



ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

ชื่อหัวข้อ เกมสับันไดงูสามมิติ
3D Snake Games

ชื่อนักศึกษา	1. นายจันทน์ พันธุ์ไธ	รหัสประจำตัว	48035539
	2. นายรินทร์ สินธุ	รหัสประจำตัว	48035552
	3. นางสาวปิยวรรณ ฟองศรี	รหัสประจำตัว	48035560
	4. นายอนุชิต อนุพันธ์	รหัสประจำตัว	48035592

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชา คอมพิวเตอร์

อาจารย์ที่ปรึกษา อ.ไพบูลย์ พวงวงศ์ตระกูล

อาจารย์ที่ปรึกษาร่วม อ.พรพิมล ฉายรัมย์

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อ.สุระชัย พิมพ์สาลี	
2. อ.ไพบูลย์ พวงวงศ์ตระกูล	
3. ผศ.สุชิน อาจหาญ	
4. อ.ปิยะ จิตธรรมมาภิรมย์	
5. อ.ปิยะ ศุภวราสุวัฒน์	

วัน/เดือน/ปีที่สอบ วันศุกร์ที่ 4 เดือนพฤษภาคม พ.ศ. 2550 เวลา 10.00 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(รศ.สุรสิทธิ์ รัตรี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 30 เดือน พ.ค. พ.ศ. 50



เอกสารนี้เป็นเอกสารสำหรับงานเพื่อการศึกษา ห้ามมิให้นำไปใช้โดยไม่ได้รับอนุญาต

เกมสับันไดงูสามมิติ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เกมส์บันไดงูสามมิติ

3D SNAKE GAMES



จำนงค์
นรินทร์
ปิยวรรณ
อนุชิต

พันธุ์วิไล
สินธุ
พงศ์ศรี
อนุพันธ์

รฟ.
๗ 3247
254๙
เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

75189

๗ ๔ ต.ก. 2550

b. 118 16247
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต
สาขาวิชาคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เกมสับันไจรูสามมิติ
3D Snake Games

วัตถุประสงค์

1. เพื่อศึกษาการสร้างเกมสับันไจรูสามมิติ ด้วยโปรแกรม 3ds max และ Dark Basic Professional
2. เพื่อออกแบบรูปร่างของตัวละครและฉากของเกมสับันไจรูสามมิติ
3. เพื่อสร้างเกมสับันไจรูสามมิติ
4. เพื่อทดลองเกมสับันไจรูสามมิติ และปรับปรุงแก้ไขส่วนที่บกพร่องของเกม
5. เพื่อนำเกมสับันไจรูที่สร้างขึ้นไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้เกี่ยวกับการสร้างเกมสับันไจรูสามมิติ ด้วยโปรแกรม 3ds max และ Dark Basic Professional
2. ได้รับความรู้ในการออกแบบรูปร่างตัวละครและฉากของเกมสับันไจรูสามมิติ
3. ได้รับความรู้ในการสร้างเกมสับันไจรูสามมิติ
4. ได้รับทราบถึงจุดต่าง ๆ ของเกม ที่ต้องปรับปรุงแก้ไข และพัฒนาต่อไปในอนาคต
5. ได้เกมสับันไจรูสามมิติที่สามารถเล่นได้ตรงตามหลักการและวัตถุประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เกมส์บันไดงูสามมิติ	
นักศึกษา	นายจ่านงค์	พันธุ์วิไล
	นายนรินทร์	สินชู
	นางสาวปิยวรรณ	พองศรี
	นายอนุชิต	อนุพันธ์
อาจารย์ที่ปรึกษา	อาจารย์ไพบุลย์	พวงวงศ์ตระกูล
อาจารย์ที่ปรึกษาร่วม	ดร. พรพิมล	ฉายรัมย์
หลักสูตร	ครุศาสตรบัณฑิตศึกษาศาสตร์บัณฑิต	
สาขาวิชา	คอมพิวเตอร์	
ปีการศึกษา	2549	

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้นำเสนอการสร้างเกมส์บันไดงูสามมิติ ซึ่งเป็นการพัฒนาจากเกมส์ที่เล่นบนแผ่นกระดาน ให้อยู่ในรูปแบบของเกมส์สามมิติที่สามารถเล่นได้บนเครื่องคอมพิวเตอร์ โดยใช้โปรแกรม 3ds max สร้างฉาก โมเดลของตัวละครการทำภาพเคลื่อนไหวและใช้โปรแกรม Dark Basic Professional ในการควบคุมมุกกล้องของเกมส์ โหลดโมเดลตัวกรและวัตถุต่างๆ ที่ดัดสร้างขึ้นมาจากโปรแกรม 3ds max ซึ่งจะทำให้ผู้เล่นสนุกสนานพร้อมทั้งเห็นความแตกต่างระหว่างเกมส์รูปแบบใหม่และรูปแบบเก่า

II

Thesis Title	3D SNAKE GAMES	
Students	Mr. Jumnong	Phanwilai
	Mr. Narin	Sinchoo
	Miss Piyawan	Fongsri
	Mr. Anuchit	Anupan
Advisor	Mr. Paiboon	Pongwongtragull
Co-Advisor	Dr. Pornpimon	Chayratsami
Education Level	Bachelor of Science in Industrial Education	
Program in	Computer	
Academic Year	2006	

ABSTRACT

The thesis presents construction 3D SNAKE GAMES which develop of board game playing to formed of 3D computer games. 3ds max is program use to creating background character modeling animation and Dark Basic Professional is use to control and viewport load character model and object etc. whiche a player will enjoy with differentiate a between of formed new games and formed new games and formed classic games.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงได้ดีนั้น เนื่องมาจากความร่วมมือร่วมใจของสมาชิกภายในกลุ่มทุกท่าน คณะผู้จัดทำขอขอบพระคุณท่านอาจารย์ไพบุลย์ พวงวงศ์ตระกูล ดร. พรพิมล ฉายรัศมีและอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่านเป็นอย่างมากที่ได้กรุณาให้คำปรึกษาและให้คำแนะนำในการแก้ไข ปัญหาต่างๆ ตลอดจนจนถึงข้อมูลที่เป็นประโยชน์ต่อการทำโครงการ และในการจัดทำปริญญานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณบิดา มารดา และผู้มีพระคุณสำหรับพวกเราที่ได้ให้การสนับสนุนทุกสิ่งทุกอย่าง ทางด้านการศึกษาตลอดมาจนถึงปัจจุบัน และสุดท้ายต้องขอขอบคุณเพื่อนๆ ที่เป็นกำลังใจให้เสมอมา



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 จุดมุ่งหมายของโครงการ	1
1.3 สมมุติฐานของการจัดทำโครงการ	1
1.4 ขีดความสามารถของโครงการ	2
1.5 ขั้นตอนการทำโครงการ	2
1.6 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 การนำโปรแกรม 3ds max เข้ามาช่วยในการทำงาน	3
2.2.1 ขั้นตอนในการสร้างงานในโปรแกรม 3ds max	3
2.2.2 โครงสร้างของโมเดลแบบต่างๆ	4
2.2.3 โครงสร้างแบบ Poly	5
2.3 คำสั่งพื้นฐานโปรแกรม Dark Basic Professional Education	18
2.3.1 การแสดงผลบนหน้าจอ	18
2.3.2 การรับค่าจากคีย์บอร์ด	18
2.3.3 การรับค่าจากเมาส์	19
2.3.4 การตัดสินใจกับการวนทำงานซ้ำ	19

สารบัญ (ต่อ)

เรื่อง	หน้า
2.3.5 ตัวแปรใน Dark Basic Professional Education	21
2.3.6 ตัวแปรชนิดอาร์เรย์	21
2.3.7 ตัวแปรชนิดอาร์เรย์หลายมิติ	22
2.3.8 เครื่องหมายคณิตศาสตร์	22
2.3.9 การสร้างกลุ่มคำสั่งของ Dark Basic Professional Education	23
2.3.10 ระบบสามมิติ	24
2.3.11 การทำให้วัตถุต่างๆ เคลื่อนไหว	25
2.3.12 ฟังก์ชันอำนวยความสะดวกของ Dark Basic Professional Education ที่ใช้สำหรับการทำให้โมเดลสามมิติเคลื่อนไหว	26
2.3.13 การหมุนวัตถุ	27
2.3.14 วิธีห่อหุ้มค่าตำแหน่งของวัตถุหรือมุมของวัตถุที่หมุนไปจากมุมอ้างอิง	27
2.3.15 การตรวจสอบการชนกันของวัตถุ	27
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	28
3.1 กล่าวนำ	28
3.2 การออกแบบและการสร้างเกมส์บนโดเมนสามมิติ	28
3.2.1 ส่วนของตัวละคร	28
3.2.2 ฉากของเกมส์	42
3.2.3 ส่วนของโปรแกรม	45
บทที่ 4 การทดสอบและผลการทดสอบ	49
4.1 กล่าวนำ	49
4.2 การทดสอบการทำงานของเกมส์บนโดเมนสามมิติ	49
4.2.1 วิธีการทดสอบเกมส์บนโดเมนสามมิติ	49
4.2.2 ผลการทดสอบจากผู้เล่นเกมส์บนโดเมนสามมิติ	55
4.2.3 สรุปผลการทดสอบ	57

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุป	58
5.1 สรุป	58
5.2 ปัญหาและวิธีการแก้ไข	58
5.3 แนวทางการพัฒนา	59
บรรณานุกรม	50
ภาคผนวก ก คู่มือการใช้งานเกมส์บันไดงูสามมิติ	51
ภาคผนวก ข ผังงาน	57
ภาคผนวก ค รหัสต้นฉบับของโปรแกรม	59
ประวัติผู้แต่ง	94



สารบัญรูป

รูปที่	หน้า
2.1 หน้าตาของโปรแกรม 3ds max	6
2.2 เลือกที่คำสั่งสร้างกล่อง	7
2.3 กำหนดรายละเอียดของกล่องที่จะสร้าง	7
2.4 สร้างกล่องที่ Viewport Top	8
2.5 กรอกราคาทุกแกนใน Coordinate display ให้เป็น 0	8
2.6 เลือกใช้คำสั่ง Convert TO : > Convert to Editable Poly	9
2.7 การคลิกเลือก Symetry จากช่อง Modify List	9
2.8 เลือกทำงานที่ Polygon	10
2.9 แดรกเมาส์เลือก Face ซีกขวาของกล่องทั้งหมด	10
2.10 หน้าต่างยืนยันการลบ	11
2.11 กดปุ่ม <Delete> ลบ Face	11
2.12 การเลือก Symmetryและการเลือก Flip	11
2.13 ผลจากการใส่ Symmetry	11
2.14 การเลือกทำงานที่ Editable Poly	12
2.15 ผลลัพธ์ที่ได้จากการ Symmetry เข้ามาในโมเดล	13
2.16 เลือก Face ด้านหลังกับด้านล่างกล่อง	14
2.17 ปรับค่าที่หน้าต่าง Extrude Polygons	14
2.18 ผลลัพธ์จากการปรับค่า Extrude Polygons	15
2.19 การย้ายตำแหน่งของโมเดลเพื่อให้ได้ตัวละครตามที่เรากำลังต้องการ	15
2.20 รูปร่างของตัวละคร	16
2.21 การ map ภาพโมเดล	16
2.22 หน้าต่าง Edit UVW	17
2.23 โมเดลที่สมบูรณ์ หลังจากการ map	17
2.24 สร้าง Box เพื่อขึ้นรูปของฉาก	18
2.25 หน้าต่าง Material Editor	19
2.26 ผลที่ได้จากการทำ Material	19
2.27 ลักษณะฉากที่สมบูรณ์	20

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1 ลักษณะของตัวละครรูปเต่า	32
3.2 ลักษณะของตัวละครรูปหุ่นยนต์	33
3.3 โมเดลที่สมบูร์นจากการสร้างด้วยโปรแกรม 3ds max	33
3.4 ลักษณะฉากที่จะใช้ในเกมส์	34
3.5 ลักษณะฉากที่จะใช้ในเกมส์	35
4.1 หน้าต่างแรกของเกมส์บนโดงสามมิติ	39
4.2 หน้าต่างเมนูหลักของเกมส์	40
4.3 หน้าต่างเลือกตัวละครที่ใช้แทนตัวผู้เล่น	40
4.4 หน้าต่างรายละเอียดเมนู CREDIT	41
4.5 หน้าต่างตัวละครที่ใช้แทนตัวผู้เล่น	41
4.6 เริ่มต้นเข้าสู่เกมส์บนโดงสามมิติ	42
4.7 ปุ่มควบคุมเกมส์บนคีย์บอร์ด	42
4.8 ปุ่มควบคุมเกมส์บนคีย์บอร์ด	43
4.9 ผู้เล่นเดินไปเจอ	43
4.10 ถูกงูกินและตกไปยังช่องที่กำหนดไว้	43
4.11 ผู้เล่นคนต่อไปเริ่มเล่นเกมส์	44
4.12 ผู้เล่นเข้าเส้นชัย	44

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันเราจะเห็นได้ว่า มีเกมส์คอมพิวเตอร์ต่างๆ กำเนิดขึ้นมามากมาย และรูปแบบของเกมส์เห็นได้อย่างชัดเจนว่า มีการปรับปรุงและพัฒนาอย่างต่อเนื่อง นับตั้งแต่การสร้างเกมส์ การออกแบบดีไซน์การใช้เทคนิคและโปรแกรมกราฟฟิกต่างๆ ที่ทำให้เกมส์นั้นดูมีความสมจริงมากยิ่งขึ้น รวมไปถึงขั้นตอนและวิธีการเล่นที่จะดึงดูดความสนใจจากผู้ที่ยิยมเล่นเกมส์คอมพิวเตอร์ ดังนั้นกลุ่มผู้จัดทำจึงต้องการสร้างเกมส์ในรูปแบบของเกมส์สามมิติที่กำลังเป็นที่นิยมในปัจจุบัน โดยได้เลือกพัฒนาจากเกมส์บันไดงูที่เล่นบนกระดานให้เป็นเกมส์ที่เล่นได้บนเครื่องคอมพิวเตอร์ โดยใช้ชื่อว่า "เกมส์บันไดงูสามมิติ"

1.2 จุดมุ่งหมายของโครงการ

จากที่ได้กล่าวไว้ในข้างต้นว่าเกมส์บันไดงูนั้น จะมีรูปแบบการเล่นจะอยู่ในลักษณะของกระดาน มีอุปกรณ์ในการเล่นคือ ลูกเต๋าและตัวละครในรูปแบบต่างๆ ที่ใช้ประกอบในการเล่น ดังนั้นเพื่อให้เกมส์มีความน่าสนใจมากยิ่งขึ้น ทางกลุ่มผู้จัดทำจึงต้องการสร้างเกมส์บันไดงูให้อยู่ในรูปแบบของเกมส์สามมิติที่เล่นได้บนเครื่องคอมพิวเตอร์ ซึ่งจะทำให้เกมส์มีความน่าสนใจและผู้เล่นจะได้รับความแปลกใหม่ในการเล่นเกมส์บันไดงูมากยิ่งขึ้น

1.3 สมมุติฐานของการจัดทำโครงการ

เมื่อผู้เล่นได้เล่นเกมส์บันไดงูสามมิติที่กลุ่มผู้จัดทำได้สร้างขึ้นแล้ว จะทำให้ผู้เล่นได้เห็นถึงเปลี่ยนแปลงของเกมส์บันไดงูสามมิติ ที่มีความแตกต่างจากเกมส์บันไดงูที่ผู้เล่นเคยเล่นบนกระดาน นอกจากนี้ยังเป็นจุดเริ่มต้นให้ผู้สนใจในการพัฒนาเกมส์ ได้พัฒนาเกมส์บันไดงูสามมิตินี้ต่อไปได้

1.4 ขีดความสามารถของโครงการ

โครงการนี้มีขีดความสามารถดังนี้

1. มีตัวละครที่สามารถเลือกเล่นได้ 8 ตัว
2. สามารถเลือกเล่นได้ครั้งละ 4 คน
3. ผู้เล่นสามารถเลือกตัวละครได้ครั้งละ 1 ตัว ต่อผู้เล่น 1 คน

4. ไม่กำหนดเวลาในการเล่น
5. สามารถแข่งขันกับคอมพิวเตอร์ได้
6. สามารถมองเห็นตำแหน่งของตัวละครอื่นๆ ได้

1.5 ขั้นตอนการทำโครงการ

โครงการนี้จะเป็นส่วนของซอฟต์แวร์ทั้งหมด ซึ่งการทำงานในระยะแรกจะเริ่มต้นจากการออกแบบฉากและตัวละครที่จะใช้เล่นบนเกมส์โดยเขียนเป็น Storyboard ขึ้นมาก่อน ต่อมาจะเป็นขั้นตอนของการออกแบบฉากและตัวละครที่จะใช้เล่นบนเกมส์ หลังจากนั้นจะเป็นขั้นตอนของการศึกษาการใช้งานโปรแกรม 3ds max และโปรแกรม Dark Basic Professional Education เพื่อที่จะนำมาสร้างเป็นเกมส์ตามที่ได้ออกแบบไว้ในเบื้องต้น และสุดท้ายเมื่อเกมส์เสร็จสมบูรณ์แล้วจะเป็นขั้นตอนของการทดสอบเล่นเกมส์ เพื่อหาข้อบกพร่องและทำการแก้ไขให้ดียิ่งขึ้น

1.6 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปฏิญานิพนธ์ ชี้ความสามารถของโครงการและเนื้อหาในบทต่างๆ โดยสังเขป

บทที่ 2 ประกอบด้วยทฤษฎีต่างๆ เกี่ยวกับโครงสร้างของโมเดลแบบต่างๆ การใส่ลดทอนลงบนโมเดลด้วย Texture Map และคำสั่งต่างๆ ในโปรแกรม Dark Basic Professional

บทที่ 3 กล่าวถึงเนื้อหาที่เกี่ยวกับแผนผังการทำงานของเกมส์บันไดงูสามมิติ ตลอดจนการออกแบบและการสร้างส่วนประกอบต่างๆ ของเกมส์ เช่น การสร้างตัวละครและฉาก รวมไปถึงโปรแกรมที่ใช้ควบคุมการทำงานต่างๆ ในเกมส์

บทที่ 4 ประกอบด้วยการทดลองและผลการทดลองการทำงานของเกมส์บันไดงูสามมิติ

บทที่ 5 เป็นการสรุปผลการจัดทำโครงการ ปัญหาที่เกิดขึ้นและแนวทางในการแก้ไข รวมทั้งแนวทางการพัฒนา

ภาคผนวก ก คู่มือวิธีการเล่นเกมส์บันไดงูสามมิติ

ภาคผนวก ข ผังงานการทำงานของเกมส์บันไดงูสามมิติ

ภาคผนวก ค รหัสต้นฉบับเกมส์บันไดงูสามมิติ

ภาคผนวก ง ประวัติผู้แต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญานิพนธ์ในบทนี้เป็นทฤษฎีและหลักการที่จะนำมาใช้ประกอบการสร้างโครงการ โดยจะประกอบด้วย โครงสร้างของโมเดลแบบต่างๆ การสร้างโมเดลด้วย โครงสร้างแบบ Mesh & Poly การใส่ลวดลายบนโมเดลด้วย Texture Map คำสั่งต่างๆ ในโปรแกรม Dark Basic Professional Education

2.2 การนำโปรแกรม 3ds max เข้ามาช่วยในการทำงาน

โปรแกรม 3ds max เป็นโปรแกรมที่ใช้ในการสร้างงานในลักษณะที่เป็น 3 มิติ ด้วยพัฒนาการของตัวโปรแกรมในปัจจุบันที่สามารถสร้างผลงานให้อยู่ในระดับเดียวกับโปรแกรม 3D ขั้นสูง จึงทำให้เป็นที่นิยมอย่างกว้างของผู้ที่ต้องการสร้างผลงาน 3D ด้วยตนเอง ซึ่งในโปรแกรมนี้จะมีเครื่องมือต่างๆ ให้เราสามารถสร้างงาน 3D ได้อย่างไม่ยากนัก

2.2.1 ขั้นตอนในการสร้างงานในโปรแกรม 3ds max

โดยขั้นตอนหลักๆ ในการสร้างผลงานสักชิ้นหนึ่งในโปรแกรมนี้ไม่ว่าจะเป็นงานชิ้นใหญ่หรืองานชิ้นเล็กก็จะมีขั้นตอนหลักๆ อยู่ 7 ขั้นตอนคือ

1. Storyboard เริ่มต้นด้วยการคิดในสิ่งที่อยากจะทำแล้วเขียนลงบนกระดาษ ซึ่งก็นับว่าเป็นขั้นตอนของการออกแบบทั้งหมด หลังจากเสร็จขั้นตอนนี้แล้วคุณก็จะได้ Storyboard เพื่อเอามาใช้เป็นแนวทางในการทำงาน
2. Modeling ก็เป็นขั้นตอนแรกที่เริ่มทำงานในคอมพิวเตอร์ ด้วยการใช้เครื่องมือในโปรแกรมสร้างสิ่งต่างๆ ที่ออกแบบเอาไว้ให้เป็นรูปร่าง หลังจากจบขั้นตอนนี้แล้วเราก็จะได้วัตถุทุกอย่างที่มีอยู่ใน Storyboard เช่น โมเดลตัวละครและอุปกรณ์ประกอบฉากต่างๆ กับดินน้ำมันกล่าวคือยังไม่มีในรายละเอียดของพื้นผิว มีแต่รูปทรงเท่านั้น เรามักจะเรียกวัตถุที่สร้าง
3. Texturing เป็นขั้นตอนต่อมาที่จะนำ "โมเดล" ที่สร้างไว้มาใส่รายละเอียดของพื้นผิว ไม่ว่าจะเป็นส่วนของความมันวาว การสะท้อนหรือลวดลายที่จะปรากฏบนพื้นผิว

4. Animation ขั้นตอนนี้เป็นขั้นตอนที่ผู้ใช้งานหลายๆ คนชอบมากที่สุดคือ การจับโมเดลที่มีอยู่มาใส่การเคลื่อนไหว อาจจะได้ตั้งแต่การเคลื่อนไหวที่ง่ายๆ เช่น กำหนดให้กล้องนั้นวิ่งไปตามเส้นทางที่มักจะใช้บ่อยๆ ในงานออกแบบสถาปัตยกรรมหรือการเคลื่อนไหวที่ซับซ้อน เช่น การวิ่ง เดินและกระโดดของตัวละคร

2.2.2 โครงสร้างของโมเดลแบบต่างๆ

วัตถุต่างๆ ในโลกนี้ มีรูปทรงที่แตกต่างกันออกไปมากมาย บางวัตถุมีลักษณะเป็นเหลี่ยมๆ มุมๆ ส่วนบางวัตถุก็โค้ง กลม มน และบางวัตถุก็มีรูปทรงเป็นแบบฟรีฟอร์มที่ไม่สามารถกำหนดรูปทรงที่แน่นอนได้เลย อย่างเช่น ผ้า เป็นต้น ดังนั้นการที่จะสร้างวัตถุต่างๆ เหล่านี้ขึ้นมาเป็นโมเดล 3 มิติ ในโปรแกรมคอมพิวเตอร์ก็จะต้องใช้วิธีที่แตกต่างกันออกไป ซึ่งในโปรแกรม 3ds max จะมีโครงสร้างของวัตถุมาให้หลายๆ แบบ เพื่อให้เราเลือกใช้งานได้ตามความเหมาะสมสำหรับนำมาสร้างเป็นโมเดลรูปทรงที่ต้องการ เช่น

1. โครงสร้างแบบ Mesh & Poly เป็นโครงสร้างที่ประกอบขึ้นมาจากจุดและเส้นตรงที่เชื่อมต่อกันจนเกิดเป็นแผ่นระนาบขึ้นมา และเมื่อนำระนาบมาต่อกันเป็นรูปทรงต่างๆ ก็จะเป็นชิ้นงาน 3 มิติ ถือว่าเป็นโครงสร้างที่นิยมใช้งานมากที่สุดในปัจจุบัน เพราะความง่ายในการขึ้นโมเดลเป็นรูปทรงต่างๆ แถมในปัจจุบันโครงสร้างแบบ Mesh & Poly ยังถูกพัฒนาไปจนถึงขั้นที่สามารถขึ้นโมเดลเป็นรูปทรงฟรีฟอร์มได้สะดวกมากขึ้นนับว่าเป็นโครงสร้างที่เหมาะสมกับผู้เริ่มต้นมากที่สุดโครงสร้างหนึ่งเลยทีเดียว

2. โครงสร้างแบบ Spline เป็นโครงสร้างที่มีจุดและเส้นเป็นส่วนประกอบเช่นเดียวกับ Mesh และ Poly แต่จะมีความแตกต่างตรงลักษณะของเส้นที่สามารถดัดให้เป็นเส้นโค้งได้อย่างอิสระก่อนนำไปสร้างพื้นผิว และด้วยความสามารถของเส้นที่ดัดโค้งได้มากมายนี้เอง จึงทำให้โครงสร้างแบบนี้เป็นโครงสร้างที่ผู้ใช้งานนิยมนำไปสร้างรูปทรงที่มีลักษณะฟรีฟอร์ม

