา
- 6) **Rotate 180** หมุนภาพทั้งหมดไป 180 องศา
- 7) **Rotate...** หมุนภาพทั้งหมดไปตามองศาที่ต้องการ
- 8) **Resize Canvas...** กำหนดขนาดของภาพโดยรวมทั้งหมดใหม่โดยขนาดของภาพภายในไม่เปลี่ยนแปลงและสามารถทำให้ขยายไปในทิศทางที่ต้องการได้
- 9) **Stretch...** กำหนดขนาดภาพโดยรวมทั้งหมดใหม่ โดยที่ภาพภายในจะมีขนาดเปลี่ยนแปลงตามที่เรากำหนดใหม่
- 10) **Scale...** สามารถปรับเปลี่ยนขนาดของรูปภาพทั้งหมดได้ตามต้องการ โดยที่ขนาดโดยรวมของภาพยังไม่เปลี่ยนแปลง

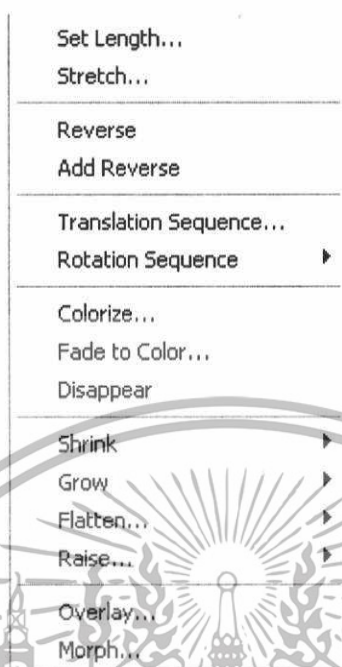
- Images

	Cycle Left	Shift+Ctrl+L
	Cycle Right	Shift+Ctrl+R
	Black and White	Shift+Ctrl+W
	Colorize...	Shift+Ctrl+C
	Colorize Partial...	Shift+Ctrl+O
	Shift Hue...	Shift+Ctrl+H
	Intensity...	Shift+Ctrl+I
	Invert	Shift+Ctrl+V
	Fade...	Shift+Ctrl+F
	Transparency...	Shift+Ctrl+T
	Blur...	Shift+Ctrl+B
	Outline...	Shift+Ctrl+U
	Boundary...	Shift+Ctrl+N
	Crop...	Shift+Ctrl+P

รูปที่ 2.39 เมนู Images ใน Sprite Editor

- 1) **Cycle Left** เลื่อนภาพทั้งหมดวนไปด้านซ้ายโดยที่ลำดับการเล่นภาพไม่เปลี่ยนแปลง
- 2) **Cycle Right** เลื่อนภาพทั้งหมดวนไปด้านขวาโดยที่ลำดับการเล่นภาพไม่เปลี่ยนแปลง
- 3) **Black and White** เปลี่ยนภาพทั้งหมดเป็นสีขาวดำ
- 4) **Colorize...** เปลี่ยนภาพทั้งหมดเป็นสีตามที่ต้องการ
- 5) **Colorize Partial...** เปลี่ยนภาพทั้งหมดเป็นสีตามที่ต้องการ โดยเปรียบเทียบสีเดิมกับสีใหม่
- 6) **Shift Hue...** เปลี่ยนภาพทั้งหมดเป็นสีตามที่ต้องการ
- 7) **Intensity...** เพิ่มความสว่างและความคมชัดของภาพทั้งหมด
- 8) **Invert** เปลี่ยนทุกสีในภาพทั้งหมดเป็นสีตรงข้าม
- 9) **Fade...** นำสีที่ต้องการมาผสมกับภาพทั้งหมดซึ่งสามารถเลือกปริมาณสีที่ต้องการได้ตั้งแต่ 0-256
- 10) **Transparency...** ทำให้ภาพทั้งหมดโปร่งใส ซึ่งสามารถเลือกปริมาณความโปร่งใสได้
- 11) **Blur...** ทำให้ภาพทั้งหมดเบลอได้ 9 ระดับ
- 12) **Outline...** สามารถสร้างเส้นโดยรอบภาพทั้งหมดได้
- 13) **Boundary...** สามารถสร้างเส้นภายในรูปภาพทั้งหมดบริเวณของรูปภาพ
- 14) **Crop...** สามารถตัดส่วนพื้นหลังที่เกินภาพทั้งหมดออกมาได้ โดยจะเหลือเฉพาะภาพที่ใช้

- Animation



รูปที่ 2.40 รูปเมนู Animation ใน Sprite Editor

- 1) **Set Length...** ใช้กำหนดภาพทั้งหมดที่ต้องการทำแอนิเมชัน
- 2) **Stretch...** ใช้ขยายภาพให้เป็นจำนวนเท่าที่ต้องการ โปรแกรมจะเพิ่มภาพให้โดยอัตโนมัติโดยการแทรกภาพจนครบ
- 3) **Reverse** ทำให้ลำดับภาพทั้งหมดกลับกัน
- 4) **Add Reverse** เพิ่มภาพทั้งหมดในลำดับที่กลับกันต่อท้ายภาพปัจจุบัน
- 5) **Translation Sequence...** ใช้สร้างแอนิเมชันในการเลื่อนภาพให้ไปอยู่ตำแหน่งที่ต้องการ สามารถกำหนดจำนวนเฟรมได้
- 6) **Rotation Sequence...** ใช้สร้างแอนิเมชันภาพหมุน
- 7) **Colorize...** ใช้สร้างแอนิเมชันเปลี่ยนสีภาพ สามารถกำหนดจำนวนเฟรมได้
- 8) **Fade to Color...** ใช้สร้างแอนิเมชันทำให้ภาพค่อยๆ จางไปเป็นสีที่ต้องการได้ สามารถกำหนดจำนวนเฟรมได้
- 9) **Disappear** ใช้สร้างแอนิเมชันทำให้ภาพค่อยๆ จางหายไป สามารถกำหนดจำนวนเฟรมได้
- 10) **Shrink** ใช้สร้างแอนิเมชันทำให้ภาพค่อยๆ ลดขนาดลง สามารถกำหนดจำนวนเฟรมและทิศทางได้

- 11) **Grow** ใช้สร้างแอนิเมชันทำให้ภาพค่อยๆเพิ่มขนาดขึ้น สามารถกำหนดจำนวนเฟรมและทิศทางได้
- 12) **Flatten...** ใช้สร้างแอนิเมชันทำให้ภาพจากปกติค่อยๆถูกบีบลงไปในทิศทางที่ต้องการ สามารถกำหนดจำนวนเฟรมได้
- 13) **Raise...** ใช้สร้างแอนิเมชันทำให้ภาพจากที่ถูกบีบในทิศทางที่กำหนดจนกลายเป็นภาพปกติสามารถกำหนดจำนวนเฟรมได้
- 14) **Overlay...** ใช้สร้างแอนิเมชันที่สามารถนำภาพอื่นๆมาซ้อนทับภาพปกติได้และสามารถกำหนดจำนวนเฟรมได้
- 15) **Morph...** ใช้สร้างแอนิเมชันที่สามารถทำให้ภาพปกติค่อยๆกลายเป็นภาพใหม่ที่เราเลือกได้และสามารถกำหนดจำนวนเฟรมได้

- บริเวณไอคอน

ไอคอนรูปภาพมีการใช้งานเหมือนกับที่ได้อธิบายไปแล้ว



รูปที่ 2.41 รูปบริเวณส่วนแสดงภาพ

- บริเวณส่วนแสดงภาพ

- 1) Show Preview ทำให้แสดงภาพตัวอย่างออกมาทางด้านล่าง
- 2) กรอบสี่เหลี่ยมใหญ่ จะแสดงภาพเคลื่อนไหวหรือภาพนิ่งบนพื้นหลังที่เป็นสีเดียว
- 3) Speed ใช้ตรวจสอบความเร็วในการเคลื่อนไหวของภาพเคลื่อนไหว
- 4) ปุ่ม Background Color สามารถเลือกสีพื้นหลังได้ตามใจชอบเพื่อให้ภาพปรากฏอย่างชัดเจน

- บริเวณส่วนแสดงภาพย่อย



รูปที่ 2.42 รูปบริเวณส่วนแสดงภาพย่อย

แสดงรูปภาพทั้งหมด ซึ่งชื่อของ Sub-image จะเริ่มตั้งแต่ image0 ไปจนถึงภาพสุดท้าย คือ image (n-1) ซึ่ง n คือจำนวนรูปทั้งหมด

2.2.2.2 Sounds Resource

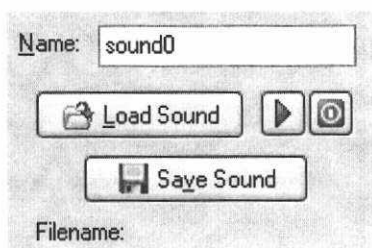


รูปที่ 2.43 หน้าจอคำสั่ง Sounds Resource

ใช้สำหรับนำไฟล์เสียงเข้ามาเพื่อเก็บไว้ใช้ในเกม

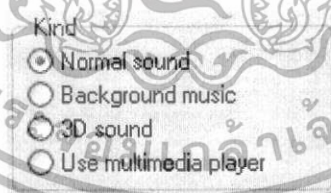
เมื่อคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add sound จะปรากฏหน้าต่าง Sound Properties ขึ้นมาดังรูป เพื่อให้เราได้คุณลักษณะต่างๆลงไปดังนี้

- Sound Properties



รูปที่ 2.44 รูปหน้าจอคำสั่ง Sound Properties

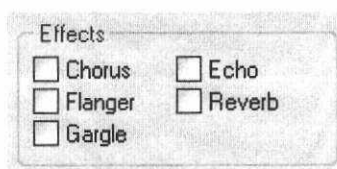
- 1) **Name** ตั้งชื่อรีซอร์ส ของเสียงนี้
- 2) **Load sound** โหลดไฟล์เสียงเข้ามา
- 3) **Save Sound** สามารถเซฟเสียงออกไปเก็บไว้ได้
- 4)  **(Play)** ใช้ทดสอบเสียง
- 5)  **(End)** ใช้หยุดเล่นเสียง
- 6) **File name** แสดงชื่อ ไฟล์ที่เลือกมาพร้อมนามสกุล
- 7) **Kind** ชนิดของเสียงแต่ละประเภทควรเลือกให้เหมาะสมกับไฟล์ที่นำเข้ามา



รูปที่ 2.45 รูปหน้าจอคำสั่ง Kind

- 8) **Normal sound** ใช้เล่นเสียงทั่วไป เสียงเอฟเฟ็กต์ต่างๆ
- 9) **Background music** ใช้เล่นดนตรีประกอบเกม ไม่สามารถเล่นเพลงซ้อนได้
- 10) **3D sound** ในกรณีที่ใช้ฟังก์ชันพิเศษจากสคริปต์ GML

- 11) **Use multimedia player** เรียกใช้ media player ในการเล่นเสียงซึ่งจะสามารถทำให้เล่นไฟล์ mp3 ได้ แต่ไม่สามารถใส่เอฟเฟกต์หรือปรับความดังได้
- 12) **Effects** สามารถใส่เอฟเฟกต์ต่างๆลงไปได้



รูปที่ 2.46 รูปหน้าจอคำสั่ง Effects

- 13) **Volume** ใช้กำหนดความดังของเสียงในเกม

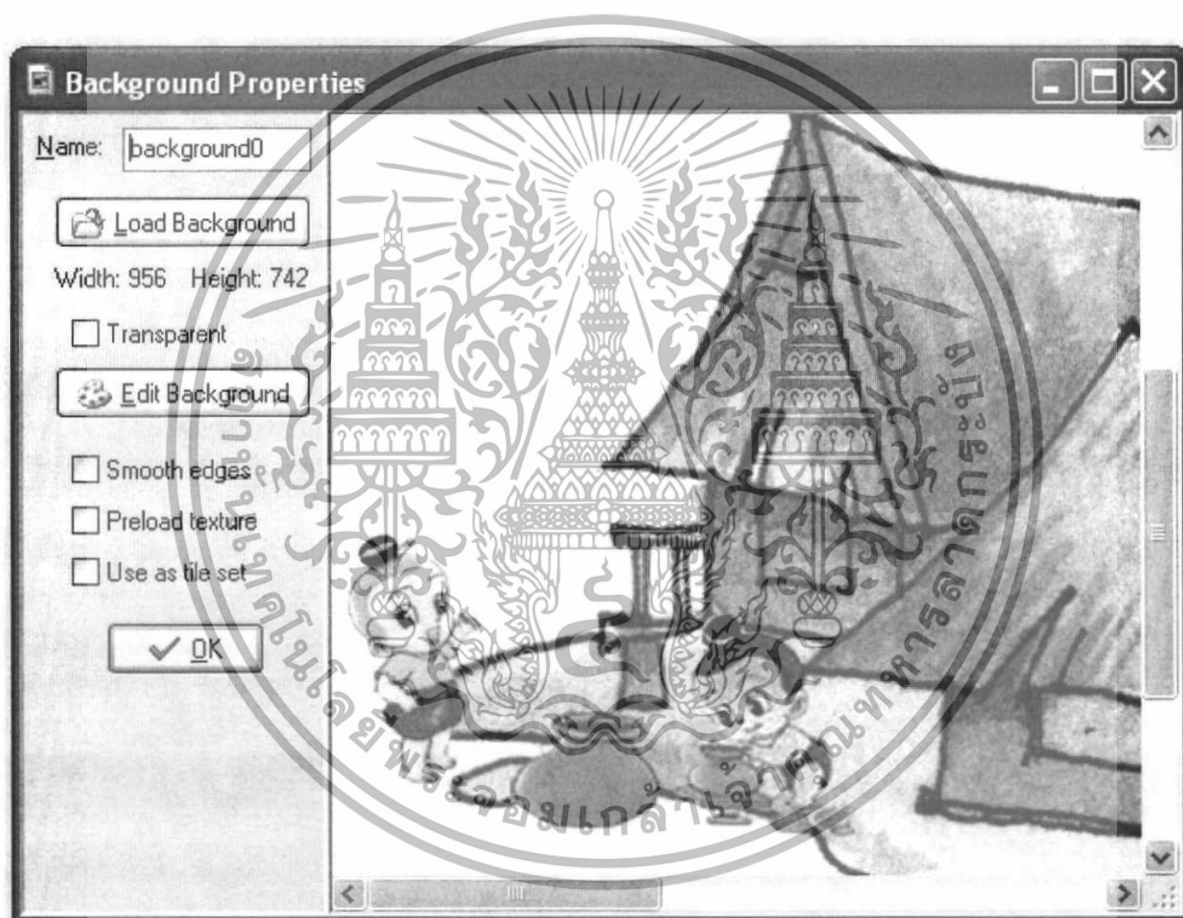


รูปที่ 2.47 รูปหน้าจอคำสั่ง Volume

- 14) **Pan** ใช้กำหนดตำแหน่งเสียงจากลำโพงซ้ายและขวาได้
- 15) **Preload** รีซอร์สทุกอันที่เช็กลูกจะถูกโหลดเข้าไปสู่หน่วยความจำทันทีเมื่อรันเกมขึ้นมา
- 16) ปุ่ม **Edit sound**
- 17) ปุ่ม **OK** ยืนยันการตั้งค่าทั้งหมด

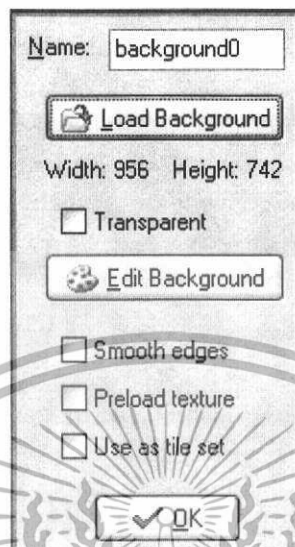
2.2.2.3 Background Resource

ใช้สำหรับนำภาพพื้นหลังของเกมเข้ามาเก็บไว้เพื่อนำไปใช้ ในเกมเมื่อคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add Background จะปรากฏหน้าต่าง Background Properties ขึ้นมาดังรูปด้านล่าง เพื่อให้เราได้คุณสมบัติต่างๆดังต่อไปนี้



รูปที่ 2.48 หน้าจอหลักคำสั่ง Background Properties

- Background Properties



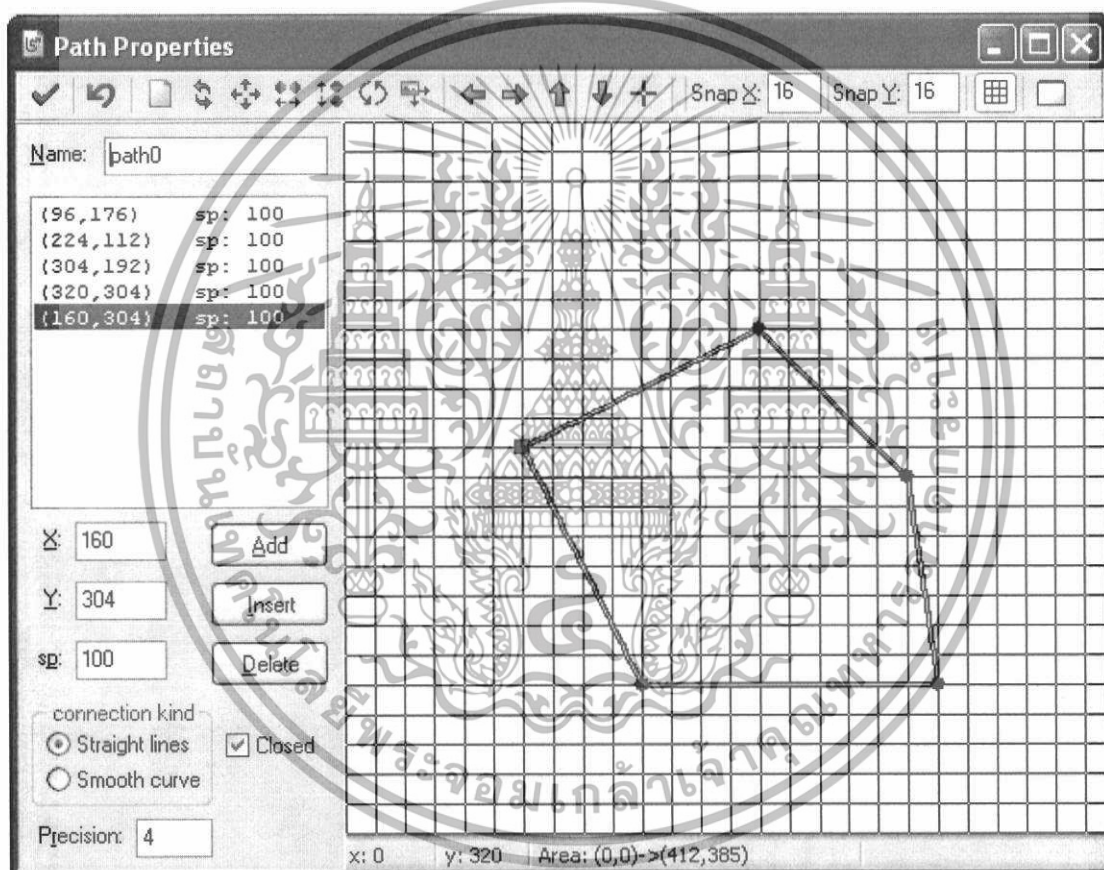
รูปที่ 2.49 รูปหน้าจอคำสั่ง Background Properties ด้านซ้าย

- 1) **Name** ตั้งชื่อให้รหัสพื้นหลังของเรา
- 2) **Load Background** โหลดไฟล์ภาพพื้นหลังเข้ามา
- 3) **Width** แสดงความกว้างของภาพ
- 4) **Height** แสดงความสูงของภาพ
- 5) **Transparent** กำหนดให้ส่วนที่เป็นพื้นหลังโปร่งใส
- 6) **Edit Background** ใช้แก้ไขรูปภาพ
- 7) **Smooth edges** ทำให้ขอบของภาพดูเรียบขึ้น
- 8) **Preload texture** ถ้าถูกเช็คจะทำให้ตอนเริ่มรันเกม ภาพพื้นหลังนี้จะถูกโหลดเข้าสู่หน่วยความจำของวีดิโอการ์ดทันที
- 9) **Use as tile set** เมื่อภาพที่โหลดเข้ามามีลักษณะเป็นแบบเป็นชิ้นๆ สามารถตั้งค่าการเป็นชิ้นๆตามขนาดที่ต้องการเพื่อนำไปใช้งาน
- 10) **พื้นที่ด้านขวา** แสดงภาพพื้นหลังที่โหลดเข้ามาขนาดเท่าจริง

2.2.2.4 Paths Resource

ใช้สำหรับสร้างเส้นทางที่แน่นอนในการเคลื่อนที่ของวัตถุใดๆในเกม เช่น การเคลื่อนที่ของศัตรูในเกม เป็นต้น

คลิก Add Path จะปรากฏหน้าต่าง Path Properties ขึ้นมาดังรูปข้างล่างเพื่อให้เราได้คุณลักษณะต่างๆลงไปดังนี้



รูปที่ 2.50 รูปหน้าจอหลักคำสั่ง Path Properties

- Path Properties

Name:

(96,176)	sp: 100
(224,112)	sp: 100
(304,192)	sp: 100
(320,304)	sp: 100
(160,304)	sp: 100

X:

Y:

sp:

