สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์ II

CHARACTERISTIC ANALYSIS OF SHADOW UNDER SUNLIGHT THROUGH SOME OBJECTS BY COMPUTER SIMULATION II

ชุณห์ธน รัตนพรสมปอง

CHUNTHON RATTANAPORNSOMPONG

วีรยุทธ กิพิบาน

VIRAYOUTH KIPHIBANE

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน 33902
วัน, เดือน, ปี 20 ก.ย. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHARACTERISTIC ANALYSIS OF SHADOW UNDER SUNLIGHT THROUGH SOME OBJECTS BY COMPUTER SIMULATION II

CHUNTHON RATTANAPORNSOMPONG

VIRAYOUTH KIPHIBANE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1998

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์ **II**		
	CHARACTERISTIC ANALYSIS OF SHADOW UNDER SUNLIGHT		
	THROUGH SOME OBJECTS BY COMPUTER SIMULATION **II**		
ชื่อนักศึกษา	นายชุณห์ธน รัตนพรสมปอง	**รหัสประจำตัว**	38014123
	นายวีรยุทธ กิพิบาน	**รหัสประจำตัว**	38014476
หลักสูตร	วิศวกรรมศาสตรบัณฑิต	**สาขาวิชา**	วิศวกรรมเครื่องกล

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(อ.มณฑล ใจกุศล)

(อ.สกนธ์ คล่องบุญจิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์ II
นักศึกษา	นายชุณห์ธน รัตนพรสมปอง
	นายวีรยุทธ กิพิบาน
อาจารย์ที่ปรึกษา	อ.มณฑล ใจกุศล
	อ.สกนธ์ คล่องบุญจิต
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมเครื่องกล
พ.ศ.	2541

บทคัดย่อ

พลังงานเป็นสิ่งสำคัญที่จะทำให้อุปกรณ์ต่าง ๆ ทำงานได้ ตั้งแต่อดีตจนมาถึงปัจจุบันหลายหน่วยงานได้มีความพยายามที่จะพัฒนาการใช้พลังงานให้เกิดประโยชน์ได้สูงสุดโดยใช้วิธีการที่แตกต่างกันไป การออกแบบโปรแกรมคอมพิวเตอร์ก็เป็นวิธีการหนึ่งที่ใช้ช่วยในการคำนวณหาค่าที่เหมาะสมในการติดตั้งระบบเพื่อให้นำพลังงานมาใช้ได้อย่างมีประสิทธิภาพและคุ้มค่าที่สุด สำหรับในประเทศไทยวิธีการนี้ยังไม่เป็นที่นิยมมากนัก ดังนั้นจึงได้นำความรู้ทางด้านการออกแบบโปรแกรมคอมพิวเตอร์นี้มาพัฒนาและประยุกต์ใช้งาน โดยโครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมคอมพิวเตอร์ที่ใช้ในการคำนวณหาตำแหน่งของดวงอาทิตย์ในวันและเวลาที่ต้องการ โดยสามารถรับข้อมูลลักษณะวัตถุรูปทรงเรขาคณิตต่าง ๆ เพื่อศึกษาลักษณะของเงาที่เกิดจากแสงอาทิตย์ที่ส่องมาและจำลองลักษณะของเงาที่เกิดขึ้น พร้อมทั้งแสดงผลทางจอคอมพิวเตอร์ ซึ่งวิธีการนี้สามารถนำไปประยุกต์ใช้เพื่อการพัฒนาการวางตำแหน่งของอาคาร หรือออกแบบสิ่งก่อสร้างและอาจนำไปประยุกต์ใช้ในด้านอื่น ๆ ได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Characteristic Analysis of Shadow under Sunlight through Some Object by Computer Simulation **II**
Student	Mr.Chunthon Rattanapornsompong
	Mr.Virayouth Kiphibane
Thesis Advisor	Mr.Monthon Jaikusol
	Mr.Sakon Klongboonjit
Degree	Bachelor of Engineering in Mechanical Engineering
Year	1998