3. โครงสร้างแบบ NURBS เป็นโครงสร้างอีกชนิดหนึ่งที่เหมาะสมสำหรับงานขึ้นโมเดลในลักษณะฟรีฟอร์ม ลักษณะของส่วนประกอบต่างๆ จะคล้ายกันกับโครงสร้างแบบ Spline แต่จะมีขอบเขตของคำสั่งสำหรับตกแต่งแก้ไขโมเดลที่ยืดหยุ่นมากขึ้นไปอีก จนได้ผลลัพธ์ออกมาเป็นโมเดลที่เรียบโค้งสวยงามในภาพตัวอย่าง แต่โมเดลเนี่ยๆ แบบนี้ก็ต้องแลกมาด้วยเวลาในการทำงานที่มากขึ้นจากความซับซ้อนของคำสั่งเป็นเงาตามตัว

จากโครงสร้างโมเดลแบบต่างๆ ที่กล่าวมาล้วนแต่เป็นโครงสร้างที่มีความสำคัญในการสร้างงานสามมิติทั้งสิ้น แต่ในการสร้างเกมสับันโด่งสามมิตินี้จะอาศัยการโครงสร้างแบบ Mesh & Poly ในการสร้างเกมส์ ดังนั้นจึงขออธิบายรายละเอียดในส่วนต่างๆ ของโครงสร้างแบบ Mesh & Poly ดังนี้

2.2.3 โครงสร้างแบบ Poly

เป็นโครงสร้างที่ประกอบขึ้นมาจากจุดและเส้นตรงที่เชื่อมต่อกันจนเกิดเป็นแผ่นระนาบขึ้นมา และเมื่อนำระนาบมาต่อกันเป็นรูปทรงต่างๆ ก็จะเป็นขึ้นงาน 3 มิติ ถือว่าเป็นโครงสร้างที่นิยมใช้งานมากที่สุด ในปัจจุบันเพราะความง่ายในการขึ้นโมเดลเป็นรูปทรงต่างๆ แรกในปัจจุบันโครงสร้างแบบ Mesh และ Poly ยังถูกพัฒนาไปจนถึงขั้นที่สามารถขึ้นโมเดลเป็นรูปทรงฟรีฟอร์มได้สะดวกมากขึ้นนับว่าเป็นโครงสร้างที่เหมาะสมกับผู้เริ่มต้นมากที่สุดโครงสร้างหนึ่ง Mesh และ Poly จะมี Concept ในการทำงานเหมือนกับการปั้น โดยเริ่มจากสร้างวัตถุที่มีรูปทรงใกล้เคียงกับผลงานที่ต้องการเสียก่อน จากนั้นจึงใช้คำสั่งที่โปรแกรมให้มาเพิ่มเติมรายละเอียด และดึงให้ยืดๆ หดๆ จนได้ผลลัพธ์ที่เสร็จสมบูรณ์

2.2.3.1 ส่วนประกอบของ Poly

ส่วนประกอบและคำสั่งในการแก้ไขตกแต่งโครงสร้างแบบ Mesh และ Poly ซึ่งมีลักษณะที่คล้ายคลึงกันเป็นอย่างมาก โดยแต่ละส่วนประกอบจะมีชื่อเรียกทางเทคนิคที่แตกต่างกันออกไปตามรายละเอียดดังนี้

1. Vertex คือจุดที่วางเรียงกันเป็นรูปทรงต่างๆ
2. Edge คือเส้นที่เชื่อมต่อระหว่างจุดหนึ่งไปอีกจุดหนึ่ง
3. Polygon คือระนาบที่เกิดจากการนำเส้นมาวางเรียงต่อกัน
4. Face ส่วนประกอบนี้จะอยู่ในโครงสร้างแบบ Mesh ซึ่งก็คือ Polygon ที่ถูกแบ่งครึ่ง
5. Border คือเส้นขอบของวัตถุในกรณีที่ว่าวัตถุมีลักษณะเหมือนถูกตัดไป
6. Element คือวัตถุทั้งชิ้นที่ทำงานอยู่ ซึ่งเมื่อทำงานไปเรื่อยๆ เราอาจจะนำวัตถุอื่นๆ มาใส่เพิ่ม

ลงไปจนมี Element หลายๆ ชิ้นได้

เนื่องจากโครงสร้างแบบ Mesh และ Poly มีความคล้ายคลึงกันอย่างมากทั้งในเรื่องของเครื่องมือที่ใช้แก้ไขตกแต่งและรูปแบบการทำงาน เรียกได้ว่าถ้ารู้จักการทำงานของ Poly ไประดับหนึ่งแล้ว ก็จะสามารถทำงานกับ Mesh ได้ด้วยวิธีเดียวกัน ดังนั้นเราจึงเลือกเอาการทำงานของ Poly ซึ่งเป็นโครงสร้างที่ใหม่กว่า และนิยมใช้กันอย่างมากในปัจจุบันมาอธิบายเพียงอย่างเดียว

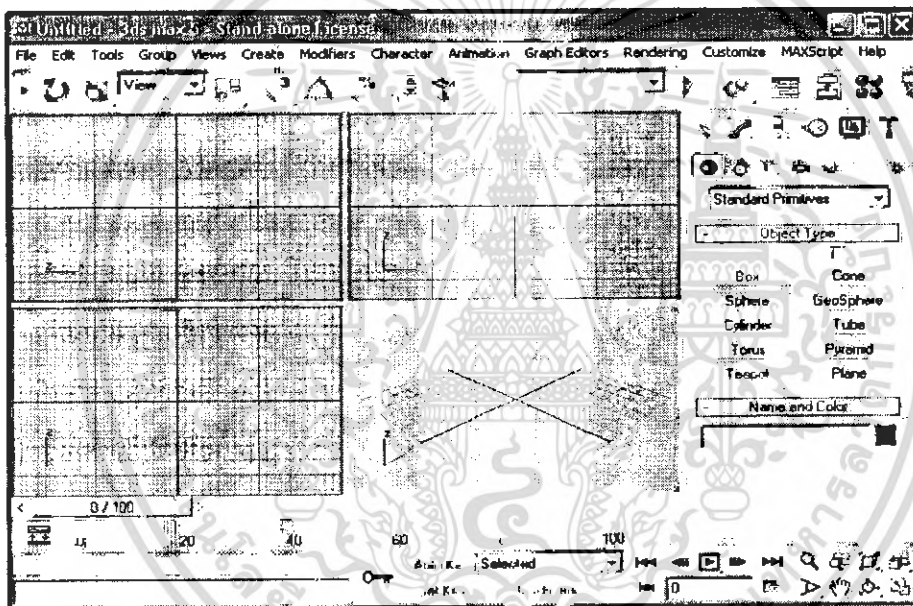
2.2.3.2 ชุดคำสั่งที่ใช้สำหรับแก้ไขตกแต่ง Poly

หลังจากที่ได้ Convert เปลี่ยนโครงสร้างของวัตถุมาเป็น Poly แล้ว โปรแกรมจะใส่ชุดคำสั่งในการแก้ไขตกแต่งมาให้โดยชุดคำสั่งดังกล่าวนี้จะถูกเก็บไว้เป็นชุดๆ ใน  (Modify Panel) ชุดคำสั่งสำหรับแก้ไขตกแต่งรูปทรงของ Poly ในตอนปกติที่ไม่ได้เลือกทำงานกับส่วนประกอบย่อยใดๆ เลย จะแบ่งออกเป็น 4 หมวดใหญ่ๆ คือ

1. Selection เป็นชุดคำสั่งสำหรับกำหนด Option ในการเลือกแบบต่างๆ
2. Soft Selection เป็นการทำงานแบบพิเศษที่จะช่วยเปลี่ยนการย้ายตำแหน่งปกติให้มีการกระจายน้ำหนักออกไปรอบๆ จนวัตถุดูเหมือนจะยืดหยุ่นได้
3. Edit Geometry เป็นชุดเครื่องมือหลักๆ ในการแก้ไขตกแต่งเพิ่มเติมรายละเอียดให้โมเดล
4. Subdivisions Surface เป็นการทำงานแบบพิเศษอีกชนิดหนึ่งที่จะช่วยให้โมเดลแบบเหลี่ยมๆ จาก Poly ดูเรียบโค้งสวยงาม

2.2.3.3 วิธีการสร้างโมเดลจากโครงสร้างแบบ Poly

1. เริ่มต้นเข้าสู่โปรแกรม 3ds max



รูปที่ 2.1 หน้าตาของโปรแกรม 3ds max

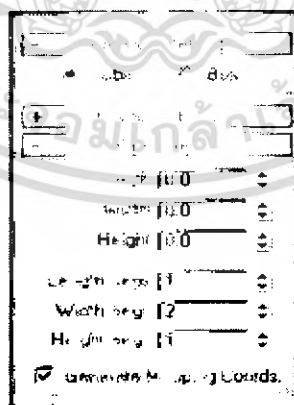
ในการสร้างตัวละครนั้นจะเริ่มจากการสร้างกล่องขึ้นมาหลังจากนั้นจะเป็นการกำหนดรายละเอียดต่างๆ เพื่อใช้ในการกำหนดสัดส่วนของตัวละคร พร้อมทั้งใส่คำสั่ง Symmetry เข้าไปด้วยเพื่อช่วยในการแก้ไขจากด้านใดด้านหนึ่งและจะส่งผลไปยังด้านตรงข้าม ซึ่งก็หมายความว่าหลังจากที่ใส่ Symmetry เข้าไปแล้วในเวลาที่ต้องการจะแก้ไขโมเดลตัวละครทางด้านซ้าย โมเดลทางด้านขวาก็จะถูกแก้ไขไปด้วย ทำให้การทำงานมีความสะดวกมากยิ่งขึ้น โดยขั้นตอนในการสร้างตัวละครมีรายละเอียดดังนี้

1. คลิกปุ่ม  >  >  เป็นการใช้น้ำสั่ง Box สร้างกล่องขึ้นมาสักหนึ่งใบ หลังจากนั้นกำหนดให้ไปวางอยู่ตรงกลางด้วย Coordinate display ดังรูปที่ 2.2



รูปที่ 2.2 เลือกที่คำสั่งสร้างกล่อง

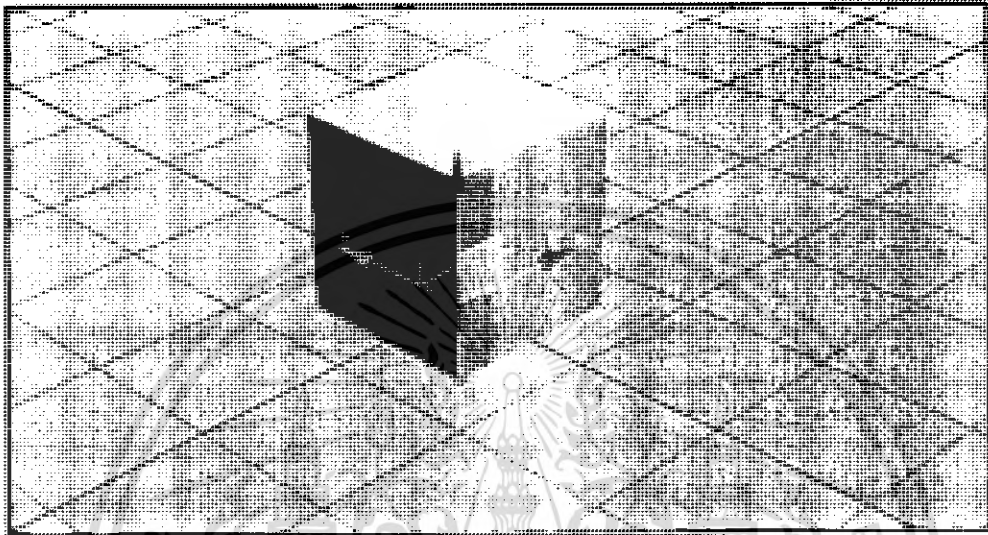
2. กำหนดรายละเอียดของกล่องที่จะสร้างดังนี้ Creation Method เลือก Cube Width - Segs = 2 ดังรูปที่ 2.3



รูปที่ 2.3 กำหนดรายละเอียดของกล่องที่จะสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แดรกเมาส์สร้างกล่องที่ Viewport Top ดังรูปที่ 2.4



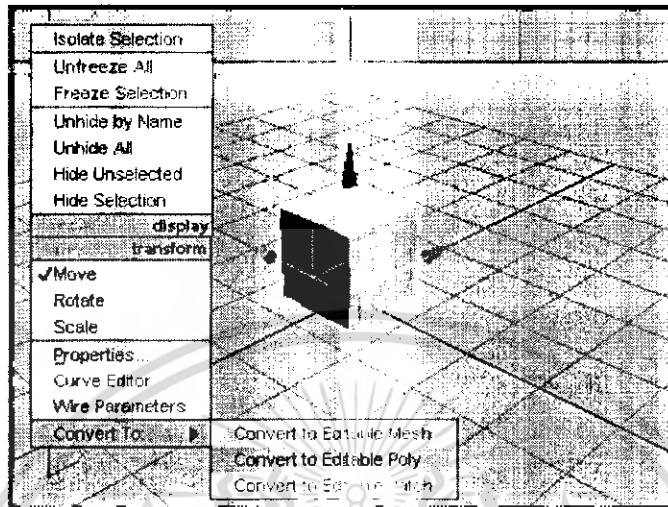
รูปที่ 2.4 สร้างกล่องที่ Viewport Top

4. คลิกเลือก  ให้ทำงานจากนั้นกรอกค่าทุกแกนใน Coordinate display ให้เป็นศูนย์ หลังจากนั้นกล่องที่จะสร้างเป็นโมเดลตัวละครจะย้ายไปอยู่ที่จุด Origin Point ดังรูปที่ 2.5



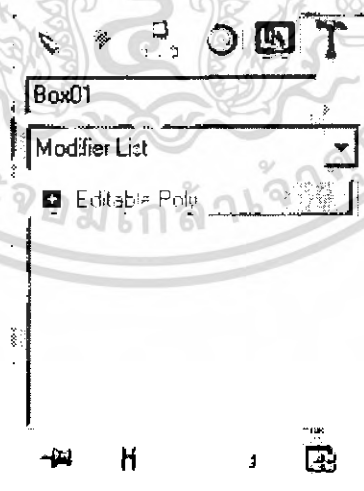
รูปที่ 2.5 กรอกค่าทุกแกนใน Coordinate display ให้เป็น 0

5. คลิกขวาที่โมเดลและเลือกคำสั่ง Convert TO : > Convert to Editable Poly ซึ่งหลังจากเสร็จสิ้นขั้นตอนนี้จะได้กล่องแบบ Polygon เพื่อจะเอาไว้ใช้กับขั้นตอนต่อไปในการขึ้นโมเดลให้เป็นตัวละครตามที่ต้องการดังรูปที่ 2.6



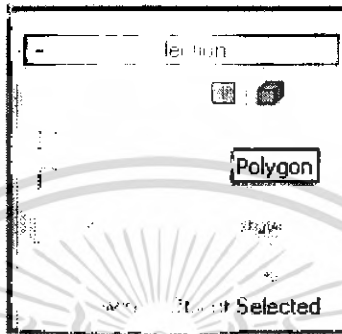
รูปที่ 2.6 เลือกใช้คำสั่ง Convert TO : > Convert to Editable Poly

6. หลังจากเปลี่ยนกล่องที่สร้างให้เป็น Polygon แล้ว โปรแกรมจะเปิด Modify Panel ให้ทำงานโดยอัตโนมัติ ในการทำงานเพียงแค่ลบโมเดลออกไปซีกหนึ่งและหลังจากนั้นก็ทำการคลิกเลือก Symmetry จากช่อง Modify List ใส่ไปได้เลย ดังรูปที่ 2.7

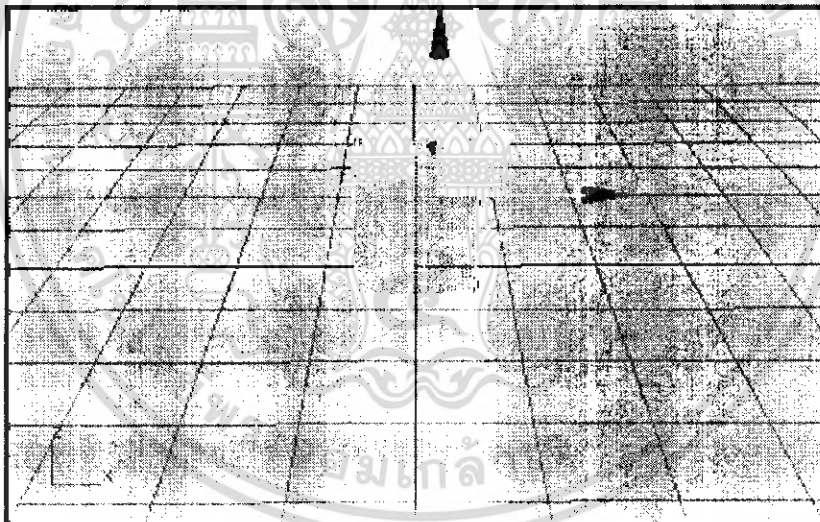


รูปที่ 2.7 การคลิกเลือก Symetry จากช่อง Modify List

7. ใน Viewport Front ให้เลือกทำงานที่  จากนั้นแดรกเมาส์เลือก Face ซีกขวาของกล่องทั้งหมดดังรูปที่ 2.8 และ 2.9

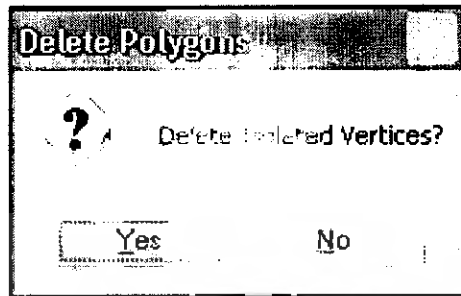


รูปที่ 2.8 เลือกทำงานที่ Polygon

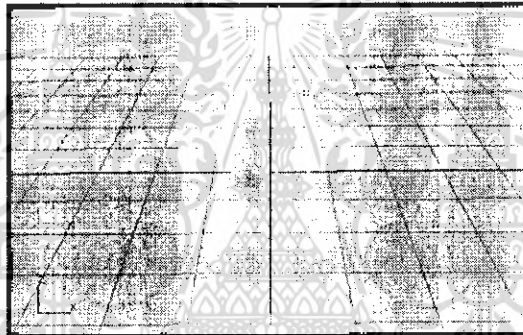


รูปที่ 2.9 แดรกเมาส์เลือก Face ซีกขวาของกล่องทั้งหมด

8. กดปุ่ม <Delete> ลบ Face ที่เลือกทิ้งไป จะมีหน้าต่างให้ยืนยันว่า ต้องการลบจริงหรือเปล่า ให้คลิกปุ่ม ☒ Yes ไปเลย จากนั้นโมเดลจะหายไปซีกหนึ่ง ดังรูปที่ 2.10 และ 2.11

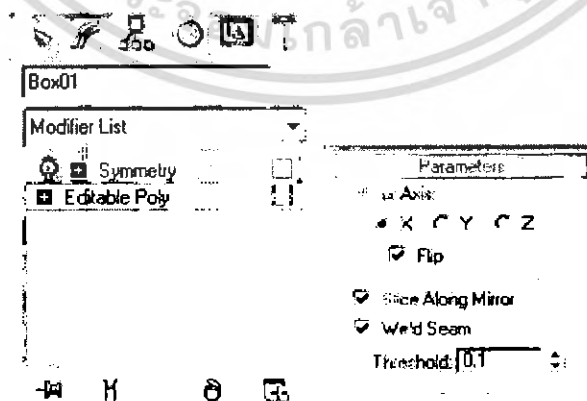


รูปที่ 2.10 หน้าต่างยืนยันการลบ



รูปที่ 2.11 กดปุ่ม <Delete> ลบ Face

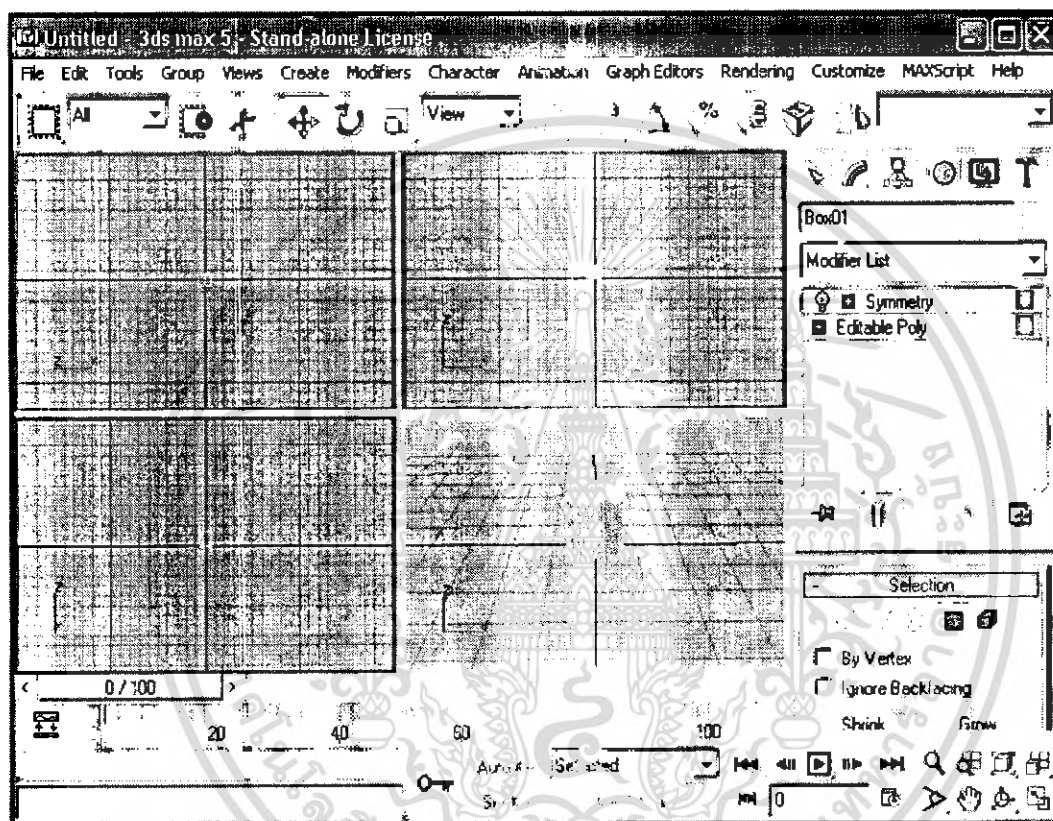
9. คลิกเลือก Symmetry จาก Modify List และคลิกเลือก Flip ที่ Parameters ดังรูปที่ 2.12



รูปที่ 2.12 การเลือก Symmetry และการเลือก Flip

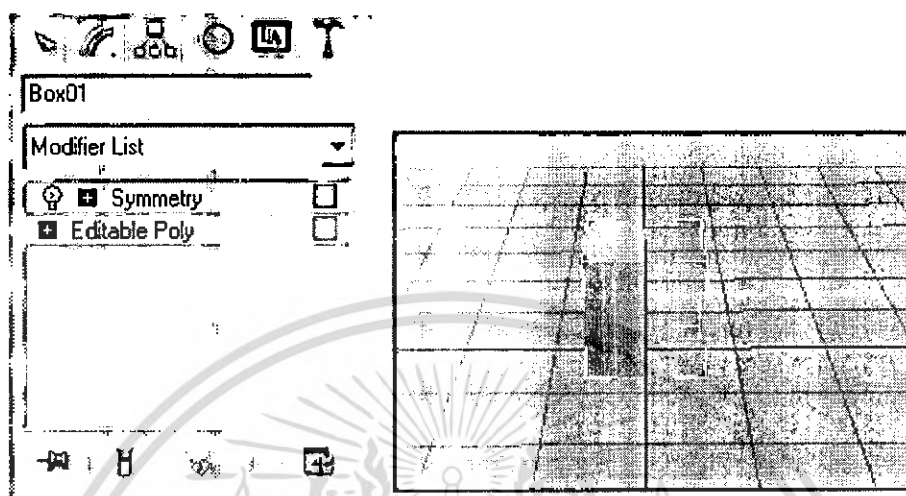
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. หลังจากที่ทำตามลำดับขั้นตอนที่ 9 แล้วจะเห็นว่า จะเป็นการช่วยให้โมเดลของเรากลับมา
กลับมีสองฝั่งเช่นเคย แต่หากมีการแก้ไขฝั่งใดอีกฝั่งหนึ่งก็จะถูกแก้ไขไปด้วยเป็นผลมาจาก
การที่เราใส่ Symmetry ดังรูปที่ 2.13