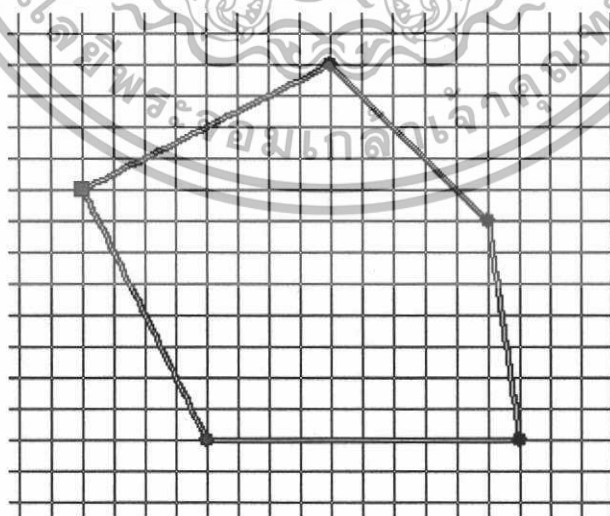
connection kind
☒ Straight line
☐ Smooth curve
☒ Closed

Precision:

รูปที่ 2.51 หน้าจอคำสั่ง Path Properties ด้านซ้าย

- 1) Name ตั้งชื่อ path
- 2) X: ข้ายตำแหน่งแกน X ของจุดที่เราเลือก
- 3) Y: ข้ายตำแหน่งแกน Y ของจุดที่เราเลือก
- 4) sp: กำหนดค่าร้อยละของความเร็ววัตถุ
- 5) Add เพิ่มจุดในการเชื่อมต่อ path ขึ้นมาอีกหนึ่งจุด
- 6) Insert แทรกจุดตรงบริเวณจุดเดิมที่เราเลือกอยู่
- 7) Delete ลบจุดที่เลือกออก
- 8) Straight line เส้นเชื่อมต่อระหว่างจุดจะเป็นเส้นตรง
- 9) Smooth curve เส้นเชื่อมต่อระหว่างจุดจะเป็นเส้นโค้ง
- 10) Closed จุดสุดท้ายจะสร้างเส้นเชื่อมต่อกับจุดแรกโดยอัตโนมัติ
- 11) Precision: เมื่อเรากำหนดให้เส้นเป็นเส้นโค้ง สามารถปรับความโค้งได้ตามต้องการ 1-8 ระดับ
- 12) ☒ (Close the form and saving) ปิดหน้าต่างนี้จะยืนยันการแก้ไข

- 13)  (Undo) กลับไปสู่สิ่งที่กระทำครั้งที่แล้ว
- 14)  (Clear the path) ลบ path ทั้งทั้งหมด
- 15)  (Reverse the path) สลับจุดทุกจุด
- 16)  (Shift the path) เคลื่อนย้าย path ทั้งหมด โดยใส่ค่าทั้งแนวนอนและแนวตั้ง
- 17)  (Mirror the path horizontally) กลับค่า path ตามแนวนอน
- 18)  (Flip the path vertically) กลับค่า path ตามแนวตั้ง
- 19)  (Rotate the path) หมุน path ตามองศาที่ต้องการ
- 20)  (Scale the path) ย่อและขยาย
- 21)  (Shift the view to left) เลื่อนมุมมองไปด้านซ้าย
- 22)  (Shift the view to right) เลื่อนมุมมองไปด้านขวา
- 23)  (Shift the view to top) เลื่อนมุมมองไปด้านบน
- 24)  (Shift the view to bottom) เลื่อนมุมมองไปด้านล่าง
- 25)  (Center the view around the path) เลื่อนมุมมองให้อยู่กลาง path ที่ทำเอาไว้
- 26)  (Snap X) เปลี่ยนขนาดกริดในแกน X
- 27)  (Snap Y) เปลี่ยนขนาดกริดในแกน Y
- 28)  (Toggle the showing of the grid) เปิดหรือปิดกริด
- 29)  (Indicate the room to show as background) นำเอาพื้นหลังของ room ในเกมมาเป็นพื้นหลังในการสร้าง path เมื่อต้องการความแม่นยำในการวางตำแหน่งใน room ของเกม



รูปที่ 2.52 รูปหน้าจอคำสั่ง Path Properties ด้านขวา

2.2.2.5 Script Resource

สำหรับรีซอร์สนี้มีไว้ให้ผู้ใช้เขียนสคริปต์ต่างๆ และจัดเก็บไว้อย่างเป็นหมวดหมู่เพื่อเรียกใช้งานตามต้องการอาศัยความรู้ด้านโปรแกรมมิ่งพอสมควรเพื่อที่จะศึกษาภาษา GML เมื่อคุณเข้าใจโครงสร้างภาษาและสามารถเขียนสคริปต์ได้แล้ว คุณจะสร้างเกมที่มีความซับซ้อนได้

เมื่อเรากดคลิกขวาที่ไฟล์เดอร์นี้แล้วเลือก Add script จะปรากฏหน้าต่าง Script Properties ขึ้นมาดังรูปด้านล่างเพื่อให้เราสามารถเขียนสคริปต์ที่เราต้องการลงไปดังนี้



รูปที่ 2.53 รูปหน้าจอหลักคำสั่ง Script Resource

- Script Properties

- 1)  (Close and Saving) ปิดหน้าต่างนี้จะยืนยันการแก้ไข
- 2)  (Undo) กลับไปสู่สิ่งที่กระทำครั้งที่แล้ว
- 3)  (Redo) กลับไปที่การกระทำเมื่อตอนก่อน Undo มาครั้งที่แล้ว

- 4)  (Cut) ลบ text และทำสำเนาไปที่ clipboard
- 5)  (Copy) ทำสำเนาไปที่ clipboard
- 6)  (Paste) วาง text ที่ทำสำเนาไว้ที่ clipboard
- 7)  (Search) ค้นหาที่ต้องการ
- 8)  (Replace string) ค้นหาที่ต้องการและแทนที่คำใหม่ได้
- 9)  (Go to a given line) ไปที่บรรทัดตามที่กำหนด
- 10)  (Check the script for the syntax error) ตรวจสอบว่ามีจุดใดบ้างที่เขียนผิดไปจากโครงสร้างภาษา GML
- 11) Name: (Name) ตั้งชื่อสคริปต์นี้
- 12) พื้นที่ตรงกลาง คือบริเวณที่เขียนสคริปต์
- 13) แสดงฟังก์ชันที่เราสามารถเรียกใช้งานได้ ซึ่งจะแสดงขึ้นมาตามตัวอักษรที่เราพิมพ์
ดังรูป



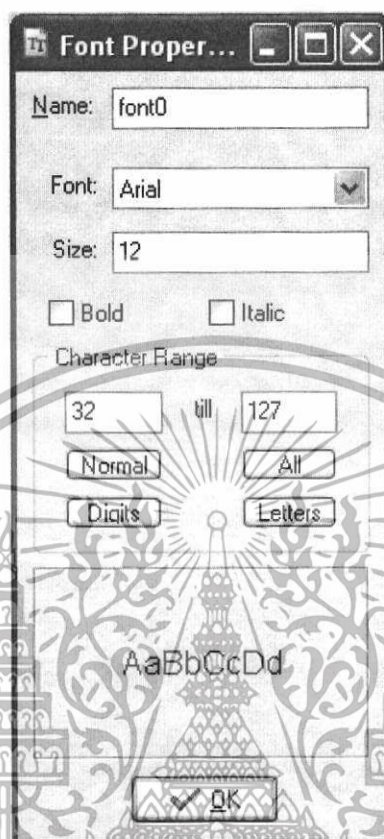
รูปที่ 2.54 รูปหน้าจอฟังก์ชันของ Script Resource

2.2.2.6 Fonts Resource

ใช้สำหรับตั้งค่าฟอนต์หรือตัวอักษรภายในเกม สำหรับ Game Maker เวอร์ชัน 6.1 นั้นจะมีปัญหากับฟอนต์ภาษาไทยบ้าง ซึ่งเวลาที่เราเขียนคำที่มีสระและวรรณยุกต์ซ้อนกันสองชั้น โปรแกรมจะเขียนทับกันเป็นชั้นเดียว อาจทำให้อ่านลำบากขึ้นบ้าง สาเหตุเนื่องมาจากการเปลี่ยนแปลงลักษณะการแสดงผลของตัวอักษรให้สามารถปรับแต่งได้หลายรูปแบบ การแก้ไขจำเป็นต้องอาศัยความรู้ด้านการเขียนสคริปต์ โดยเราต้องสร้างฟอนต์ที่เป็น sprite ขึ้นมาเพื่อใช้งานเองแล้วเขียนสคริปต์โดยใช้ฟังก์ชัน `front_add_sprite (spr, first, prop, sep)` ในการเรียกใช้งานภาพฟอนต์แล้วจึงเขียนสคริปต์อื่นๆ เพื่อแสดงตัวอักษรให้เป็นไปตามที่ควรจะเป็น

เมื่อคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add Font จะปรากฏหน้าต่าง Font Properties ขึ้นมา ดังรูปหน้าถัดไปเพื่อให้เราใส่คุณลักษณะต่างๆ ลงไปดังนี้

- Font Properties



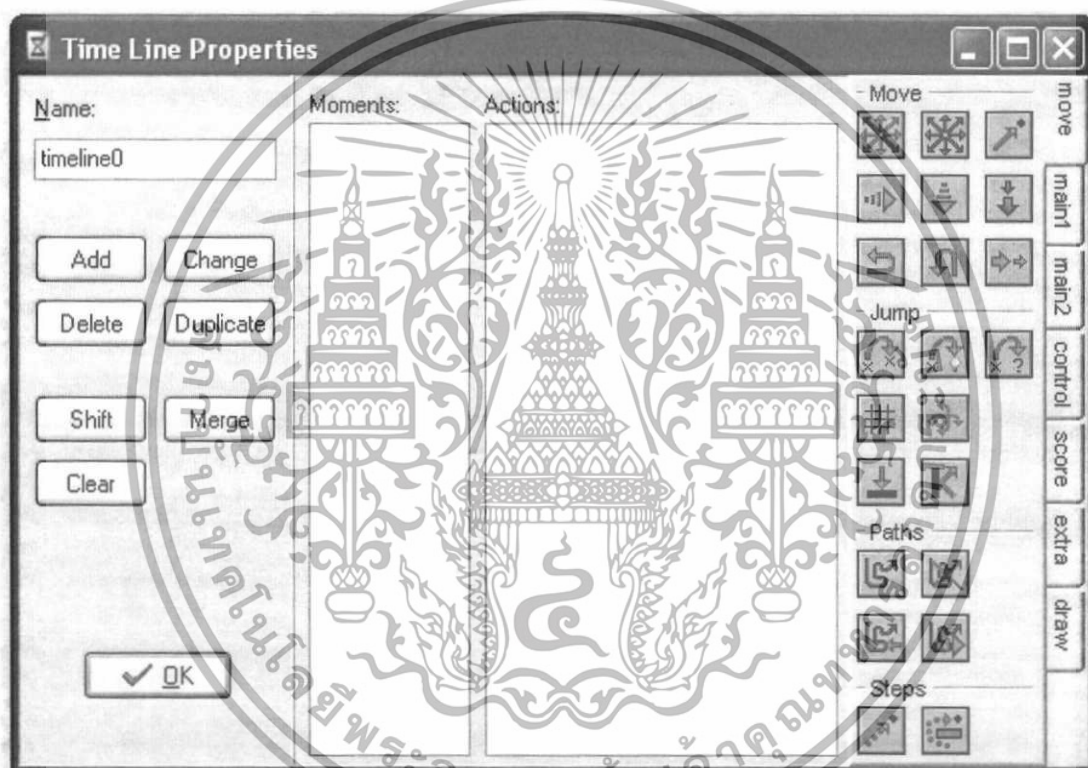
รูปที่ 2.55 หน้าจอคำสั่ง Font Properties

- 1) Name: ตั้งชื่อฟอนต์
- 2) Font: เลือกฟอนต์ที่มีอยู่ในวินโดวส์
- 3) Size: กำหนดขนาดของฟอนต์
- 4) Bold กำหนดให้เป็นตัวหนา
- 5) Italic กำหนดให้เป็นตัวเอียง
- 6) Character Range ใช้กำหนดตัวอักษรทั้งหมดที่ต้องการให้แสดงผลของฟอนต์ที่เลือก
- 7) กรอบสี่เหลี่ยมด้านล่าง แสดงฟอนต์ที่เราเลือกและขนาดตามที่กำหนด
- 8) ปุ่ม OK เพื่อยืนยันการแก้ไขและปิดหน้าต่าง

2.2.2.7 Time Lines Resource

ใช้สำหรับกำหนดเวลาที่แน่นอนสำหรับวัตถุในเกม เพื่อที่จะให้วัตถุนั้นๆทำตามคำสั่งที่เราต้องการเมื่อถึงเวลาตามที่เราได้ตั้งค่าไว้

เมื่อเรากดคลิกขวาที่ฟลอร์เดอร์นี้แล้วเลือก Add Time Line จะปรากฏหน้าต่าง Time Line Properties ขึ้นมาดังรูปด้านล่างเพื่อให้เราใส่คุณลักษณะต่างๆลงไปดังนี้



รูปที่ 2.56 รูปหน้าจอหลักคำสั่ง Time Lines Resource

- Time Line Properties

1) **Name** ตั้งชื่อรีซอร์สนี้

2) **Add** (Add) เพิ่มช่วงเวลาที่จะทำให้วัตถุในเกมกระทำการสิ่งต่างๆ

3) **Delete** (Delete) ให้ลบช่วงเวลาที่เราไม่ต้องการ

4) **Change** (Change) สามารถเปลี่ยนจำนวน step ที่เลือกให้เป็นไปตามที่ต้องการ

5) **Duplicate** (Duplicate) สามารถทำสำเนา step ที่เลือกเพื่อสร้าง step ใหม่

6) **Shift** (Shift) เลือกช่วง step สองช่วง และสามารถขยับเวลาช่วง step นั้นๆ เพิ่มขึ้นหรือลดลงได้ โดยใส่ค่าที่ช่อง By amount

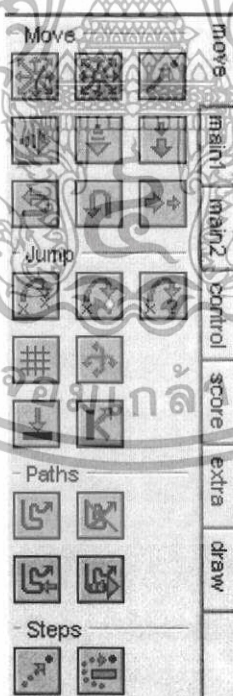
7) **Merge** (Merge) สามารถรวม step สองช่วงเข้าด้วยกัน Action ก็จะรวมกันด้วย

8) **Clear** (Clear) ลบ step ทั้งหมด

9) **OK** (ปุ่ม OK) ยืนยันการแก้ไขและปิดหน้าต่างนี้

10) **Moments** (Moment) แสดงช่วงเวลาที่เหตุการณ์ต่างๆขึ้นเป็น step

11) **(Actions)** สิ่งที่เกิดขึ้นทั้งหมดเมื่อถึง step ดังกล่าว เรียงลำดับการเกิดจากบนลงล่าง

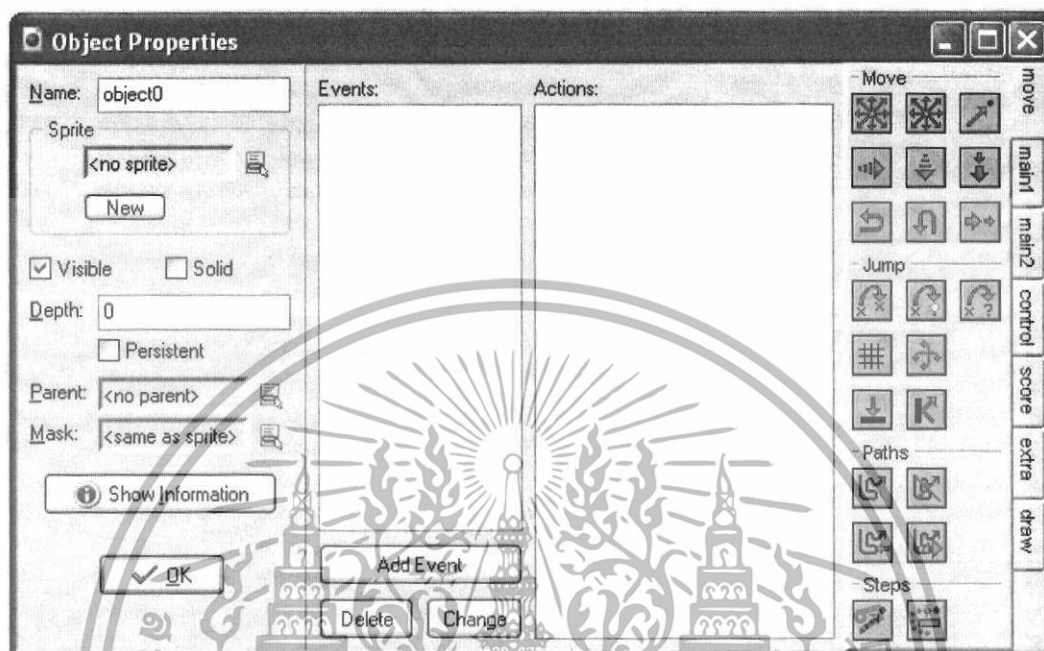


รูปที่ 2.57 หน้าจอคำสั่ง Actions

12) Actions Library รวบรวม Action ทั้งหมดแบ่งไว้เป็นหมวดหมู่

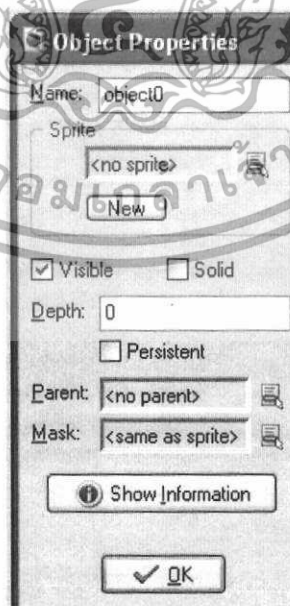
2.2.2.8 Object Resource

ใช้สำหรับสร้างวัตถุในเกม เมื่อเรากดคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add Object จะปรากฏหน้าต่าง Object Properties ขึ้นมาดังรูปด้านล่างเพื่อให้เราใส่คุณลักษณะต่างๆลงไปดังนี้



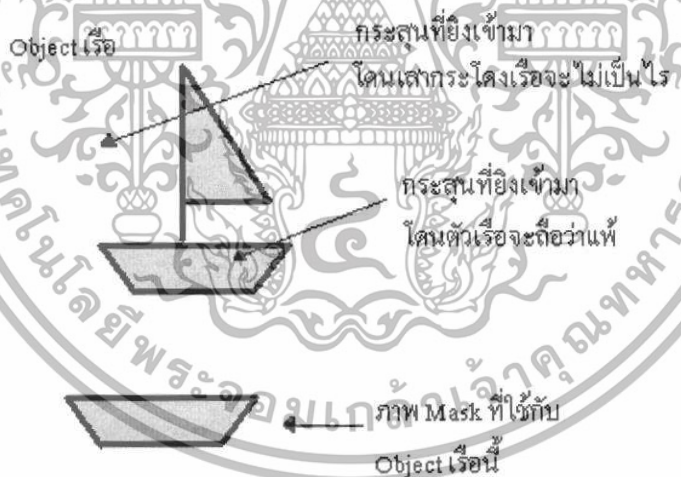
รูปที่ 2.58 หน้าจอหลักคำสั่ง Time Lines Resource

- Object Properties



รูปที่ 2.59 หน้าจอหลักคำสั่ง Object Properties

- 1) **Name:** ตั้งชื่อวัตถุ
- 2) **Sprite** เลือกภาพ sprite เพื่อนำมาใช้กับวัตถุนี้
- 3) ปุ่ม **New** เพิ่ม sprite ใหม่ขึ้นมา
- 4) ปุ่ม **Edit** เปิดภาพ sprite ที่เราเลือกอยู่ขึ้นมาเพื่อแก้ไข
- 5) **Visible** เมื่อรันเกมจะสามารถทำให้มองเห็นภาพ sprite ที่เราเลือกขณะที่อยู่ใน room ได้ ถ้าเช็คถูกออกจะมองไม่เห็นภาพแต่ภาพยังคงอยู่
- 6) **Solid** วัตถุจะมีสภาพเป็นของแข็ง ซึ่งเมื่อมีการชนกันกับวัตถุอื่นที่มีสภาพเดียวกัน ก็จะสามารถเช็คเงื่อนไขต่างๆที่เกี่ยวข้องได้
- 7) **Depth:** กำหนดความลึกของวัตถุ
- 8) **Persistent** การคงสภาพวัตถุไว้อย่างเดิม แม้มีการเปลี่ยน room
- 9) **Parent:** สามารถกำหนดวัตถุที่เป็นตัวหลักได้
- 10) **Mask:** ขอบเขตที่วัตถุใช้เช็คว่าชนกันระหว่างวัตถุ



รูปที่ 2.60 รูปแสดงตัวอย่างการ Mask

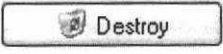
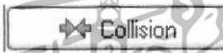
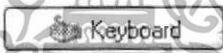
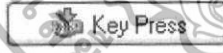
- 11) ปุ่ม **Show Information** แสดงข้อมูลต่างๆของวัตถุนั้นๆเป็น text สามารถเซฟออกไปได้
- 12) ปุ่ม **OK** ยืนยันการแก้ไขและปิดหน้าต่างนี้