ABSTRACT

Energy is an important thing to make devices operated. From past to present, many organizations try to develop the use of energy in difference methods to make maximum benefits. The design of computer's program is one of many ways which is very interesting and useful in calculating the satisfy values for systems. And it will make the maximum efficiency. In Thailand , this way is not favor to be developed and applied. This project is the development of computer's program which is used for calculating the position of the sun at any time and we can input any geometric shape in order to study the shape of the shadow of the sunlight and to make a model of its shapes which will be resulted to the computer's screen. Moreover, this method can be applied to develop the construction's designing and to other fields later.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับข้อมูลที่เกี่ยวข้องและจำเป็นต้องใช้ แหล่งข้อมูลเบื้องต้น รวมถึงแนวความคิดในการนำไปประยุกต์ใช้งานของ การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์จาก อ. มณฑล ใจกุศล ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ โดยทางผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อ. สกนธ์ คล่องบุญจิต ที่ช่วยให้คำปรึกษาและคำแนะนำในการทำงานและการเตรียมตัวสำหรับการนำเสนอปริญญานิพนธ์

ขอขอบพระคุณ อ. ธวัชชัย นาคพิพัฒน์ ที่ให้คำแนะนำในด้านการค้นหาข้อมูล

ขอขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ให้ความช่วยเหลือและคำแนะนำต่างๆ

ขอขอบคุณน้อง ๆ ชุมนุมวิชาการที่คอยเป็นกำลังใจ และให้ใช้สถานที่ทำงาน

สุดท้ายขอขอบคุณภาควิชาวิศวกรรมเครื่องกล ที่ได้ให้ทุนสนับสนุนในการทำปริญญานิพนธ์ครั้งนี้