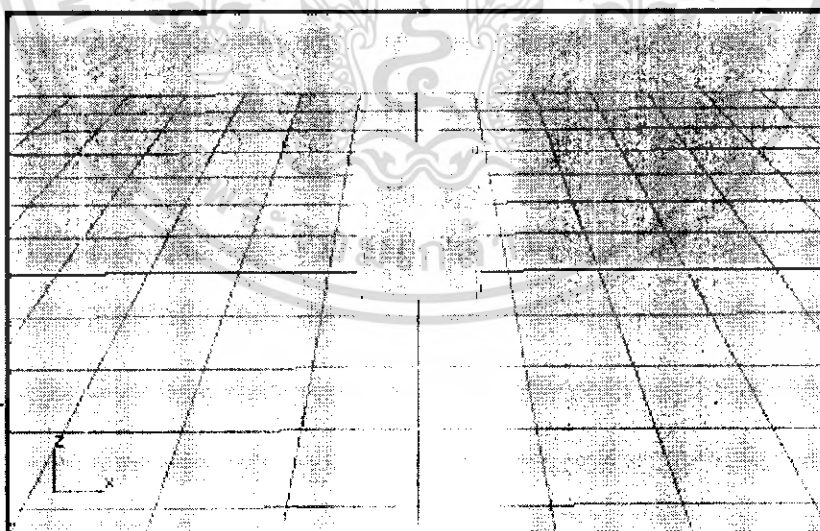
รูปที่ 2.13 ผลจากการใส่ Symmetry

11. คลิกเลือกทำงานใน Editable Poly เพื่อที่จะทำการดัดแปลงรูปร่างของ กล้องให้เป็นตัว
ละครซึ่งจะเห็นว่าโมเดลด้านที่ Symmetry นั้นหายไปต้องคลิกที่ปุ่ม  ก็เพื่อที่จะสั่ง
ให้โปรแกรมโชว์ผลลัพธ์จากคำสั่งที่ใส่ไว้ใน Modify Panel ทั้งหมด ดังรูปที่ 2.14



รูปที่ 2.14 การเลือกทำงานที่ Editable Poly

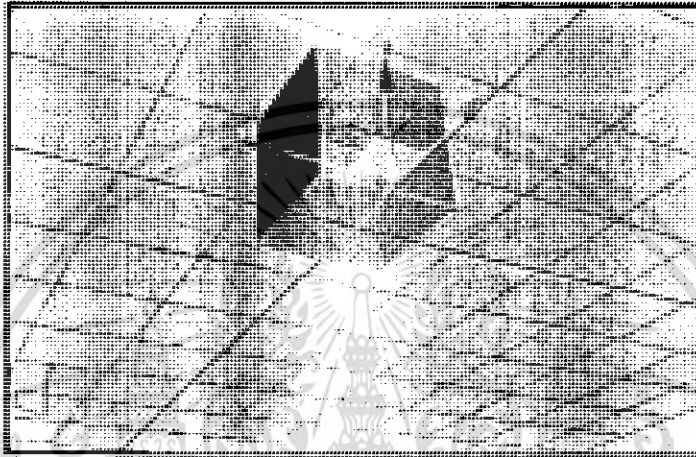
12. โปรแกรมจะแสดงผลที่ได้จากการ Symmetry เข้ามาในโมเดลด้วย โดยสามารถเลือกทำงานใน Editable Poly ได้ตามที่ต้องการ ดังรูปที่ 2.15



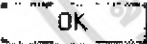
รูปที่ 2.15 แสดงผลลัพธ์ที่ได้จากการ Symmetry เข้ามาในโมเดล

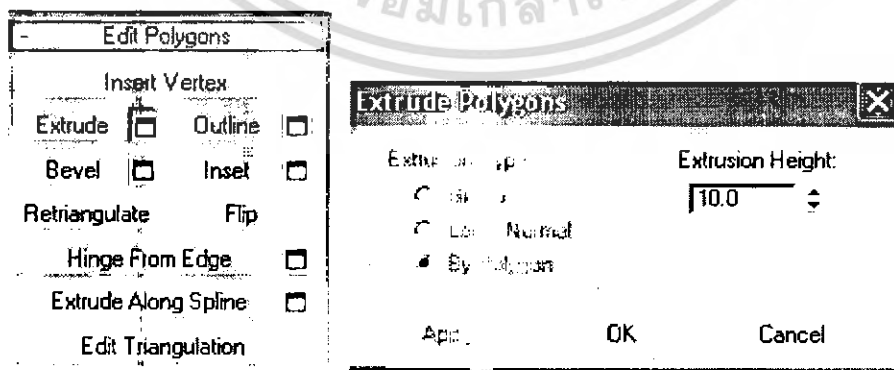
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. ใน Modify Panel ให้คลิกที่ปุ่ม  แล้วคลิกเลือก Face ด้านหลังกับด้านล่างกล่องดังรูปที่ 2.16



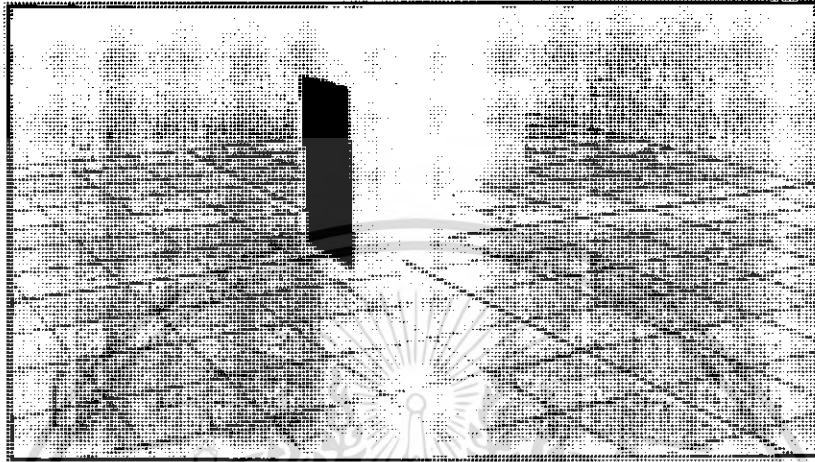
รูปที่ 2.16 เลือก Face ด้านหลังกับด้านล่างกล่อง

14. คลิกที่ปุ่ม  ด้านหลังปุ่ม Extrude ที่หน้าต่าง Extrude Polygons ปรับค่า ดังนี้ Extrude Type เลือก By Polygon และ Extrude Height กำหนดความยาวตามที่ ต้องการแต่จะต้อง ระวังอย่าให้ส่วนที่ Extrude ออกมายาวเกินไป แล้วคลิกปุ่ม  ดังรูปที่ 2.17



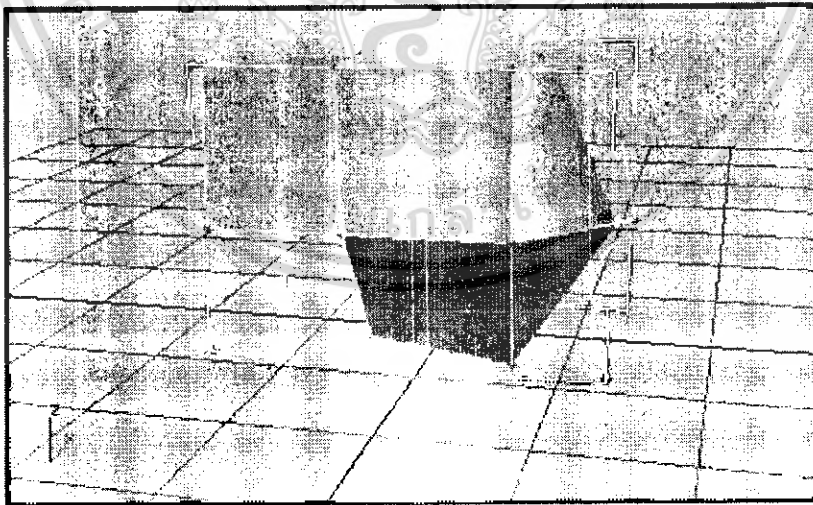
รูปที่ 2.17 ปรับค่าที่หน้าต่าง Extrude Polygons

15. ผลลัพธ์ที่ได้ออกมา ก็จะเป็นรูปทรงที่เป็นต้นแบบให้เราดัดแปลงส่วนต่างๆ ดังรูปที่ 2.18



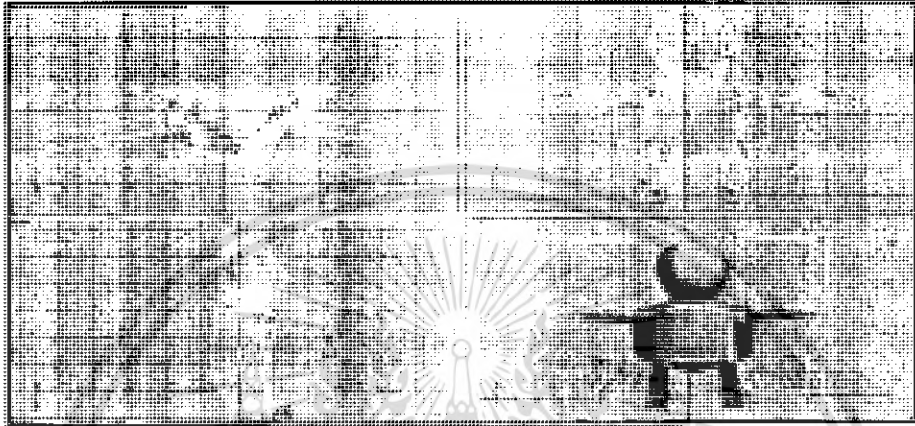
รูปที่ 2.18 ผลลัพธ์จากการปรับค่า Extrude Polygons

16. คลิกเลือกทำงานในระดับ : แล้วใช้เครื่องมือ  ย้ายตำแหน่งในทุกๆ Viewport ให้ได้รูปทรงต่างๆ ตามที่เราได้ออกแบบไว้ ดังรูปที่ 2.19



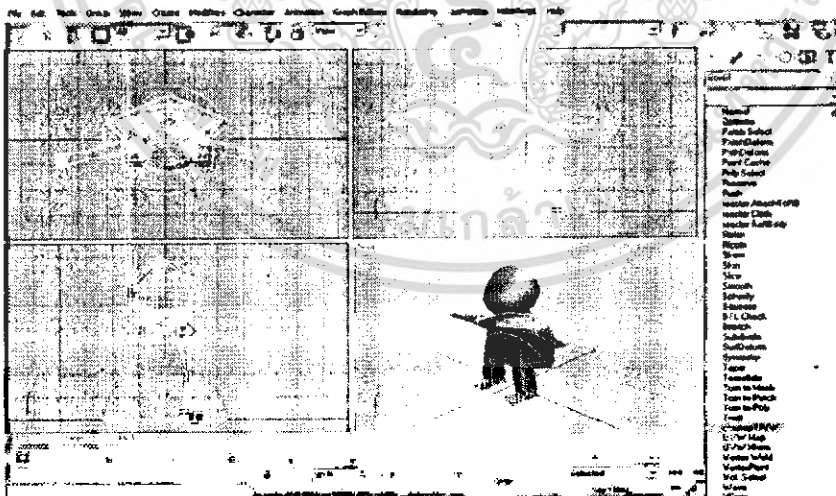
รูปที่ 2.19 การย้ายตำแหน่งของโมเดลเพื่อให้ได้ตัวละครตามที่เราต้องการ

17. หลังจากที่เราได้ทำงานกับเครื่องมือต่างๆ เสร็จสิ้นแล้วเราก็จะได้โมเดลตัวละครที่ใช้เล่นในเกมส์ตามที่เราได้ออกแบบไว้ ดังรูปที่ 2.20



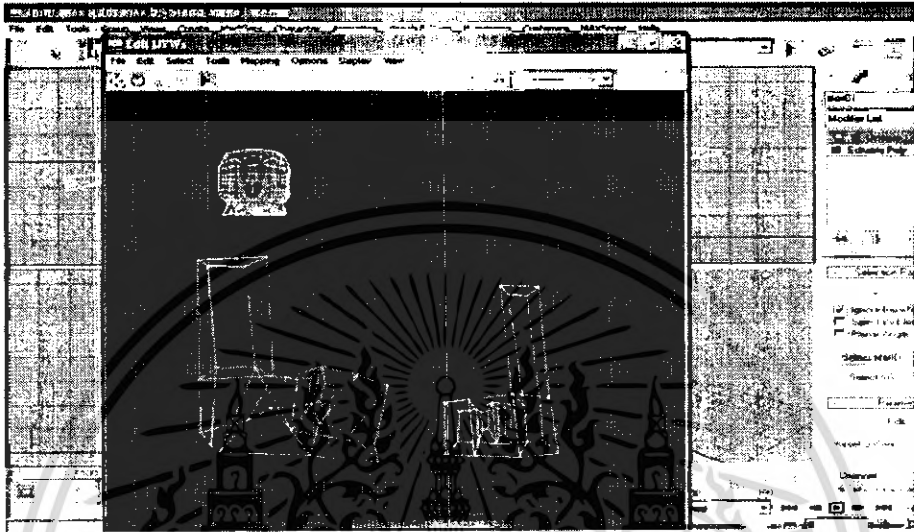
รูปที่ 2.20 รูปร่างของตัวละคร

18. เมื่อได้รูปร่างโมเดลออกมาแล้ว ขั้นตอนต่อไปจะเป็นวิธีการ map ภาพ โดยการคลิกที่ปุ่ม  และเลือกที่ Uvmap UVW ดังแสดงในรูปที่ 2.21



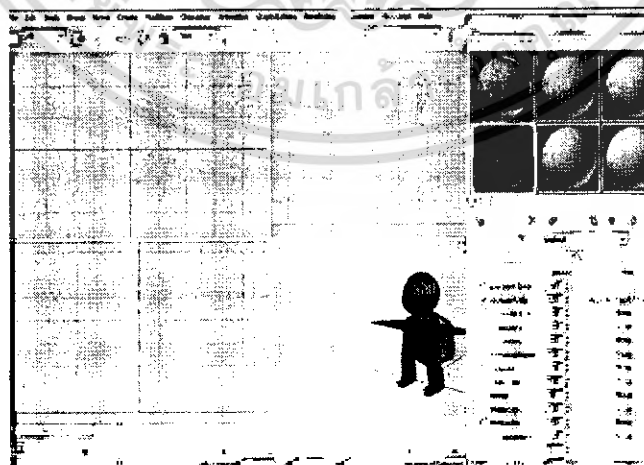
รูปที่ 2.21 วิธีการ map ภาพ

19. หลังจากทำตามขั้นตอนที่ 18 แล้วก็จะปรากฏหน้าต่าง Edit UVW ขึ้นมาดังแสดงในรูปที่ 2.22



รูปที่ 2.22 หน้าต่าง Edit UVW

20. ทำการพรีนสกรีนหน้าต่าง Edit UVW ไปตกแต่งในโปรแกรม Photoshop แล้วก็ทำการบันทึกไฟล์ให้เป็นนามสกุล .bmp หลังจากนั้นให้นำภาพที่ตกแต่งจากขั้นตอนที่ 4 ไปทำการ map ลงในโมเดลที่สร้างไว้ก็จะได้โมเดลที่สมบูรณ์ขึ้นมาดังแสดงในรูปที่ 2.23

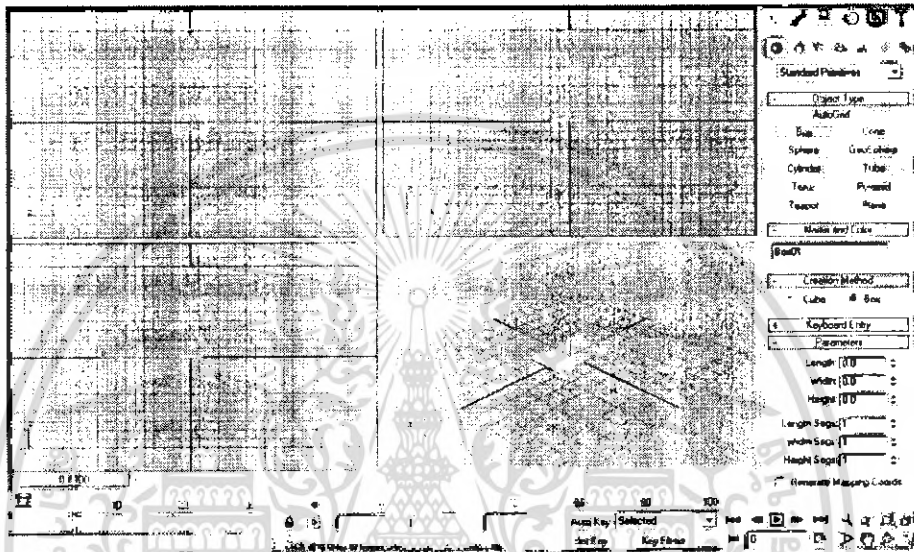


รูปที่ 2.23 โมเดลที่สมบูรณ์ หลังจากการ map

2.2.3.4 วิธีการสร้างฉากจากโครงสร้าง Poly

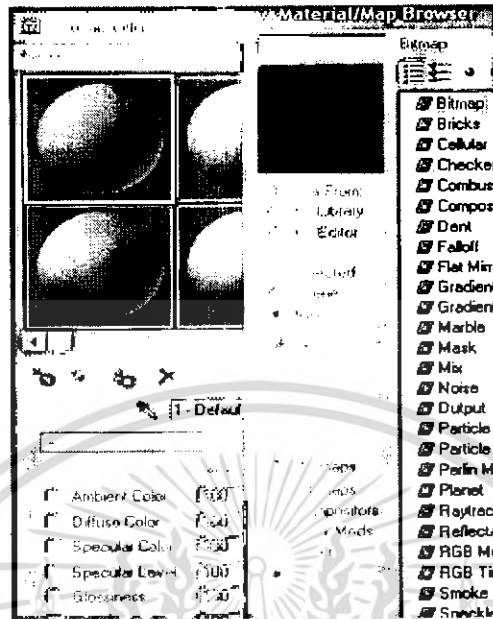
เมื่อผ่านขั้นตอนของการออกแบบฉากแล้ว ต่อมาจะเป็นขั้นตอนของการสร้างฉากที่สามารถใช้งานได้จริงในเกมส์ด้วยโปรแกรม 3ds max ซึ่งมีวิธีการสร้างดังต่อไปนี้

1. ทำการสร้าง Box ขึ้นมาเพื่อเป็นพื้นฐานของฉากดังแสดงในรูปที่ 2.24



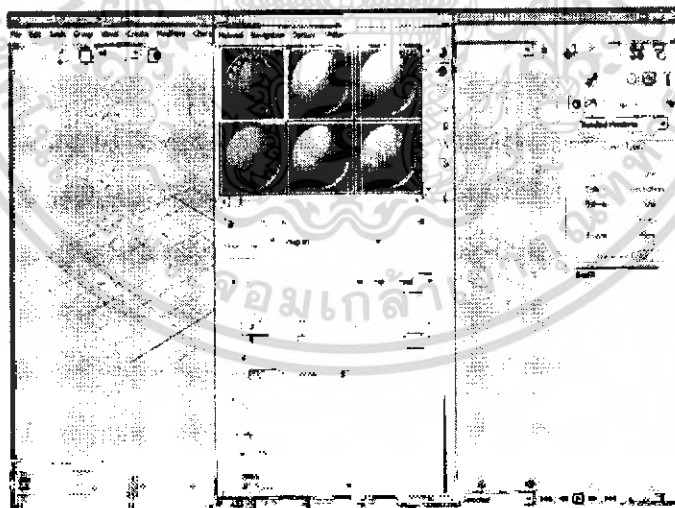
รูปที่ 2.24 สร้าง Box เพื่อขึ้นรูปของฉาก

2. เมื่อได้รูปร่างพื้นฐานของฉากที่เราต้องการแล้ว ในขั้นตอนต่อไปจะเป็นวิธีการ map ภาพโดยการคลิกที่ปุ่มและเลือกที่ Uvmap UVW จะปรากฏหน้าต่าง Edit UVW และทำการปรับสกรีนที่หน้าต่าง Edit UVW ไปตกแต่งในโปรแกรม Photoshop แล้วทำการบันทึกไฟล์ให้เป็นนามสกุล .bmp นำภาพจากหน้าต่าง Edit UVW
3. หลังจากนั้นให้คลิกที่ปุ่ม  (Material Editor) ก็จะปรากฏหน้าต่างขึ้นมา และทำการเลือกไปยัง Bitmap ดังแสดงในรูปที่ 2.25



รูปที่ 2.25 หน้าต่าง Material Editor

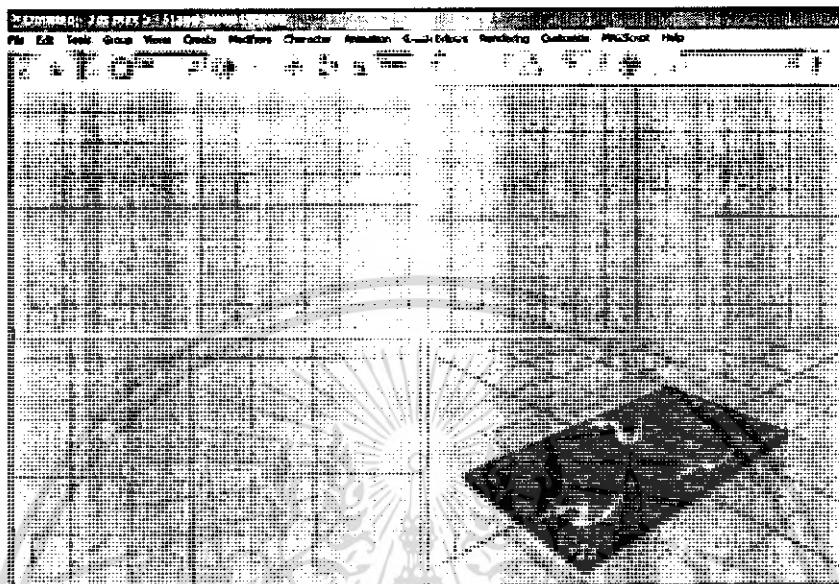
4. จะปรากฏผลของการ Material ดังแสดงในรูปที่ 2.26



รูปที่ 2.26 ผลที่ได้จากการทำ Material

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นให้คลิกที่ปุ่ม  และปุ่ม  จะปรากฏจากที่สมบูรณ์ขึ้นมา ดังแสดงในรูปที่ 2.28



รูปที่ 2.28 ลักษณะจากที่สมบูรณ์

2.3 คำสั่งพื้นฐานในโปรแกรม Dark Basic Professional Education

2.3.1 การแสดงผลบนหน้าจอ

2.3.1.1 Text X,Y "String"

ใช้สำหรับโชว์ตัวอักษรตรงตำแหน่งที่ระบุในพารามิเตอร์ตัวที่ 1 และ 2 โดยตำแหน่งอ้างอิงของข้อความคือตัวอักษรทางซ้ายสุดของตัวอักษร คำสั่งนี้ไม่สามารถนำค่าที่อยู่ในตัวแปรออกมาแสดงได้เองจึงต้องอาศัยฟังก์ชัน Str\$ (ตัวแปร) เพื่อเปลี่ยนตัวเลขให้เป็นตัวอักษรเสียก่อนจึงจะพิมพ์ออกมาได้เช่น Text 100,100" คำตอบ คือ "Str\$(ans) สมมติว่าตัวแปร ans มีค่าเท่ากับ 10 คำสั่งนี้จะแสดงคำว่า "คำตอบคือ10" นั้นเอง

2.3.1.2 Center Text X,Y, "string"

คำสั่งนี้คล้ายกับคำสั่ง Text เพียงแต่จุดอ้างอิงของข้อความจะเปลี่ยนไปอยู่ตรงตัวอักษรตัวกลางของข้อความแทน

2.3.2 การรับค่าจากคีย์บอร์ด

วิธีการรับค่าจากคีย์บอร์ดของ Dark Basic มีอยู่ด้วยกันหลายวิธีด้วยกัน และ Dark Basic ก็มีฟังก์ชันคอยอำนวยความสะดวกให้ผู้พัฒนาโปรแกรมได้เลือกใช้ตามความเหมาะสม และต่อไปนี้จะมาทำความรู้จักกับเครื่องมือต่างๆ ที่จำเป็นต่อการเขียนเกมส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1 คำสั่ง Input String, Variable

คำสั่งนี้จะแสดงตัวหนังสือที่เรากำหนดไว้ในพารามิเตอร์ ตัวแรกที่ชื่อว่า String พารามิเตอร์ตัวแรกนี้ ต้องเป็นชนิดตัวหนังสือเท่านั้นหรืออาจจะไม่ใช่เลยก็ได้ ส่วนพารามิเตอร์ตัวที่สองเป็นชื่อของตัวแปรที่จะให้เก็บข้อมูลจากคีย์บอร์ด ชนิดของข้อมูลจะขึ้นอยู่กับชนิดของตัวแปร (Variable) เมื่อเจอคำสั่งนี้โปรแกรมก็จะหยุดรอข้อมูลจนกว่าผู้ใช้จะกดปุ่ม Enter ตัวอย่างวิธีใช้มีดังนี้