- Events

ในส่วนนี้จะเป็นการกำหนดเงื่อนไขต่างๆที่จะเกิดขึ้นกับวัตถุ

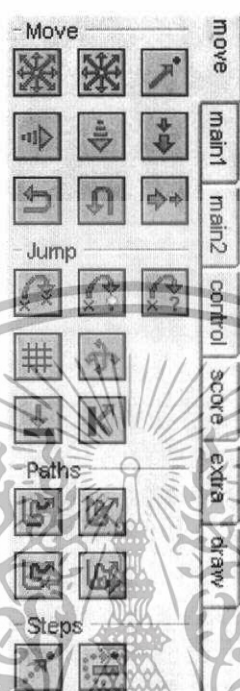


รูปที่ 2.62 รูปปุ่มการเลือก Events

- 2)  **Create** (Create) เมื่อวัตถุถูกสร้างขึ้นมาใน room จะทำ Action ที่อยู่ในเหตุการณ์นี้ทันทีเพียงรอบเดียว
- 3)  **Destroy** (Destroy) เมื่อวัตถุถูกทำลาย จะทำ Action ที่อยู่ในเหตุการณ์นี้ทันทีเพียงรอบเดียว
- 4)  **Alarm** (Alarm) เมื่อถึงเวลาที่ได้อ้างไว้ใน set an alarm clock จะมีเหตุการณ์นี้ซึ่งสามารถตั้งเวลาได้สูงสุด 12 alarm และจะทำ Action ที่อยู่ในเหตุการณ์นี้ทันทีเพียงรอบเดียว
- 5)  **Step** (Step) Action ภายใน event นี้จะถูกตรวจสอบอยู่เสมอขณะที่เกมรันอยู่และอนิเมชันนี้ยังไม่ถูกทำลาย
- 6)  **Collision** (Collision) เมื่อมีวัตถุที่เลือกเข้ามาชน จะทำ Action ที่อยู่ในเหตุการณ์นี้ทุกขณะที่วัตถุที่เลือกเกิดการซ้อนทับกันอยู่
- 7)  **Keyboard** (Keyboard) เมื่อกดปุ่มค้างไว้จะทำ Action ที่อยู่ในเหตุการณ์นี้ตลอดเวลาที่ยังกดค้างไว้อยู่
- 8)  **Mouse** (Mouse)
- 9)  **Other** (Other) เหตุการณ์อื่นๆ
- 10)  **Draw** (Draw) ใช้วาดภาพ sprite หรือ text ลงไปใน room
- 11)  **Key Press** (Key Press) คล้ายกับคีย์บอร์ดแต่เหตุการณ์เกิดขึ้นเฉพาะตอนกดคีย์บอร์ดลงไปเท่านั้น
- 12)  **Key Release** (Key Release) คล้ายกับคีย์บอร์ดแต่เหตุการณ์เกิดขึ้นเฉพาะตอนปล่อยคีย์บอร์ดออกจากปุ่มเท่านั้น
- 13)  **Cancel** (Cancel) ยกเลิกและปิดหน้าต่าง

- Action Library

ส่วนของ Action Library ซึ่งแบ่งออกเป็นหมวดหมู่ดังต่อไปนี้



รูปที่ 2.63 หน้าจอคำสั่ง Action Library

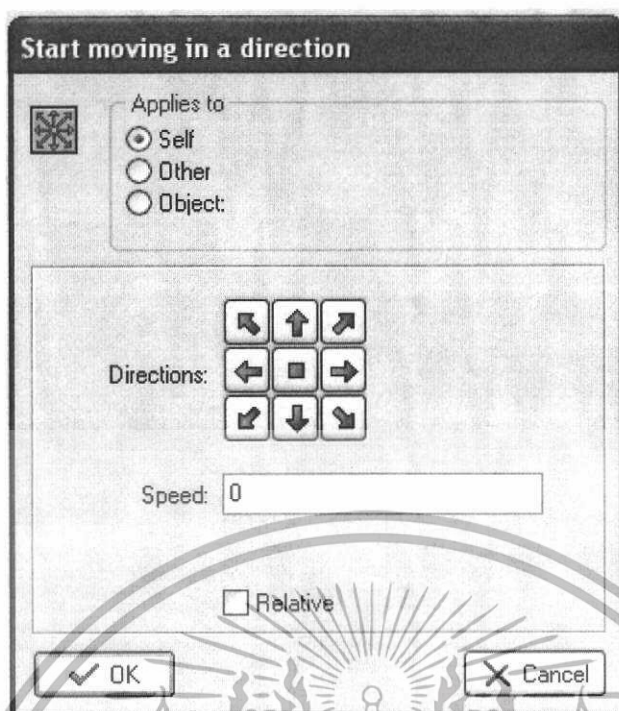
- Move

- Move

- 1)  (Start moving in a direction) ทำให้วัตถุเคลื่อนที่ไปในทิศทางที่ต้องการหรือ
สุ่มทิศทาง

Direction คือทิศทางการเคลื่อนที่ของวัตถุมีให้เลือก 8 ทิศ ส่วนจุดตรงกลางจะทำให้วัตถุหยุดนิ่งอยู่ที่เดิม ในกรณีที่เรเลือกหลายทิศทางจะเป็นการสุ่มทิศทางที่เลือกทั้งหมด

Speed คือความเร็วในการเคลื่อนที่ของวัตถุ ซึ่งจะมีหน่วยเป็น pixel/step หรือ 1 step นั้นจะเคลื่อนที่ไปได้กี่ pixel



รูปที่ 2.64 หน้าจอคำสั่ง Start moving in a direction

- 2)  (Set direction and speed of motion) ทำให้วัตถุเคลื่อนที่ไปในทิศทางที่ต้องการ ซึ่งสามารถกำหนดทิศทางเป็นองศาได้
- 3)  (Move towards a point) ทำให้วัตถุเคลื่อนที่ไปในทิศทางของตำแหน่ง (X,Z) ที่เรากำหนดไว้ตามความเร็วที่กำหนด
- 4)  (Set the horizontal speed) ทำให้วัตถุเคลื่อนที่ในแนวนอน
- 5)  (Set the vertical speed) ทำให้วัตถุเคลื่อนที่ในแนวตั้ง
- 6)  (Set the gravity) ทำให้วัตถุมีแรงโน้มถ่วงในทิศทางที่ต้องการ
- 7)  (Reverse horizontal direction) ทำให้ทิศทางของวัตถุสะท้อนกลับในแนวนอนโดยความเร็วเท่าเดิม
- 8)  (Reverse vertical direction) ทำให้ทิศทางของวัตถุสะท้อนกลับในแนวตั้งโดยความเร็วเท่าเดิม
- 9)  (Set the friction) ใส่ค่า friction เพื่อทำให้วัตถุเกิดความฝืดขึ้น

- Jump

- 1)  (Jump to given position) ใส่ค่า X และค่า Y เพื่อย้ายวัตถุไปในตำแหน่งที่กำหนด
- 2)  (Jump to the start position) ย้ายวัตถุไปที่ตำแหน่งเริ่มต้นที่วัตถุนั้นถูกสร้างขึ้นมา
- 3)  (Jump to a random position) สุ่มย้ายวัตถุไปตำแหน่งใดๆใน room
- 4)  (Snap to grid) เมื่อจุด origin ของวัตถุอยู่ใกล้ค่า snap ที่เราตั้งไว้ วัตถุจะย้ายไปที่จุดนั้นทันที
- 5)  (Wrap when moving outside) สำหรับย้ายวัตถุที่เคลื่อนที่ออกไปนอก room ให้ไปอยู่ฝั่งตรงกันข้ามในแกนเดียวกัน
- 6)  (Move to contact position) เมื่อวัตถุเคลื่อนที่ปะทะกับอีกวัตถุหนึ่ง เราสามารถใช้ Action นี้ทำให้วัตถุเข้าไปสัมผัสที่ผิวของอีกวัตถุหนึ่งได้
- 7)  (Bounce against objects) ทำให้วัตถุกระเด็นกลับตามธรรมชาติเมื่อชนอีกวัตถุหนึ่ง

- Path

- 1)  (Set a path for the instance) เมื่อสร้าง path แล้ว สามารถนำวัตถุเข้าสู่เส้นทางใน path ได้
- 2)  (End a path for the instance) ใช้งานการทำงานของ path
- 3)  (Set the position on the path) ใช้กำหนดให้วัตถุไปปรากฏที่ตำแหน่งใดๆใน path ซึ่งค่าจะอยู่ระหว่าง 0-1
- 4)  (Set the speed for the path) กำหนดความเร็วให้กับวัตถุเมื่อวัตถุนั้นอยู่ใน path ใดๆ

- Steps

- 1)  **(Perform a step towards a point)** ใช้ใน step event ทำให้วัตถุเคลื่อนที่ไปที่ตำแหน่ง (x,y) ที่กำหนดไว้ด้วยความเร็วที่กำหนดไว้และสามารถเลือกได้ว่าจะหยุดเมื่อปะทะกับ Solid หรือปะทะกับวัตถุทั้งหมด
- 2)  **(Step towards a point avoiding objects)** ใช้ใน step event ทำให้วัตถุสามารถหลบหลีกสิ่งกีดขวางและหาทางเพื่อให้ไปถึงที่เป้าหมายกำหนดไว้ (x,y) ให้ได้ด้วยความเร็วที่กำหนดไว้สามารถเลือกได้ว่าจะให้หลบหลีกเฉพาะ Solid หรือวัตถุทั้งหมด

● Main 1

- Object

- 1)  **(Create an instance of an object)** ใช้สร้างวัตถุใดๆขึ้นมาใน room ในตำแหน่ง (x,y) ที่กำหนด
- 2)  **(Create an instance of an object with a speed and direction)** ใช้สร้างวัตถุใดๆขึ้นมาใน room ที่ตำแหน่ง (x,y) พร้อมกำหนดทิศทางเป็นองศา และความเร็วในการเคลื่อนที่
- 3)  **(Create instance of random object)** ใช้สร้างวัตถุใดๆขึ้นมาใน room ที่ตำแหน่ง (x,y) โดยสามารถสุ่มเลือกวัตถุได้ 4 ชนิด
- 4)  **(Change the instance)** ใช้เปลี่ยนวัตถุเดิมเป็นอีกวัตถุหนึ่งที่เลือกไว้ ซึ่งค่าตัวแปรอยู่ภายในจะยังคงเหมือนเดิม
- 5)  **(Destroy the instance)** ใช้ทำลาย instance
- 6)  **(Destroy the instance at the position)** ใช้ทำลาย instance ที่ตำแหน่ง (x,y) ตามที่กำหนด

- Sprite

- 1)  (Change the sprite) ใช้เปลี่ยนภาพ sprite เป็นภาพอื่นได้สามารถใส่ค่าเพื่อเลือก subimage หรือภาพย่อยๆ โดยเริ่มตั้งแต่ 0 เป็นต้นไป
- 2)  (Transform the sprite) สามารถกำหนดขนาดของ sprite ได้
- 3)  (Set sprite blending) สามารถเปลี่ยนสีให้กับ sprite ได้และสามารถกำหนดค่าความโปร่งใสได้โดยใส่ค่าที่ alpha ซึ่งค่านี้ คือ ปกติ

- Sound

- 1)  (Play a sound) ใช้เล่นเสียงที่เราเข้ามาเก็บไว้
- 2)  (Stop a sound) ใช้หยุดเล่นเสียงโดยให้เลือกเสียงที่ต้องการหยุดเล่น
- 3)  (If a sound is playing) เป็นประ โยคเงื่อนไข ซึ่งถ้าเงื่อนไขดังกล่าวเป็นจริงจึงจะทำ Action ที่อยู่ในเงื่อนไขนี้

- Room

- 1)  (Go to previous room) ในกรณีที่เกมเรามีหลาย room ใช้คำสั่งนี้ทำให้กลับไปยัง room ก่อนหน้า
- 2)  (Go to next room) ในกรณีที่เกมเรามีหลาย room ใช้คำสั่งนี้ทำให้ไปสู่ room ถัดไป
- 3)  (Restart the current room) กลับไปเล่นที่ room ปัจจุบันอีกครั้งหนึ่ง
- 4)  (Go to a different room) คำสั่งนี้สามารถเลือก room ที่ต้องการจะไปได้
- 5)  (If previous room exists) เป็นประ โยคเงื่อนไข หมายถึง ยังมีroom ก่อนหน้านี้อยู่แล้ว...{ต่อด้วย Action}
- 6)  (If next room exists) เป็นประ โยคเงื่อนไข หมายถึง ยังมีroom ถัดไป แล้ว...{ต่อด้วย Action}

- Main2

- Timing

- 1)  (Set an alarm clock) ใช้ในการหน่วงเวลาเพื่อที่จะไปที่ alarm event ที่กำหนดไว้
- 2)  (Sleep for a while) หยุดเกมชั่วคราวตามเวลาที่กำหนดไว้
- 3)  (Set a time line) ทำให้วัตถุสามารถใช้ time line ที่ตั้งไว้ได้
- 4)  (Set a time line position) กำหนดตำแหน่งใหม่ของ time line ที่กำลังทำอยู่

- Info

- 1)  (Display a message) แสดงไอคอนบล็อกข้อความ
- 2)  (Show the game information) ใช้แสดงข้อมูลต่างๆ ในเกม
- 3)  (Show a video file) ใช้เล่นไฟล์วิดีโอจากภายนอก

- Game

- 1)  (Restart the game) เล่นเกมทั้งหมดใหม่อีกครั้ง
- 2)  (End the game) ออกจากเกม
- 3)  (Save the game) เซฟเกม
- 4)  (Load the game) โหลดเกมที่เรเซฟไว้

- Resources

- 1)  (Replace a sprite from file) ใช้โหลดภาพจากภายนอกเข้ามาแทนที่ sprite ในโปรแกรม
- 2)  (Replace a sound from file) ใช้โหลดไฟล์เสียงจากภายนอกเข้ามาแทนที่
- 3)  (Replace a background from file) ใช้โหลดภาพพื้นหลังจากภายนอกเข้ามาแทนที่

● Control

- Questions

- 1)  (If a position is collision free) เป็นเงื่อนไข หมายถึง ถ้าตำแหน่ง x และ y ที่กำหนดไม่มีวัตถุใดๆอยู่ ณ ตำแหน่งนั้นๆสามารถเลือกได้เฉพาะ Solid หรือวัตถุทั้งหมด
- 2)  (If there is a collision at a position) เป็นเงื่อนไข หมายถึง ถ้ามีการชนกันที่ตำแหน่ง x และ y ที่กำหนดสามารถเลือกได้เฉพาะ Solid หรือวัตถุทั้งหมด
- 3)  (If there is an object at a position) เป็นเงื่อนไข หมายถึง ถ้ามีออบเจกต์ที่ตำแหน่ง x และ y ที่กำหนดสามารถเลือกได้ว่าเป็นออบเจกต์ใด
- 4)  (If the number of instances is a value) เป็นเงื่อนไข หมายถึง ถ้ามีจำนวนวัตถุตามที่กำหนด สามารถเลือกออบเจกต์ใด
- 5)  (With a change perform next action) เป็นเงื่อนไข หมายถึง มีโอกาสที่จะเป็นจริงคล้ายกับการทอดลูกเต๋า
- 6)  (If the user answers yes to a question) เป็นเงื่อนไข หมายถึง ให้ผู้เล่นตอบคำถาม
- 7)  (If an expression is true) เป็นเงื่อนไข หมายถึง ถ้าเงื่อนไขที่กำหนดไว้เป็นจริง
- 8)  (If a mouse button is pressed) เป็นเงื่อนไข หมายถึง ถ้ามีการคลิกเมาส์ค้างไว้สามารถเลือกได้ว่าจะเป็นคลิกขวา คลิกซ้าย คลิกกลาง หรือไม่คลิก

- 9)  (If instance is aligned with grid) เป็นเงื่อนไข หมายถึง ถ้าวัตถุ snap ที่ตำแหน่งที่กำหนดไว้ สามารถกำหนด snap ได้แนวนอนและแนวตั้ง

- Other

- 1)  (Start of block) เครื่องหมายปีกกาเปิด ใช้เริ่มถัดจากเงื่อนไข
- 2)  (End of block) เครื่องหมายปีกกาปิด ใช้จบประโยคเงื่อนไข
- 3)  (Else) เมื่อเงื่อนไขที่ผ่านมาเป็นเท็จ เงื่อนไขนี้จะเป็นจริงทันที
- 4)  (Repeat next action) เป็นเงื่อนไข หมายถึง ให้ทำ action ภายในวนซ้ำเป็นจำนวนกี่รอบก็ได้ตามที่กำหนด
- 5)  (Exit the current event) เมื่อโปรแกรมรันมาเจอกับ action นี้ก็จะออกจาก event นั้นทันที
- 6)  (Call the inherited event) ใช้ action นี้เมื่อเราตั้ง parent ไว้ที่ขอบเจ็ดใดๆ โปรแกรมจะทำ action เดียวกันใน parent ซึ่งอยู่ใน event เดียวกัน

- Code

- 1)  (Execute a piece of code) ใช้เขียนสคริปต์ต่างๆไป
- 2)  (Execute a script) ใช้สำหรับเรียกสคริปต์ที่เราเขียนไว้เรียบร้อยแล้ว
- 3)  (Comment) ใช้สำหรับใส่ข้อมูลหรือโน้ตสั้นๆไว้เพื่อเตือนความจำ

- Variables

- 1)  (Set the value of a variable) การตั้งค่าตัวแปรมีความสำคัญมาก เนื่องจากค่าทุกค่าในเกมจะถูกเก็บไว้ในตัวแปรที่เราตั้งขึ้น
- 2)  (If a variable has a value) เป็นเงื่อนไข หมายถึง ถ้าค่าตัวแปรเป็นไปอย่างที่กำหนด สามารถเลือกเครื่องหมายปฏิบัติการมากกว่า น้อยกว่า หรือเท่ากับได้
- 3)  (Draw the value of a variable) ใช้ใน draw event สำหรับแสดงค่าตัวแปรที่กำหนดบนหน้าจอตามตำแหน่งที่ต้องการ

- Score

- Score

- 1)  (Set the score) ใช้กำหนดคะแนนตามต้องการ
- 2)  (If score has a value) เป็นเงื่อนไข หมายถึง ถ้าคะแนนเป็นไปอย่างที่กำหนด สามารถเลือกเครื่องหมายปฏิบัติการมากกว่า น้อยกว่า หรือเท่ากับได้
- 3)  (Draw the value of score) ใช้ใน draw event สำหรับแสดงค่า score ที่กำหนด บนหน้าจอตามตำแหน่งที่ต้องการ
- 4)  (Display the high score table) สำหรับเก็บ high score
- 5)  (Clear the high score table) สำหรับลบ high score ทั้งหมดที่บันทึกไว้

- Lives

- 1)  (Set the number of lives) ใช้กำหนดค่าชีวิตตามต้องการ
- 2)  (If lives is a value) เป็นเงื่อนไข หมายถึง ถ้าค่าชีวิตเป็นไปอย่างที่กำหนด สามารถเลือกเครื่องหมายปฏิบัติการมากกว่า น้อยกว่า หรือเท่ากับได้
- 3)  (Draw the number of lives) ใช้ใน draw event สำหรับแสดงค่า lives ที่กำหนดบนหน้าจอตามตำแหน่งที่ต้องการ
- 4)  (Draw the lives as image) ใช้ใน draw event สำหรับแสดงจำนวนชีวิตเป็นรูปภาพที่เราต้องการ ที่กำหนดบนหน้าจอตามตำแหน่งที่ต้องการ

- Health

- 1)  (Set the health) ใช้กำหนดค่าพลังตามต้องการ ค่าจะอยู่ระหว่าง 1-100
- 2)  (If health is a value) เป็นเงื่อนไข หมายถึง ถ้าค่าพลังเป็นไปอย่างที่กำหนด สามารถเลือกเครื่องหมายปฏิบัติการมากกว่า น้อยกว่า หรือเท่ากับได้
- 3)  (Draw the health bar) ใช้ใน draw event สำหรับแสดงค่า health บนหน้าจอ เป็นแถบพลัง

- 4)  (Set the window caption information) สามารถกำหนดข้อความบนส่วนหัว window ได้ว่าจะให้แสดงค่า score, lives หรือ health หรือไม่

● Extra

- Particles

- 1)  (Create the particle system) ทุกครั้งที่สร้าง particle ต้องใช้คำสั่งนี้ก่อนทุกครั้งเสมอเพื่อประกาศให้โปรแกรมทราบสามารถกำหนดความลึกของ particle ได้
- 2)  (Destroy the particle system) เมื่อไม่ต้องการใช้ particle อีกต่อไปให้ใช้คำสั่งนี้ทำลาย particle system ที่สร้างขึ้นมาทิ้ง
- 3)  (Clear all particles in the system) เมื่อต้องการลบการแสดงผลของ particle บนหน้าจอทั้งหมดทันที
- 4)  (Create a type of particle) ใช้สร้างรูปแบบของ particle
- 5)  (Set the color for a particle type) สำหรับกำหนดสีให้กับ particle
- 6)  (Set the life time for a particle type) สำหรับกำหนดระยะเวลาของ particle
- 7)  (Set the motion for a particle type) สำหรับกำหนดความเร็วของ particle
- 8)  (Set the gravity of a particle type) สำหรับทำให้ particle แต่ละชิ้นที่เกิดขึ้นมีแรงโน้มถ่วง
- 9)  (Create secondary particles) สำหรับสร้าง particle ต่อเนื่องจาก particle แต่ละชิ้นที่ถูกสร้างออกมาตอนแรก
- 10)  (Create a particle emitter) สำหรับสร้างตำแหน่งของ particle ซึ่งสามารถเซตค่าแตกต่างกันได้ 8 แบบ
- 11)  (Destroy an emitter) ใช้ทำลาย emitter ทิ้ง โดยการเลือก id ที่ต้องการ
- 12)  (Burst a number of particles from an emitter) ใช้แสดง particle ตามตำแหน่งขอบที่เลือก