ชุณห์ธน รัตนพรสมปอง

วีรยุทธ กิพิบาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญภาพ.....VI

บทที่ 1 บทนำ.....1

บทที่ 2 ทฤษฎีเบื้องต้น.....2

2.1 ทฤษฎีเบื้องต้นเกี่ยวกับการเกิดเงา.....2

2.1.1 ความสัมพันธ์ของตำแหน่งดวงอาทิตย์กับโลก.....2

2.1.2 การคำนวณในเรื่องเวลา.....6

2.1.3 เส้นทางการเดินของดวงอาทิตย์.....7

2.1.4 วิธีการหาเงา.....8

2.2 การสร้างและการแปลงภาพแบบสองมิติและสามมิติ.....10

2.2.1 การสร้างรูปสองมิติ.....11

2.2.2 การสร้างรูปสามมิติ.....11

2.2.3 ขั้นตอนการสร้างภาพสามมิติในระบบแสดงผลสองมิติ.....13

2.2.3.1 การคลิปป์กับวิวโวลุม.....13

2.2.3.2 การโพรเจ็กต์.....14

2.2.4 หลักพื้นฐานในการเขียนโปรแกรมเพื่อสร้างภาพสองมิติ.....18

2.2.5 หลักพื้นฐานในการเขียนโปรแกรมเพื่อสร้างภาพสามมิติ(อย่างง่าย).....18

2.2.5.1 การสร้างภาพสามมิติแบบ Perspective.....18

2.2.5.2 การสร้างภาพสามมิติแบบ Orthographic.....20

2.2.5.3 การสร้างภาพสามมิติแบบ Oblique.....20

2.2.6 หลักพื้นฐานในการเขียนโปรแกรมเพื่อการแปลงภาพสองมิติ.....22

2.2.6.1 การย้ายภาพ.....22

2.2.6.2 การย้ายหรือย่อภาพ.....23

2.2.6.3 การหมุนภาพ.....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.6.4 ระบบโคออร์ดิเนตแบบ Homogeneous และการใช้เมตริก
ในการคำนวณ.....26
2.2.7 หลักพื้นฐานในการเขียนโปรแกรมเพื่อการแปลงภาพสามมิติ.....27
2.2.7.1 การเขียนโปรแกรมเปลี่ยนระบบโคออร์ดิเนต.....29
2.2.7.2 การเขียนโปรแกรมเพื่อทำหารโพเจ็กต์.....29
2.2.8 สรุปขั้นตอนในการเขียนโปรแกรมเพื่อสร้างภาพสามิติบนจอภาพแบบ
สองมิติ.....30
บทที่ 3 แนวคิดและขั้นตอนการทำงานของโปรแกรม.....31
บทที่ 4 เครื่องมืออุปกรณ์และผลการทดลอง.....40
4.1 เครื่องมือ และ อุปกรณ์.....40
4.2 วิธีการทดลอง.....40
4.3 ผลการทดลอง.....40
4.3.1 วัตถุรูปทรงกล่อง.....41
4.3.2 วัตถุรูปทรงกระบอก.....42
4.3.3 วัตถุรูปทรงปิรามิด.....44
4.3.4วัตถุรูปทรงบ้าน.....46
บทที่ 5 สรุปผลการวิจัย.....48

ภาคผนวก คำอธิบายการทำงานของฟังก์ชั่นต่าง ๆ.....50
แผนภาพการทำงานของฟังก์ชั่นต่าง ๆ.....52
โปรแกรม Solar.C.....76

บรรณานุกรม.....206

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงความสัมพันธ์ของการเคลื่อนที่ของโลกรอบดวงอาทิตย์.....2

2.2 แสดงตำแหน่งของเส้นอาร์คติค (Arctic Circle)และเส้นแอนตาร์คติค (AntarcticCircle) 3

2.3 แสดงลักษณะของ Latitude L, Hour angle h, Declination δ.....3

2.4 แสดงเส้นทางเดินของดวงอาทิตย์ และมุม Zenith **Z**, Altitude angle α, Azimuth angle ϕ.....5

2.5 แสดงมุมที่เกิดจากพื้นผิว ได้แก่มุม Tilt angle **S**, Surface Azimuth angle Ψ, Incedent angle **I**.....5

2.6 แสดงเส้นทางเคลื่อนที่ของดวงอาทิตย์พาดผ่านท้องฟ้าในช่วง 1 ปี.....7

2.7 แสดง Sun Path Diagram ของประเทศไทย.....9

2.8 แสดงมุม Profile angle β และความสัมพันธ์กับมุม Altitude angle α Azimuth angle ϕ และ Surface azimuth angle Ψ.....10

2.9 ขั้นตอนในการสร้างภาพสองมิติ.....11

2.10 เส้นตรง AB และ perspective projection A'B'.....12

2.11 เส้นตรง AB และ parallel projection A'B'.....12

2.12 แสดงระบบมือขวาและระบบมือซ้ายของระบบโคออร์ดิเนต.....13

2.13 การโพรเจ็กต์แบบ perspective 15

2.14 ตัวอย่างของ orthographic parallel projection.....16

2.15 ตัวอย่างการ โพรเจ็กต์แบบ Isometric.....16

2.16 ตัวอย่างการ โพรเจ็กต์แบบ Oblique.....17

2.17 ตัวอย่างการ โพรเจ็กต์แบบCavalier.....17

2.18 ตัวอย่างการโพรเจ็กต์แบบ Cabinet.....18

2.19 การคำนวณหาจุดตัดของโพรเจ็กต์เตอร์และระนาบของการ โพรเจ็กต์ของรูปสามมิติแบบ Perspective.....19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

2.20 (ก)แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P(0,0,1) ไปยังจุด P' (1cosα, 1sinα,0).....20

(ข) แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P(0,0,1) ยังจุด P' (1cosα, 1sinα,0).....21

2.21 แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P ไปยังจุด P'(มองตามระนาบ YZ และ XZ).....21