- Input “มีเท่าไรใส่มาให้หมด” แล้วรอข้อมูลจากคีย์บอร์ด เมื่อผู้ใช้กด Enter ข้อมูลก็จะถูกเก็บไว้ในตัวแปรที่ชื่อ Buffer
- คำสั่ง variable = Scancode() คำสั่งนี้จะส่งค่าของปุ่มต่างๆที่ผู้ใช้กด โดยค่าที่ส่งออกมาจะเป็นชนิดตัวเลขจำนวนเต็มเพื่อตรวจสอบเพื่อดูว่าผู้ใช้กำลังกดปุ่มใดในขณะนั้นนั่นเอง ค่าของปุ่มต่างๆ ที่ จะคืนออกมาให้
- Variable = KEYSTATE (หมายเลขคีย์) เป็นการตรวจสอบการกดคีย์ที่ระบุตรงตารางพารามิเตอร์หมายเลขคีย์ เมื่อใดที่ผู้ใช้กดคีย์ที่มีหมายเลขตรงกับที่ระบุไว้ ฟังก์ชันจะคืนค่า 1 (ค่าความจริงเป็นจริง) ออกมา แต่ถ้าผู้ใช้ไม่ได้กดคีย์ที่ระบุ ฟังก์ชันก็จะคืนค่า 0 (ค่าความจริงเป็นเท็จ) ออกมาแทน
- คำสั่ง Variable = upkey() เมื่อผู้ใช้กดปุ่มขึ้น ฟังก์ชันจะคืนค่า 1 ออกมา แต่ถ้าไม่ได้กดปุ่มขึ้นก็จะคืนค่า 0 แทน
- คำสั่ง Variable = downkey() เมื่อผู้ใช้กดปุ่มลง ฟังก์ชันจะคืนค่า 1 ออกมา แต่ถ้าไม่ได้กดปุ่มซ้ายก็จะให้ค่า 0 แทน
- คำสั่ง Variable = Leftkey() เมื่อผู้ใช้กดปุ่มซ้าย ฟังก์ชันจะคืนค่า 1 ออกมา แต่ถ้าไม่ได้กดปุ่มซ้ายก็จะให้ค่า 0 แทน
- คำสั่ง Variable = righthkey() เมื่อผู้ใช้กดปุ่มขวา ฟังก์ชันจะคืนค่า 1 ออกมา แต่ถ้าไม่ได้กดปุ่มขวาก็จะให้ค่า 0 แทน

2.3.3 การรับค่าจากเมาส์

- คำสั่ง Variable = mousex() จะให้ค่าตำแหน่งของเมาส์ในแกน X
- คำสั่ง Variable = mousey() จะให้ค่าตำแหน่งของเมาส์ในแกน Y
- คำสั่ง Variable = mousez() จะให้ค่า 90 เมื่อหมุน Mouse Scroll ขึ้นและค่า 0 เมื่อหมุนลง
- คำสั่ง Variable = mousemovex() จะเป็นคำสั่งที่ใช้จับการเคลื่อนที่ของเมาส์ในแกน X หากผู้ใช้เคลื่อนที่เมาส์ไปทางขวาจะคืนค่าที่มากกว่าศูนย์กลาง หากผู้ใช้เคลื่อนที่เมาส์ไปทางซ้ายก็จะคืนค่าน้อยกว่าศูนย์ออกมา หากผู้ใช้เคลื่อนที่เมาส์ไปทางซ้ายก็จะคืนค่าน้อยกว่าศูนย์ออกมาและหากไม่มีการเปลี่ยนแปลงตำแหน่งในแกน X ก็คืนค่าศูนย์ออกมา

- คำสั่ง Variable = mousemovey() เป็นคำสั่งที่จัดการเคลื่อนที่ของเมาส์ในแกน Y หากผู้ใช้เคลื่อนที่เมาส์ขึ้นด้านบนจะคืนค่าที่น้อยกว่าศูนย์ออกมาและหากไม่มีการเปลี่ยนแปลงตำแหน่งในแกน Y ก็จะได้คืนค่าศูนย์ออกมา

- คำสั่ง Variable = mousemovez() เป็นคำสั่งที่ใช้ในการหมุนเมาส์ครอลล์หากหมุนขึ้นค่าที่คืนออกมาจะเพิ่มขึ้นไปเรื่อยๆ และถ้าหากหมุนลงค่าที่คืนออกมาออกมาจะลดลงไปเรื่อยๆ

2.3.4 การตัดสินใจกับการวนทำงานซ้ำ

2.3.4.1 รูปแบบ IF...endif

1. คำสั่งจะทำงานเมื่อเงื่อนไขเป็นจริงหรือมีค่าที่ไม่ใช่ศูนย์ เรายังสามารถที่จะให้โปรแกรมทำงานได้หลายๆ คำสั่งโดยใช้เครื่องหมาย “,” คั่นระหว่างคำสั่ง เช่น

```
if a=0 Then Print "a is zero": a=a+1
```

รูปแบบคำสั่งแบบที่ 1

If เงื่อนไข Then คำสั่ง

2. ทุกคำสั่งภายในจะทำงานเรียงลำดับจากบนลงล่างและจากซ้ายไปขวาเมื่อเงื่อนไขเป็นจริง

รูปแบบคำสั่งแบบที่ 2

If เงื่อนไข

คำสั่ง1

คำสั่ง2

Endif

3. ถ้าเงื่อนไขเป็นจริงหรือมีค่าไม่เป็นศูนย์คำสั่ง 1 จะทำงาน แต่ถ้าเงื่อนไขเป็นเท็จหรือมีค่าเป็นศูนย์คำสั่ง2 จะทำงานแทน

รูปแบบคำสั่งแบบที่ 3

If เงื่อนไข

คำสั่ง1

else

คำสั่ง2

endif

2.3.4.2 รูปแบบ Select ... endselect

```
select ตัวแปร
    case ค่าคงที่1 : คำสั่ง1 : endcase
    case ค่าคงที่2 : คำสั่ง2 : endcase
    case default : คำสั่ง13 : endcase
endselect
```

โปรแกรมจะเลือกทำคำสั่งใน Case ที่มีค่าที่ตรงกับค่าในตัวแปร แต่ถ้าหากไม่มีค่าคงที่ใดตรงกับค่าในตัวแปรโปรแกรมจะเลือกทำคำสั่งใน Case default แทน

2.3.4.3 รูปแบบ while ... endwhile

```
while เงื่อนไข
```

```
    คำสั่งที่1
```

```
    คำสั่งที่2
```

```
    ...
```

```
endwhile
```

โปรแกรมจะทำงานตามคำสั่งภายใน เมื่อเงื่อนไขมีค่าเป็นจริงหรือมีค่าไม่เป็นศูนย์ หากเรากำหนดเงื่อนไขไม่รัดกุมโปรแกรมอาจจะออกจากลูปไม่ได้หรืออาจจะไม่ยอมทำงานตามคำสั่งภายในเลย

2.3.4.4 รูปแบบ Do...Loop

```
Do
```

```
    คำสั่งที่1
```

```
    คำสั่งที่2
```

```
Loop
```

Do..Loop เป็นการวนทำคำสั่งซ้ำแบบไม่ต้องกำหนดเงื่อนไขมันจะวนทำคำสั่งภายในไปตลอดกาลไม่มีที่สิ้นสุด แต่ถึงอย่างไรผู้พัฒนาก็สามารถที่จะออกจากลูปนี้ได้โดยใช้คำสั่ง break เช่น

Do

$D=d+1$

If $d > 10$ Then break

Loop

2.3.5 ตัวแปรใน Dark Basic Professional Education

ตัวแปรของ Dark Basic Professional Education มีด้วยกัน 3 ชนิดคือ

1. ตัวแปรชนิดจำนวนเต็ม

$A = 0$

2. ตัวแปรชนิดจำนวนจริง

$a\# = 10.0$ เติม “#” ไว้ข้างหลังชื่อตัวแปร

3. ตัวแปรชนิดตัวอักษร

$a\$ = \text{"string"}$ เติม “\$” ไว้ข้างหลังชื่อตัวแปร

เราต้องกำหนดชื่อโดยใช้ตัวอักษรภาษาอังกฤษหรือตัวเลขเท่านั้น และต้องไม่มีเครื่องหมายหรือสัญลักษณ์อยู่ในชื่อตัวแปร อักษรใหญ่หรือเล็กถือเป็นตัวเดียวกัน เช่น a และ A จะถือว่าเป็นตัวแปรตัวเดียวกัน แต่ถ้า a และ $a\#$ จะถือว่าเป็นตัวแปรคนละตัวกัน

ตัวแปรทั้งสามชนิดไม่ต้องประกาศ ก่อนนำมาใช้เหมือนกับภาษาอื่นๆ เราสามารถกำหนดชื่อแล้วใช้ได้ทันที

2.3.6 ตัวแปรชนิดอาร์เรย์

ลักษณะการประกาศตัวแปรอาร์เรย์ dim ชื่อ (จำนวนอาร์เรย์) เช่น หากต้องการประกาศตัวแปรชนิดต่างๆ เป็นจำนวนสิบตัว

Dim $a(9)$

Dim $a\#(9)$

Dim $a\$(9)$

รูปแบบการอ้างถึงข้อมูลในตัวแปรอาร์เรย์โดยการกำหนดตำแหน่งไว้ในวงเล็บตำแหน่งแรก ของอาร์เรย์ จะเริ่มที่ศูนย์ และควรระวังการอ้างถึงข้อมูล ณ ตำแหน่งที่ไม่ได้ประกาศไว้จะทำให้โปรแกรมของเราเกิดข้อผิดพลาดขึ้น เช่น หากเราประกาศไว้ $\text{dim } a(9)$ แต่พอเราอ้างข้อมูลจากอาร์เรย์ตัวนี้กลับอ้างถึง $a(10)$ ด้วยแบบนี้จะทำให้เกิดข้อผิดพลาดในระบบโปรแกรม ตัวอย่างของการกำหนดค่าให้กับอาร์เรย์

$A(0) = 20$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 ตัวแปรอาเรย์หลายมิติ

หากเรามีความต้องการที่จะใช้อาเรย์ที่มีมากกว่ามิติเดียวก็สามารถทำได้โดยกำหนดตัวเลขเพิ่มไว้ในวงเล็บแล้วใช้เครื่องหมาย “,” คั่นเอาไว้

Dim a(2,2)

ตัวอย่างข้างต้นเป็นการประกาศตัวแปรชนิดจำนวนเต็ม แบบสองมิติ ซึ่งจะสามารถเก็บข้อมูลได้เท่ากับ (มิติที่หนึ่ง+1) x (มิติที่สอง+1) จากตัวอย่างนี้ก็จะสามารถเก็บข้อมูลได้ (2+1) x (2+1) เท่ากับ 9 ตัวนั่นเองทั้งนี้ เพราะการอ้างถึงตำแหน่งข้อมูลจะเริ่มจาก 0

2.3.8 เครื่องหมายทางคณิตศาสตร์

การเขียนโปรแกรมอะไรก็ตามล้วนแล้วแต่ต้องมีการคำนวณเข้ามาเกี่ยวข้องทั้งนั้น Dark Basic จึงเลี่ยงไม่ได้ที่จะต้องมียี่ห้อเครื่องมือเพื่อไว้คำนวณหาผลลัพธ์ทางคณิตศาสตร์และเปรียบเทียบค่าต่างๆ เครื่องหมายเหล่านั้นมีดังนี้

1. + เครื่องหมายบวกใช้รวมผลของค่าสองค่า
2. - เครื่องหมายลบหาค่าผลต่างระหว่างค่าสองค่า
3. * เครื่องหมายคูณหาผลคูณ
4. / เครื่องหมายหารหาผลหาร
5. = เครื่องหมายเท่ากับใช้เปรียบเทียบค่าสองค่า หากเท่ากันจะให้ค่าความจริงเป็นจริงและหากไม่เท่ากันจะให้ค่าความจริงเป็นเท็จ
6. <> เครื่องหมายไม่เท่ากับใช้เปรียบเทียบค่าสองค่า หากเท่ากันจะให้ค่าความจริงเป็นเท็จและหากไม่เท่ากันจะให้ค่าความจริงเป็นจริง
7. > เครื่องหมายมากกว่าใช้เปรียบเทียบค่าสองค่า หากค่าทางซ้ายมากกว่าจะให้ค่าความจริงเป็นจริงและหาก ทางขวามากกว่าจะให้ค่าความจริงเป็นเท็จ
8. < เครื่องหมายน้อยกว่าใช้เปรียบเทียบค่าสองค่า หากค่าทางซ้ายมากกว่าจะให้ค่าความจริงเป็นเท็จและหากค่าทางขวามากกว่าจะให้ค่าความจริงเป็นจริง
9. > = เครื่องหมายมากกว่าหรือเท่ากับ หากค่าทางซ้ายมากกว่าทางขวาหรือเท่ากันจะให้ค่าความจริงเป็นจริง แต่ถ้าค่าทางซ้ายน้อยกว่าค่าทางขวาจะให้ค่าความจริงเป็นเท็จ
10. < = เครื่องหมายน้อยกว่าหรือเท่ากับ หากค่าทางซ้ายน้อยกว่าหรือเท่ากันจะให้ค่าความจริงเป็นเท็จ แต่ถ้าค่าทางซ้ายมากกว่าจะให้ค่าความจริงเป็นเท็จ

2.3.9 การสร้างกลุ่มคำสั่งของ Dark Basic fessional Education

2.3.9.1 การสร้าง Label ใน Dark Basic fessional Education

ชื่อลาเบล : นี่คือรูปแบบการกำหนดบรรทัดแรกของลาเบล เราเพียงแต่กำหนดชื่อลาเบลแล้วต่อด้วยเครื่องหมาย “;” ก็เป็นการสร้างชื่อลาเบลแล้ว ประโยชน์ของลาเบลก็คือใช้สำหรับกระโดดไปทำงานตรงบรรทัดที่ระบุเป็นลาเบลนั่นเอง โดยใช้ฟังก์ชัน Gosub ชื่อลาเบล หรือ Goto ชื่อลาเบล

ข้อแตกต่างของ Gosub กับ Goto นั้นมีอยู่ว่า Gosub นั้นจะกระโดดกลับมาทำคำสั่งด้านล่างต่อไป โดยที่เราต้องมีคำสั่ง Return กำหนดบรรทัดสิ้นสุดลาเบลด้วย แต่ถ้าเป็น Goto จะไม่กลับมาอีกแล้วจะกระโดดไปแล้วไปเลยมีรูปแบบดังนี้

```
Label1 :
    คำสั่งที่1
    คำสั่งที่2
    ...
Return
```

ในส่วนของ Return นั้นจะมีหรือไม่ก็ได้ แต่ถ้าไม่มีโปรแกรมจะไม่กระโดดกลับไปตำแหน่งนั้นเอง และการเรียกก็มีวิธีดังนี้ Gosub label1 จะกระโดดกลับมา และ Goto Label1 จะไม่กระโดดกลับมา

2.3.9.2 การสร้าง Function

มีรูปแบบดังนี้

```
Function ชื่อ (ตัวแปร 1, ตัวแปร2)
    คำสั่งที่1
    คำสั่งที่2
    ...
endfunction
```

ตัวแปรที่อยู่ในวงเล็บคือพารามิเตอร์ที่จะส่งเข้ามาเพื่อใช้ภายในฟังก์ชันจะมีหรือไม่ก็ได้ ในส่วนของพารามิเตอร์นี้ หากไม่มีก็เขียนวงเล็บเปิดและปิดติดกันไปเลย และอย่าลืมว่าวงเล็บเปิดต้องติดกับชื่อของฟังก์ชันเลยจะไม่มีเว้นวรรค

คำสั่งปิดฟังก์ชัน endfunction เราจะสามารถคืนค่าออกไปได้โดยระบุไว้ต่อจากค่านี้จะมีหรือไม่ก็ได้ หลังจาก endfunction แล้วต้องเว้นวรรคหนึ่งจึงจะเป็นค่าที่จะส่งออกไป

การเรียกใช้งานฟังก์ชันที่เราสร้างขึ้นก็เช่น mod (num1, num2) เป็นต้น โดยฟังก์ชันนี้ชื่อว่า mod และมีพารามิเตอร์สองตัวก็คือ num1 และ num2 และหากฟังก์ชันมีการคืนค่าออกมาด้วยเราสามารถกำหนดตัวแปรขึ้นมาให้เป็นชนิดเดียวกับค่าที่จะส่งออกมาจากฟังก์ชัน แล้วใช้เครื่องหมายเท่ากับ เพื่อเก็บค่านั้นจากฟังก์ชันได้เลยเช่น หากฟังก์ชันจะคืนค่าที่เป็นชนิดจำนวนเต็มก็จะมีรูปแบบดังนี้

A = mod (8, 2)

2.3.10 ระบบ 3 มิติ

การเตรียมระบบแสดงผลของ Dark Basic เนื่องจาก Dark Basic เป็น Interpreter ซึ่งอำนวยความสะดวกอย่างมากในเรื่องของการแสดงผลติดต่อกับ DirectX ซึ่งถือเป็นข้อดีของ Dark Basic ในการพัฒนาเกมส์ ฟังก์ชันที่เอาไว้สำหรับกำหนดคุณสมบัติการแสดงผล มีดังนี้

1. SET DISPLAY MODE Width, Height, Depth

เป็นฟังก์ชันสำหรับกำหนดโหมดของหน้าจอโดยปกติหากไม่ได้กำหนด Dark Basic จะกำหนดให้เป็นโหมด 648x480x16

2. CHECK DISPLAY MODE (Width, Height, Depth)

ใช้สำหรับตรวจสอบหาโหมดที่สนับสนุนโดย ส่งรายละเอียดโหมดเข้าไปแล้วฟังก์ชันจะคืนค่า 0 ออกมาหากโหมดนั้นไม่สนับสนุน

2.3.10.1 วัตถุสามมิติ

วัตถุสามมิติหรือเรียกอีกอย่างหนึ่งคือโมเดลสามมิติหรือวัตถุนั้นเอง ซึ่ง Dark Basic ก็มีฟังก์ชันที่ใช้สำหรับสร้างโมเดลหลายรูปทรงด้วยกัน แต่ละฟังก์ชันก็จะมีวิธีการใช้งานไม่ต่างกันเท่าใดนัก

1. MAKE OBJECT SPHERE Object Number, Radius Value การสร้างรูปทรงต่างๆ เราต้องมีหมายเลขวัตถุประจำรูปทรงนั้นๆ โดยจะต้องไม่ซ้ำกัน เราสามารถใช้ตัวเลขได้ตั้งแต่ 1 ถึง 65535 เท่านั้น แต่ก็มีจำนวนที่เพียงพอแล้วกับเกมส์ๆ หนึ่ง ส่วนพารามิเตอร์ คือการกำหนดขนาดของรูปทรงกลมจากขนาดของรัศมีนั่นเอง ตัวอย่างการใช้งานฟังก์ชัน

- MAKE OBJECT SPHERE 1, 50 จากตัวอย่างนี้ เป็นการสร้างรูปทรงกลมขนาดเส้นผ่านศูนย์กลางเท่ากับ 50 และถูกกำหนดให้มีเลขอ้างอิงเท่ากับ 1
- MAKE OBJECT CUBE Object Number, Size Value คำสั่งนี้ใช้สำหรับสร้างโมเดลถั่ว เพียงระบุเลขที่วัตถุและขนาดของวัตถุในพารามิเตอร์ตัวที่สอง
- MAKE OBJECT BOX Object Number, Width, Height, Depth คำสั่งนี้เป็นการสร้างกล่องสี่เหลี่ยมโดยระบุหมายเลขวัตถุและความกว้าง ความสูงและความลึก

2.3.10.2 วิธีการโหลดไฟล์สามมิติเข้ามาใช้ในเกมส์

การใช้ฟังก์ชันสร้างรูปทรงสามมิติต่างๆ ภายในโปรแกรมมีข้อจำกัดและเป็นไปได้ยาก เมื่อเราต้องการรูปทรงสามมิติที่มีความซับซ้อนเราไม่สามารถที่จะสร้างโมเดลนก ปลา หรือคน โดยใช้ฟังก์ชันเหล่านี้ได้ Dark Basic จึงมีวิธีที่จะทำให้เราสามารถโหลดโมเดลจากที่สร้างไว้แล้วโดยโปรแกรมสามมิติทั่วไปมาใช้งานในเกมส์ได้ โดยใช้ฟังก์ชันต่อไปนี้

Load object "ชื่อของไฟล์", Object number ใช้โหลดไฟล์ .x ที่ทำขึ้นโดยโปรแกรมสามมิติทั่วไป วิธีการทำโมเดลสามมิติต้องอาศัยทักษะการใช้โปรแกรมอยู่พอสมควร ซึ่งโปรแกรมที่ใช้ในการสร้างที่นิยมก็คือ 3ds studio Max Load mesh "ชื่อของไฟล์", mesh number ใช้ดึงไฟล์สามมิติเข้ามาแล้วเรายังสามารถโหลดเข้ามาอยู่ในรูปแบบของ Mesh นอกจากโหลดเข้ามาเป็นวัตถุแล้วเรายังสามารถโหลดเข้ามาให้อยู่ในรูปแบบของ Mesh ได้เพื่อที่จะใช้สร้างวัตถุพร้อมกับปะพื้นผิวต่อไป

2.3.11 การทำให้วัตถุต่างๆ เคลื่อนไหว

เมื่อเราสามารถใส่พื้นผิวลงลายให้แก่วัตถุได้แล้วก็ถึงคราวของการทำให้มันเคลื่อนไหว ในส่วนของการเคลื่อนที่การหมุนของวัตถุในสามมิตินั้น เราจำเป็นต้องอ้างอิงระบบพิกัดจากถึงสามระบบด้วยกัน พิกัดที่ใหญ่ที่สุดก็คือ

2.3.11.1 ระบบพิกัดโลก

ซึ่งจะเป็นพิกัดที่อยู่หนึ่งไม่มีการย้ายไม่มีการหมุน เพราะต้องเป็นพิกัดสำหรับให้ทุกอย่างในเกมส์อ้างอิงตำแหน่งและจะมีเพียงหนึ่งพิกัดเท่านั้น สมมติว่ามีลูกบอลอยู่ลูกหนึ่งวางนิ่งอยู่ตรงตำแหน่ง (0,0,0) แล้วมีกล่องเลื่อนผ่านบอลไป สิ่งที่เราเห็นไม่ใช่กล่องที่เลื่อนตำแหน่ง แต่เป็นลูกบอลที่เคลื่อนที่แทนทั้งที่ความจริงแล้ว กล่องต่างหากที่เคลื่อนที่ พิกัดนี้เองที่เป็นพิกัดที่ใหญ่ที่สุด

2.3.11.2 ระบบพิกัดท้องถิ่น

เป็นระบบพิกัดที่มีได้หลายพิกัดแล้วแต่จำนวนของโมเดล กล่อง หรือแสงสิ่งเหล่านี้แล้วแต่มีพิกัดท้องถิ่นเป็นตัวของตัวเองทั้งสิ้น โดยจุด (0, 0, 0) ของระบบนี้จะอยู่ตรงจุดอ้างอิง (Pivot) ของสิ่งๆ นั้น การหมุนหรือย้ายตำแหน่งก็เปรียบเทียบกับกับการย้ายหรือหมุนระบบพิกัดท้องถิ่นนี้เอง

2.3.11.3 ระบบพิกัดมุมกล้อง

เราจะสามารถเห็นสิ่งต่างๆ ได้นั้นก็ต้องอาศัยมุมกล้องโดยเราจะคำนวณหาสิ่งที่อยู่ภายในกรอบรูปปริมาตรที่มีปลายยอดชี้เข้าหากกล้องและ Range ของกล้องก็จะสิ้นสุดตรงบานของปริมาตรนั่นเอง ทุกอย่างที่อยู่ด้านในของปริมาตรก็จะถือว่าสิ่งนั้นอยู่ในระยะมุมกล้องด้วย

2.3.12 ฟังก์ชันอำนวยความสะดวกของ Dark Basic fessional Education ที่ใช้สำหรับการทำให้โมเดลสามมิติเคลื่อนไหว

2.3.12.1 POSITION OBJECT Object Number, X, Y, Z

เราสามารถย้ายวัตถุต่างๆ ไปได้ทุกที่บนระบบพิกัดโลกโดยใช้ฟังก์ชันนี้ แล้วระบุหมายเลขวัตถุและตำแหน่งที่จะย้ายไป หากเราทำการเปลี่ยนตำแหน่งที่ละนิดและแสดงผลออกมา ก็จะเป็นการทำภาพเคลื่อนไหวได้

2.3.12.2 MOVE OBJECT Object Number, Step Value

เป็นอีกฟังก์ชันที่ใช้สำหรับย้ายวัตถุไปทางทิศศูนย์กลางของวัตถุหรือจะพูดอีกนัยหนึ่ง คือจะเคลื่อนที่ไปทางด้านหน้าของวัตถุนั้นเองโดยระบุหมายเลขวัตถุและค่าคงที่ที่จะให้เคลื่อนที่ไปจากตำแหน่งเดิม