- 13)  (Stream particles from an emitter) การตั้งค่าและผลลัพธ์เหมือนกับ Burst a number of particles from an emitter ทุกประการเพียงแต่คำสั่งนี้จะสร้างparticle ใน ทุกๆ step

- CD

- 1)  (Play a CD) สั่งให้เล่นแผ่นซีดี
- 2)  (Stop the CD) สั่งให้หยุดเล่นแผ่นซีดี
- 3)  (Pause the CD) สั่งให้หยุดเล่นแผ่นซีดีชั่วคราว
- 4)  (Resume the CD) สั่งให้เล่นแผ่นซีดีต่อไปหลังจากหยุดชั่วคราว
- 5)  (If a CD exists in the drive) เงื่อนไข หมายถึง ถ้ามีแผ่นซีดีอยู่ในไดรฟ์
- 6)  (If the CD is playing) เงื่อนไข หมายถึง ถ้ามีแผ่นซีดีกำลังถูกเล่น

- Other

- 1)  (Set the mouse cursor) สามารถเปลี่ยนรูปเมาส์ให้เป็นภาพอื่นๆที่ต้องการได้
- 2)  (Open a webpage in browser) สามารถสั่งเปิดเว็บไซต์ได้โดยใส่ที่อยู่ของเว็บ
ลงไป

● Draw

- Drawing

- 1)  (Draw a sprite image) ใช้ใน draw event สำหรับแสดงภาพ sprite
- 2)  (Draw a background image) ใช้ใน draw event สำหรับแสดงภาพพื้นหลัง
- 3)  (Draw a text) ใช้ใน draw event สำหรับแสดงข้อความต่างๆ
- 4)  (Draw a text transformed) ใช้ใน draw event คล้ายกับคำสั่ง Draw a text แต่
สามารถกำหนดสัดส่วน เพิ่มลดขนาดได้

- 5)  (Draw a rectangle) ใช้ใน draw event สำหรับวาดสี่เหลี่ยม
- 6)  (Draw a horizontal gradient) ใช้ใน draw event สำหรับวาดสี่เหลี่ยมซึ่งมีการไล่สีตามแนวนอน
- 7)  (Draw a vertical gradient) ใช้ใน draw event สำหรับวาดสี่เหลี่ยมซึ่งมีการไล่สีตามแนวตั้ง
- 8)  (Draw an ellipse) ใช้ใน draw event สำหรับวาดวงกลม
- 9)  (Draw a gradient ellipse) ใช้ใน draw event สำหรับวาดวงกลมซึ่งมีการไล่สีจากด้านในไปด้านนอก
- 10)  (Draw a line) ใช้ใน draw event สำหรับวาดเส้นตรง
- 11)  (Draw an arrow) ใช้ใน draw event สำหรับวาดเส้นตรงซึ่งมีหัวเป็นลูกศร

- Setting

- 1)  (Set the colors) ก่อนที่จะมีการวาดสิ่งต่างๆลงไปบนหน้าจอ ใช้คำสั่งนี้หน้าหน้าก่อน ก็จะสามารถกำหนดสีของสิ่งที่เราจะวาดต่อไปได้
- 2)  (Set a font for drawing text) ก่อนที่จะมีการแสดงตัวอักษรสิ่งต่างๆลงไปบนหน้าจอ ใช้คำสั่งนี้หน้าหน้าก่อน ก็จะสามารถเลือกฟอนต์ที่เราตั้งไว้แล้ว
- 3)  (Change full screen mode) ใช้เปลี่ยนหน้าจอเป็น full screen เต็มจอ หรือแบบ window หรือ switch

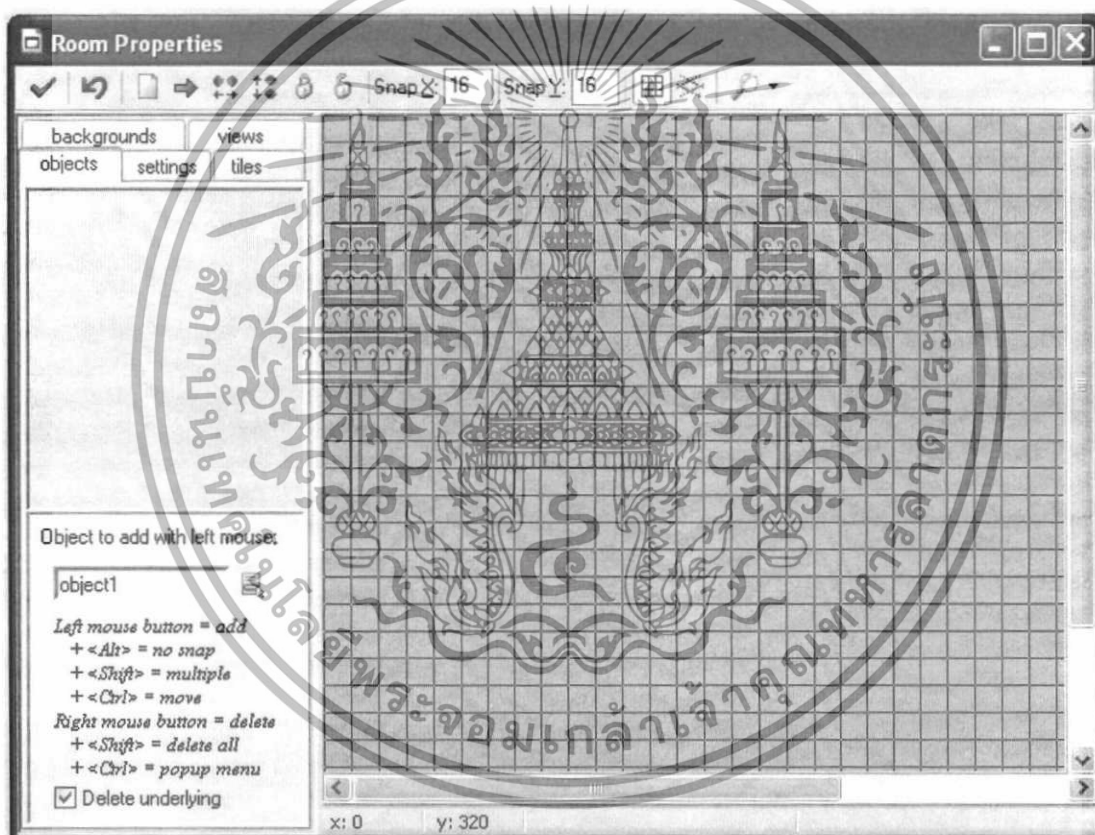
- Other

- 1)  (Take a snapshot image of the game) สามารถเซฟภาพหน้าจอของเกมออกไปเป็นไฟล์ภาพได้
- 2)  (Create an effect) สามารถใช้คำสั่งนี้ สร้างเอฟเฟกต์ในเกมได้ซึ่งเป็นเอฟเฟกต์สำเร็จรูป

2.2.2.9 Room Resource

Room คือหน้าจอหรือขอบเขตที่ใช้สำหรับเล่นเกมซึ่งจะแสดงภาพกราฟิกทุกอย่างในเกม เราสามารถนำวัตถุต่างๆที่สร้างขึ้นเข้าไปวางใน room ตามตำแหน่งที่ต้องการได้ รวมถึงการสร้างพื้นหลังของเกม การสร้างขอบเขตมุมมองที่ต้องการได้ การกำหนดขนาดหน้าต่างในการเล่นเกม และการกำหนดความเร็ว

เมื่อเรากดคลิกขวาที่โฟลเดอร์นี้แล้วเลือก Add room จะปรากฏหน้าต่าง room properties ขึ้นมาดังรูป



รูปที่ 2.65 หน้าจอคำสั่ง Room Properties

- Room Properties

เมนูบาร์

- 1)  (Close the form and saving) ยืนยันการแก้ไขและการปิดหน้าต่าง
- 2)  (Undo) กลับไปที่การกระทำครั้งที่แล้ว
- 3)  (clear) ลบ instance ทั้งหมดใน room
- 4)  (shift) เลื่อน instance ทั้งหมดไปในแนวนอนและแนวตั้งได้ตามค่าที่กำหนด
- 5)  (Sort all instance by x) เมื่อมีวัตถุอยู่ใน room เป็นจำนวนมาก ซึ่งซ้อนทับกันอยู่ใน depth เดียวกัน วัตถุที่ตำแหน่ง x ใน room น้อยจะอยู่ด้านล่าง ส่วนวัตถุที่ตำแหน่ง x ใน room มากจะอยู่ด้านบนของวัตถุที่มีค่า x น้อยกว่า
- 6)  (Sort all instance by y) เมื่อมีวัตถุอยู่ใน room เป็นจำนวนมาก ซึ่งซ้อนทับกันอยู่ใน depth เดียวกัน วัตถุที่ตำแหน่ง y ใน room น้อยจะอยู่ด้านล่าง ส่วนวัตถุที่ตำแหน่ง y ใน room มากจะอยู่ด้านบนของวัตถุที่มีค่า y น้อยกว่า
- 7)  (Lock all instance) ทำให้ไม่สามารถลบ instance ใดๆใน room ได้ เป็นการป้องกันการเผลอลบโดยไม่ได้ตั้งใจ
- 8)  (Unlock all instance) ทำให้สามารถแก้ไข instance ใน room ได้ตามปกติ
- 9)  Snap X: 16 (Snap X) กำหนดระยะกริดในแกน x ที่จะใช้ snap วัตถุที่นำมาวางใน room
- 10)  Snap Y: 16 (Snap Y) กำหนดระยะกริดในแกน y ที่จะใช้ snap วัตถุที่นำมาวางใน room
- 11)  (Toggle the showing of the grid) เปิดหรือปิดกริดใน room
- 12)  (Turn the grid into an isometric grid) เปลี่ยนมุมมองของกริดเป็นแบบ isometric กรณีต้องการทำเกมในรูปแบบ isometric
- 13)  (Indicate which aspects of the room) สามารถเลือกให้แสดงหรือไม่แสดงสิ่งต่างๆใน room ได้

บทที่ 3

ขั้นตอนการดำเนินการ

การสร้างเกมที่ดีนั้นควรที่จะมีการวางแผนให้รอบคอบ ก่อนที่จะทำการสร้างเกมรวมถึงการคำนึงถึงความเป็นไปได้ว่าจะสามารถพัฒนาได้หรือไม่ ดังนั้นการเขียนโครงสร้างของเกมก่อนการพัฒนาจึงเป็นเรื่องที่ควรกระทำโดยเฉพาะกับตัวเกมที่มีขนาดใหญ่ มีแผนงานทั้งหมดดังต่อไปนี้

1) ขั้นตอนการวิเคราะห์และออกแบบเกม

- การวิเคราะห์ความต้องการว่าต้องการจะสร้างเกมในรูปแบบใด เนื้อหาทั้งหมดมีลักษณะอย่างไร ออกแบบกติกาการเล่น ระบบของเกม ว่าควรมีรูปแบบใด
- ออกแบบหน้าจอส่วนที่ใช้ติดต่อกับผู้เล่น ซึ่งต้องเป็นรูปแบบที่ใช้ง่าย และดึงดูดความสนใจของผู้เล่น
- ออกแบบวิธีการจัดเก็บข้อมูลที่จะต้องใส่ภายในเกม โดยที่ต้องใช้พื้นที่ในการจัดเก็บน้อย เป็นระเบียบและสามารถนำมาใช้ได้ง่ายเพื่อเพิ่มประสิทธิภาพการทำงานให้กับเกม
- ออกแบบวิธีการเลือกใช้อาร์ตเวิร์ค,ซอฟต์แวร์ รวมถึงComponent และเครื่องมือ ต่างๆที่นำมาใช้ในการสร้างเกม

2) ขั้นตอนการลงมือปฏิบัติการจริง

- ด้านภาพและกราฟิก การทำงานในส่วนนี้จะใช้โปรแกรมทางด้านกราฟิกเข้ามาช่วยในการสร้างฉากที่ได้ทำการออกแบบไว้แล้ว
- ด้านโปรแกรมมิ่ง การทำงานในส่วนนี้จะเกี่ยวข้องกับการเขียนคำสั่งต่างๆโดยมีการนำฉากตัวละคร และเสียงประกอบ ที่ได้สร้างไว้มาใช้ในส่วนนี้ซึ่งมีการจัดลำดับขั้นตอนการทำงานดังนี้
 - ใช้โปรแกรม Game Maker ในส่วนของการเขียนและแก้ไข ส่วนต่างๆของเกม
 - นำฉาก ตัวละคร และเสียงมาประกอบเข้าด้วยกัน
 - ใช้คำสั่งที่ทำให้เกิดการกระทำตามที่ต้องการ

3) ขั้นตอนการทดสอบเกม การทำงานในส่วนนี้จะขั้นตอนในการทดสอบเกมที่เรารผลิตขึ้นว่ามีความสมบูรณ์มากน้อยเพียงใด เพื่อหาข้อผิดพลาดและประเมินประสิทธิภาพของเกม เมื่อพบข้อผิดพลาดก็จะทำการแก้ไขโดยทันที

3.1 ขั้นตอนการวิเคราะห์และการออกแบบ

การออกแบบระบบเกมเป็นส่วนที่สำคัญที่สุดในขั้นตอนการออกแบบทั้งหมด เพราะระบบเกม คือ ทุกสิ่งทุกอย่างตั้งแต่แนวคิดของกม การดำเนินเรื่องของเกม จนถึงการทำงานของระบบภายในเกม โดยการออกแบบมีรายละเอียดดังต่อไปนี้

3.1.1 การออกแบบระบบเกมและการกำหนดโครงสร้างของเกม

เป็นการกำหนดลักษณะของเกมแต่ละเกม ตามที่เราต้องการเพื่อให้ภายในเกมนั้นเป็นไปตามที่เราต้องการ โดยแบ่งรูปแบบเกมออกเป็นทั้งหมด 3 เกม ซึ่งมีลักษณะและวิธีการเล่นที่แตกต่างกันดังนี้

3.1.1.1 เกมกินเมือง

รูปแบบของเกมเป็นการฝึกการใช้ทักษะความคิดในการเล่น โดยกำหนดออกเป็นสองฝ่ายคือฝ่ายผู้เล่นและฝ่ายคอมพิวเตอร์เพื่อแข่งขันกันกินเมือง

ซึ่งกติกาการเล่นเกมนี้นักเล่นจะต้องกินเมือง(จุด)ให้ได้มากกว่าอีกฝ่ายถึงจะถือว่าชนะ โดยที่เราหมุนจุดตามลูกศรที่กำหนดไปยังทิศที่มีจุดว่างซึ่งการเล่นหนึ่งครั้งจะหมุนได้ 1 ทิศเท่านั้น ทิศตามเข็มนาฬิกาไปยังจุดที่ว่างอื่นๆโดยจะมีเส้นเชื่อมและสีที่บ่งบอกจำนวนจุดของผู้เล่นแต่ละฝ่าย ในการเล่นนี้จะกระทำโดยการผลัดกันคนละครั้งทั้งสองฝ่ายจนกว่า จุดทั้งหมดที่มีอยู่ไม่มีจุดใดว่างหรือฝ่ายใดฝ่ายหนึ่งกินจุดของอีกฝ่ายได้จนหมดจะถือว่าเป็นการจบเกมและเริ่มเกมใหม่ ซึ่งในเกมนี้มีการแบ่งระดับความยากง่ายถึง 3 ระดับ

3.1.1.2 เกมโยนลูกเต๋าร้อยห่วง

รูปแบบของเกมมีลักษณะคล้ายเกมกีฬา เป็นการโยนลูกเต๋าร้อยห่วงให้ลงห่วงเพื่อทำคะแนน เน้นความสนุกสนานเพลิดเพลิน

ซึ่งกติกาการเล่นผู้เล่นจะต้องทำการโยนลูกเต๋าร้อยห่วงให้ลงห่วง โดยที่ใช้ปุ่ม Space bar ในการบังคับความแรงของลูก ปุ่มลูกศร บังคับทิศทางในการโยนลูกเพื่อทำคะแนน และปุ่มตัวเลขในการเปลี่ยนระยะการยิงลูก

3.1.1.3 เกมเป็ดน้อยมหัศจรรย์

รูปแบบของเกมมีลักษณะเป็นเกมแนว Shooting อาศัยทักษะการบังคับหลบหลีกเป็นหลัก

ซึ่งกติกาการเล่นผู้เล่นจะต้องอาศัยทักษะการบังคับหลบหลีก และยังต่อสู้กับศัตรูเพื่อทำคะแนนและเพิ่มระดับความแรงของกระสุน

3.1.2 การออกแบบรูปภาพและกราฟิก

รูปแบบของภาพกราฟิกในเกม มีลักษณะของรูปภาพเป็น 2 มิติ ใช้สีสดใสเพื่อดึงดูดความสนใจผู้เล่น ในเกมประกอบด้วยหน้าจอต่างๆดังนี้ หน้าจอหลักแสดงเมนูต่างๆเพื่อเข้าสู่เกม หน้าจอขณะเล่นเกม หน้าจอโหลดเกมเพื่อเข้าสู่การเล่นเกมที่ต่อจากครั้งที่แล้ว หน้าจอเพื่อออกจากเกม ซึ่งโปรแกรมที่นำมาใช้สำหรับสร้างและตกแต่งภาพที่เลือกใช้สำหรับเกมนี้ได้แก่ โปรแกรม Adobe Photoshop CS

3.1.3 การออกแบบ Interface สำหรับติดต่อและโต้ตอบกับผู้เล่น

ในเกม เราใช้อุปกรณ์ที่ใช้ติดต่อกับผู้เล่น 2 อย่าง คือ เมาส์ และ คีย์บอร์ด โดยรูปแบบการใช้งานของแต่ละเกมจะแตกต่างกันออกไป

3.1.4 การออกแบบการจัดเก็บข้อมูล

ข้อมูลที่เป็นที่จำเป็นที่ใช้ในเกมนี้ที่จะต้องจัดเก็บประกอบไปด้วยพารามิเตอร์ที่จำเป็นต่างๆ ของเกม สำหรับไฟล์ทั้งหมดนั้นต้องการให้เข้าถึงได้อย่างรวดเร็ว และมีความซับซ้อนน้อย จึงเลือกใช้วิธีเก็บแบบText File เพื่อให้มีการเข้าถึงข้อมูลได้ง่าย และยังใช้เนื้อที่น้อยกว่าการเก็บแบบDatabase

3.1.5 การกำหนดฮาร์ดแวร์, ซอร์ฟแวร์ และเครื่องมือที่จะใช้

การเขียนเกมนี้ ได้เลือกใช้โปรแกรม Game Maker ในการสร้างเกม โดย Game Maker จะเป็นผู้ติดต่อกับฮาร์ดแวร์ต่างๆ โดยตรงโดยเฉพาะอุปกรณ์แสดงผลทั้งภาพและเสียง ส่วนซอร์ฟแวร์ที่ใช้ในการสร้างฉากและภาพได้แก่ Photoshop CS

3.2 ขั้นตอนการสร้างฉากและกราฟิกทั้งหมดที่ใช้ในเกม

ภาพกราฟิกที่ใช้เป็น 2 มิติ ซึ่งใช้ Photoshop CS ในการสร้างภาพกราฟิกโดยภาพที่ได้จะเก็บเป็นซึ่งการสร้างภาพกราฟิก ต้องสร้างรูปที่เป็นฉากหลังและรูปที่เป็นตัวละครแยกกัน จะมีภาพฉากหลังและตัวละครแบ่งออกเป็น 3 เกม ดังนี้