2.22 การย้ายจุดจากจุด (1,2) ไปยังจุด (6,9) ได้ระยะทาง (5,7).....22

2.23 แสดงการย่อภาพ.....24

2.24 แสดงลักษณะการหมุนภาพ.....25

3.1 แผนภาพแสดงการทำงานของโปรแกรม Solar.exe.....32

3.2 แสดงลักษณะของโปรแกรม Solar.exe.....33

3.3 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุรูปทรงกล่อง.....33

3.4 แสดงผลวัตถุรูปทรงกล่องที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป.....34

3.5 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงกรวย.....34

3.6 แสดงผลวัตถุทรงกรวยที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป.....35

3.7 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงกระบอก.....35

3.8 แสดงผลวัตถุทรงกระบอกที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป.....36

3.9 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงปิรามิด.....36

3.10 แสดงผลวัตถุทรงปิรามิดที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป.....37

3.11 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุรูปทรงกล่อง.....37

3.12 แสดงผลวัตถุทรงกลมที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป.....38

3.13 แสดงการป้อนข้อมูลวัน และเวลา ที่ใช้ในการคำนวณของดวงอาทิตย์.....38

3.14 แสดงผลลักษณะเงาของวัตถุในวัน และเวลาที่ป้อนเข้าไป.....39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันมนุษย์ได้นำทรัพยากรธรรมชาติมาใช้เพื่อสนองความต้องการด้านต่าง ๆ ซึ่งมีแนวโน้มสูงขึ้นเพื่อใช้ผลิตพลังงาน แต่ด้วยข้อจำกัดที่ว่าปริมาณทรัพยากรเหล่านั้นมีอยู่จำกัด จึงต้องคิดหาวิธีเพื่อให้การใช้พลังงานเกิดประสิทธิภาพและคุ้มค่าที่สุด ในประเทศไทยก็ได้มีการศึกษาทางด้านนี้อยู่เหมือนกัน แต่ยังไม่ค่อยเป็นที่แพร่หลายเท่าใดนัก จึงทำให้การนำความรู้เรื่องนี้ไปใช้เพื่อประโยชน์ในเรื่องการประหยัดพลังงานให้กับอาคารบ้านเรือนซึ่งเป็นแหล่งใช้พลังงานที่สำคัญได้ไม่ดีเท่าที่ควร

การจะออกแบบแผงกันแดด การตัดสินใจติดกระจกกรองแสงที่บริเวณไหน การวางตำแหน่งต้นไม้ หรือการจะหันตัวอาคารไปในทิศทางใด อย่างไร ก็เป็นวิธีการหนึ่งที่ช่วยได้ โดยต้องพิจารณาจากทั้งทิศทางแดด ทิศทางลม ตำแหน่งและทางเดินของดวงอาทิตย์ โดยต้องพิจารณาว่าจะวางตำแหน่งอาคารอย่างไร จึงจะทำให้มีการรับรังสีที่แผ่จากดวงอาทิตย์น้อยที่สุดในฤดูร้อน และให้ได้รับลมเต็มที่ โดยแผงกันแดดที่ออกแบบขึ้นนี้ต้องช่วยลดแสงจ้าและความร้อนซึ่งมาจากดวงอาทิตย์โดยตรง หรือสะท้อนจากพื้นดินและอาคารข้างเคียง สำหรับการออกแบบแผงกันแดดลักษณะต่าง ๆ จะให้ผลกับรูปลักษณะของตัวอาคารด้วย เช่นการเลือกใช้ชนิดของแผงกันแดด คือ อาจเป็นบานเกล็ดเล็กหรือใหญ่ หนา บาง แท่งตัน ตรงหรือเอียง แยกจากตัวอาคารหรือเป็นส่วนหนึ่งของตัวอาคาร ซึ่งนอกจากจะเป็นส่วนเสริมให้อาคารมีความสวยงามเพิ่มขึ้นจนเป็นที่สะดุดตาของผู้ที่พบเห็นแล้ว ยังจะได้ประโยชน์จากการกันแดดกับการประหยัดพลังงานด้วย