2.3.12.3 POINT OBJECT Object Number, X, Y, Z

ฟังก์ชันนี้ใช้สำหรับชี้จุดให้วัตถุมุ่งไปหา มักใช้ควบคุมทิศทางการเคลื่อนที่ให้กับฟังก์ชัน Move Object โดยระบุหมายเลขวัตถุที่จะมุ่งไปจุดนี้แล้วระบุจุดที่จะมุ่งไปในพิกัดโลกตรงพารามิเตอร์ X, Y, Z

2.3.13 การหมุนวัตถุ

การหมุนวัตถุเราสามารถหมุนรอบแกนต่างๆ หากหมุนรอบแกน X มุมที่เปลี่ยนคือมุมของแกน Z กับแกน Y นั่นก็คือ เมื่อหมุนรอบแกนอะไรแกนนั้นก็อยู่กับที่ กำหนดค่าได้ถึง 360.0 หากกำหนดค่ามากกว่านี้จะทำให้โปรแกรมเกิดข้อผิดพลาดขึ้นได้ ซึ่งมีฟังก์ชันดังนี้

2.3.13.1 Rotate object object number, X, Y, Z

เป็นฟังก์ชันที่ทำการระบุหมายเลขวัตถุในพารามิเตอร์ตัวแรกแล้ว กำหนดค่ามุมที่จะหมุนตรงพารามิเตอร์ของแกนที่ต้องการ หากไม่ต้องการให้หมุนรอบแกนใดก็กำหนดให้พารามิเตอร์ตัวนั้นเป็นศูนย์

2.3.13.2 XROTATE OBJECT Object Number, X

ใช้สำหรับหมุนรอบแกน X อย่างเดียวแกน X จะอยู่กับที่

2.3.13.3 YROTATE OBJECT Object Number, Y

ใช้สำหรับหมุนรอบแกน Y อย่างเดียวแกน Y จะอยู่กับที่

2.3.13.4 ZROTATE OBJECT Object Number, Z

ใช้สำหรับหมุนรอบแกน Z อย่างเดียวแกน Z จะอยู่กับที่

หากเราต้องการให้มุมที่หมุนไปแล้วนั้นเป็นมุมที่ใช้ในการอ้างอิงหรือให้มุมนั้นกลายเป็นมุมศูนย์กลางไว้ใช้ในการที่โมเดลของเรา กลับหน้ากลับหลังเราก็สามารถหมุน 180 องศาแล้วใช้ฟังก์ชัน

- Fix object Pivot object number

พารามิเตอร์มีเพียงตัวเดียวคือ หมายเลขวัตถุเมื่อใช้ฟังก์ชันนี้ โมเดลของเราก็จะมีค่าองศาอ้างอิงเปลี่ยนไป

2.3.14 วิธีขอทราบค่าตำแหน่งของวัตถุหรือมุมของวัตถุที่หมุนไปจากมุมอ้างอิง

เมื่อเราต้องการทราบว่าวัตถุนี้อยู่ที่พิกัดเท่าไรของระบบพิกัดโลก เราสามารถใช้ฟังก์ชันดังต่อไปนี้

- ตัวแปร = OBJECT POSITION X (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ตำแหน่งของวัตถุในแกน X ของระบบพิกัดโลกนั่นเอง
- ตัวแปร = OBJECT POSITION Y (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ตำแหน่งของวัตถุในแกน Y ของระบบพิกัดโลกนั่นเอง
- ตัวแปร = OBJECT POSITION Z (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ตำแหน่งของวัตถุในแกน Z ของระบบพิกัดโลกนั่นเอง
- ตัวแปร = OBJECT ANGLE X (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ค่า มุมที่หมุนรอบแกน X นั่นเอง
- ตัวแปร = OBJECT ANGLE Y (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ค่า มุมที่หมุนรอบแกน Y นั่นเอง
- ตัวแปร = OBJECT ANGLE Z (Object Number) ค่าที่ส่งออกมาให้แก่ตัวแปรก็คือ ค่า มุมที่หมุนรอบแกน Z นั่นเอง

2.3.15 การตรวจสอบการชนกันของวัตถุ

2.3.15.1 SET OBJECT COLLISION ON Object Number

เป็นการเปิดการตรวจสอบการชนกันของวัตถุ ที่ระบุตรงพารามิเตอร์ Object number โดยปกติ วัตถุจะอยู่ในสถานะนี้อยู่แล้วเมื่อเราสร้างวัตถุนั้นขึ้นมา

2.3.15.2 SET OBJECT COLLISION OFF Object Number

เป็นการเปิดการตรวจสอบการชนกันของวัตถุ ที่ระบุตรงพารามิเตอร์ Object number โปรแกรมจะไม่ตรวจสอบการชนกับวัตถุที่อยู่ในสถานะนี้

2.3.15.3 MAKE OBJECT COLLISION BOX Object Number, x1, y1, z1, x2, y2, z2, flag

เป็นฟังก์ชันสำหรับสร้างพื้นที่ตรวจการชน เป็นลักษณะของกล่องครอบวัตถุเอาไว้โดยกำหนดจุดของ มุมสองมุม พารามิเตอร์ตัวสุดท้ายให้กำหนดเป็นค่าใดค่าหนึ่ง หากไม่ได้มีการกำหนดค่าให้พารามิเตอร์นี้ โปรแกรมจะทำงานไม่ได้

2.3.15.4 DELETE OBJECT COLLISION BOX Object Number

ฟังก์ชันสำหรับลบพื้นที่ตรวจจับการชนกันที่สร้างขึ้นมาจากฟังก์ชัน Make object Collision box เมื่อลบกล่องแล้วโปรแกรมก็จะไม่ตรวจจับการชนกับวัตถุนั้นๆ มีพารามิเตอร์เพียงตัวเดียวคือ หมายเลขวัตถุ ที่ต้องการ

2.3.15.5 SET OBJECT COLLISION TO SPHERES Object Number

กำหนดรูปทรงพื้นที่ของการตรวจจับการชนกันของวัตถุให้กลายเป็นรูปทรงกลม มีพารามิเตอร์เพียงตัวเดียวคือ

2.3.15.6 Object Number

นอกจากฟังก์ชันนี้ ยังมีการกำหนดอีกหลายแบบและมีลักษณะการใช้งานและวิธีการใช้งานเหมือนกัน คือ SET OBJECT COLLISION TO BOXES Object Number และ SET OBJECT COLLISION TO POLYGONS Object Number SET GLOBAL COLLISION ON เป็นคำสั่งเปิดการตรวจสอบการชนกันระหว่างทุกๆ วัตถุในโปรแกรม

2.3.15.7 SET GLOBAL COLLISION OFF

เป็นคำสั่งปิดการตรวจสอบการชนกันระหว่างทุกๆ วัตถุในโปรแกรม

ตัวแปร = OBJECT COLLISION (หมายเลขวัตถุที่ 1, หมายเลขวัตถุที่ 2)

ฟังก์ชันนี้จะส่งค่า 0 ออกมา หากวัตถุที่ 1 ไม่ชนกับวัตถุที่ 2 และจะส่งค่า 1 ออกมาหากวัตถุที่ 1 ชนกับวัตถุที่ 2 และจะส่งหมายเลขวัตถุที่ 1 ที่กำลังชนอยู่ออกมาหากเรากำหนดหมายเลขวัตถุที่ 2 เป็น 0 แต่การกำหนดให้หมายเลขวัตถุตัวที่ 2 เป็นศูนย์นั้นมีความเสี่ยงอยู่ว่าเราจะตรวจสอบการชนได้เพียงวัตถุต่อวัตถุเท่านั้น ไม่สามารถตรวจการชนพร้อมกันหลายๆ วัตถุ เพราะฟังก์ชันจะส่งหมายเลขวัตถุสุดท้ายที่เกิดการชนกันออกมาเท่านั้น เราจึงไม่สามารถตรวจว่าชนกับวัตถุอื่นอีกหรือไม่ เมื่อชนกับวัตถุใดวัตถุหนึ่งอยู่แล้ว

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การออกแบบและการสร้างเกมส์บนโด่งสามมิติแบ่งออกเป็น 3 ส่วนคือ ส่วนของตัวละคร ฉากของเกมส์ และโปรแกรม โดยทั้ง 3 ส่วนนี้จะเป็นการทำงานร่วมกันของโปรแกรม 3ds max และโปรแกรม Dark Basic ในการสร้างตัวละครและฉากนั้นได้ใช้โปรแกรม 3ds max เป็นเครื่องมือในการทำงานนอกจากนั้นการควบคุมทิศทางการเดินของตัวละครรวมถึงตำแหน่งของมุกกล้องจะใช้โปรแกรม Dark Basic ในการพัฒนาเกมส์ ซึ่งเนื้อหาปริญญาณิพนธ์ในบทนี้จะได้กล่าวถึงการออกแบบของเกมส์บนโด่งสามมิติทั้ง 3 ส่วน

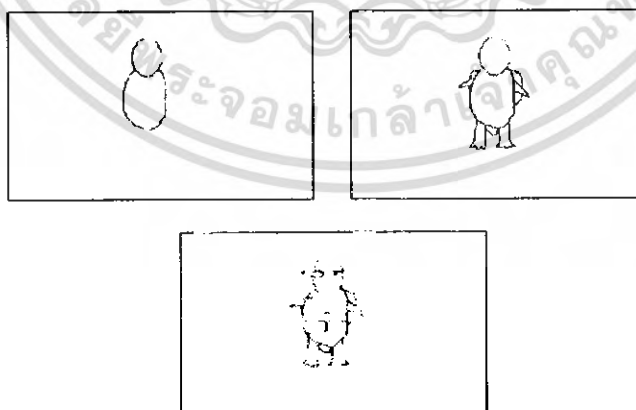
3.2 การออกแบบและการสร้างเกมส์บนโด่งสามมิติ

การออกแบบและการสร้างเกมส์บนโด่งสามมิติดังได้กล่าวแล้วข้างต้นว่าแบ่งออกเป็น 3 ส่วน ดังนั้นจะได้กล่าวถึงการออกแบบทีละส่วน ดังนี้

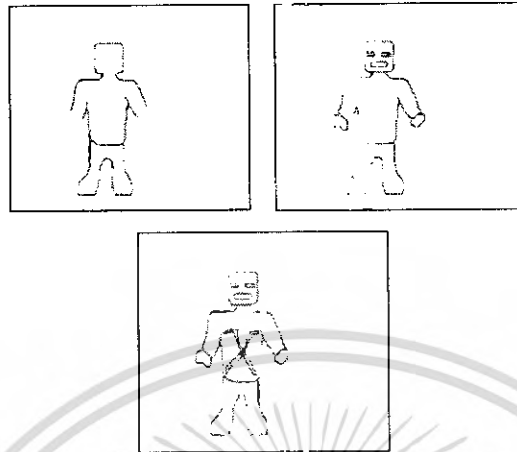
3.2.1 ส่วนของตัวละคร

3.2.1.1 ขั้นตอนของการออกแบบตัวละคร

ในขั้นตอนนี้เป็นขั้นตอนของการคิดและออกแบบลักษณะตัวละครที่จะใช้แทนตัวผู้เล่นแต่ละคน ซึ่งในขั้นแรกเป็นการร่างแบบที่ต้องการบน Storyboard เสียก่อน โดยจะมีลักษณะดังรูปที่ 3.1 และ 3.2

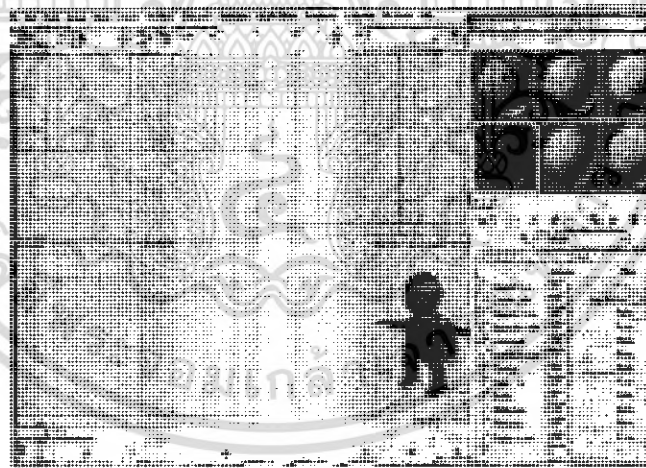


รูปที่ 3.1 ลักษณะของตัวละครรูปเต่า



รูปที่ 3.2 ลักษณะของตัวละครรูปหุ่นยนต์

หลังจากที่ได้ออกแบบตัวละครแล้วให้นำมาสร้างบนโปรแกรม 3ds max จะได้โมเดลที่สามารถใช้งานได้จริง ดังรูปที่ 3.3



รูปที่ 3.3 โมเดลที่สมบูรณ์จากการสร้างด้วยโปรแกรม 3ds max

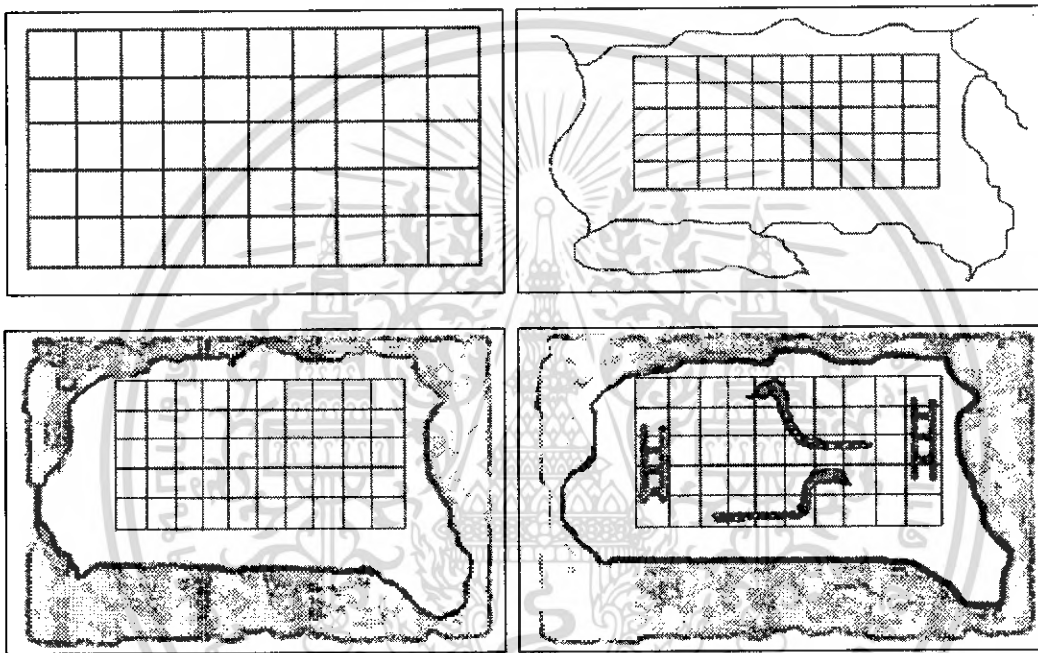
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนของฉากในเกมส์

3.2.2.1 ขั้นตอนของการออกแบบฉากที่ใช้ในเกมส์

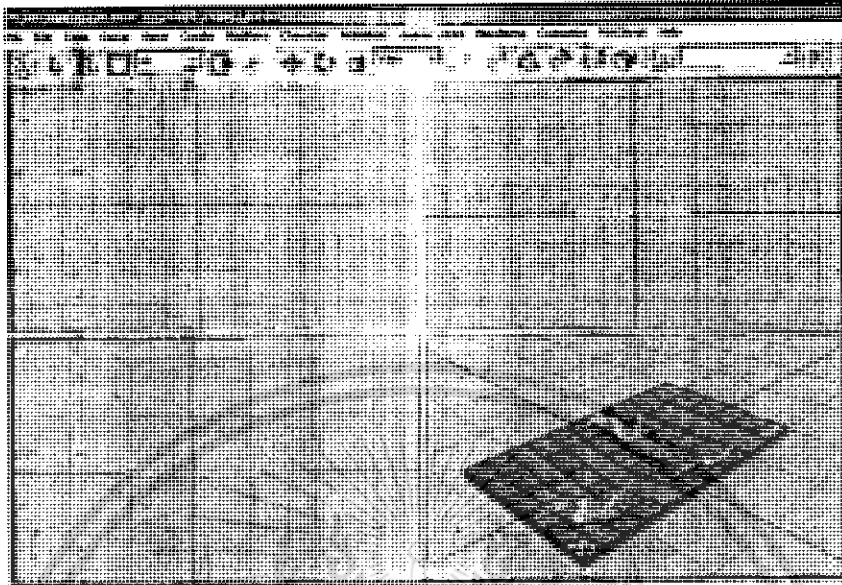
ในส่วนฉากของเกมส์บนโดเมนสามมิติได้ออกแบบลักษณะของฉากเพื่อใช้เป็นเส้นทางในการเดินของตัวละคร ซึ่งก่อนที่จะเป็นฉากจริงที่ใช้ในการเล่นได้ ก็ต้องทำการออกแบบร่างบน Story Board เสียก่อน ดังรูปที่

3.4



รูปที่ 3.4 ลักษณะฉากที่จะใช้ในเกมส์

หลังจากที่ได้ออกแบบฉากแล้วให้นำมาสร้างบนโปรแกรม 3ds max จะได้โมเดลที่สามารถใช้งานได้จริง ดังรูปที่ 3.5



รูปที่ 3.5 ลักษณะฉากที่จะใช้ในเกมส์

3.2.3 ส่วนของโปรแกรม

ส่วนของโปรแกรมที่ใช้ในควบคุมเกมส์บนโดงูสามมิตินั้น จะใช้โปรแกรม Dark Basic Professional Education เป็นเครื่องมือในการเขียนคำสั่งต่างๆ ที่จำเป็นสำหรับการดึงวัตถุที่สร้างขึ้นมาจากโปรแกรม 3ds max เข้ามาแสดงในเกมส์บนโดงูสามมิตินอกจากนั้นยังสามารถใช้สิ่งให้วัตถุต่างๆ รวมไปถึงฉากของเกมส์บนโดงูสามมิติ มุมกล้องและตัวละครที่สร้างขึ้นมาสามารถเคลื่อนไหวได้โดยการโหลดวัตถุขึ้นมา และใช้ฟังก์ชันในการควบคุมวัตถุสามมิติ ซึ่งต่อไปนี้จะเป็นการอธิบายคำสั่งสำคัญที่ใช้ในเกมส์บนโดงูสามมิติว่าคำสั่งแต่ละส่วนทำงานอย่างไร ดังนี้

คำสั่งในการโหลดภาพ

```
load bitmap "Title\Main.bmp",PitM.Loading :sync:sleep 2000 // โหลดภาพที่มีชื่อ
```

```
Main.bmp
```

```
Sync //แสดงภาพที่ โหลด
```

```
show mouse // แสดงเมาส์บนจอภาพ
```

```

rem Start Program Main service
start: // เข้าสู่หน้า menu
do
    cls
    load bitmap "Title\Menu.bmp",PitM.Loading //โหลดภาพที่ชื่อ Menu.bmp
    text 20,20,str$(dasx#)
    text 50,20,str$(dasy#)
    if mousex() >= 39 and mousex() <= 159 and mousey() >= 115 and mousey() <=
136 // เมาส์ในการเลือกเมนู
        sleep 700
        gosub vs_to_vs // กระโดดไปโปรแกรมย่อย gosub vs_to_vs
        if vsMode = 2
            exit
        else
            goto start
        endif
    endif
    if mousex() >=495 and mousex() <= 608 and mousey() >= 115 and mousey() <=
136 then if mouseclick() = 1 then gosub castitle // เมาส์ในการเลือกเมนู castitle
        if mousex() >=501 and mousex() <= 614 and mousey() >= 186 and mousey() <=
205 then if mouseclick() = 1 then end // เมาส์ในการเลือกเมนู end
    sync
loop
cls

```

คำสั่งที่ใช้ในการกำหนด Speed ให้กับ object

Model:

```
select action01.Loading
  case 1:
    load object "Model\Model1.x",action01.Loading // ทำการโหลด Model1.x จน case 8
    set object speed action01.Loading,800    // ให้โมเดลมีความเร็ว 800
  endcase
```

_Control: // ควบคุม

```
select ComComman.Loading
  case 1:
    if K=2
      select action01.Loading // เลือก action01.Loading ก็จะทำ case
        case 1:
          Player1.Start = 21
          Player1.Stop = 39
        endcase
    endcase
```

```
if ch1# >= 2937 and ch1# <= 3138 // ch1# อยู่ระหว่าง 2937-3138
  gosub MV1 //กระโดดไปโปรแกรมย่อย gosub MV1
  ComComman.Loading = 0
  position object action01.Loading,-330,0,-356 //ตำแหน่งของ
    //action01.Loadingให้เท่ากับ -330,0,-356
  yrotate object action01.Loading,180 //หมุน 180
  point camera 0.1,object position x(action01.Loading),object position
  y(action01.Loading),object position z(action01.Loading)
  ch1# = 5361
endif
```

```

rem Vs To Vs Main Form Easy
if CheckEventPlayer.Loading=1
gosub MakeObject
    load object "Screen\sky.X",20
    sync on : sync rate 0 : color backdrop rgb(0,0,64) : backdrop on
    gosub Model
rem position camera 0.1,object position x(21),800,object position z(21)
load object "Model\1.X",101 //โหลด Model ชื่อ Model\1.X ให้ค่าเป็น 101
load object "Model\2.X",102 //โหลด Model ชื่อ Model\1.X ให้ค่าเป็น 101
load object "Model\3.X",103 //โหลด Model ชื่อ Model\1.X ให้ค่าเป็น 101
load object "Model\4.X",104 //โหลด Model ชื่อ Model\1.X ให้ค่าเป็น 101
rem rotate camera 0.1,0,0
load bsp "Screen\map.pk3","map.bsp" //โหลด Model ชื่อ map.pk3
set bsp camera collision 1,0.1,0.75,0 // ความคมกล้องในแนว collision
set bsp collision threshold 1,0.001 // ความคมกล้องในแนว threshold
rem set object speed action01.Loading,800
rem set object speed action02.Loading,800
rem set object speed action03.Loading,800
rem set object speed action04.Loading,800
load image "Tao01\1.JPG",1 //โหลด ภาพ ชื่อ Tao01\1.JPG ให้ค่าเป็น 1
load image "Tao01\2.JPG",2 //โหลด ภาพ ชื่อ Tao01\2.JPG ให้ค่าเป็น 2
load image "Tao01\3.JPG",3 //โหลด ภาพ ชื่อ Tao01\3.JPG ให้ค่าเป็น 3
load image "Tao01\4.JPG",4 //โหลด ภาพ ชื่อ Tao01\4.JPG ให้ค่าเป็น 4
load image "Tao01\5.JPG",5 //โหลด ภาพ ชื่อ Tao01\5.JPG ให้ค่าเป็น 5
load image "Tao01\6.JPG",6 //โหลด ภาพ ชื่อ Tao01\6.JPG ให้ค่าเป็น 6
rem play object
action01.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading
    position camera 0.1,912,150,-767
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position z(action01.Loading)

```

บทที่ 4

การทดสอบและผลการทดสอบ

4.1 กล่าวนำ

ในส่วนของบทนี้จะกล่าวถึงการทดสอบและผลการทดสอบของเกมสบันโดงสามมิติทั้งหมด ไม่ว่าจะเป็นส่วนของโมเดล ฉาก และโปรแกรม ซึ่งผลที่ได้จากการทดสอบเกมสบันโดงสามมิตินี้ได้มาจากการประเมินของผู้ทดลองเล่นเกมสบันโดงสามมิติว่ามีความยากง่ายในการเล่นมากน้อยแค่ไหน ได้รับความสนุกสนานจากเกมนี้นักแค่ไหน

4.2 การทดสอบการทำงานของเกมนี้นักโดงสามมิติ

หลักในการทำงานของเกมนี้นักโดงสามมิตินั้นคือ เมื่อเราเปิดเกมส์ขึ้นมา โปรแกรมก็จะโหลดเข้าสู่หน้าหลักของเกมส์ โดยผู้เล่นสามารถเข้ามาเลือกตัวละครที่จะใช้แทนตัวผู้เล่นแต่ละคนได้ในหน้าหลักนี้ และเมื่อผู้เล่นเลือกตัวละครที่ใช้แทนตัวผู้เล่นได้ ก็จะเริ่มเข้าสู่เกมส์ โดยในขั้นแรกจะให้ผู้เล่นคนที่หนึ่งคลิกหมุนลูกเต๋าและเดินตามแต้มที่หมุนได้ แล้วเดินไปตามจำนวนแต้มที่หมุนได้ หากเจอบันไดก็ให้เข้าไปในช่องถัดไป หากเจอตัวผู้เล่นจะตกไปยังช่องก่อนหน้า หลังจากนั้นผู้เล่นคนที่หนึ่งจะทอยลูกเต๋าและกระทำเช่นเดียวกันผู้เล่นคนที่หนึ่ง เป็นเช่นนี้ไปเรื่อยๆ จนกว่าจะมีผู้เล่นคนใดถึงเส้นชัย เกมส์จึงจะจบลง