3.2.1 เกมกินเมือง

3.2.1.1 รูปภาพที่ใช้ในการสร้างหน้าจอหลัก

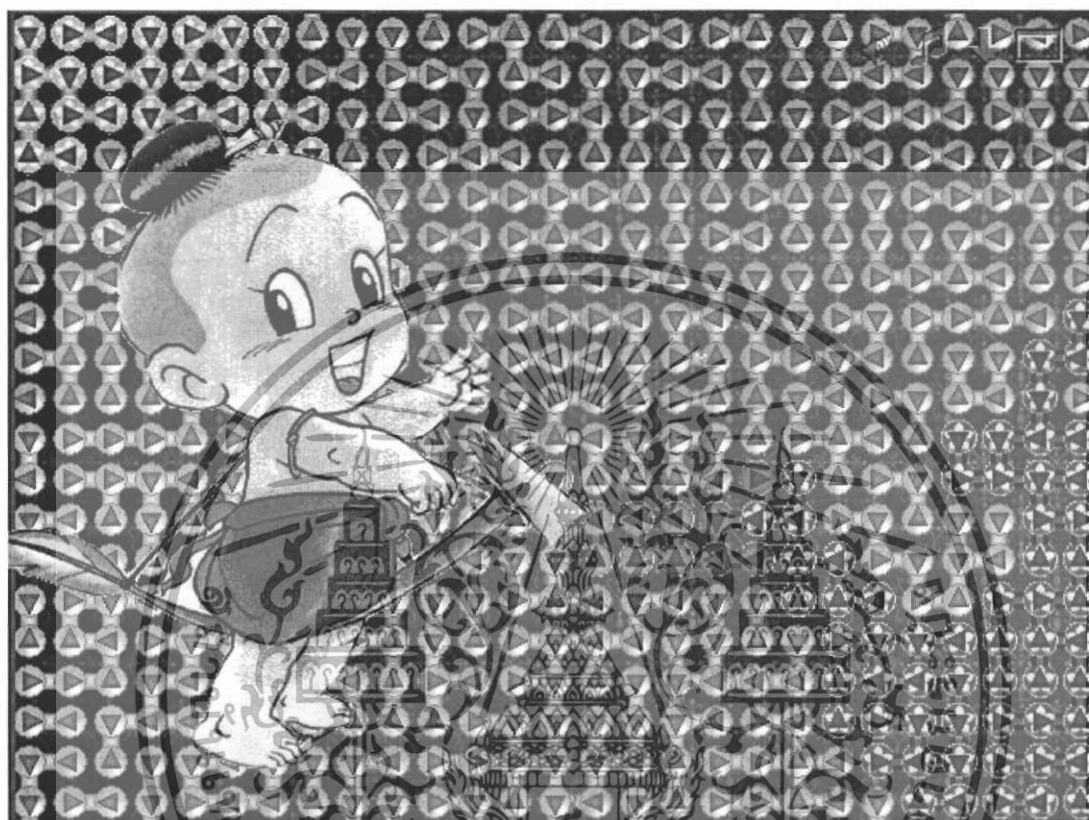
จะเป็นหน้าจอแรกเมื่อผู้เล่นทำการ โหลดเกมเพื่อเข้าเล่นเกม





รูปที่ 3.2 รูปฉากแอนิเมชันก่อนเข้าสู่หน้าจอหลักเกมกินเมือง

หลังจากจบการแสดงผลแอนิเมชัน จะเข้าสู่หน้าจอหลัก



รูปที่ 3.3 รูปฉากหลังของหน้าจอหลักเกมกินเมือง

เล่นเกม

วิธีเล่น

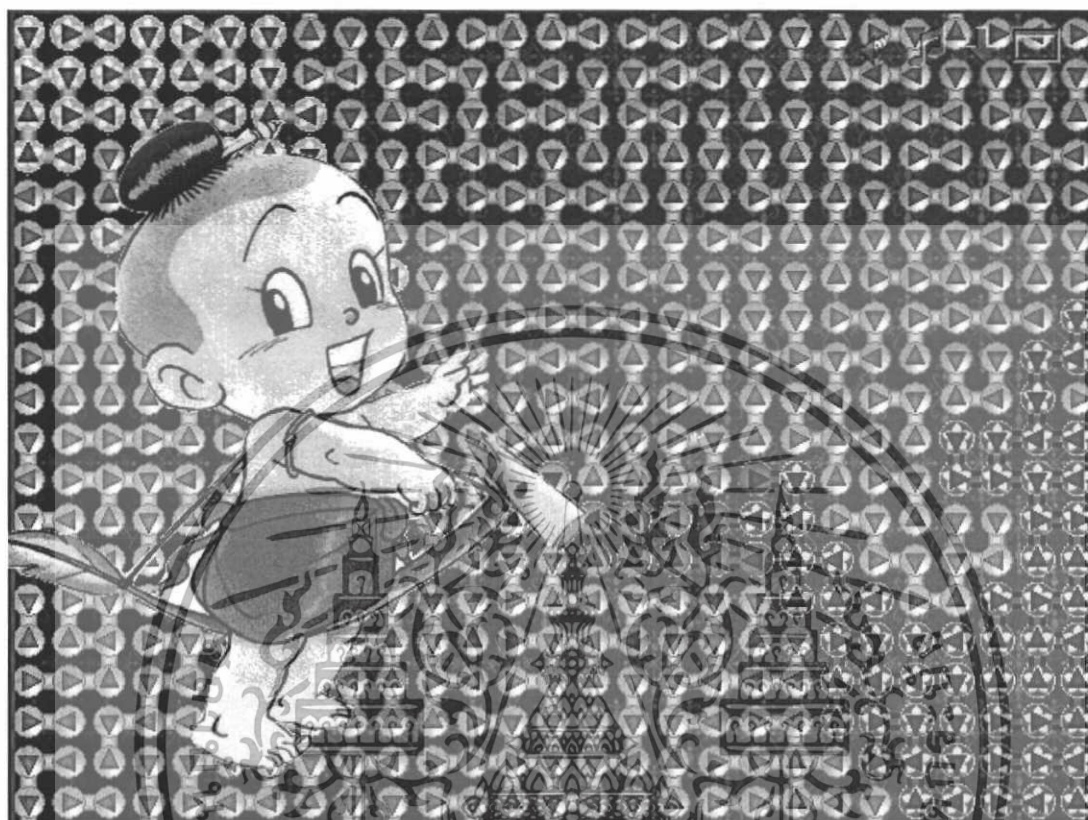
ออก

รูปที่ 3.4 รูปปุ่มต่างๆของหน้าจอหลักเกมกินเมือง



รูปที่ 3.6 รูปปุ่มออฟชั่นหน้าหลักเกมกินเมือง

3.2.1.2 รูปภาพที่ใช้ในหน้าจอการเลือกระดับการเล่นเกม



รูปที่ 3.7 รูปฉากหลังของหน้าจอการเลือกระดับเกมกินเมือง

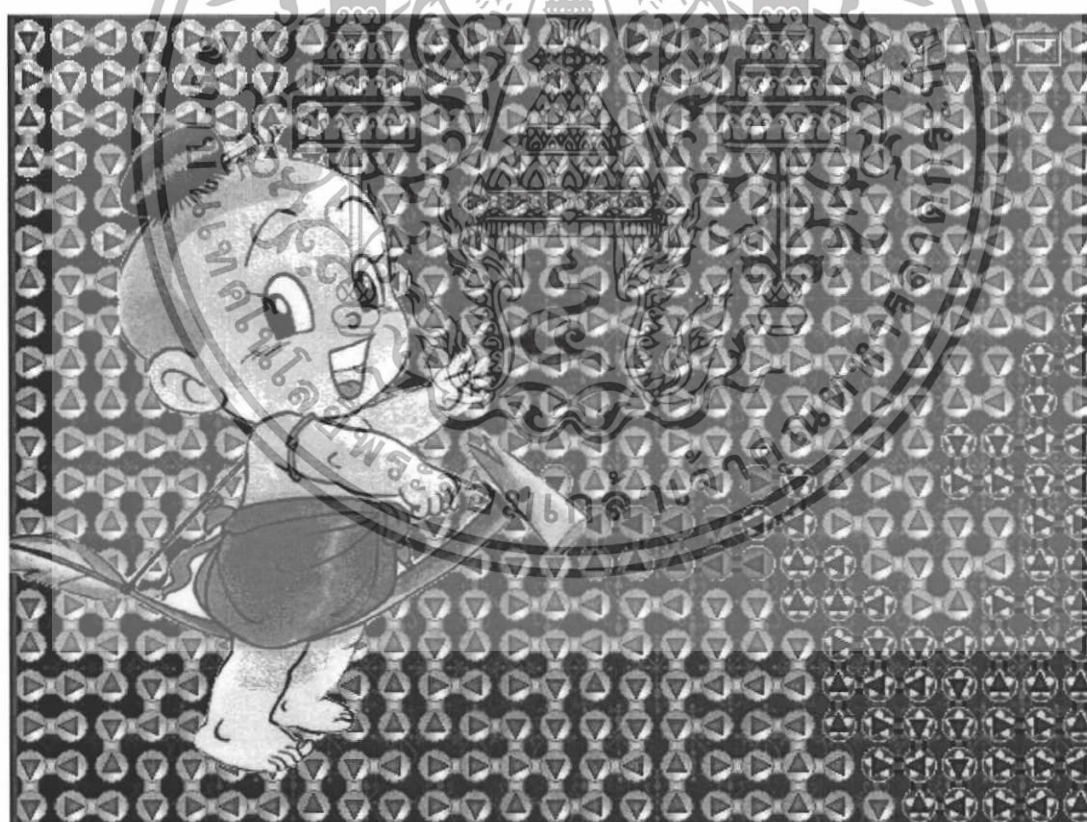


รูปที่ 3.8 รูปปุ่มต่างๆของหน้าการเลือกระดับเกมกินเมือง



รูปที่ 3.9 รูปตัวละครหน้าจอการเด็กกระดานเกมกินเมือง

3.2.1.3 รูปภาพหน้าจอแสดงวิธีการเล่นเกม

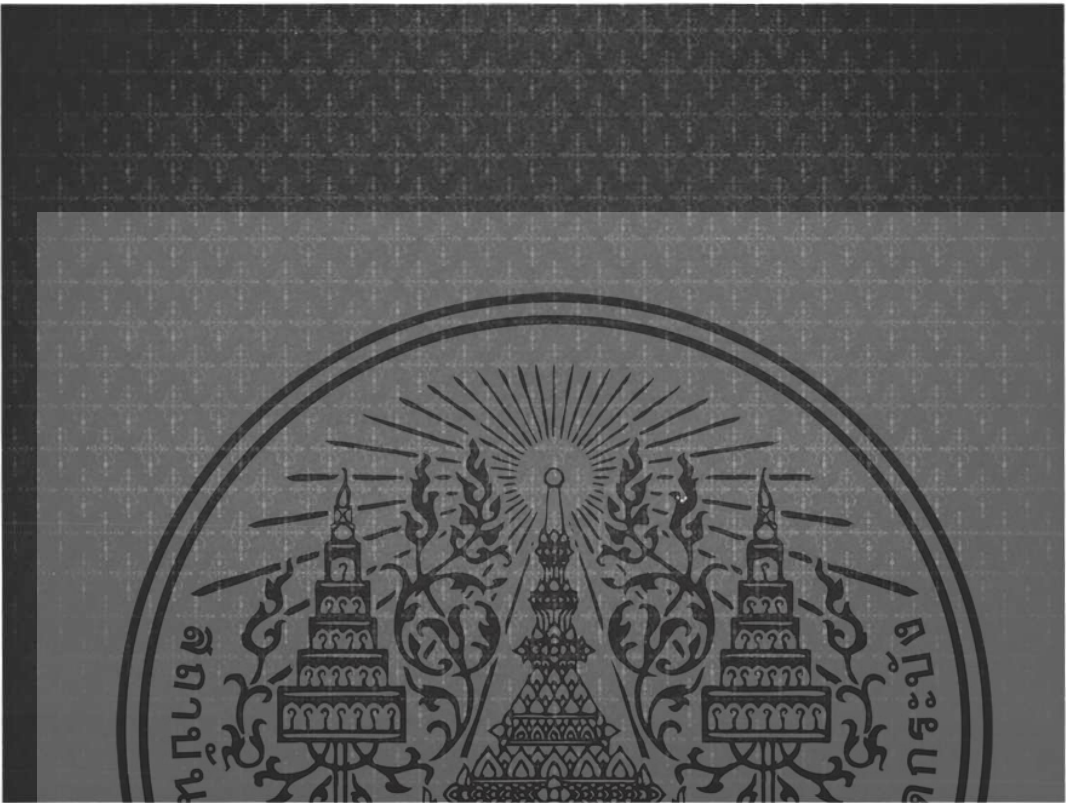


รูปที่ 3.10 รูปฉากหลังของหน้าจอแสดงวิธีการเล่นเกมกินเมือง



รูปที่ 3.12 รูปปูมต่างๆของหน้าจอวิธีการเล่นเกมกินเมือง

3.2.1.4 รูปภาพของเกม



รูปที่ 3.13 รูปฉากหลังของเกมกินเมือง

เสียง:เปิด

เสียง:ปิด

รูปที่ 3.14 รูปปุ่มปรับเสียงเกมกินเมือง



รูปที่ 3.15 รูปปุ่มปรับเพลงเกมกินเมือง



รูปที่ 3.16 รูปตัวนับคะแนนเกมกินเมือง

รูปที่ 3.17 รูปตัวนับจำนวนรอบเกมกินเมือง



รูปที่ 3.18 รูปปุ่มการออกจากเกมกินเมือง

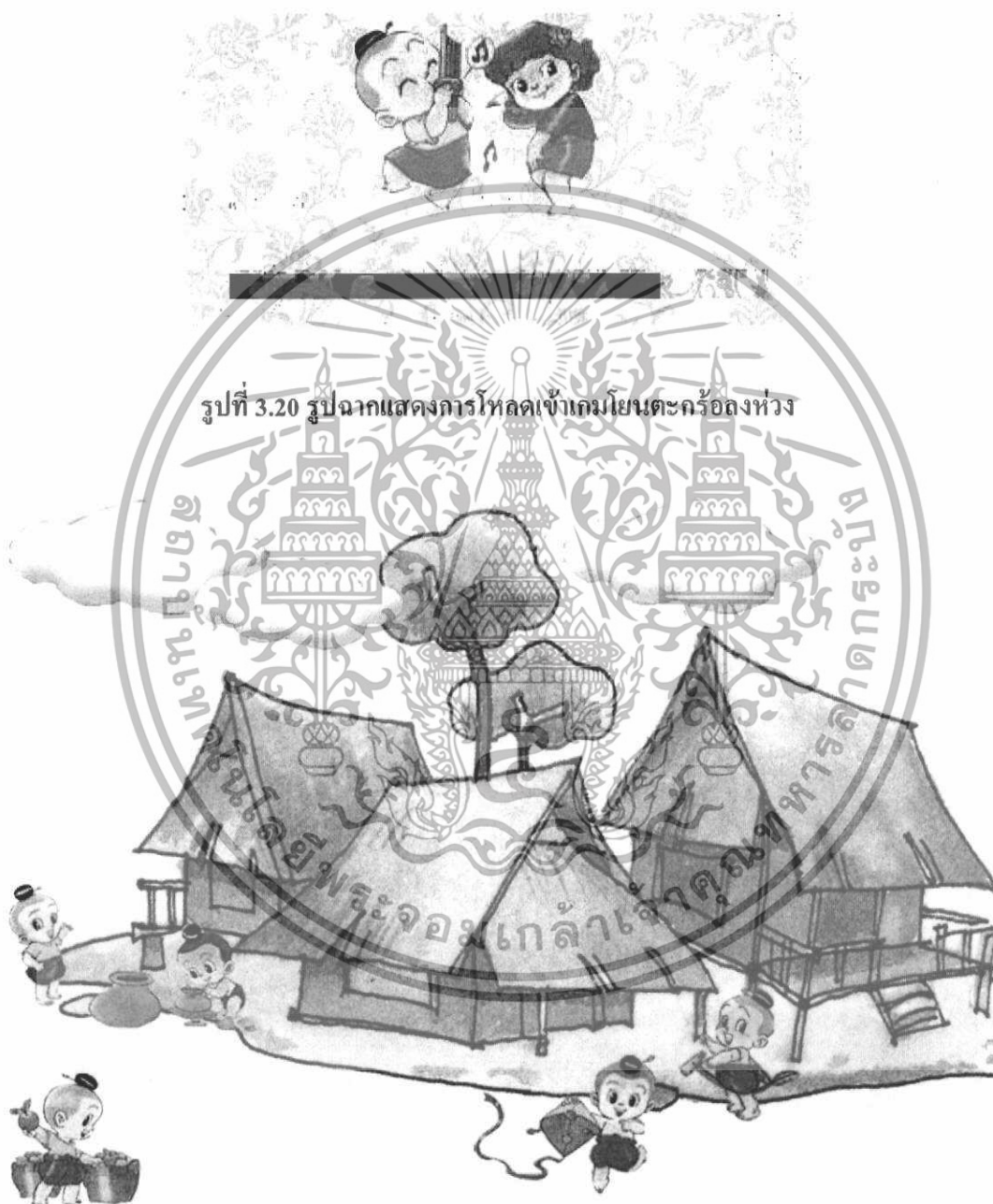


รูปที่ 3.19 รูปตัวเมืองของเกมกินเมือง

3.2.2 เกมโยนตะกร้อลงห่วง

3.2.2.1 รูปภาพที่ใช้ในการสร้างหน้าจอหลัก

จะเป็นหน้าจอแรกเมื่อผู้เล่นทำการ โหลดเกมเพื่อเข้าเล่นเกม



รูปที่ 3.21 รูปฉากหลังของหน้าจอหลักเกมโยนตะกร้อลงห่วง



รูปที่ 3.22 รูปตัวละครของฉากหลังในหน้าจอหลักเกมโยนตะกร้อลงห่วง

3.2.2.2 รูปภาพของเกม



รูปที่ 3.23 รูปตัวละครของฉากเกมโยนตะกร้อลงห่วง



รูปที่ 3.24 รูปตะกร้อ

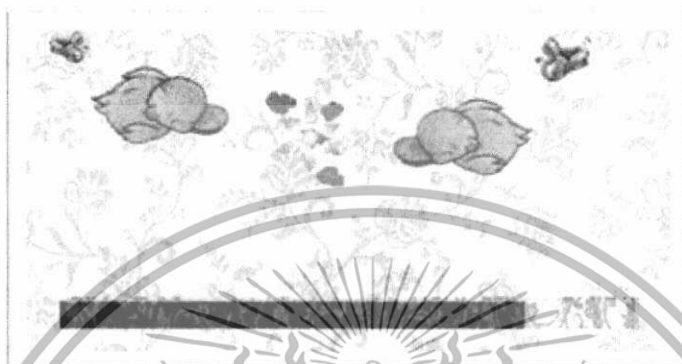


รูปที่ 3.25 รูปห้วงลักษณะต่างๆ

3.2.3 เกมเบ็ดน้อยมหัศจรรย์

3.2.3.1 รูปภาพที่ใช้ในการสร้างหน้าจอหลัก

จะเป็นหน้าจอแรกเมื่อผู้เล่นทำการโหลดเกมเพื่อเข้าเล่นเกม



รูปที่ 3.26 รูปฉากแสดงการโหลดเข้าเกมเบ็ดน้อยมหัศจรรย์

3.2.3.2 รูปภาพของเกม



รูปที่ 3.27 รูปฉากหลังของเกมเบ็ดน้อยมหัศจรรย์



รูปที่ 3.28 รูปตัวละครของฉากเกมเปิดน้อยมหัศจรรย์

SCORE:



รูปที่ 3.29 รูปแถบพลังและคะแนนของฉากเกมเปิดน้อยมหัศจรรย์

3.3 ขั้นตอนการสร้างเกม

3.3.1 เกมกินเมือง

3.3.1.1 หน้าจอหลัก

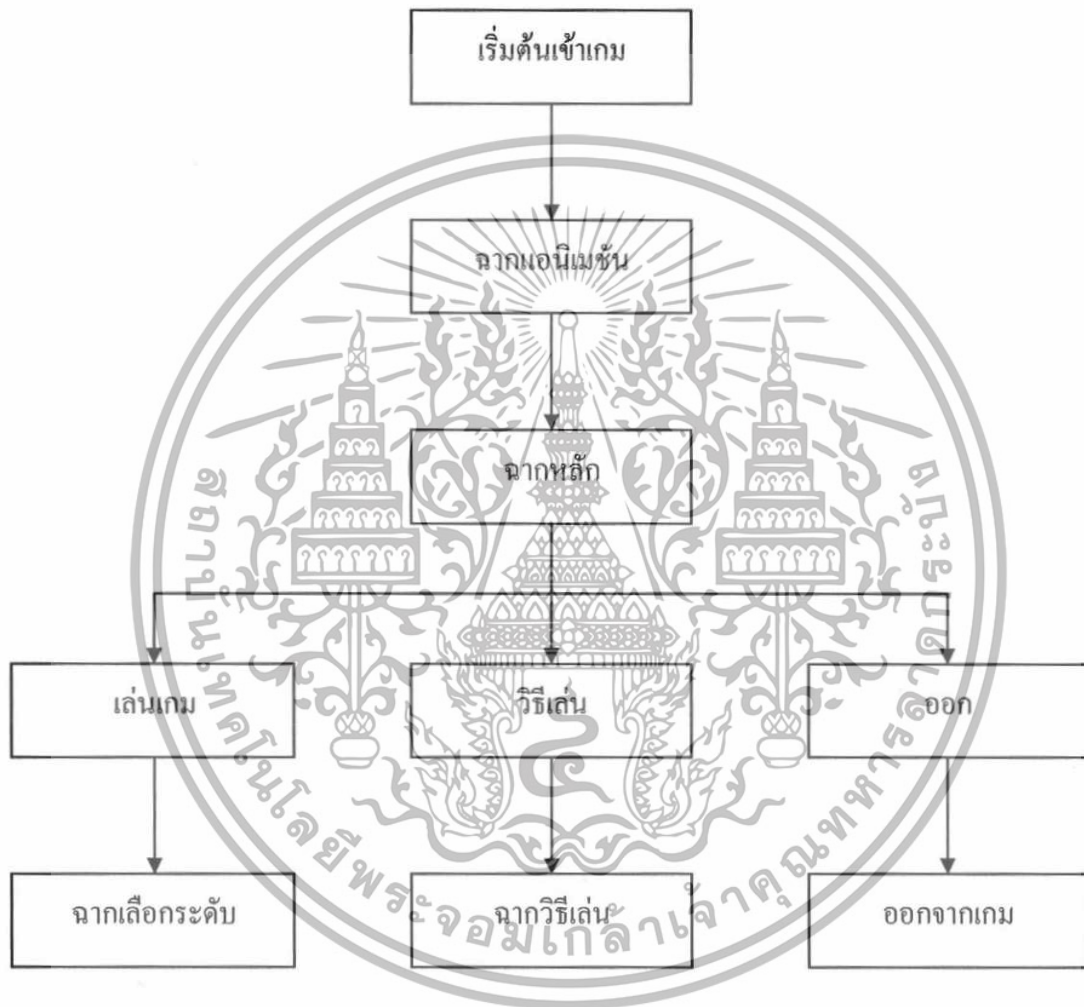
ในหน้าจอหลักมีปุ่มให้เลือกทั้งหมด 3 ปุ่ม ได้แก่