ด้วยเหตุนี้ต้องรู้ธรรมชาติและลักษณะการโคจรของดวงอาทิตย์ จึงทำการศึกษาสมการต่างๆ ที่ใช้คำนวณเส้นทางเดิน และตำแหน่งของดวงอาทิตย์ ในวันและเวลาต่างๆ ที่ต้องการ ซึ่งจะได้เป็นค่าของมุมต่างๆ เพื่อนำไปหาลักษณะของเงาที่เกิดขึ้นในวันและเวลานั้น แล้วนำไปสร้างแบบจำลองของเงาด้วยคอมพิวเตอร์ โดยสามารถรับข้อมูลลักษณะวัตถุรูปทรงเรขาคณิตต่าง ๆ ได้

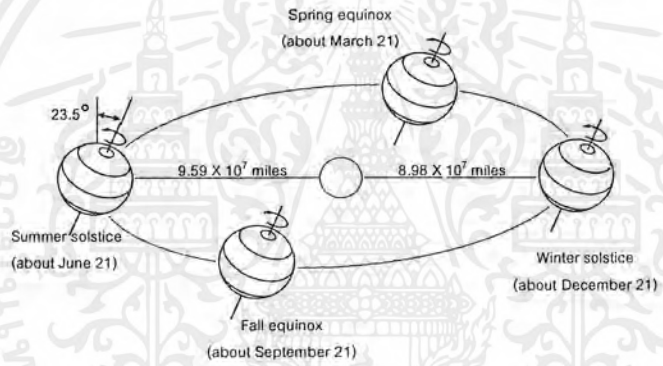
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้น

2.1 ทฤษฎีเบื้องต้นเกี่ยวกับการเกิดเงา

2.1.1 ความสัมพันธ์ของตำแหน่งดวงอาทิตย์กับโลก (Sun Positions in Relation To the Earth)

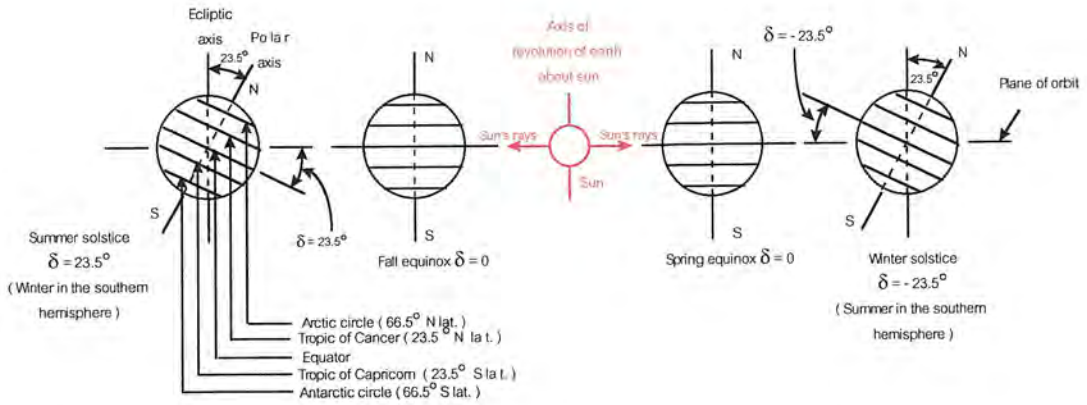
วงโคจรของโลกรอบดวงอาทิตย์เป็นรูปวงรี ซึ่งมีระยะห่างประมาณ 5×1.8^8 Km. (9.3×10^7 miles) ภาพที่ 2.1 แสดงความสัมพันธ์ของตำแหน่งโลกรอบดวงอาทิตย์ตลอดระยะเวลา 1 ปี ขณะที่โลกหมุนรอบตัวเองจะใช้เวลา 1 วัน และหมุนรอบดวงอาทิตย์จะใช้เวลา 1 ปี ส่วนดวงอาทิตย์จะหมุนรอบตัวเองจะใช้เวลาประมาณ 1 เดือน แกนหมุนของโลกจะเอียงทำมุม 23.5° กับแนวตั้งฉากของระนาบวงโคจร ในการอ้างอิงระนาบของการโคจรทำได้โดย ถ้าซีกโลกเหนือเอียงไปทางดวงอาทิตย์จะเป็นฤดูร้อน และถ้าซีกโลกใต้เอียงไปทางดวงอาทิตย์จะเป็นฤดูหนาว ด้วยเหตุนี้จึงเกิดฤดูกาลต่าง ๆ บนโลก