4.2.1 วิธีการทดสอบเล่นเกมสบันโดงสามมิติ

4.2.1.1 วิธีการเล่นเกมสบันโดงสามมิติ

1. เริ่มต้นเข้าสู่เกมส์ จะปรากฏหน้าต่างแรกของเกมส์ขึ้นมา ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 หน้าต่างแรกของเกมนี้นักโดงสามมิติ

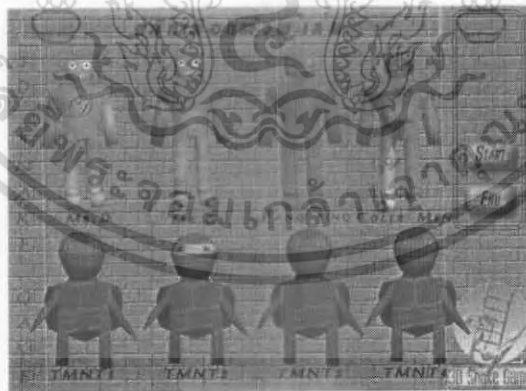
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าต่างเมนูหลักต่างๆ ของเกมส์บันไดงูสามมิติ แสดงในรูปที่ 4.2



รูปที่ 4.2 หน้าต่างเมนูหลักของเกมส์

2.1 เมนู VS TO VS คือ เมนูที่ใช้เลือกในการที่ผู้เล่นต้องการเล่นเกมสลับระหว่างผู้เล่นด้วยกันเอง ซึ่งเมื่อผู้เล่นเลือกคลิกเมนูนี้ก็จะปรากฏหน้าต่างที่ให้ผู้เล่นเลือกตัวละครที่ใช้แทนตัวผู้เล่นแต่ละคน ดังรูปที่ 4.3



รูปที่ 4.3 หน้าต่างเลือกตัวละครที่ใช้แทนตัวผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เมนู VS TO COM คือ เมนูที่ใช้เลือกในกรณีที่ผู้เล่นต้องการเล่นเกมส์ระหว่างผู้เล่นกับคอมพิวเตอร์ ซึ่งเมื่อผู้เล่นเลือกคลิกเมนูนี้ก็จะปรากฏหน้าต่างให้ผู้เล่นเลือกตัวละครดังแสดงในรูปที่ 4.3 เช่นเดียวกับ เมนู VS TO VS

2.3 เมนู CREDIT คือ เมนูแสดงรายละเอียดต่างๆ ของผู้จัดทำเกมส์บันไดงูสามมิติซึ่งเมื่อคลิกเข้าไปจะปรากฏหน้าต่างดังรูปที่ 4.4

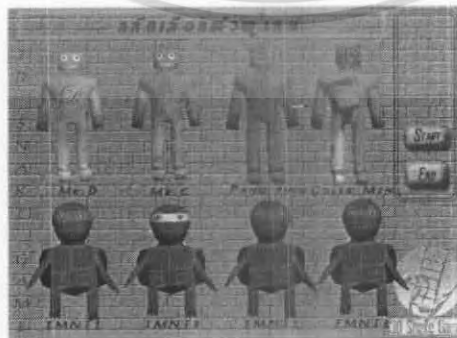


รูปที่ 4.4 หน้าต่างรายละเอียดเมนู CREDIT

2.4 เมนู EXIT คือ เมนูที่ใช้ออกจากเกมส์บันไดงูสามมิติ

4.2.1.2 การทดสอบเกมส์บันไดงูสามมิติ

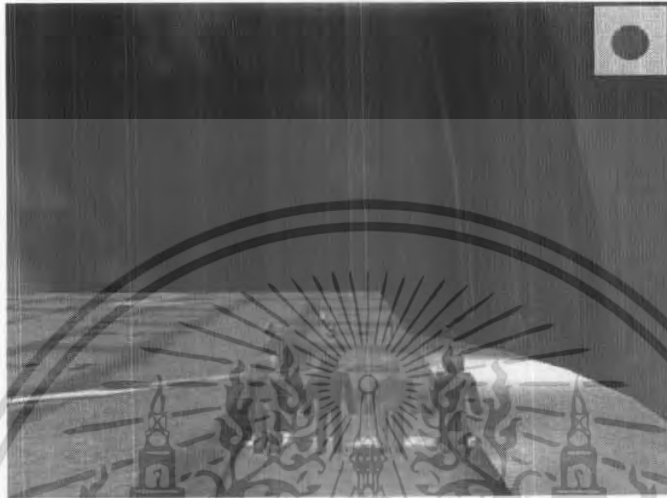
1. กรณีเลือกเมนู VS TO VS จะปรากฏหน้าต่างให้ผู้เล่นเลือกตัวละครที่จะใช้แทนตัวผู้เล่น (โดยขีดจำกัดของเกมส์จะสามารถเล่นได้เพียง 4 คน) ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 หน้าต่างตัวละครที่ใช้แทนตัวผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อผู้เล่นแต่ละคนเลือกตัวละครเรียบร้อยแล้วให้คลิกที่ปุ่ม Start ก็จะเข้าสู่หน้าต่างของเกมส์ ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 เริ่มต้นเกมสลับกันต่อสู้สามมิติ

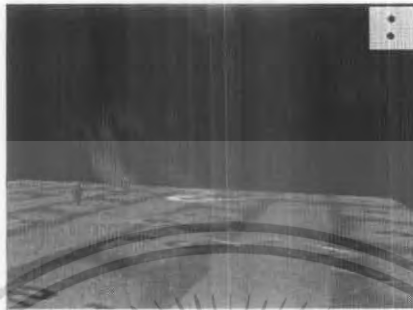
3. ผู้เล่นนั้นสามารถเริ่มเล่นเกมได้โดยการทอยลูกเต๋าซึ่งทำได้โดยการคลิกที่รูปของลูกเต๋าทิศทางขวามือบนหน้าจอที่เห็นดังรูปที่ 4.6
4. หลังจากทอยลูกเต๋ได้แล้ว ให้ผู้เล่นเดินตามแผนที่ที่ทอยได้โดยใช้ปุ่มบนคีย์บอร์ดควบคุมดังรูปที่ 4.7



รูปที่ 4.7 ปุ่มควบคุมเกมสลับกันต่อสู้สามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หลังจากตัวละครเดินไปจนถึงแต้มที่หอยได้แล้ว หากพบว่ามันไดก็ให้ตัวละครขึ้นบันไดและ
ให้ไปปรากฏในช่องถัดไป ดังแสดงในรูปที่ 4.8

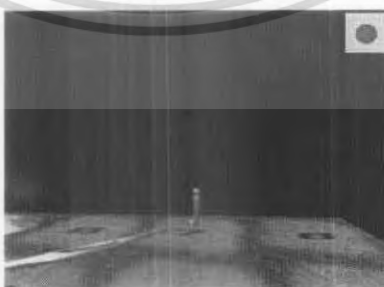


รูปที่ 4.8 แสดงการเดินทางของตัวละคร

6. ในกรณีที่ตัวละครเดินไปแล้วเจอ ก็จะถูกกฏกั้นและจะตกไปยังช่องก่อนหน้าดังรูปที่ 4.9
และ 4.10



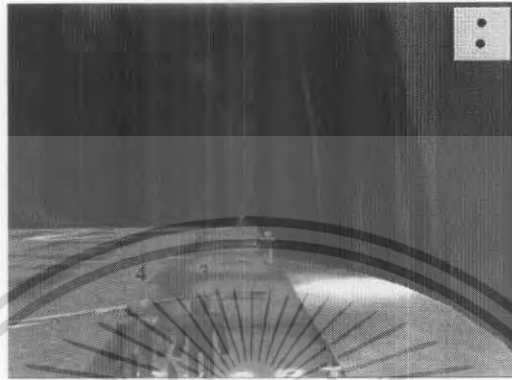
รูปที่ 4.9 ผู้เล่นเดินไปเจอ



รูปที่ 4.10 ถูกกฏกั้นและตกไปยังช่องที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมื่อผู้เล่นคนแรกเล่นเสร็จแล้ว ก็จะถึงคิวของผู้เล่นคนที่สองทอยลูกเต๋า และเดินตามแต้มที่ทอยได้ ซึ่งวิธีการก็จะเหมือนกับผู้เล่นคนแรกทุกประการ แสดงดังรูปที่ 4.11



รูปที่ 4.11 ผู้เล่นต่อไปเริ่มเล่นเกม

8. ให้ผู้เล่นดำเนินการเล่นไปเช่นนี้ ไปจนกว่าจะมีผู้เล่นคนใดคนหนึ่งได้เข้าสู่เส้นชัยเกมก็จะถึงจุดจบ ดังแสดงในรูปที่ 4.12



รูปที่ 4.12 ผู้เล่นเข้าเส้นชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดสอบจากผู้ทดลองเล่นเกมสับันโดงสามมิติ

4.2.2.1 ผลการประเมินจากการทดลองของผู้ทดลองเล่นเกมสับันโดงสามมิติ คนที่ 1

แบบประเมิน การทดสอบเกมสับันโดงสามมิติ					
รายการประเมิน	ระดับความคิดเห็น				
	ดีมาก	ดี	ปานกลาง	พอใช้	ปรับปรุง
1. ความน่าสนใจของเกมสับันโดง			/		
2. ความเหมาะสมของฉากและตัวละคร			/		
3. ความสมบูรณ์ของเกมสับันโดง				/	
4. โดยภาพรวมผู้เล่นได้รับความสนุกสนาน			/		
ข้อเสนอแนะ.....					
.....					

4.2.2.2 ผลการประเมินจากการทดลองของผู้ทดลองเล่นเกมสับันโดงสามมิติ คนที่ 2

แบบประเมิน การทดสอบเกมสับันโดงสามมิติ					
รายการประเมิน	ระดับความคิดเห็น				
	ดีมาก	ดี	ปานกลาง	พอใช้	ปรับปรุง
1. ความน่าสนใจของเกมสับันโดง		/			
2. ความเหมาะสมของฉากและตัวละคร			/		
3. ความสมบูรณ์ของเกมสับันโดง				/	
4. โดยภาพรวมผู้เล่นได้รับความสนุกสนาน			/		
ข้อเสนอแนะ.....					
.....					

4.2.2.3 ผลการประเมินจากการทดลองของผู้ทดลองเล่นเกมส์คนที่ 3

แบบประเมิน การทดสอบเกมส์บันไดงูสามมิติ					
รายการประเมิน	ระดับความคิดเห็น				
	ดีมาก	ดี	ปานกลาง	พอใช้	ปรับปรุง
1. ความน่าสนใจของเกมส์		/			
2. ความเหมาะสมของฉากและตัวละคร			/		
3. ความสมบูรณ์ของเกมส์				/	
4. โดยภาพรวมผู้เล่นได้รับความ สนุกสนาน		/			
ข้อเสนอแนะ.....					
.....					

4.2.2.4 ผลการประเมินจากการทดลองของผู้ทดลองเล่นเกมส์คนที่ 4

แบบประเมิน การทดสอบเกมส์บันไดงูสามมิติ					
รายการประเมิน	ระดับความคิดเห็น				
	ดีมาก	ดี	ปานกลาง	พอใช้	ปรับปรุง
1. ความน่าสนใจของเกมส์	/				
2. ความเหมาะสมของฉากและตัวละคร		/			
3. ความสมบูรณ์ของเกมส์			/		
4. โดยภาพรวมผู้เล่นได้รับความ สนุกสนาน		/			
ข้อเสนอแนะ.....					
.....					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.5 ผลการประเมินจากการทดลองของผู้ทดลองเล่นเกมส้นที่ 5

แบบประเมิน การทดสอบเกมส้นไฉงสามมิติ					
รายการประเมิน	ระดับความคิดเห็น				
	ดีมาก	ดี	ปานกลาง	พอใช้	ปรับปรุง
1. ความน่าสนใจของเกมส้น			/		
2. ความเหมาะสมของฉากและตัวละคร			/		
3. ความสมบูรณ์ของเกมส้น			/		
4. โดยภาพรวมผู้เล่นได้รับความ สนุกสนาน			/		
ข้อเสนอแนะ.....					
.....					

4.2.3 สรุปผลการทดลอง

จากที่ได้ให้ผู้ทดลองทั้ง 5 คนได้ทดลองเล่นเกมส้นไฉงสามมิติแล้ว สรุปผลได้วาระดับความน่าสนใจของเกมส้นอยู่ในเกณฑ์ที่ดี เพราะเป็นรูปแบบการเล่นเกมส้นไฉงที่แปลกใหม่ เกมส้นมีมิติมากขึ้นกว่าเดิมที่เคยเล่นบนกระดาน ส่วนความเหมาะสมของฉากและตัวละครความสนุกสนานที่ได้รับจากการเล่นเกมส้นนี้ จากการประเมินพบว่าอยู่ในเกณฑ์ปานกลาง แต่สิ่งที่ต้องปรับปรุงในเกมส้นไฉงสามมิตินี้ก็คือความสมบูรณ์ของเกมส้น ที่ให้เกมส้นสามารถเล่นกับคอมพิวเตอร์ได้

บทที่ 5

บทสรุป

5.1 สรุป

เกมส์บันไดงูสามมิติ เป็นเกมส์ที่สร้างขึ้นจากโปรแกรมหลักๆ 2 โปรแกรมคือ โปรแกรม 3ds max และโปรแกรม Dark Basic Professional โดยผู้จัดทำได้ใช้โปรแกรม 3ds max ในการสร้างฉากและตัวละครที่ใช้แทนตัวผู้เล่นแต่ละคน และได้ใช้โปรแกรม Dark Basic Professional เขียนโปรแกรมให้เกมส์ทำงานตามรูปแบบที่ได้วางไว้ในตอนต้น

ซึ่งเกมส์บันไดงูสามมิตินี้มีข้อจำกัดอย่างหนึ่งคือสามารถเล่นได้เพียง 4 คนเท่านั้น ส่วนวิธีการเล่นก็ไม่ได้แตกต่างไปจากการเล่นเกมบันไดงูบนกระดานมากนัก วิธีการเล่นก็คือผู้เล่นจะต้องเลือกตัวละครให้แทนตัวผู้เล่นแต่ละคนก่อน เมื่อได้ตัวละครตามต้องการแล้วก็จะเริ่มเล่น โดยให้ผู้เล่นคนแรกจะทำการหมุนลูกเต๋าก่อน ได้แต้มเท่าใดให้เดินไปตามแต้มที่หมุนลูกเต๋าคือได้ ซึ่งถ้าหากเดินไปแล้วพบบันไดก็ให้ขึ้นบันได และตัวละครก็จะไปปรากฏในช่องต่อไปตามที่ได้โปรแกรมไว้ กรณีที่เดินไปพบงู ตัวละครก็จะถูกกินและจะไปปรากฏในช่องก่อนหน้าตามที่ได้โปรแกรมไว้เช่นกัน สำหรับผู้เล่นคนที่สองและคนต่อไป ก็ทำเช่นเดียวกับผู้เล่นคนแรก จะสิ้นสุดเกมส์ก็ต่อเมื่อมีผู้เล่นคนใดคนหนึ่งเดินไปถึงเส้นชัย โปรแกรมก็จะถามผู้เล่นว่าต้องการเล่นเกมใหม่หรือต้องการออกสู่วินโดวส์

5.2 ปัญหาและวิธีการแก้ไข

จากการดำเนินการสร้างและการทดสอบเกมส์ปรากฏว่ามีปัญหาเกิดขึ้นหลายประการ ซึ่งสามารถสรุปได้ดังนี้

1. ในการสร้างตัวละครมี Polygon มากเกินไป เป็นสาเหตุให้เวลาในการโหลดภาพค่อนข้างใช้เวลานาน
วิธีการแก้ไข ลบ Polygon ที่ไม่จำเป็นออกให้มากที่สุด หรือเลือก MSmooth เฉพาะบริเวณที่ต้องการรายละเอียดชัดเจน
2. เนื่องจาก ตารางที่ใช้เดิน สร้างจากโปรแกรม 3ds max เป็นพื้นเดียวกันทั้งหมดจึงทำให้การกำหนดระยะและขนาดช่องนั้นได้ยาก เมื่อนำตัวละครลงมาเดินจึงต้องหน้าพิกัด x,y ของตารางที่ใช้เดิน
วิธีการแก้ไข หาพิกัดของช่องตารางแต่ละช่องโดยอ้างอิงจากโปรแกรม 3ds max ในพิกัด x,y นำมาใส่ไปชี้ที่ภาพก็จะมีพิกัดของ x,y แสดงออกมา

3. Model บางตัวเมื่อโหลดเข้าไปในโปรแกรม Dark basic Professional บางครั้งจะมองไม่เห็นวัตถุหรือตัว Model
วิธีการแก้ไข กำหนด พิกัดหรือตำแหน่งของวัตถุใหม่อีกครั้งเพราะบางครั้งในการกำหนดวัตถุอาจจะทำกับตำแหน่งหรือพิกัดกับวัตถุอื่นอยู่

5.3 แนวทางการพัฒนา

1. สำหรับเกมส์บันไดงูสามมิตินี้ รูปแบบในการเล่นขณะนี้เห็นได้ว่าจะอยู่ในลักษณะของการเล่นแบบ VS TO VS เท่านั้น ในการพัฒนาต่อไป ผู้พัฒนาอาจเพิ่มให้มีการเล่นแบบ VS TO COM ได้ โดยการเขียนโปรแกรมเพิ่มเติมเข้าไปในส่วนนี้ เพื่อให้สามารถเล่นได้ทั้ง VS TO VS และ VS TO COM
2. ในอนาคตสำหรับผู้ที่ต้องการพัฒนาเกมส์บันไดงูสามมิติสามารถพัฒนารูปแบบฉากให้สามารถเลือกได้มากกว่า 1 ฉาก ซึ่งวิธีการสร้างผู้พัฒนาสามารถสร้างฉากเพิ่มโดยใช้โปรแกรม 3ds max ในการสร้างฉาก และเขียนโปรแกรมให้สามารถเลือกฉากที่ต้องการเล่นได้ โดยใช้โปรแกรม Dark Basic Professional ได้ตามความต้องการของผู้พัฒนา
3. ในการพัฒนาขั้นต่อไปของเกมส์บันไดงูสามมิตินี้ ผู้พัฒนาอาจพัฒนาให้เกมส์สามารถเล่นผ่านเครือข่ายอินเทอร์เน็ตได้ โดยทำให้มีลักษณะของเกมส์ออนไลน์ทั่วไปๆ ไป ซึ่งก็สามารถทำได้ไม่ยากนัก หากแต่ต้องศึกษาตรงส่วนนี้เพิ่มเติมสักเล็กน้อยเท่านั้น

บรรณานุกรม

ปิยะบุตร สุทธิตารา. 2547. **3ds max 6 basic**. กรุงเทพฯ : บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

พูนศักดิ์ ฐนพันธ์พานิช. 2545. **เทคนิคการสร้าง Character modeling บน 3ds max 4.2**.
กรุงเทพฯ : เอส.พี.ซี. บัคส์.

ฐิตารัตน์ รัชตะวราณ. 2545. **Photoshop เพื่องานกราฟฟิกและตกแต่งภาพ**. นนทบุรี : อินโฟเพรส.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน
เกมส์บันไดงูสามมิติ



ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำแนะนำเบื้องต้น

สำหรับการใช้งานเกมส์บนโดงูสามมิติ คอมพิวเตอร์ที่จะต้องมีการ์ดจอแสดงผลทางด้านกราฟฟิกพอสมควร

2. ปุ่มที่ใช้ควบคุมการเล่นในเกมส์บนโดงูสามมิติ

- 2.1 ปุ่มเลข 1,2,3,4 เพื่อควบคุมการเดินของผู้เล่นคนที่ 1,2,3,4 ตามลำดับ
- 2.2 ปุ่มควบคุมทิศทาง ซ้าย-ขวา,บน-ล่าง ใช้ในการควบคุมกล้องให้สามารถแพนดูได้ทั้งฉาก
- 2.3 เมาส์จะใช้คลิกลูกเต๋าให้ทำการ Random แต้มเดินให้กับตัวละคร

3. การติดตั้งและใช้งานเกมส์บนโดงูสามมิติ

- 3.1 ทำการติดตั้งเกมส์บนโดงูสามมิติลงเครื่องคอมพิวเตอร์
- 3.2 การเริ่มเล่นเกมส์ให้ดับเบิลคลิกที่ไฟล์นามสกุล .exe จะเข้าสู่หน้าจอแรกของเกมส์ดังรูป ก.1



รูปที่ ก.1 หน้าแรกของเกมส์

หลังจากนั้นโปรแกรมจะโหลดเข้าสู่หน้าต่างเมนูหลักของเกมส์ดังรูปที่ ก.2

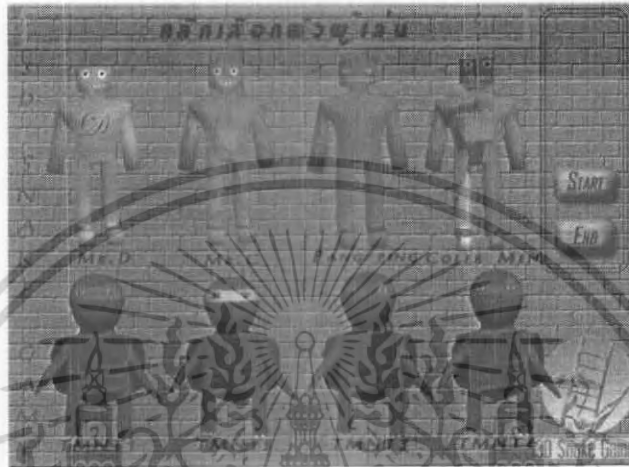


รูปที่ ก.2 เมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 วิธีการเล่นเกมสึบนโดงสามมิติ

1. กรณีที่ผู้เล่นเลือก VS TO VS ก็จะเข้าสู่หน้าต่างสำหรับเลือกตัวละครที่ใช้แทนตัวผู้เล่นแต่ละคนโดยสามารถเลือกได้เพียง 4 ตัวเท่านั้น ดังรูปที่ ก.3



รูปที่ ก.3 หน้าต่างเลือกตัวละครที่ใช้แทนตัวผู้เล่น

2. ผู้เล่นเลือกตัวละครที่ใช้แทนตัวผู้เล่นแต่ละคนเรียบร้อยแล้ว ให้คลิกปุ่ม start ด้านขวามือบนหน้าจอ ก็จะเข้าสู่หน้าจอเพื่อเริ่มเล่นเกมสึ ดังรูปที่ ก.4

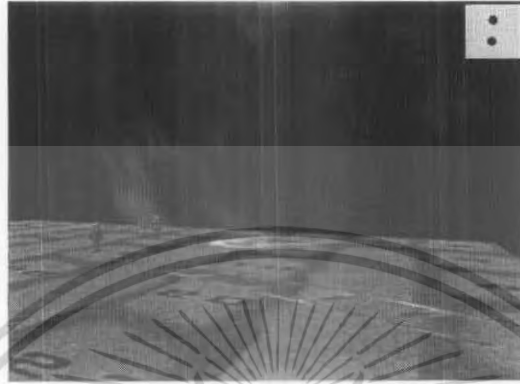


รูปที่ ก.4 เริ่มเล่นเกมสึ

3. ให้ผู้เล่นคนแรกคลิกลูกเต๋าเพื่อ Random แต้มเดิน

4. เมื่อทอยลูกเต๋ได้แล้วก็ดูว่าได้แต้มเท่าใดและใช้ไปบนคีย์บอร์ดควบคุมให้ตัวเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กรณีที่ตัวละครเดินไปเจอบันได ตัวละครก็จะไปปรากฏยังช่องถัดไปตามที่ได้โปรแกรมไว้ละครเดินไปตามแต้มที่ Random ได้ ดังรูปที่ ก.5



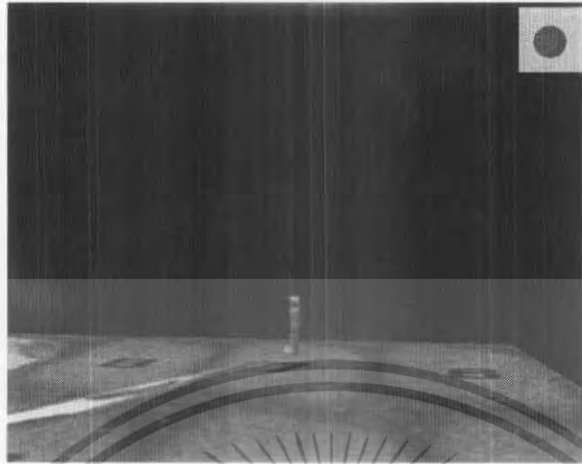
รูปที่ ก.5 กรณีผู้เล่นเดินไปเจอบันได

6. กรณีที่ตัวละครเดินไปเจออุ้ง ตัวละครก็จะถูกงูกินและจะไปปรากฏยังช่องก่อนหน้าตามที่ได้โปรแกรมไว้ ดังรูปที่ ก.6 และ ก.7