ปุ่ม **เล่นเกม** **วิธีเล่น** **ออก**



รูปที่ 3.30 รูปหน้าจอหลักเกมกินเมือง

มีแผนภาพการทำงานดังนี้



รูปที่ 3.31 แผนภาพการทำงานในหน้าจอหลักเกมกินเมือง

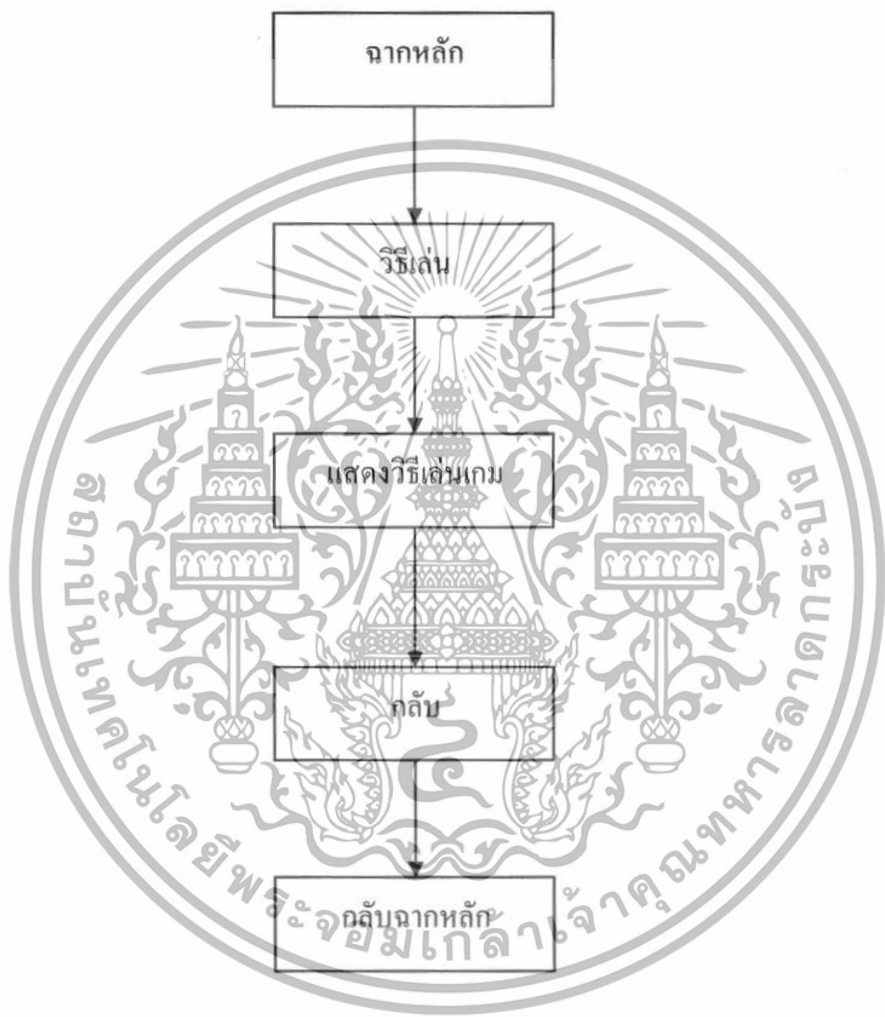
3.3.1.2 หน้าจอแสดงวิธีการเล่น

ในหน้าจอแสดงวิธีการเล่นนี้ จะมีภาพแสดงรายละเอียดวิธีการเล่นเกมกินเมืองพร้อมภาพตัวอย่าง เมื่อผู้เล่นทำความเข้าใจวิธีการเล่นแล้วต้องการกลับไปหน้าจอหลักก็สามารถกดปุ่ม **กลับ**



รูปที่ 3.32 รูปหน้าจอวิธีการเล่นเกมกินเมือง

มีแผนภาพการทำงานดังนี้

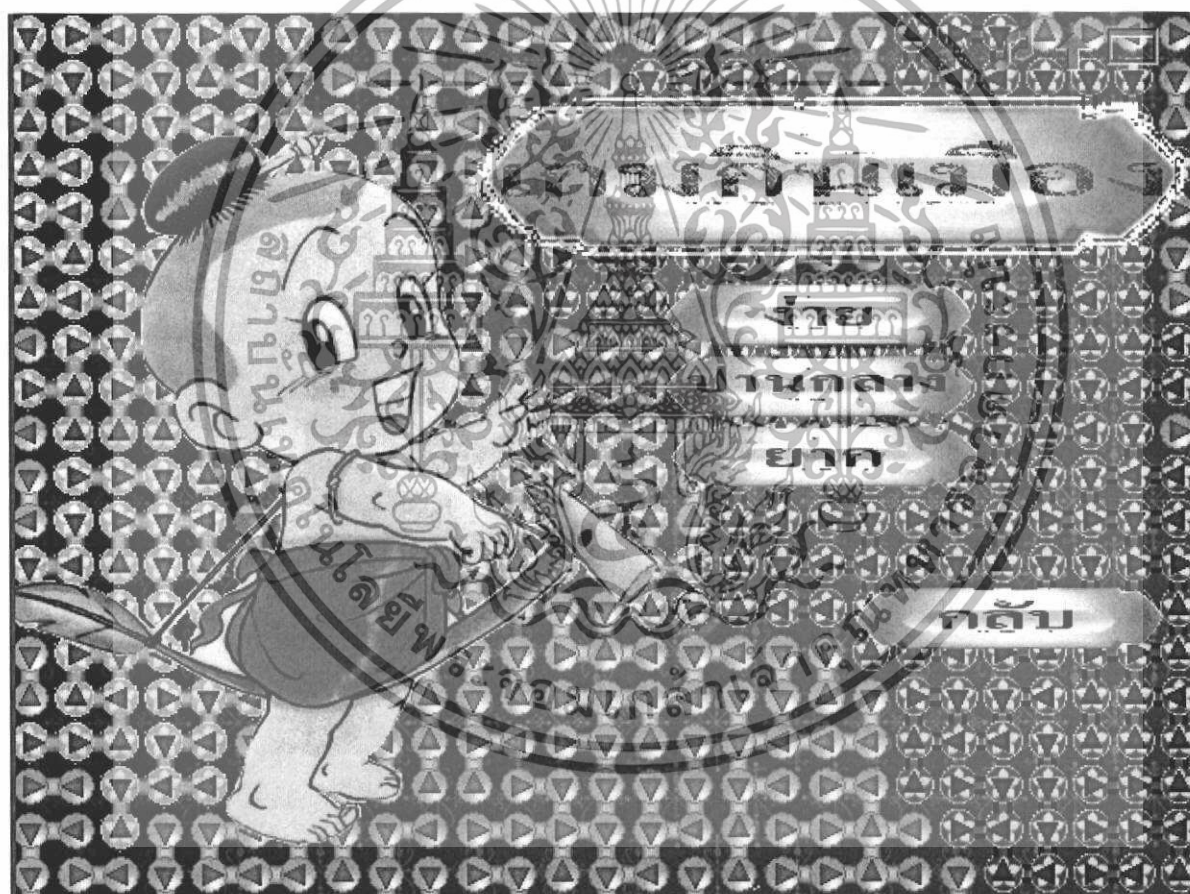


รูปที่ 3.33 แผนภาพการทำงานในหน้าจอวิธีการเล่นเกมกินเมือง

3.3.1.3 หน้าจอการเลือกระดับเกม

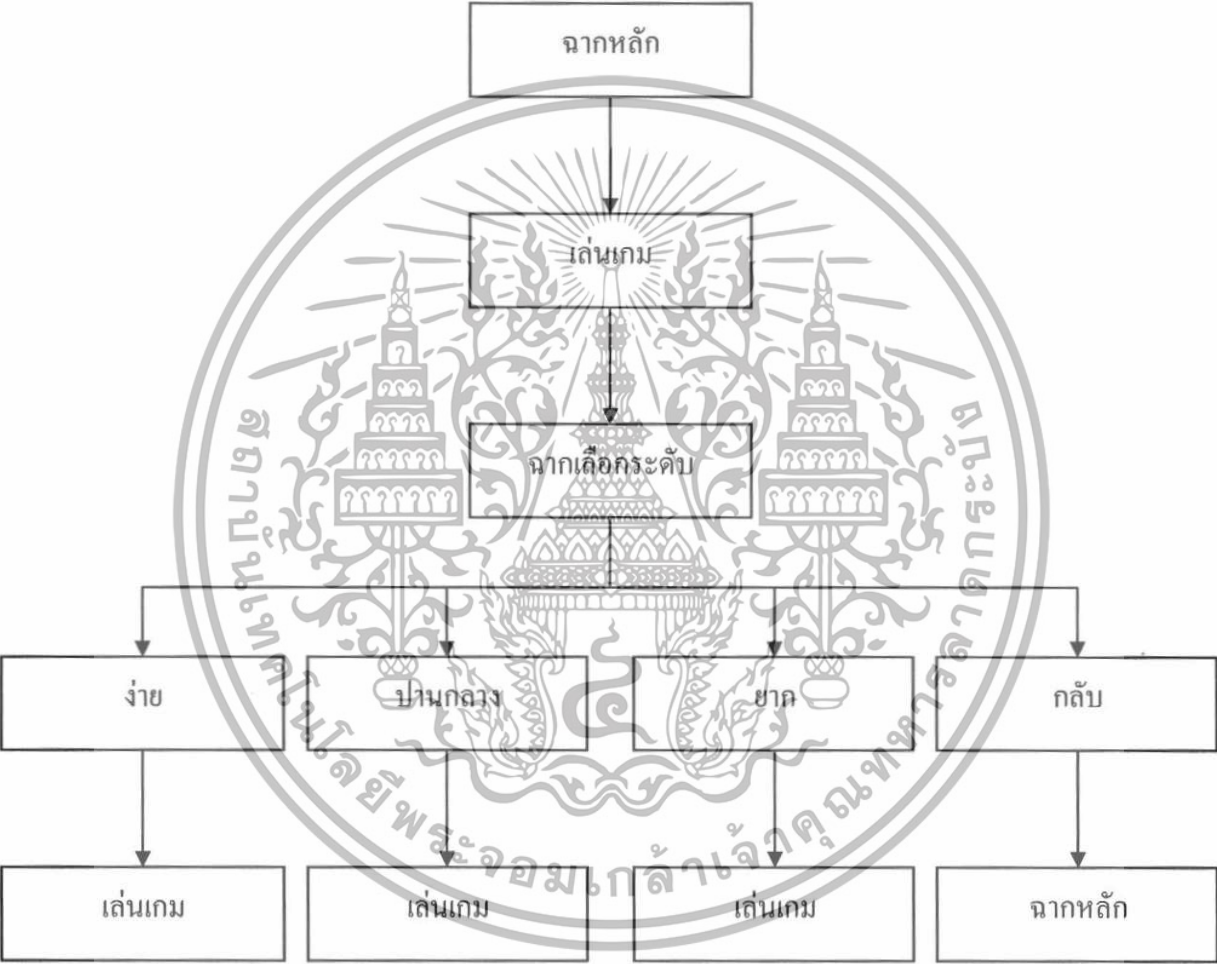
ในหน้าจอนี้จะเป็นการเลือกระดับในการเข้าเล่นเกมกินเมือง แบ่งเป็น 3 ระดับ ตามความยากง่ายโดยสามารถกดเลือกที่ปุ่ม **ง่าย** **ปานกลาง** **ยาก**

เมื่อผู้เล่นต้องการออกจากหน้าจอเลือกระดับกลับไปสู่หน้าจอหลักสามารถกดปุ่ม **กลับ**



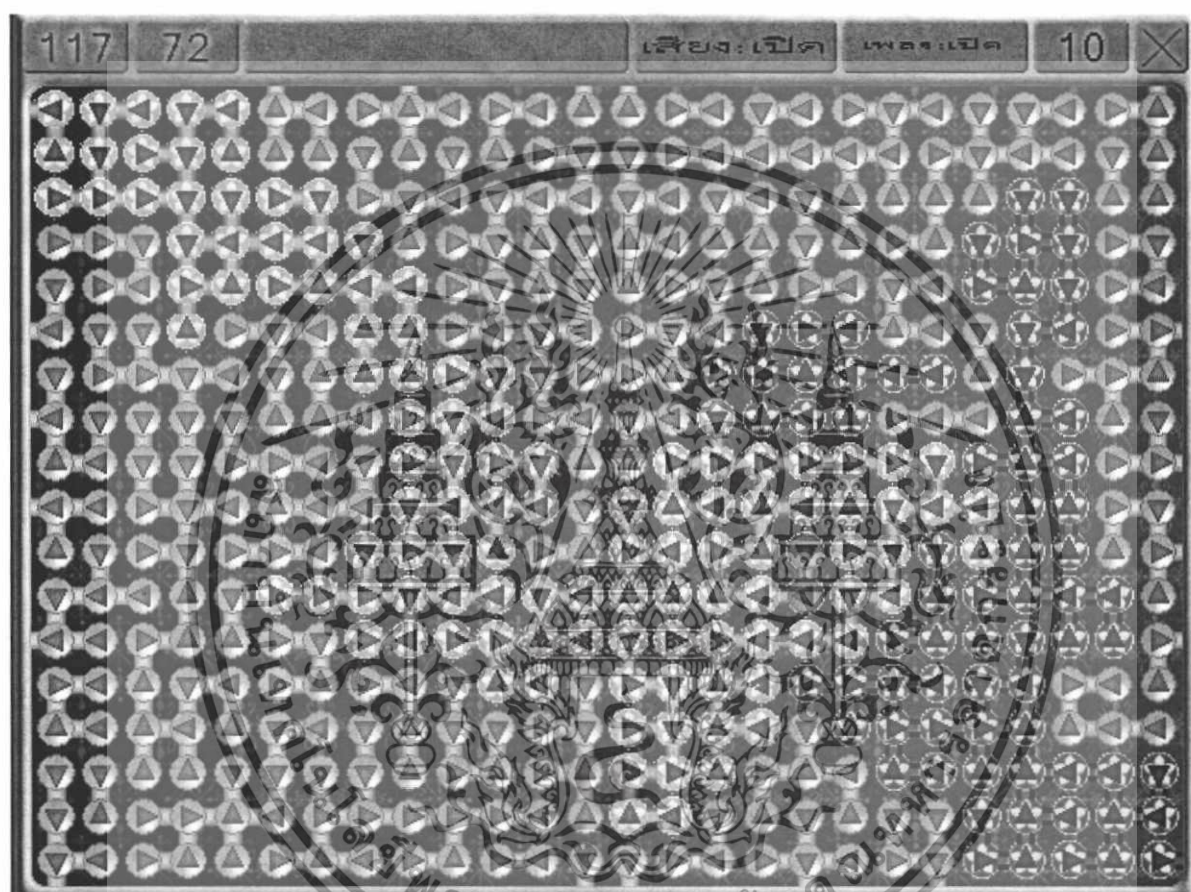
รูปที่ 3.34 รูปหน้าจอการเลือกระดับเกมกินเมือง