ภาพที่ 2.1 แสดงความสัมพันธ์ของตำแหน่งของโลกรอบดวงอาทิตย์

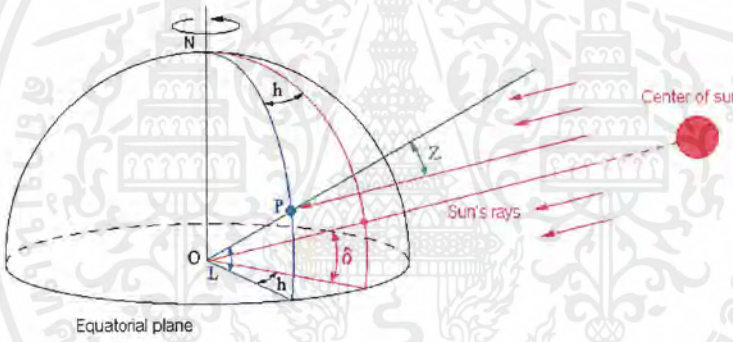
ในภาพที่ 2.2 ฤดูหนาวดวงอาทิตย์จะอยู่ห่างจากโลกมากที่สุดประมาณวันที่ 21 ของเดือนธันวาคม และเป็นช่วงที่ขั้วโลกเหนือเอียงห่างออกไปจากดวงอาทิตย์ 23.5° เหตุนี้เอง บริเวณเหนือเส้นอาร์คติค (Arctic Circle) จะมืดโดยสมบูรณ์ และบริเวณใต้เส้นแอนตาร์คติค (Antarctic Circle) จะสว่างตลอด ส่วนในฤดูร้อนดวงอาทิตย์จะอยู่ห่างจากโลกมากที่สุดเช่นกันซึ่งจะอยู่ประมาณวันที่ 21 ของเดือนมิถุนายนเป็นช่วงที่ขั้วโลกเหนือเอียงเข้าหาดวงอาทิตย์ 23.5° ด้วยเหตุนี้บริเวณเหนือเส้นอาร์คติค (Arctic Circle) จะสว่างตลอด และบริเวณใต้เส้นแอนตาร์คติค (Antarctic Circle) จะมืดโดยสมบูรณ์ ส่วนในฤดูใบไม้ผลิและฤดูใบไม้ร่วงมีกลางวันเท่ากับกลางคืนจะอยู่ประมาณวันที่ 21 ของเดือนมีนาคม และวันที่ 21 ของเดือนกันยายนตามลำดับ ขั้วโลกเหนือและขั้วโลกใต้จะมีระยะห่างจากดวงอาทิตย์เท่ากัน ทุกแห่งบนโลกจะมีกลางวันและกลางคืน 12 ชั่วโมงเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 แสดงตำแหน่งของเส้นอาร์คติค (Arctic Circle) และ เส้นแอนตาร์คติค (Antarctic Circle)

จากข้างต้นรังสีที่ส่องมาบนโลกจึงเป็นฟังชันก์ทางเรขาคณิต เราจึงต้องศึกษามุมพื้นฐานต่างๆ ที่เกิดขึ้นดังแสดงได้เป็นความสัมพันธ์ระหว่างดวงอาทิตย์กับผิวโลกดังภาพที่ 2.3 โดยกำหนดให้จุด P เป็นจุดบนผิวโลก