รูปที่ ก.6 ผู้เล่นเดินไปเจออุ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 ผู้เล่นถูกูกินและตักมายังช่องก่อนหน้าที่กำหนดไว้

7. สำหรับผู้เล่นคนต่อไปจะกระทำเช่นเดียวกับผู้เล่นคนแรกทำเช่นนี้ไปเรื่อยจนครบจำนวนผู้เล่นทุกคน
8. หากมีผู้เล่นคนใดคนหนึ่งเดินถึงเส้นชัยเกมก็จะสิ้นสุดลง ดังแสดงในรูปที่ ก.8



รูปที่ ก.8 มีผู้เล่นเข้าเส้นชัย

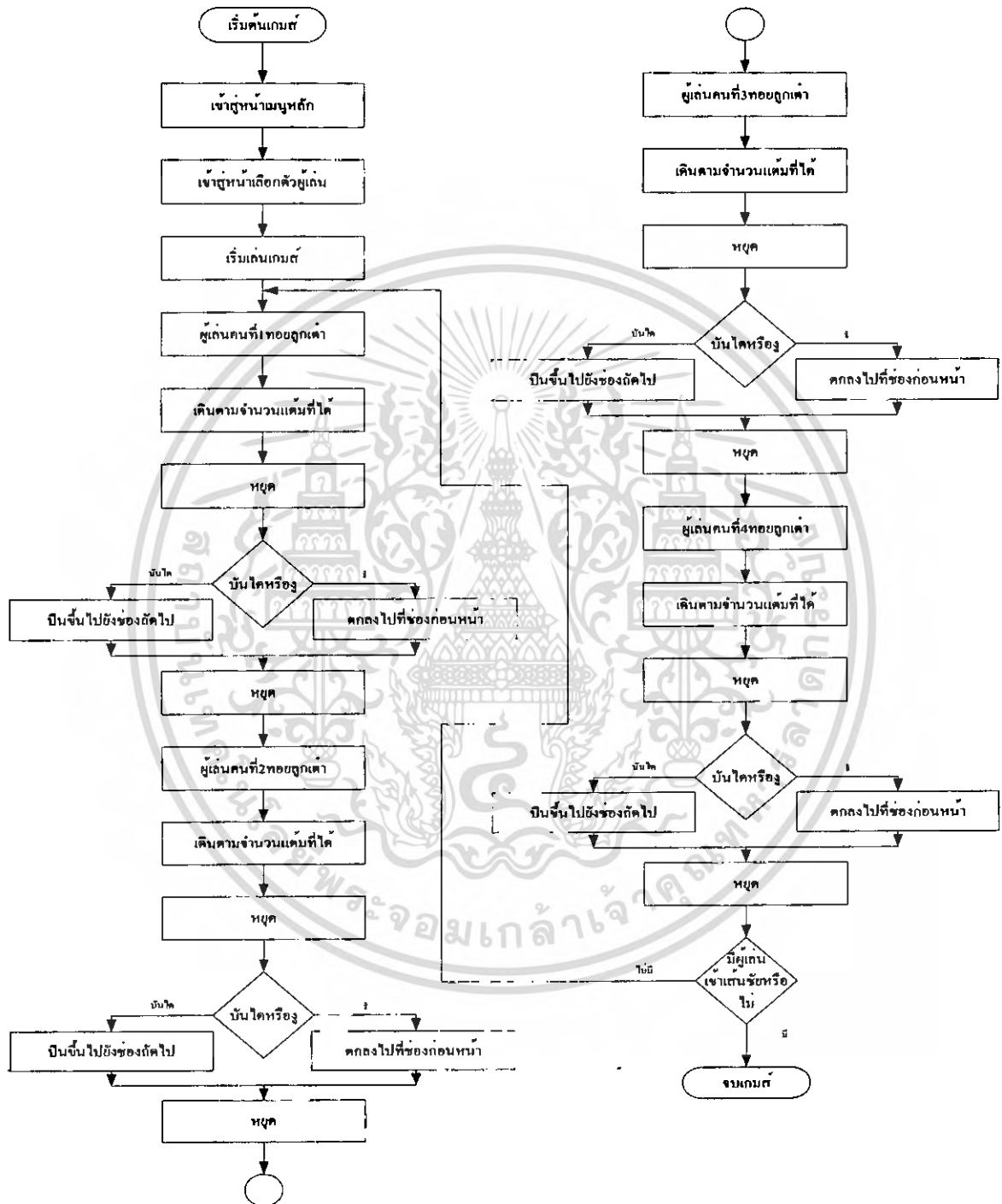
9. ในขณะที่เล่น ผู้เล่นสามารถใช้ปุ่มควบคุมที่ได้กล่าวไปแล้วข้างต้นในข้อ 2 ดูตัวละครของผู้เล่นและฉากได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังงานของเกมส์บันไดสามมิติ



รูปที่ ข.1 ผังงานของเกมส์บันไดสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค
รหัสต้นฉบับของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของเกมส์

Rem Project: Snake

Rem Created: 24/3/2007 18:43:31

Rem ***** Main Source File *****

sync on:sync rate 100

hide mouse

autocam on

set camera range 0.1,15000

type Player

 Loading as integer

 Camera as float

 Start as integer

 Stop as integer

endtype

 action01 as Player

 action02 as Player

 action03 as Player

 action04 as Player

 Player1 as Player

 Player2 as Player

 PlayerCou as Player

 playerComman as Player

 ComComman as Player

 CheckEventPlayer as Player

 CheckEventPlayer.Loading = 0

 PitM as Player

rem variable Model

 action01.Loading = 1

 action02.Loading = 2

 action03.Loading = 3

 action04.Loading = 4

 ComComman.Loading = 1

 PlayerCou.Loading = 160

 Player1.Start = 21

```

Player1.Stop = 39
rem Variable Camera
PitM.Loading = 0
imag1 = 1
imag2 = 2
imag3 = 3
imag4 = 4
imag5 = 5
imag6 = 6
imag7 = 7
imag8 = 8
vsMode = 0
player_select = 0
RndCCom = 0
playerComman.Loading = 0
ComComman.Loading = 0
Toy = 99
Scene01 = 5
Check01 = 1
Check02 = 1
Check03 = 1
Check04 = 1
RndToy = 1
strFile$=""
ch1# = 0
ch2# = 0
ch3# = 0
ch4# = 0
P = 0
rem Load Title Heard Programming
load bitmap "Title\Main.bmp",PitM.Loading :sync:sleep 2000
sync
show mouse
rem Start Program Main service
start:
do

```

```

cls
load bitmap "Title\Menu.bmp",PitM.Loading
text 20,20,str$(dasx#)
text 50,20,str$(dasy#)
if mousex() >= 39 and mousex() <= 159 and mousey() >= 115 and mousey() <= 136
    sleep 700
    gosub vs_to_vs
    if vsMode = 2
        exit
    else
        goto start
    endif
endif
if mousex() >=495 and mousex() <= 608 and mousey() >= 115 and mousey() <= 136 then if
mouseclick() = 1 then gosub castitle
    if mousex() >=501 and mousex() <= 614 and mousey() >= 186 and mousey() <= 205 then if
mouseclick() = 1 then end
    sync
loop
cls
rem Reload Game Over Action on Key
rem Vs To Vs Main Form Easy
if CheckEventPlayer.Loading=1
gosub MakeObject
    load object "Screen\sky.X",20
    sync on : sync rate 0 : color backdrop rgb(0,0,64) : backdrop on
    gosub Model
rem position camera 0.1,object position x(21),800,object position z(21)
    load object "Model\1.X",101
    load object "Model\2.X",102
    load object "Model\3.X",103
    load object "Model\4.X",104
rem rotate camera 0.1,0,0
    load bsp "Screen\map.pk3","map.bsp"
    set bsp camera collision 1,0.1,0.75,0
    set bsp collision threshold 1,0.001

```

```

rem set object speed action01.Loading,800
rem set object speed action02.Loading,800
rem set object speed action03.Loading,800
rem set object speed action04.Loading,800

load image "Tao01\1.JPG",1
load image "Tao01\2.JPG",2
load image "Tao01\3.JPG",3
load image "Tao01\4.JPG",4
load image "Tao01\5.JPG",5
load image "Tao01\6.JPG",6

rem play object action01.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading
position camera 0.1,912,150,-767
point camera 0.1,object position x(action01.Loading),object position y(action01.Loading),object position
z(action01.Loading)
rem Main loop
do
control camera using arrowkeys 0.1,1,1
s# = camera position x(0.1)
ss# = camera position y(0.1)
sss# = camera position z(0.1)
Mu# = mousex()
MY# = mousey()
text 50,70,str$(s#)
text 50,90,str$(ss#)
text 50,110,str$(sss#)

gosub Toi
PASTE IMAGE RndToy, 563, 6

rem point camera 0.1, object position x(21),0,-600
rem point camera 0.1,object position x(Toy),object position y(Toy),object position z(Toy)
K = scancode()
select K
case 2:
ComComman.Loading = 1
endcase
case 3:
ComComman.Loading = 2

```

```

endcase
case 4:
    ComComman.Loading = 3
endcase
case 5:
    ComComman.Loading = 4
endcase
endselect
gosub _Control
text 300,300,str$(ch1#)
text 300,330,str$(ch2#)
text 300,360,str$(ch3#)
text 300,390,str$(ch4#)
text 100,100,strFile$
text 50,50,str$(K)
text 50,200,str$(xxx1#)
text 50,220,str$(yyy1#)
text 50,240,str$(zzz1#)
text 50,260,str$(P)
position object 101,object position x(action01.Loading),object position y(action01.Loading)+5,object
position z(action01.Loading)
position object 102,object position x(action02.Loading),object position y(action02.Loading)+5,object
position z(action02.Loading)
position object 103,object position x(action03.Loading),object position y(action03.Loading)+5,object
position z(action03.Loading)
position object 104,object position x(action04.Loading),object position y(action04.Loading)+5,object
position z(action04.Loading)
sync
loop
endif
rem end vs mode on the game
vs_to_vs:
checkout1 = 0
load image "Title\b1on.jpg",1
load image "Title\b1on.jpg",2
load image "Title\b1on.jpg",3

```

```

load image "Title\b1on.jpg",4
load image "Title\b1on.jpg",5
load image "Title\b1on.jpg",6
load image "Title\b1on.jpg",7
load image "Title\b1on.jpg",8
load image "Title\b1off.jpg",9
load image "Title\b1off.jpg",10
load image "Title\b1off.jpg",11
load image "Title\b1off.jpg",12
load image "Title\b1off.jpg",13
load image "Title\b1off.jpg",14
load image "Title\b1off.jpg",15
load image "Title\b1off.jpg",16

while checkout1 = 0
    c# = mousex()
    y# = mousey()
    cls
    text 20,50, str$(c#)
    text 20,55, str$(y#)
    load bitmap "Title\Model.bmp",PitM.Loading
    PASTE IMAGE imag1, 130,238
    PASTE IMAGE imag2, 271,239
    PASTE IMAGE imag3, 400,239
    PASTE IMAGE imag4, 511,238
    PASTE IMAGE imag5, 144,432
    PASTE IMAGE imag6, 271,430
    PASTE IMAGE imag7, 414,430
    PASTE IMAGE imag8, 535,430
    if player_select <= 3
        if mousex() >=130 and mousex() <= 158 and mousey() >= 237 and mousey() <= 256 then if
mouseclick()=1
            if imag1 = 9 then player_select = player_select - 1
            imag1=9
            sleep 100
            player_select = player_select + 1

```

```

select player_select
  case 1:
    action01.Loading = 1
  endcase
  case 2:
    action02.Loading = 1
  endcase
  case 3:
    action03.Loading = 1
  endcase
  case 4:
    action04.Loading = 1
  endcase
endselect
endif
if mousex() >=271 and mousex() <= 298 and mousey() >= 239 and mousey() <= 257 then if
mouseclick()=1
  if imag2 = 10 then player_select = player_select - 1
  imag2=10
  sleep 100
  player_select = player_select + 1
  select player_select
    case 1:
      action01.Loading = 2
    endcase
    case 2:
      action02.Loading = 2
    endcase
    case 3:
      action03.Loading = 2
    endcase
    case 4:
      action04.Loading = 2
    endcase
  endselect
endif

```

```

if mousex() >=400 and mousex() <= 428 and mousey() >= 239 and mousey() <= 257 then if
mouseclick()=1
    if imag3 = 11 then player_select = player_select - 1
    imag3=11
    sleep 100
    player_select = player_select + 1
    select player_select
        case 1:
            action01.Loading = 3
        endcase
        case 2:
            action02.Loading = 3
        endcase
        case 3:
            action03.Loading = 3
        endcase
        case 4:
            action04.Loading = 3
        endcase
    endselect
endif
if mousex() >=511 and mousex() <= 539 and mousey() >= 238 and mousey() <= 257 then if
mouseclick()=1
    if imag4 = 12 then player_select = player_select - 1
    imag4=12
    sleep 100
    player_select = player_select + 1
    select player_select
        case 1:
            action01.Loading = 4
        endcase
        case 2:
            action02.Loading = 4
        endcase
        case 3:
            action03.Loading = 4

```

```

        endcase
    case 4:
        action04.Loading = 4
    endcase
endselect
endif

if mousex() >=144 and mousex() <= 172 and mousey() >= 432 and mousey() <= 450 then if
mouseclick()=1
    if imag5 = 13 then player_select = player_select - 1
    imag5=13
    sleep 100
    player_select = player_select + 1
    select player_select
        case 1:
            action01.Loading = 5
        endcase
        case 2:
            action02.Loading = 5
        endcase
        case 3:
            action03.Loading = 5
        endcase
        case 4:
            action04.Loading = 5
        endcase
    endselect
endif

if mousex() >=271 and mousex() <= 430 and mousey() >= 430 and mousey() <= 443 then if
mouseclick()=1
    if imag6 = 14 then player_select = player_select - 1
    imag6=14
    sleep 100
    player_select = player_select + 1
    select player_select
        case 1:
            action01.Loading = 6

```

```

endcase
case 2:
    action02.Loading = 6
endcase
case 3:
    action03.Loading = 6
endcase
case 4:
    action04.Loading = 6
endcase
endselect
endif
if mousex() >=414 and mousex() <= 442 and mousey() >= 430 and mousey() <= 448 then if
mouseclick()=1
    if imag7 = 15 then player_select = player_select - 1
    imag7=15
    sleep 100
    player_select = player_select + 1
    select player_select
    case 1:
        action01.Loading = 7
    endcase
    case 2:
        action02.Loading = 7
    endcase
    case 3:
        action03.Loading = 7
    endcase
    case 4:
        action04.Loading = 7
    endcase
endselect
endif
if mousex() >=535 and mousex() <= 563 and mousey() >= 430 and mousey() <= 449 then if
mouseclick()=1
    if imag8= 16 then player_select = player_select - 1

```

```

imag8=16
sleep 100
player_select = player_select + 1
select player_select
    case 1:
        action01.Loading = 8
    endcase
    case 2:
        action02.Loading = 8
    endcase
    case 3:
        action03.Loading = 8
    endcase
    case 4:
        action04.Loading = 8
    endcase
endselect
endif
endif
rem : start
    if mousex() >=560 and mousex() <= 611 and mousey() >= 173 and mousey() <= 196 then if
mouseclick()=1
        vsMode = 2 : checkout1 = 1 : CheckEventPlayer.Loading=1

    endif
rem : end
    if mousex() >=560 and mousex() <= 611 and mousey() >= 228 and mousey() <= 252
        if mouseclick()=1
            VsMode = 1 : checkout1 = 1
            imag1 = 1
            imag2 = 2
            imag3 = 3
            imag4 = 4
            imag5 = 5
            imag6 = 6
            imag7 = 7

```

```

    imag8 = 8
    player_select = 0
    endif
endif
sync
endwhile
return
Model:
select action01.Loading
    case 1:
        load object "Model\Model1.x",action01.Loading
        set object speed action01.Loading,800
    endcase
    case 2:
        load object "Model\Model2.x",action01.Loading
        set object speed action01.Loading,800
    endcase
    case 3:
        load object "Model\Model3.x",action01.Loading
        set object speed action01.Loading,800
    endcase
    case 4:
        load object "Model\Model4.x",action01.Loading
        set object speed action01.Loading,800
    endcase
    case 5:
        load object "Model\Model5.x",action01.Loading
        set object speed action01.Loading,1800
    endcase
    case 6:
        load object "Model\Model6.x",action01.Loading
        set object speed action01.Loading,1800
    endcase
    case 7:
        load object "Model\Model7.x",action01.Loading
        set object speed action01.Loading,1800

```

```

endcase
case 8:
load object "Model\Model8.x",action01.Loading
set object speed action01.Loading,1800
endcase
endselect

position object action01.Loading,931.984,0,-524.607
rotate object action01.Loading,0,180,0
FIX OBJECT PIVOT action01.Loading
select action02.Loading
case 1:
load object "Model\Model1.x",action02.Loading
set object speed action02.Loading,800
endcase
case 2:
load object "Model\Model2.x",action02.Loading
set object speed action02.Loading,800
endcase
case 3:
load object "Model\Model3.x",action02.Loading
set object speed action02.Loading,800
endcase
case 4:
load object "Model\Model4.x",action02.Loading
set object speed action02.Loading,800
endcase
case 5:
load object "Model\Model5.x",action02.Loading
set object speed action02.Loading,1800
endcase
case 6:
load object "Model\Model6.x",action02.Loading
set object speed action02.Loading,1800
endcase
case 7:
load object "Model\Model7.x",action02.Loading

```

```

set object speed action02.Loading,1800
endcase
case 8:
load object "Model\Model8.x",action02.Loading
set object speed action02.Loading,1800
endcase
endselect
position object action02.Loading,977.363,0,-524.607
rotate object action02.Loading,0,180,0
FIX OBJECT PIVOT action02.Loading
select action03.Loading
case 1:
load object "Model\Model1.x",action03.Loading
set object speed action03.Loading,800
endcase
case 2:
load object "Model\Model2.x",action03.Loading
set object speed action03.Loading,800
endcase
case 3:
load object "Model\Model3.x",action03.Loading
set object speed action03.Loading,800
endcase
case 4:
load object "Model\Model4.x",action03.Loading
set object speed action03.Loading,800
endcase
case 5:
load object "Model\Model5.x",action03.Loading
set object speed action03.Loading,1800
endcase
case 6:
load object "Model\Model6.x",action03.Loading
set object speed action03.Loading,1800
endcase
case 7:

```

```

load object "Model\Model7.x",action03.Loading
set object speed action03.Loading,1800
endcase
case 8:
load object "Model\Model8.x",action03.Loading
set object speed action03.Loading,1800
endcase
endselect
position object action03.Loading,886.605,0,-524.607
rotate object action03.Loading,0,180,0
FIX OBJECT PIVOT action03.Loading
select action04.Loading
case 1:
load object "Model\Model1.x",action04.Loading
set object speed action04.Loading,800
endcase
case 2:
load object "Model\Model2.x",action04.Loading
set object speed action04.Loading,800
endcase
case 3:
load object "Model\Model3.x",action04.Loading
set object speed action04.Loading,800
endcase
case 4:
load object "Model\Model4.x",action04.Loading
set object speed action04.Loading,800
endcase
case 5:
load object "Model\Model5.x",action04.Loading
set object speed action04.Loading,800
endcase
case 6:
load object "Model\Model6.x",action04.Loading
set object speed action04.Loading,800
endcase

```

```

case 7:
load object "Model\Model7.x",action04.Loading
set object speed action04.Loading,800
endcase
case 8:
load object "Model\Model8.x",action04.Loading
set object speed action04.Loading,1800
endcase
endselect
position object action04.Loading,841.226,0,-524.607
rotate object action04.Loading,0,180,0
FIX OBJECT PIVOT action04.Loading
return
castile:
checkout = 0
while checkout = 0
c# = mousex()
y# = mousey()
cls
load bitmap "Title\cadice.bmp",PitM.Loading
if mouseclick() = 1 then checkout = 1
sync
endwhile
return

MakeObject:
MAKE OBJECT BOX 9, 200, 200,10
MAKE OBJECT BOX 10, 10, 200, 200
MAKE OBJECT BOX 11, 200, 200, 10
MAKE OBJECT BOX 12, 10, 200,200
MAKE OBJECT BOX 13, 200, 200, 10
MAKE OBJECT BOX 14, 10, 200, 200
MAKE OBJECT BOX 15, 200, 200, 10
MAKE OBJECT BOX 16, 10, 200, 200
MAKE OBJECT BOX 17, 200, 200, 10
MAKE OBJECT BOX 18, 10, 200,200

```

```
MAKE OBJECT BOX 19, 200, 200, 10
```

```
rem MAKE OBJECT sphere 21,50
```

```
position object 9,912,0,600
```

```
position object 10,392,0,510
```

```
position object 11,500,0,-600
```

```
position object 12,-22,0,-510
```

```
position object 13,117,0,600
```

```
position object 14,-388,0,510
```

```
position object 15,-308,0,-600
```

```
position object 16,-795,0,-510
```

```
position object 17,-693,0,600
```

```
position object 18,-1004,0,510
```

```
position object 19,-915,0,-600
```

```
hide object 9
```

```
hide object 10
```

```
hide object 11
```

```
hide object 12
```

```
hide object 13
```

```
hide object 15
```

```
hide object 14
```

```
hide object 16
```

```
hide object 17
```

```
hide object 18
```

```
rem hide object 19
```

```
rem position object 21,0,0,0
```

```
return
```

```
_Control:
```

```
select ComComman.Loading
```

```
case 1:
```

```
if K=2
```

```
select action01.Loading
```

```
case 1:
```

```
Player1.Start = 21
```

```
Player1.Stop = 39
```

```
endcase
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 2:
    Player1.Start = 21
    Player1.Stop = 39
endcase
case 3:
    Player1.Start = 21
    Player1.Stop = 39
endcase
case 4:
    Player1.Start = 21
    Player1.Stop = 39
endcase
case 5:
    Player1.Start = 3
    Player1.Stop = 80
endcase
case 6:
    Player1.Start = 3
    Player1.Stop = 80
endcase
case 7:
    Player1.Start = 3
    Player1.Stop = 80
endcase
case 8:
    Player1.Start = 3
    Player1.Stop = 80
endcase
case 9:
    Player1.Start = 3
    Player1.Stop = 80
endcase
endselect
xxx1# = object position x(action01.Loading)
yyy1# = object position y(action01.Loading)
zzz1# = object position z(action01.Loading)

```

bx1# = object position x(9)

by1# = object position y(9)

bz1# = object position z(9)

bx2# = object position x(10)

by2# = object position y(10)

bz2# = object position z(10)

bx3# = object position x(11)

by3# = object position y(11)

bz3# = object position z(11)

bx4# = object position x(12)

by4# = object position y(12)

bz4# = object position z(12)

bx5# = object position x(13)

by5# = object position y(13)

bz5# = object position z(13)

bx6# = object position x(14)

by6# = object position y(14)

bz6# = object position z(14)

bx7# = object position x(15)

by7# = object position y(15)

bz7# = object position z(15)

bx8# = object position x(16)

by8# = object position y(16)

bz8# = object position z(16)

bx9# = object position x(17)

by9# = object position y(17)

bz9# = object position z(17)

bx10# = object position x(18)

by10# = object position y(18)

bz10# = object position z(18)

bx11# = object position x(19)

by11# = object position y(19)

bz11# = object position z(19)

che# = $\sqrt{((\text{xxx1\#}-\text{bx1\#}) * (\text{xxx1\#}-\text{bx1\#})) + ((\text{zzz1\#}-\text{bz1\#}) * (\text{zzz1\#}-\text{bz1\#}))}$

che1# = $\sqrt{((\text{xxx1\#}-\text{bx2\#}) * (\text{xxx1\#}-\text{bx2\#})) + ((\text{zzz1\#}-\text{bz2\#}) * (\text{zzz1\#}-\text{bz2\#}))}$

che2# = $\sqrt{((\text{xxx1\#}-\text{bx3\#}) * (\text{xxx1\#}-\text{bx3\#})) + ((\text{zzz1\#}-\text{bz3\#}) * (\text{zzz1\#}-\text{bz3\#}))}$

```

che3# = sqrt (((xxx1#-bx4#)*(xxx1#-bx4#))+((zzz1#-bz4#)*(zzz1#-bz4#)))
che4# = sqrt (((xxx1#-bx5#)*(xxx1#-bx5#))+((zzz1#-bz5#)*(zzz1#-bz5#)))
che5# = sqrt (((xxx1#-bx6#)*(xxx1#-bx6#))+((zzz1#-bz6#)*(zzz1#-bz6#)))
che6# = sqrt (((xxx1#-bx7#)*(xxx1#-bx7#))+((zzz1#-bz7#)*(zzz1#-bz7#)))
che7# = sqrt (((xxx1#-bx8#)*(xxx1#-bx8#))+((zzz1#-bz8#)*(zzz1#-bz8#)))
che8# = sqrt (((xxx1#-bx9#)*(xxx1#-bx9#))+((zzz1#-bz9#)*(zzz1#-bz9#)))
che9# = sqrt (((xxx1#-bx10#)*(xxx1#-bx10#))+((zzz1#-bz10#)*(zzz1#-bz10#)))
che10# = sqrt (((xxx1#-bx11#)*(xxx1#-bx11#))+((zzz1#-bz11#)*(zzz1#-bz11#)))
move object action01.Loading,1
ch1# = ch1# + 1
if che# < 64 then yrotate object action01.Loading,-90
if che1# < 64 then yrotate object action01.Loading,180
if che2# < 64 then yrotate object action01.Loading,-90
if che3# < 100 then yrotate object action01.Loading,360
if che4# < 100 then yrotate object action01.Loading,-90
if che5# < 64 then yrotate object action01.Loading,180
if che6# < 64 then yrotate object action01.Loading,-90
if che7# < 64 then yrotate object action01.Loading,360
if che8# < 100 then yrotate object action01.Loading,-90
if che9# < 100 then yrotate object action01.Loading,180
Loop object
action01.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading
else
    if K=31 then move object action01.Loading,-0.1:ch1# = ch1# - 0.1: Loop object
action01.Loading,21*160,39*160
loop object action01.Loading,0*PlayerCou.Loading,4*PlayerCou.Loading
if ch1# >= 360 and ch1# <= 463
    gosub MV1
    ComComman.Loading = 0
    position object action01.Loading,63,0,83
    yrotate object action01.Loading,360
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position
z(action01.Loading)
    ch1# = 3680
endif