มีแผนภาพการทำงานดังนี้



รูปที่ 3.35 แผนภาพการทำงานในหน้าจอการเลือกกระดานเกมกินเมือง

3.3.1.4 หน้าจอการเล่นเกมนินเมือง



รูปที่ 3.36 รูปหน้าจอการเล่นเกมนินเมือง

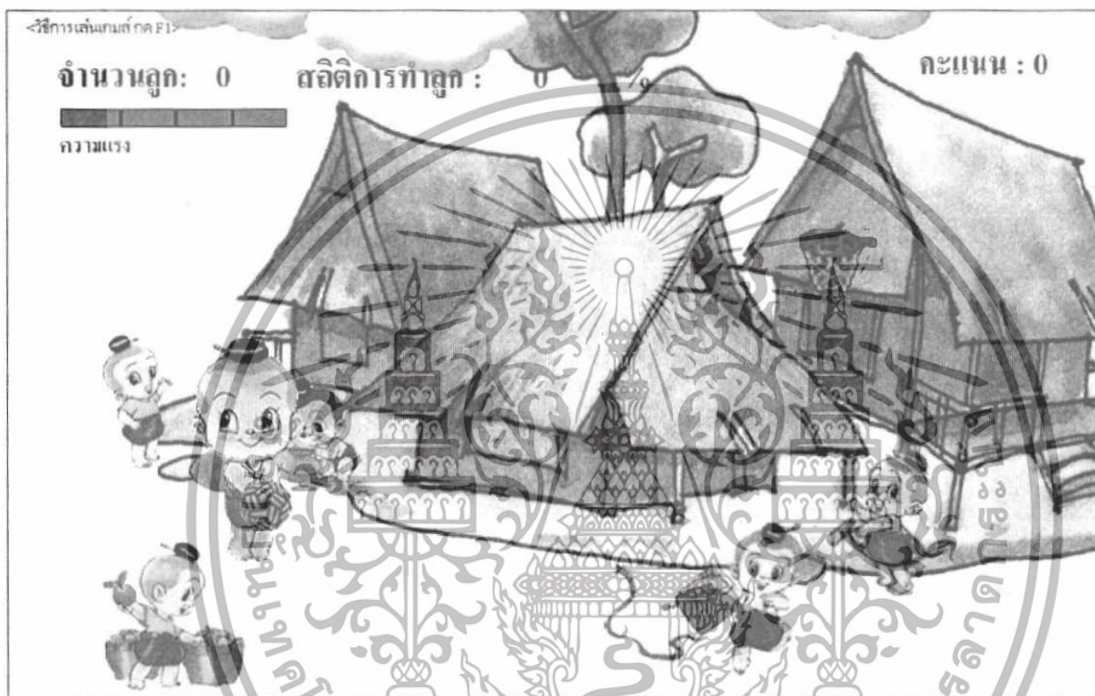
มีแผนภาพการทำงานดังนี้



รูปที่ 3.37 แผนภาพการทำงานในหน้าจอการเล่นเกมกินเมือง

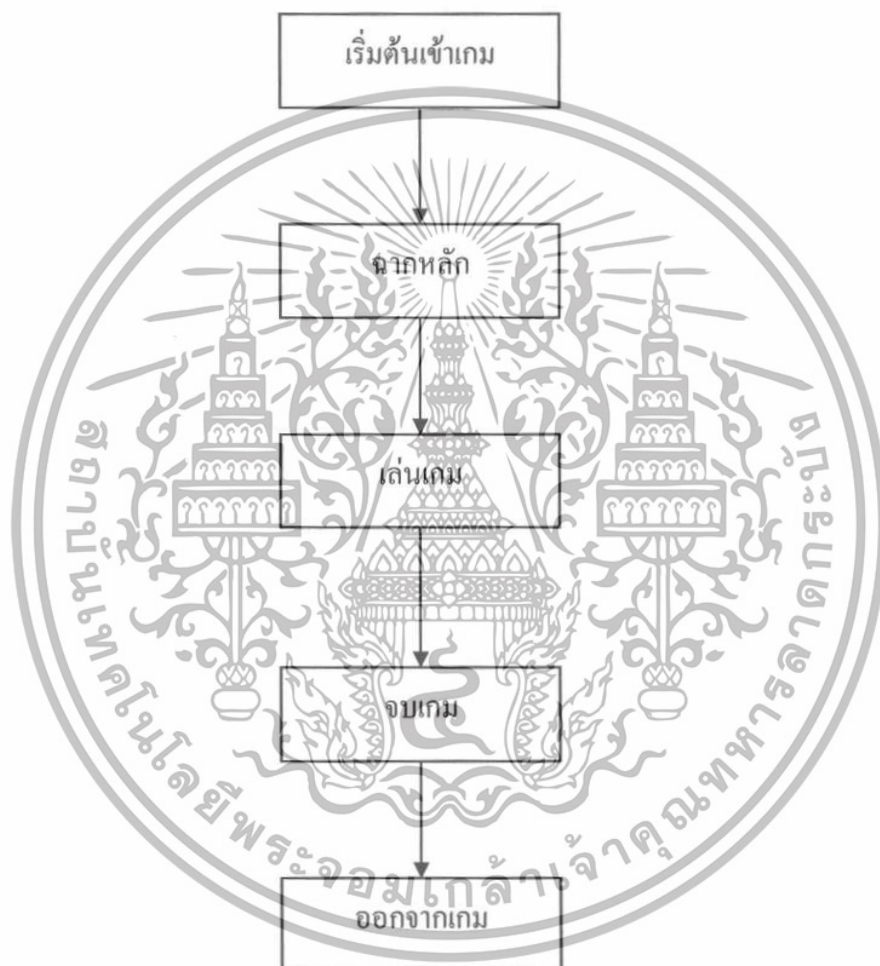
3.3.2 เกมโยนตะกร้อลงห่วง

- หน้าจอการเล่นเกมนโยนตะกร้อลงห่วง



รูปที่ 3.38 รูปหน้าจอการเล่นเกมนโยนตะกร้อลงห่วง

มีแผนภาพการทำงานดังนี้



รูปที่ 3.39 แผนภาพการทำงานในหน้าจอการเล่นเกมนโยนตะกร้อลงห่วง

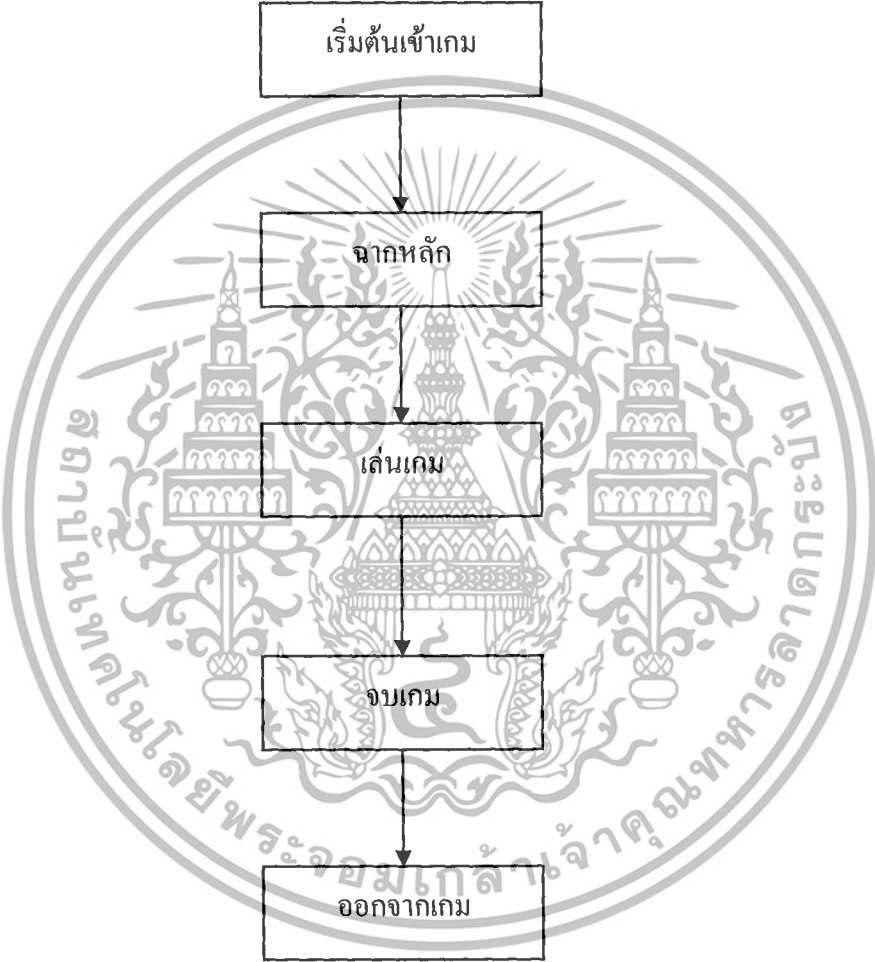
3.3.3 เกมเป็ดน้อยมหัศจรรย์

- หน้าจอการเล่นเกมเป็ดน้อยมหัศจรรย์



รูปที่ 3.40 รูปหน้าจอการเล่นเกมเป็ดน้อยมหัศจรรย์

มีแผนภาพการทำงานดังนี้



รูปที่ 3.41 แผนภาพการทำงานในหน้าจอการเล่นเกมเปิดน้อยมัทศจรรย์

3.4 ขั้นตอนการทดสอบเกม

เป็นขั้นตอนสุดท้ายของการทำเกม โดยขั้นตอนนี้เป็นการตรวจสอบความผิดพลาดและประสิทธิภาพโดยรวมของเกมทั้งหมด ว่าสามารถโต้ตอบและแสดงผลได้ตามที่ออกแบบไว้หรือไม่ถ้าตรวจสอบพบความผิดพลาดจะได้นำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขให้มีความถูกต้องมากที่สุด



บทที่ 4

ขั้นตอนการใช้งานเกม

การจะทดสอบขั้นตอนวิธีในเกมที่ได้ออกแบบไว้แล้วนั้น ทำได้โดยการสร้างโปรแกรมเกมตามขั้นตอนที่ได้กล่าวมาในบทที่ 3 โดยการเล่นเกม เพื่อตรวจสอบประสิทธิภาพของเกมว่ามาสามารถเล่นได้ดีเพียงใด

4.1 เครื่องมือที่ใช้ในการทดลอง

คอมพิวเตอร์ที่ใช้ในการทดลองในปัญหาพิเศษนี้มีลักษณะความสามารถของเครื่องเป็นดังนี้

1. เครื่องมือที่ใช้ในการพัฒนา

- โปรแกรม Game Maker

2. ลักษณะขั้นต่ำที่เครื่องต้องมี

Input: เมาส์ และ คีย์บอร์ด

Output: หน้าจอขนาด 800x600 pixels

Os: Microsoft Windows XP

ความต้องการอื่นๆ: Microsoft DirectX 8.0 Runtime ขึ้นไป

CPU 700 MHZ ขึ้นไป

RAM 128 MB ขึ้นไป

4.1.1 ขั้นตอนการใช้งานโปรแกรม

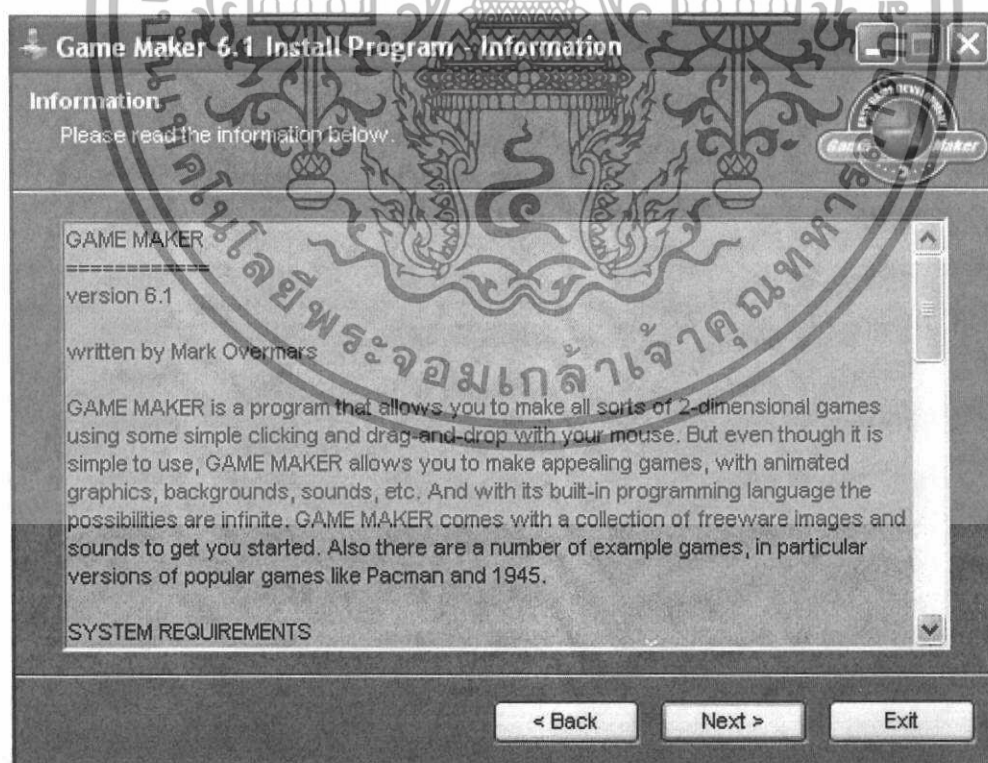
4.1.1.1 การลงโปรแกรม

การสร้างโปรแกรมเพื่อทดสอบเกมนั้นจะอ้างอิงตามลักษณะการใช้งานของผู้ใช้ทั่วไปที่ใช้งานกับระบบปฏิบัติการ Window เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่ายและใช้งานได้สะดวกยิ่งขึ้น โดยมีลักษณะการทดสอบดังนี้

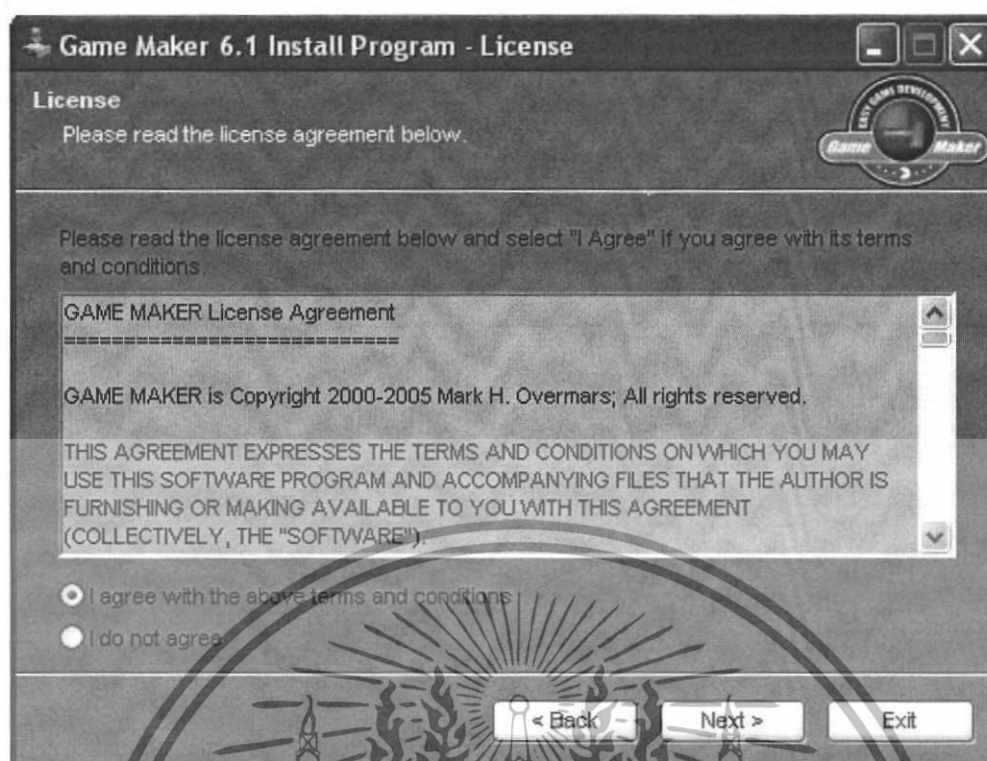
- ให้ทำการลงโปรแกรม Game Maker ก่อน
- เมื่อลงโปรแกรม Game Maker เสร็จแล้วให้ทำการลง Microsoft DirectX 8.0



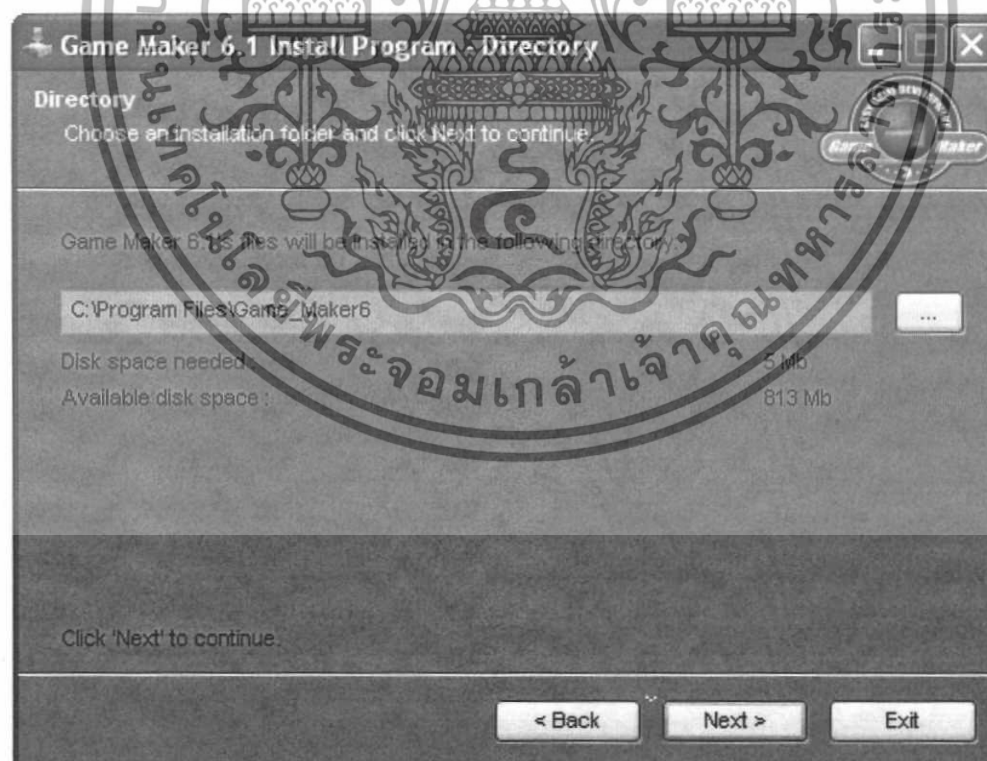
รูปที่ 4.1 รูปหน้าจอเริ่มต้นการติดตั้ง (กดปุ่ม Next)



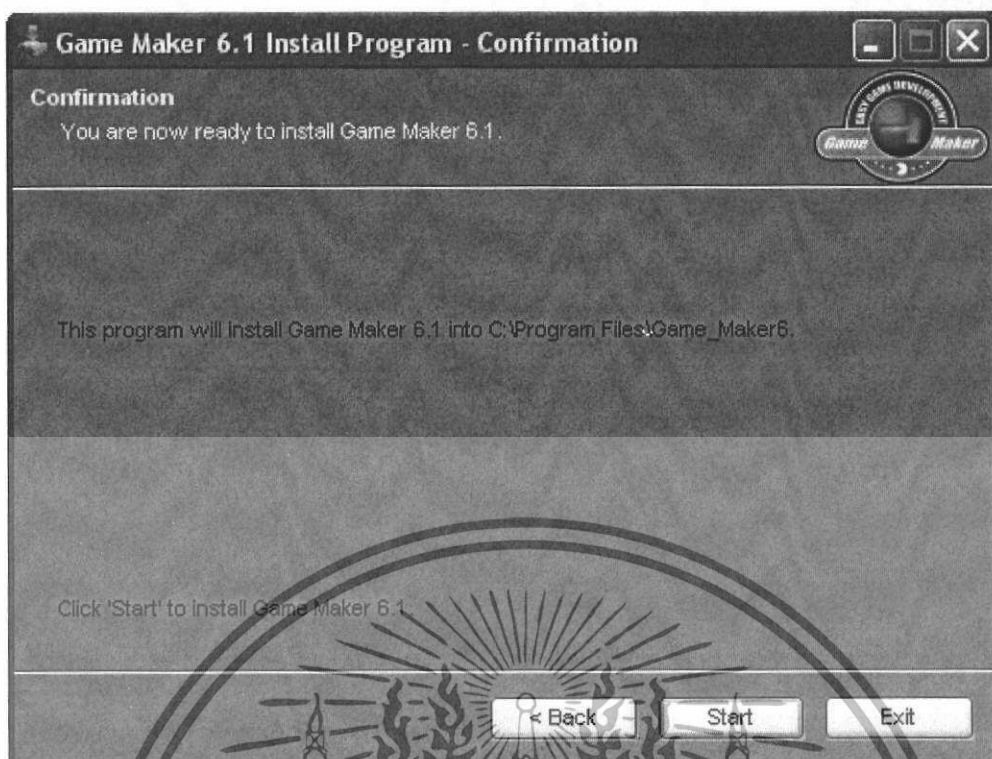
รูปที่ 4.2 รูปหน้าจอ Information (กดปุ่ม Next)



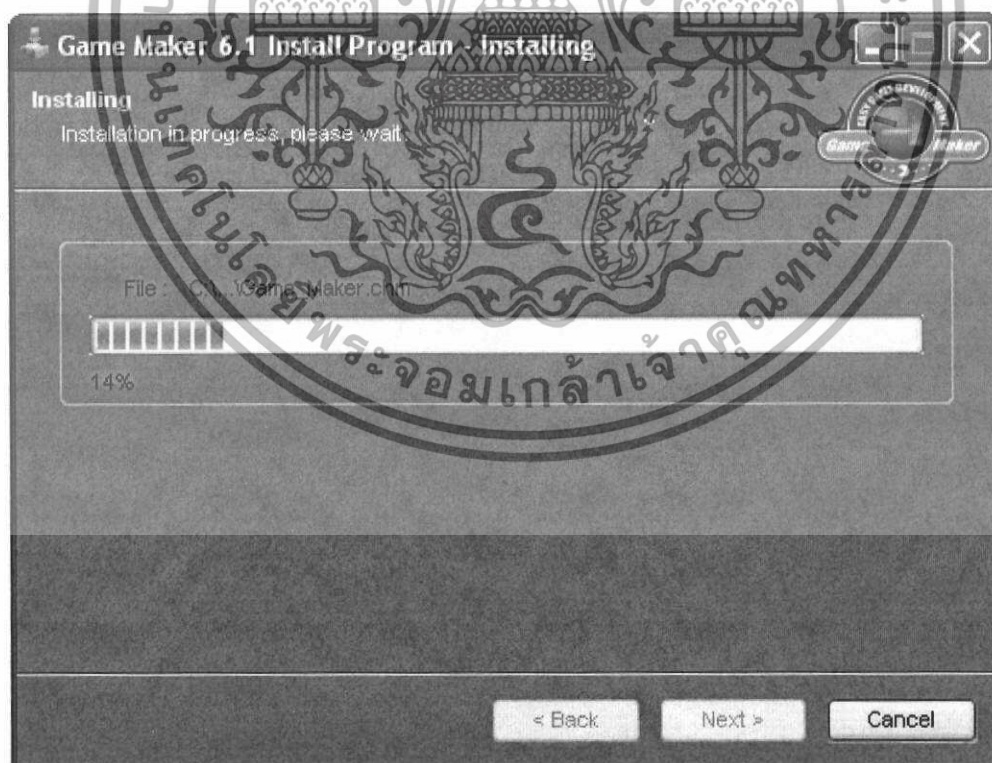
รูปที่ 4.3 รูปหน้าจอ License (เลือก I agree แล้วกดปุ่ม Next)



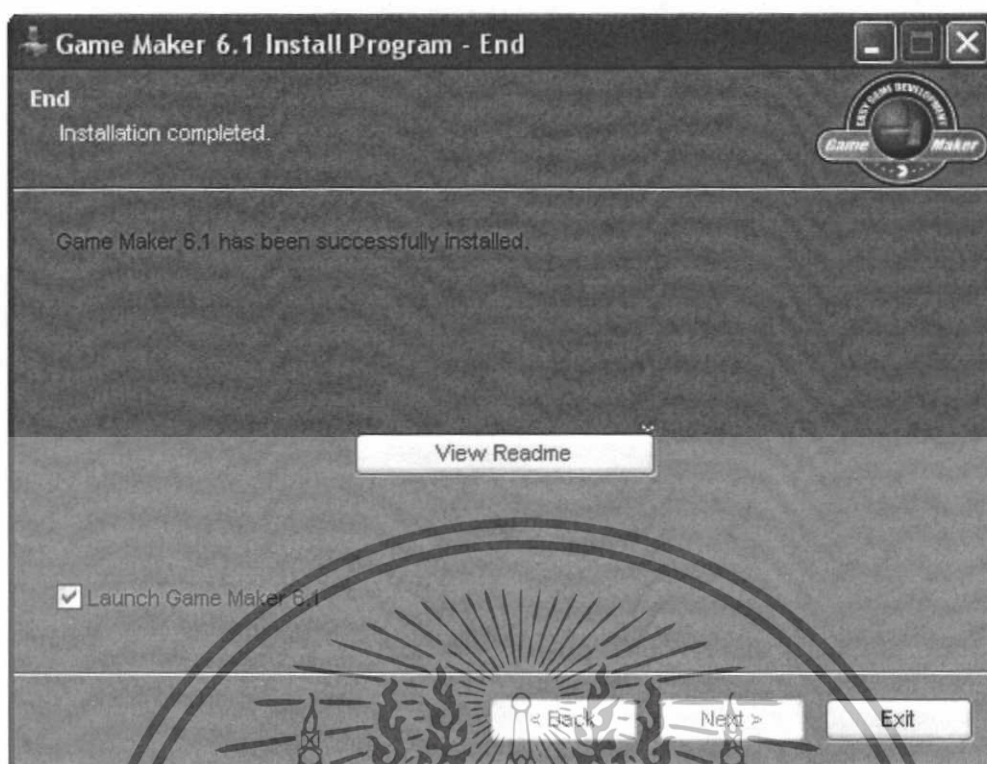
รูปที่ 4.4 รูปหน้าจอ Directory (เลือกตำแหน่งที่จะติดตั้งแล้วกดปุ่ม Next)



รูปที่ 4.5 หน้าจอ Confirmation (ยืนยันการติดตั้งแล้วกดปุ่ม Start)



รูปที่ 4.6 หน้าจอ Installing (รอการติดตั้งสักครู)

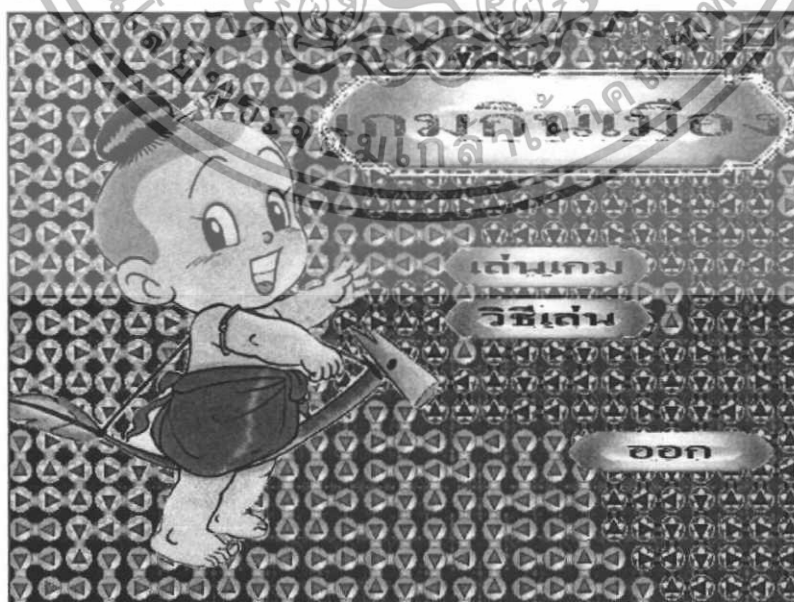


รูปที่ 4.7 หน้าจอ End (การติดตั้งเสร็จเรียบร้อยแล้วกดปุ่ม Exit)

4.1.1.2 คู่มือการเล่นเกม

1) เกมกินเมือง

เป้าหมายของเกมนี้ก็คือการให้ผู้เล่นเกมได้พัฒนาทักษะการคิด ซึ่งมีวิธีการเล่นจะอธิบายตามขั้นตอนดังนี้



รูปที่ 4.8 แสดงหน้าจอเริ่มต้นของเกมกินเมือง

โดยแต่ละส่วนจะมีรายละเอียดดังต่อไปนี้

1. **เล่นเกม** เป็นการเริ่มการเข้าเล่นเกม
2. **วิธีเล่น** เป็นการเข้าสู่การแสดงวิธีการเล่นเกม
3. **ออก** เป็นการออกจากเกม
4.  คลิกเลือกเพื่อเปิดหรือปิดเสียงเพลง
5.  คลิกเลือกเพื่อเปิดหรือปิดเสียง
6.  คลิกเพื่อเลื่อนแถบเมนูทั้งหมดขึ้นไปด้านบน(แสดงแอนิเมชัน)
7.  คลิกเพื่อขยายหน้าจอเป็นแบบเต็มจอ
8.  คลิกเพื่อย่อหน้าจอ



รูปที่ 4.9 แสดงหน้าจอวิธีการเล่นเกมกินเมือง

มีรายละเอียดดังต่อไปนี้

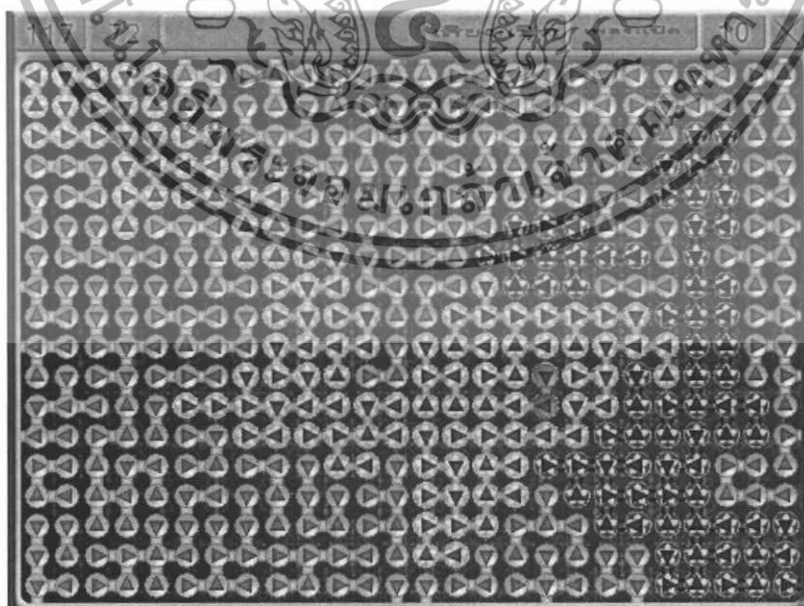
1. แสดงรายละเอียดวิธีการเล่นเกมพร้อมภาพประกอบ
2. **กลับ** เป็นการกลับสู่หน้าจอหลัก



รูปที่ 4.10 แสดงหน้าจอการเลือกระดับของเกมกินเมือง

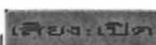
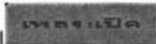
มีรายละเอียดดังต่อไปนี้

1. **ง่าย** เป็นการเลือกเข้าเล่นเกมระดับ ง่าย
2. **ปานกลาง** เป็นการเลือกเข้าเล่นเกมระดับ ปานกลาง
3. **ยาก** เป็นการเลือกเข้าเล่นเกมระดับ ยาก
4. **กลับ** เป็นการกลับสู่หน้าจอหลัก



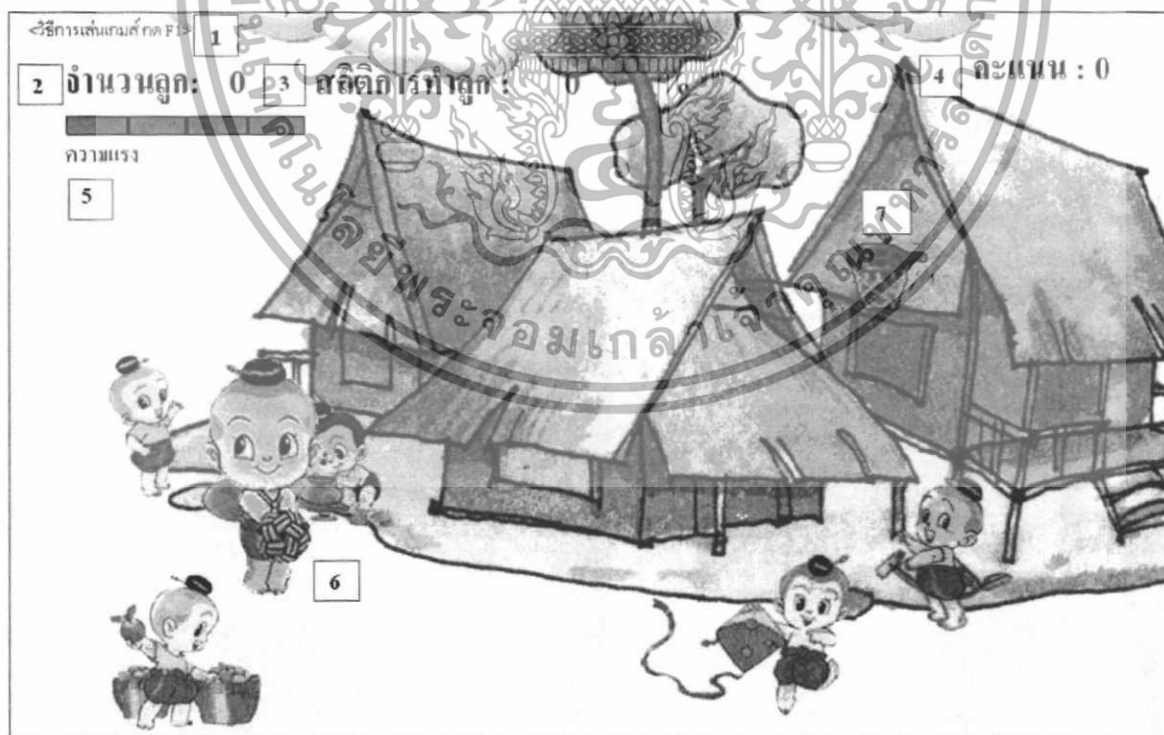
รูปที่ 4.11 แสดงหน้าจอการเล่นเกมกินเมือง

มีรายละเอียดดังต่อไปนี้

1. ตรงส่วนของเกมนั้นวิธีการเล่นให้คลิกที่เมือง(จุด)จะหมุนตามเข็มนาฬิกาเพื่อกินพื้นที่ว่างหรือเมืองของฝั่งตรงข้าม
2. ตรงปุ่ม  เป็นการเปิดหรือปิดเสียงภายในเกม
3. ตรงปุ่ม  เป็นการเปิดหรือปิดเสียงเพลงภายในเกม
4. ตรงช่อง  เป็นการแสดงจำนวนเมือง(จุด)ของฝ่ายผู้เล่น
5. ตรงช่อง  เป็นการแสดงจำนวนเมือง(จุด)ของฝ่ายตรงข้าม
6. ตรงช่อง  เป็นการแสดงจำนวนรอบของเกมที่จะเล่น
7. ตรงช่อง  เป็นการเลือกเมื่อต้องการออกจากเกม

2) เกมโยนตะกร้อลงห้วย

เป้าหมายของเกมนี้มีลักษณะคล้ายเกมกีฬาเป็นการโยนลูกตะกร้อให้ลงห้วยเพื่อทำคะแนน เน้นความสนุกสนานเพลิดเพลิน ซึ่งมีวิธีการเล่นจะอธิบายตามขั้นตอนดังนี้



รูปที่ 4.12 แสดงหน้าจอการเล่นโยนตะกร้อลงห้วย

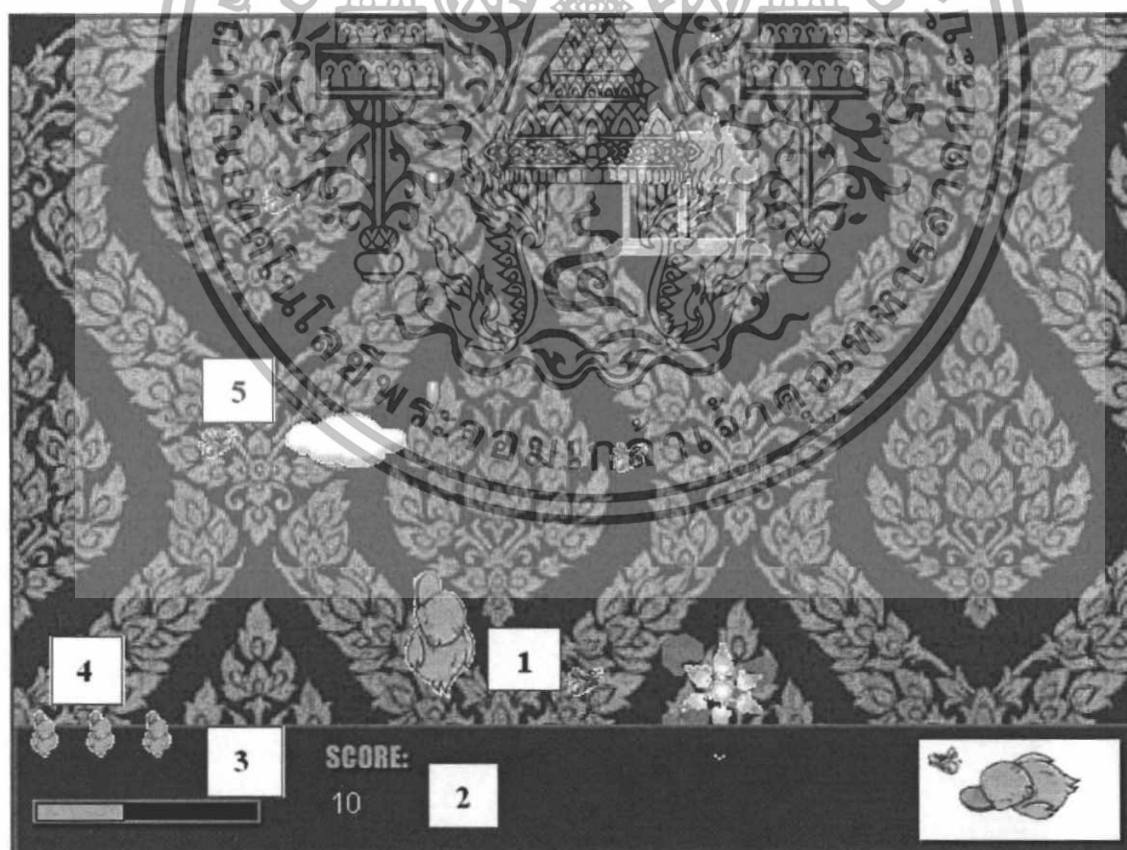
ในเกมโยนตะกร้อลงห่วงนี้จะใช้คีย์บอร์ดในการควบคุมการเล่นเกม โดยแต่ละส่วนจะมีรายละเอียดดังต่อไปนี้

1. หมายเลข 1 เป็นการแสดงข้อความบอกวิธีการเล่นเมื่อกดปุ่ม F1
2. หมายเลข 2 เป็นการแสดงจำนวนลูกที่ผู้เล่น โยนลูก
3. หมายเลข 3 เป็นการแสดงสถิติการ โยนลูกลงห่วง
4. หมายเลข 4 เป็นการแสดงคะแนนที่ผู้เล่น โยนลูกลงห่วง
5. หมายเลข 5 เป็นการแสดงแถบความแรงที่ผู้เล่นจะ โยนลูก
6. หมายเลข 6 เป็นลูกตะกร้อที่ผู้เล่นจะใช้โยน
7. หมายเลข 7 เป็นห่วงที่ผู้เล่นจะต้อง โยนลูกตะกร้อให้ลง

3) เกมเป็ดน้อยมหัศจรรย์

เป้าหมายของเกมมีลักษณะเป็นเกมแนว Shooting อาศัยทักษะการบังคับหลบหลีกเป็นหลัก

ซึ่งมีวิธีการเล่นจะอธิบายตามขั้นตอนดังนี้



รูปที่ 4.13 แสดงหน้าจอการเล่นเกมเป็ดน้อยมหัศจรรย์

ในเกมเปิดน้อยมหัสจรรย์นี้จะใช้คีย์บอร์ดในการควบคุมการเล่นเกมโดยแต่ละส่วนจะมีรายละเอียดดังต่อไปนี้

1. หมายเลข 1 เป็นการแสดงตัวละครในเกมของผู้เล่น
2. หมายเลข 2 เป็นการแสดงคะแนน
3. หมายเลข 3 เป็นการแสดงระดับเลือดของตัวละครของผู้เล่น
4. หมายเลข 4 เป็นการแสดงจำนวนชีวิตของตัวละครของผู้เล่น
5. หมายเลข 5 เป็นการแสดงตัวละครฝ่ายตรงข้ามที่ผู้เล่นต้องยิง

4.2 การออกแบบการทดลองเกม

เพื่อให้ทราบผลและสามารถประเมินผลของเกมว่ามีความถูกต้องเพียงใดนั้นจะต้องทดสอบโปรแกรมหลายๆครั้งโดยการลองเล่นเกมหลายรูปแบบ

4.2.1 ลักษณะการกดเมาส์

- การคลิกซ้าย ซึ่งในโปรแกรมเกมนี้ส่วนใหญ่จะใช้การคลิกเมาส์ซ้ายส่วนใหญ่ ก็เลยลองเปลี่ยนมาคลิกเมาส์ขวาแทนว่าจะเกิดปัญหาขึ้นหรือเปล่า
- การคลิกขวาคลิกซ้ายพร้อมกัน จะเกิดปัญหาหรือเปล่า
- การคลิกเมาส์ในส่วนที่ไม่ได้กำหนดให้คลิก ว่าเกิดปัญหาขึ้นหรือไม่

4.2.2 ลักษณะการกดคีย์บอร์ด

- คว้าเกมไหนกำหนดให้กดคีย์บอร์ดแล้วสามารถกดได้หรือเปล่า
- คว้าช่วงที่ให้ใช้เมาส์นั้นแล้วถ้าเรากดคีย์บอร์ดจะเกิดอะไรขึ้น

4.3 ผลการทดลอง

เป็นขั้นตอนสุดท้ายของการทำเกม โดยขั้นตอนนี้เป็นการตรวจสอบ ความผิดพลาดและประสิทธิภาพโดยรวมของเกมทั้งหมด ว่าเราสามารถโต้ตอบและแสดงผลได้ตามที่บอกไว้หรือไม่ ถ้าตรวจสอบพบความผิดพลาดจะได้นำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขให้มีความถูกต้องมากที่สุด

ซึ่งจากการทดลองตามลักษณะดังที่ได้กล่าวมาแล้วนั้นปรากฏว่าไม่มีเหตุการณ์ผิดปกติเกิดขึ้น

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

การทำโครงการปัญหาพิเศษนี้มีการวางแผนขั้นตอนการทำงานเป็นอย่างดี แต่ถึงแม้จะได้วางแผนแต่ละขั้นตอนไว้อย่างดีก็ยังมีปัญหาบางขั้นตอนเนื่องจากตอนแรกนั้นเราได้กำหนดการทำงานไว้สองส่วนคือส่วนสร้างไฟล์รูปภาพ กับ ส่วนของเกม ซึ่งเราได้ทำในส่วนของไฟล์รูปภาพไปแล้วแต่ปรากฏว่าต้องตัดแต่งรูปใหม่ในบางส่วนและบางส่วนที่ได้สร้างไว้นั้นก็ไม่ได้นำมาใช้เลยทำให้เสียเวลาโดยขั้นตอนต่างๆสรุปได้ดังนี้

- ขั้นตอนการออกแบบ Story Board
- ขั้นตอนการทำเกม
- ขั้นตอนการสร้างตัวละครและภาพต่างๆ

5.1 ขั้นตอนการออกแบบ Story Board

เป็นขั้นตอนที่ดำเนินไปค่อนข้างช้าเนื่องจากว่าในโครงปัญหาพิเศษนี้จะสร้างเกมทั้งหมด 3 เกม ซึ่งในการออกแบบเกมแต่ละเกมนั้นต้องมีลักษณะ มีความความท้าทาย และมีรูปแบบที่แตกต่างกัน

5.2 ขั้นตอนการทำเกม

เป็นขั้นตอนที่ช้าที่สุด เนื่องจากที่กล่าวไว้แล้วว่าต้องสร้างทั้งหมด 3 เกม ซึ่งแต่ละเกมก็ต่างรูปแบบกัน ดังนั้นกระบวนการสร้างเกมจึงมีความยากง่ายต่างกันมีการเลือกใช้เทคนิครูปแบบต่างกัน จึงทำให้การสร้างเกมนั้นมีความล่าช้า

5.3 ขั้นตอนการสร้างตัวละคร

เป็นขั้นตอนที่ค่อนข้างช้าพอสมควรเนื่องจากลักษณะการออกแบบนั้นต้องคำนึงถึงการใช้สีสันทันเพื่อเวลาที่เรานำมาใช้ในการทำเกมนั้นเกมจะได้ดูออกมาไม่แปลกตา หรือดูแล้วสับสนปวดตา เลยทำให้ต้องคอยปรับปรุงเปลี่ยนแปลงภาพอยู่ตลอดเวลา และการสร้างภาพในแต่ละภาพนั้นค่อนข้างใช้เวลานานพอสมควร จึงเลยต้องปรับเปลี่ยน แก้วใจ หรือหารูปภาพมาจากที่อื่นเพื่อให้เข้ากับเป้าหมายของปัญหาพิเศษนี้ที่จะสร้างออกมาในรูปแบบไทยๆ

5.4 ข้อเสนอแนะและแนวทางในการพัฒนา

ขั้นตอนการสร้างเกมในปัญหาพิเศษนี้ เป็นเพียงต้นแบบแนวทางหนึ่งในการเขียนเกม 2D ซึ่งเป็นการออกแบบและสร้างเกมออกมาในรูปแบบที่เป็นไทย ซึ่งจะนำไปพัฒนา ปรับปรุงให้ดีขึ้นต่อไป

ข้อเสนอแนะในการพัฒนาขั้นตอนการเขียนเกมนั้นมีดังนี้

1. เนื่องจากเกมที่สร้างขึ้นมาที่บางเกมยังไม่มี การเพิ่มระดับความยากของเกม ดังนั้นหากมีการเพิ่มระดับของเกมจะทำให้ผู้เล่นมีความท้าทายมากขึ้น
2. เนื่องจากเกมได้ออกแบบมาให้เหตุการณ์ที่เล่นนั้นจบเพียงในครั้งเดียว ดังนั้นถ้ามีการเพิ่มระดับตามข้อที่ 1 แล้ว อาจจะเพิ่มในส่วนของการบันทึกเกมเพื่อการเข้าเล่นครั้งต่อไป
3. จุดมุ่งหมายภายในเกมมีน้อย ดังนั้นหากมีการเพิ่มจุดมุ่งหมายที่มากขึ้นจะทำให้เกมสมจริงมากขึ้น
4. เหตุการณ์ในเกมที่ได้ออกแบบมาเพื่อนำเสนอ เป็นเพียงเหตุการณ์ที่มีเงื่อนไขง่ายๆ ดังนั้นหากต้องการพัฒนาให้สมจริงควรใช้เหตุการณ์ที่มีเงื่อนไขหลากหลาย สามารถเปลี่ยนแปลงได้ตามสถานการณ์แล้วแต่การกระทำของผู้เล่น
5. รูปแบบที่ออกแบบในเกมนั้น อาจจะออกแบบให้มีรูปแบบความเป็นไทยมากยิ่งขึ้นแต่ยังคงต้องออกแบบให้เหมาะสมกับเด็ก

รายการอ้างอิง

- [1] ชันยา นวลละออง, “สร้างเกมง่ายๆด้วยโปรแกรม *GAME MAKER*”, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2550.