```

```

if ch1# >= 2937 and ch1# <= 3138
    gosub MV1
    ComComman.Loading = 0
    position object action01.Loading,-330,0,-356
    yrotate object action01.Loading,180
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position
z(action01.Loading)
    ch1# = 5361
endif
if ch1# >= 4396 and ch1# <= 4587
    gosub MV1
    ComComman.Loading = 0
    position object action01.Loading,-908,0,303
    yrotate object action01.Loading,180
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position
z(action01.Loading)
    ch1# = 7462
endif
if ch1# >= 3260 and ch1# <= 3490
    gosub MV2
    ComComman.Loading = 0
    position object action01.Loading,931,0,-318
    yrotate object action01.Loading,360
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position
z(action01.Loading)
    ch1# = 208
endif
if ch1# >= 4043 and ch1# <= 4147
    gosub MV2
    ComComman.Loading = 0
    position object action01.Loading,695,0,541
    yrotate object action01.Loading,-90
    point camera 0.1,object position x(action01.Loading),object position

```

```

y(action01.Loading),object position
z(action01.Loading)
    ch1# = 1304
endif
if ch1# >= 7829 and ch1# <= 7925
    gosub MV2
    ComComman.Loading = 0
    position object action01.Loading,55,0,-76
    yrotate object action01.Loading,360
    point camera 0.1,object position x(action01.Loading),object position
y(action01.Loading),object position
z(action01.Loading)
    ch1# = 3551
endif
endif
endcase
case 2:
if K=3
    xxx1# = object position x(action02.Loading)
    yyy1# = object position y(action02.Loading)
    zzz1# = object position z(action02.Loading)
    bx1# = object position x(9)
    by1# = object position y(9)
    bz1# = object position z(9)
    bx2# = object position x(10)
    by2# = object position y(10)
    bz2# = object position z(10)
    bx3# = object position x(11)
    by3# = object position y(11)
    bz3# = object position z(11)
    bx4# = object position x(12)
    by4# = object position y(12)
    bz4# = object position z(12)
    bx5# = object position x(13)
    by5# = object position y(13)
    bz5# = object position z(13)

```

```

bx6# = object position x(14)
by6# = object position y(14)
bz6# = object position z(14)
bx7# = object position x(15)
by7# = object position y(15)
bz7# = object position z(15)
bx8# = object position x(16)
by8# = object position y(16)
bz8# = object position z(16)
bx9# = object position x(17)
by9# = object position y(17)
bz9# = object position z(17)
bx10# = object position x(18)
by10# = object position y(18)
bz10# = object position z(18)
bx11# = object position x(19)
by11# = object position y(19)
bz11# = object position z(19)
che# = sqrt (((xxx1#-bx1#)*(xxx1#-bx1#))+((zzz1#-bz1#)*(zzz1#-bz1#)))
che1# = sqrt (((xxx1#-bx2#)*(xxx1#-bx2#))+((zzz1#-bz2#)*(zzz1#-bz2#)))
che2# = sqrt (((xxx1#-bx3#)*(xxx1#-bx3#))+((zzz1#-bz3#)*(zzz1#-bz3#)))
che3# = sqrt (((xxx1#-bx4#)*(xxx1#-bx4#))+((zzz1#-bz4#)*(zzz1#-bz4#)))
che4# = sqrt (((xxx1#-bx5#)*(xxx1#-bx5#))+((zzz1#-bz5#)*(zzz1#-bz5#)))
che5# = sqrt (((xxx1#-bx6#)*(xxx1#-bx6#))+((zzz1#-bz6#)*(zzz1#-bz6#)))
che6# = sqrt (((xxx1#-bx7#)*(xxx1#-bx7#))+((zzz1#-bz7#)*(zzz1#-bz7#)))
che7# = sqrt (((xxx1#-bx8#)*(xxx1#-bx8#))+((zzz1#-bz8#)*(zzz1#-bz8#)))
che8# = sqrt (((xxx1#-bx9#)*(xxx1#-bx9#))+((zzz1#-bz9#)*(zzz1#-bz9#)))
che9# = sqrt (((xxx1#-bx10#)*(xxx1#-bx10#))+((zzz1#-bz10#)*(zzz1#-bz10#)))
che10# = sqrt (((xxx1#-bx11#)*(xxx1#-bx11#))+((zzz1#-bz11#)*(zzz1#-bz11#)))
move object action02.Loading,2
ch2# = ch2# + 2
if che# < 74 then yrotate object action02.Loading,-90
if che1# < 74 then yrotate object action02.Loading,180
if che2# < 74 then yrotate object action02.Loading,-90
if che3# < 110 then yrotate object action02.Loading,360
if che4# < 110 then yrotate object action02.Loading,-90

```

```

if che5# < 74 then yrotate object action02.Loading,180
if che6# < 74 then yrotate object action02.Loading,-90
if che7# < 74 then yrotate object action02.Loading,360
if che8# < 110 then yrotate object action02.Loading,-90
if che9# < 110 then yrotate object action02.Loading,180

Loop object

action02.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading

else

    if K=31 then move object action02.Loading,-0.1:ch2# = ch2# - 0.1: Loop object
action02.Loading,21*160,39*160

    loop object action02.Loading,0*PlayerCou.Loading,4*PlayerCou.Loading
    if ch2# >= 358 and ch2# <= 470
        gosub MV1
        ComComman.Loading = 0
        position object action02.Loading,65,0,94
        yrotate object action02.Loading,360
        point camera 0.1,object position x(action02.Loading),object position
y(action02.Loading),object position
z(action02.Loading)
        ch2# = 3807
    endif
    if ch2# >= 3051 and ch2# <= 3264
        gosub MV1
        ComComman.Loading = 0
        position object action02.Loading,-316,0,-305
        yrotate object action02.Loading,180
        point camera 0.1,object position x(action02.Loading),object position
y(action02.Loading),object position
z(action02.Loading)
        ch2# = 5418
    endif
    if ch2# >= 4535 and ch2# <= 4643
        gosub MV1
        ComComman.Loading = 0
        position object action02.Loading,-896,0,284
        yrotate object action02.Loading,180

```

```

point camera 0.1,object position x(action02.Loading),object position
y(action02.Loading),object position
z(action02.Loading)
    ch2# = 7464
endif
if ch2# >= 3353 and ch2# <= 3459
    gosub MV2
    ComComman.Loading = 0
    position object action02.Loading,977,0,-326
    yrotate object action02.Loading,360
    point camera 0.1,object position x(action02.Loading),object position
    y(action02.Loading),object position
    z(action02.Loading)
        ch2# = 200
    endif
    if ch2# >= 4173 and ch2# <= 4241
        gosub MV2
        ComComman.Loading = 0
        position object action02.Loading,681,0,568
        yrotate object action02.Loading,-90
        point camera 0.1,object position x(action02.Loading),object position
        y(action02.Loading),object position
        z(action02.Loading)
            ch2# = 1391
        endif
        if ch2# >= 7803 and ch2# <= 8315
            gosub MV2
            ComComman.Loading = 0
            position object action02.Loading,65,0,-75
            yrotate object action02.Loading,360
            point camera 0.1,object position x(action02.Loading),object position
            y(action02.Loading),object position
            z(action02.Loading)
                ch2# = 3606
            endif
        endif
    endif
endif

```

endcase

case 3:

if K=4

xxx1# = object position x(action03.Loading)

yyy1# = object position y(action03.Loading)

zzz1# = object position z(action03.Loading)

bx1# = object position x(9)

by1# = object position y(9)

bz1# = object position z(9)

bx2# = object position x(10)

by2# = object position y(10)

bz2# = object position z(10)

bx3# = object position x(11)

by3# = object position y(11)

bz3# = object position z(11)

bx4# = object position x(12)

by4# = object position y(12)

bz4# = object position z(12)

bx5# = object position x(13)

by5# = object position y(13)

bz5# = object position z(13)

bx6# = object position x(14)

by6# = object position y(14)

bz6# = object position z(14)

bx7# = object position x(15)

by7# = object position y(15)

bz7# = object position z(15)

bx8# = object position x(16)

by8# = object position y(16)

bz8# = object position z(16)

bx9# = object position x(17)

by9# = object position y(17)

bz9# = object position z(17)

bx10# = object position x(18)

by10# = object position y(18)

bz10# = object position z(18)

```

bx11# = object position x(19)
by11# = object position y(19)
bz11# = object position z(19)
che# = sqrt (((xxx1#-bx1#)*(xxx1#-bx1#))+((zzz1#-bz1#)*(zzz1#-bz1#)))
che1# = sqrt (((xxx1#-bx2#)*(xxx1#-bx2#))+((zzz1#-bz2#)*(zzz1#-bz2#)))
che2# = sqrt (((xxx1#-bx3#)*(xxx1#-bx3#))+((zzz1#-bz3#)*(zzz1#-bz3#)))
che3# = sqrt (((xxx1#-bx4#)*(xxx1#-bx4#))+((zzz1#-bz4#)*(zzz1#-bz4#)))
che4# = sqrt (((xxx1#-bx5#)*(xxx1#-bx5#))+((zzz1#-bz5#)*(zzz1#-bz5#)))
che5# = sqrt (((xxx1#-bx6#)*(xxx1#-bx6#))+((zzz1#-bz6#)*(zzz1#-bz6#)))
che6# = sqrt (((xxx1#-bx7#)*(xxx1#-bx7#))+((zzz1#-bz7#)*(zzz1#-bz7#)))
che7# = sqrt (((xxx1#-bx8#)*(xxx1#-bx8#))+((zzz1#-bz8#)*(zzz1#-bz8#)))
che8# = sqrt (((xxx1#-bx9#)*(xxx1#-bx9#))+((zzz1#-bz9#)*(zzz1#-bz9#)))
che9# = sqrt (((xxx1#-bx10#)*(xxx1#-bx10#))+((zzz1#-bz10#)*(zzz1#-bz10#)))
che10# = sqrt (((xxx1#-bx11#)*(xxx1#-bx11#))+((zzz1#-bz11#)*(zzz1#-bz11#)))
move object action03.Loading,1
ch3# = ch3# + 1
if che# < 74 then yrotate object action03.Loading,-90
if che1# < 74 then yrotate object action03.Loading,180
if che2# < 74 then yrotate object action03.Loading,-90
if che3# < 110 then yrotate object action03.Loading,360
if che4# < 110 then yrotate object action03.Loading,-90
if che5# < 74 then yrotate object action03.Loading,180
if che6# < 74 then yrotate object action03.Loading,-90
if che7# < 74 then yrotate object action03.Loading,360
if che8# < 110 then yrotate object action03.Loading,-90
if che9# < 110 then yrotate object action03.Loading,180
Loop object
action03.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading
else
if K=31 then move object action03.Loading,-0.1:ch3# = ch3# - 0.1: Loop object
action03.Loading,21*160,39*160
loop object action03.Loading,0*PlayerCou.Loading,4*PlayerCou.Loading
if ch3# >= 349 and ch3# <= 500
gosub MV1
ComComman.Loading = 0
position object action03.Loading,83,0,86

```

```

yrotate object action03.Loading,360
point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position
z(action03.Loading)
ch3# = 3557
endif
if ch3# >= 2849 and ch3# <= 3032
gosub MV1
ComComman.Loading = 0
position object action03.Loading,-316,0,-346
yrotate object action03.Loading,180
point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position z(action03.Loading)
ch3# = 5167
endif
if ch3# >= 4274 and ch3# <= 4432
gosub MV1
ComComman.Loading = 0
position object action03.Loading,-895,0,278
yrotate object action03.Loading,180
point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position
z(action03.Loading)
ch3# = 7213
endif
if ch3# >= 3080 and ch3# <= 3236
gosub MV2
ComComman.Loading = 0
position object action02.Loading,886,0,-290
yrotate object action02.Loading,360
point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position
z(action03.Loading)
ch3# = 214
endif

```

```

if ch3# >= 3903 and ch3# <= 4026
    gosub MV2
    ComComman.Loading = 0
    position object action03.Loading,698,0,532
    yrotate object action03.Loading,-90
    point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position
z(action03.Loading)
    ch3# = 1246
endif

if ch3# >= 7526 and ch3# <= 7680
    gosub MV2
    ComComman.Loading = 0
    position object action03.Loading,83,0,-57
    yrotate object action03.Loading,360
    point camera 0.1,object position x(action03.Loading),object position
y(action03.Loading),object position
z(action03.Loading)
    ch3# = 3371
endif
endif
endcase

case 4:
    if K=5
        xxx1# = object position x(action04.Loading)
        yyy1# = object position y(action04.Loading)
        zzz1# = object position z(action04.Loading)
        bx1# = object position x(9)
        by1# = object position y(9)
        bz1# = object position z(9)
        bx2# = object position x(10)
        by2# = object position y(10)
        bz2# = object position z(10)
        bx3# = object position x(11)
        by3# = object position y(11)

```

bz3# = object position z(11)
 bx4# = object position x(12)
 by4# = object position y(12)
 bz4# = object position z(12)
 bx5# = object position x(13)
 by5# = object position y(13)
 bz5# = object position z(13)
 bx6# = object position x(14)
 by6# = object position y(14)
 bz6# = object position z(14)
 bx7# = object position x(15)
 by7# = object position y(15)
 bz7# = object position z(15)
 bx8# = object position x(16)
 by8# = object position y(16)
 bz8# = object position z(16)
 bx9# = object position x(17)
 by9# = object position y(17)
 bz9# = object position z(17)
 bx10# = object position x(18)
 by10# = object position y(18)
 bz10# = object position z(18)
 bx11# = object position x(19)
 by11# = object position y(19)
 bz11# = object position z(19)
 che# = sqrt (((xxx1#-bx1#)*(xxx1#-bx1#))+((zzz1#-bz1#)*(zzz1#-bz1#)))
 che1# = sqrt (((xxx1#-bx2#)*(xxx1#-bx2#))+((zzz1#-bz2#)*(zzz1#-bz2#)))
 che2# = sqrt (((xxx1#-bx3#)*(xxx1#-bx3#))+((zzz1#-bz3#)*(zzz1#-bz3#)))
 che3# = sqrt (((xxx1#-bx4#)*(xxx1#-bx4#))+((zzz1#-bz4#)*(zzz1#-bz4#)))
 che4# = sqrt (((xxx1#-bx5#)*(xxx1#-bx5#))+((zzz1#-bz5#)*(zzz1#-bz5#)))
 che5# = sqrt (((xxx1#-bx6#)*(xxx1#-bx6#))+((zzz1#-bz6#)*(zzz1#-bz6#)))
 che6# = sqrt (((xxx1#-bx7#)*(xxx1#-bx7#))+((zzz1#-bz7#)*(zzz1#-bz7#)))
 che7# = sqrt (((xxx1#-bx8#)*(xxx1#-bx8#))+((zzz1#-bz8#)*(zzz1#-bz8#)))
 che8# = sqrt (((xxx1#-bx9#)*(xxx1#-bx9#))+((zzz1#-bz9#)*(zzz1#-bz9#)))
 che9# = sqrt (((xxx1#-bx10#)*(xxx1#-bx10#))+((zzz1#-bz10#)*(zzz1#-bz10#)))
 che10# = sqrt (((xxx1#-bx11#)*(xxx1#-bx11#))+((zzz1#-bz11#)*(zzz1#-bz11#)))

```

move object action04.Loading,1
ch4# = ch4# + 1
if che# < 105 then yrotate object action04.Loading,-90
if che1# < 114 then yrotate object action04.Loading,180
if che2# < 114 then yrotate object action04.Loading,-90
if che3# < 100 then yrotate object action04.Loading,360
if che4# < 100 then yrotate object action04.Loading,-90
if che5# < 114 then yrotate object action04.Loading,180
if che6# < 114 then yrotate object action04.Loading,-90
if che7# < 114 then yrotate object action04.Loading,360
if che8# < 100 then yrotate object action04.Loading,-90
if che9# < 100 then yrotate object action04.Loading,180
Loop object
action04.Loading,Player1.Start*PlayerCou.Loading,Player1.Stop*PlayerCou.Loading
else
    if K=31 then move object action04.Loading,-0.1:ch4# = ch4# - 0.1: Loop object
action04.Loading,21*160,39*160
    loop object action04.Loading,0*PlayerCou.Loading,4*PlayerCou.Loading
    if ch4# >= 352 and ch4# <= 498
        gosub MV1
        ComComman.Loading = 0
        position object action04.Loading,74,0,90
        yrotate object action04.Loading,360
        point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
        z(action04.Loading)
        ch4# = 3406
    endif
    if ch4# >= 2722 and ch4# <= 2876
        gosub MV1
        ComComman.Loading = 0
        position object action04.Loading,-275,0,299
        yrotate object action04.Loading,180
        point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
        z(action04.Loading)

```

```

ch4# = 4988
endif
if ch4# >= 4127 and ch4# <= 4257
    gosub MV1
    ComComman.Loading = 0
    position object action04.Loading,-905,0,285
    yrotate object action04.Loading,180
    point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
z(action04.Loading)
    ch4# = 7021
endif
if ch4# >= 2934 and ch4# <= 3082
    gosub MV2
    ComComman.Loading = 0
    position object action04.Loading,841,0,-302
    yrotate object action04.Loading,360
    point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
z(action04.Loading)
    ch4# = 224
endif
if ch4# >= 3751 and ch4# <= 3867
    gosub MV2
    ComComman.Loading = 0
    position object action04.Loading,709,0,525
    yrotate object action04.Loading,-90
    point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
z(action04.Loading)
    ch4# = 944
endif
if ch4# >= 7342 and ch4# <= 7489
    gosub MV2
    ComComman.Loading = 0
    position object action04.Loading,74,0,-84

```

```

yrotate object action04.Loading,360

point camera 0.1,object position x(action04.Loading),object position
y(action04.Loading),object position
z(action04.Loading)

ch4# = 3231

endif

endif

endcase

endselect

return

Toi:
if mousex()>=563 and mousex()<=631 and mousey()>=7 and mousey()<=70
if mouseclick()=1
rem for i=1 to 3
RndToy = rnd(100)+ 1
if RndToy >= 0 and RndToy <= 17 then RndToy = 1
if RndToy >= 18 and RndToy <= 34 then RndToy = 2
if RndToy >= 35 and RndToy <= 52 then RndToy = 3
if RndToy >= 53 and RndToy <= 70 then RndToy = 4
if RndToy >= 71 and RndToy <= 87 then RndToy = 5
if RndToy >= 88 and RndToy <= 101 then RndToy = 6
rem next i
endif
endif
return
PositionMove:

return
MV1:
load animation "Title\Movie_0002.avi",1
play animation 1
while P <= 512
P = P + 1
sync
endwhile
delete animation 1

```

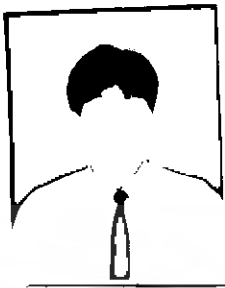
```

P = 0
return
MV2:
load animation "Title\Movie.avi",2
play animation 2
while P <= 1190
P = P + 1
sync
endwhile
delete animation 2
P=0
return

```



ประวัติผู้แต่ง



ชื่อ-สกุล	จำนงค์ พันธุ์วิไล
วัน เดือน ปีเกิด	23 มกราคม พ.ศ. 2527
ภูมิลำเนา	บ้านเลขที่ 114 หมู่ 2 ต. เวียง อ. เทิง จ. เชียงราย 57160
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดพระเกิดคงคาราม จังหวัดเชียงราย
มัธยมศึกษาตอนต้น	โรงเรียนเทิงวิทยาคม จังหวัดเชียงราย
ประกาศนียบัตรวิชาชีพ	โรงเรียนพาณิชยการกรุงธนเชียงราย จังหวัดเชียงราย
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคล วิทยาเขตเชียงราย จังหวัดเชียงราย
ปริญญาตรี	สาขาวิชาคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล. กีฬาฟุตบอล
ความสนใจพิเศษ	
คติพจน์	ทุกวันนี้เป็นวันพิเศษได้ ถ้าเราทำให้เป็นวันพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นรินทร์ สินธุ	
วัน เดือน ปีเกิด	11 กุมภาพันธ์ พ.ศ. 2528	
ภูมิลำเนา	บ้านเลขที่ 1055 หมู่ 4 ต. คลองซุด อ.เมือง จ. สตูล 91000	
ประวัติการศึกษา		
ประถมศึกษา	โรงเรียนบ้านคลองซุด	จังหวัดสตูล
มัธยมศึกษาตอนต้น	โรงเรียนพิมานพิทยาสรรค์	จังหวัดสตูล
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสตูล	จังหวัดสตูล
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคลวิทยาเขตภาคใต้	จังหวัดสงขลา
ปริญญาตรี	สาขาวิชาคอมพิวเตอร์	
	ภาควิชาวิศวกรรมคอมพิวเตอร์	
	คณะครุศาสตร์อุตสาหกรรม สจล.	
ความสนใจพิเศษ	ภาพยนตร์	
คติพจน์	ทำวันนี้ให้ดีกว่าทุกวัน	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล

ปิยวรรณ ฟองศรี

วัน เดือน ปีเกิด

7 มีนาคม พ.ศ. 2528

ภูมิลำเนา

บ้านเลขที่ 29 หมู่ 8 ต. ดันตาล อ.เส้าไห้ จ. สระบุรี 18160

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนวัดห้วยบง

จังหวัดสระบุรี

มัธยมศึกษาตอนต้น

โรงเรียนโคกกระท้อนกิตติคุณวิทยา

จังหวัดสระบุรี

ประกาศนียบัตรวิชาชีพ

วิทยาลัยอาชีวศึกษาสระบุรี

จังหวัดสระบุรี

ประกาศนียบัตรวิชาชีพชั้นสูง

วิทยาลัยเทคนิคท่าหลวงอภิมหานคร

จังหวัดสระบุรี

ปริญญาตรี

สาขาวิชาคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม สจล.

ความสนใจพิเศษ

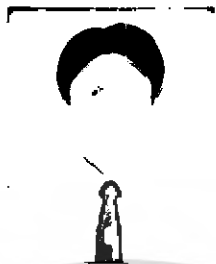
คอมพิวเตอร์

คติพจน์

หลังพายุฝน ท้องฟ้ามักจะสดใสเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล

อนุชิต อนุนพันธ์

วัน เดือน ปีเกิด

19 มีนาคม พ.ศ. 2528

ภูมิลำเนา

บ้านเลขที่ 420 หมู่ 8 ต. หนองปล้อง อ.เมือง จ. ศรีสะเกษ
33000

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนมารวิทยา

จังหวัดศรีสะเกษ

มัธยมศึกษาตอนต้น

โรงเรียนศรีสะเกษวิทยาลัย

จังหวัดศรีสะเกษ

ประกาศนียบัตรวิชาชีพ

วิทยาลัยเทคนิคศรีสะเกษ

จังหวัดศรีสะเกษ

ประกาศนียบัตรวิชาชีพชั้นสูง

วิทยาลัยเทคนิคศรีสะเกษ

จังหวัดศรีสะเกษ

ปริญญาตรี

สาขาวิชาคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม สจล.

ความสนใจพิเศษ

เขียนโปรแกรมคอมพิวเตอร์ทุกภาษา

คติพจน์

หินเท่านั้นที่อยู่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้