ภาพที่ 2.3 แสดงลักษณะของ Latitude L, Declination δ, Hour angle h

1. ละติจูด (Latitude) L คือ มุมจากระนาบเส้นศูนย์สูตรถึงจุด P ละติจูด (Latitude) ที่อยู่เหนือเส้นศูนย์สูตรจะกำหนดให้มีค่าเป็นบวก จากภาพที่ 2.3 ละติจูด (Latitude) จะเป็นมุมระหว่างเส้น OP (โดยที่ O เป็นจุดศูนย์กลางของโลก) กับโพรเจคชั่นของเส้น OP บนระนาบเส้นศูนย์สูตร

2. Declination δ คือ มุมที่รังสีของดวงอาทิตย์กระทำกับระนาบของเส้นศูนย์สูตร กำหนดให้มุม Declination ที่เอียงไปทางขั้วโลกเหนือมีค่าเป็นบวก ดังภาพที่ 2.3 สมการสำหรับการคำนวณคือ

$$\delta = 23.45\sin\left[\frac{360(284+n)}{365}\right] \qquad \text{หน่วยเป็นองศา} \qquad \textbf{(2-1)}$$

เมื่อ n = จำนวนวันใน 1 ปี ($1 < n < 365$)

ในการคำนวณเราจะใช้ค่า Declination คงที่ตลอดทั้งวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Hour Angle h คือ มุมระหว่างจุดใด ๆ บนเส้นลองติจูด (Longtitude)กับทิศทางของดวงอาทิตย์บนโลก ในภาพที่ 2.3 Hour Angle ของจุด P คือ มุมที่วัดจากเส้นโพรเจคชั่น OP บนระนาบเส้นศูนย์สูตร กับเส้นโพรเจคชั่นของเส้นผ่านศูนย์กลางระหว่างโลกกับดวงอาทิตย์บนระนาบเส้นศูนย์สูตร โดยที่ ณ. Solar Noon ของแต่ละท้องถิ่นจะมีค่า Hour Angle เป็นศูนย์ เรากำหนดให้ค่า Hour Angle มีค่าเป็นบวกในตอนบ่าย สูตรสำหรับการคำนวณคือ

$$h = \pm\left(\frac{1}{4}\right)x \text{ (จำนวนนาทีจากตอนเที่ยง)} \quad \text{หน่วยเป็นองศา} \qquad \textbf{(2-2)}$$

เมื่อ ค่า + แสดงว่าเป็นเวลาหลังเที่ยง

ค่า - แสดงว่าเป็นเวลาก่อนเที่ยง

ส่วนในการบอกตำแหน่งของดวงอาทิตย์จะต้องใช้มุมใหม่อีก 3 มุม เพิ่มจาก 3 มุมพื้นฐานข้างต้น ซึ่งมีความสำคัญอย่างมากในการคำนวณ ดังแสดงในภาพที่ 2.4 โดยโลกจะหมุนไปทางทิศตะวันออก ส่วนดวงอาทิตย์จะปรากฏโดยการสังเกตบนโลกเห็นดวงอาทิตย์เคลื่อนที่บนท้องฟ้าในลักษณะเป็นเส้นโค้ง โดยค่อย ๆ ขึ้นจากทิศตะวันออกจนกระทั่งถึงเส้น meridian (เส้นที่ลากจากขั้วโลกเหนือไปยังขั้วโลกใต้)ในตอนกลางวัน หลังจากนั้นจะคล้อยไปทางทิศตะวันตก

1. Zenith angIe Z คือ มุมระหว่างรังสีดวงอาทิตย์กระทำกับระนาบแนวดิ่ง หน่วยเป็นองศา

2. Altitude angle α คือ มุมระหว่างรังสีดวงอาทิตย์กระทำกับระนาบแนวนอนหน่วยเป็นองศา ความสัมพันธ์ระหว่าง Zenith angle กับ Altitude angle เป็นดังนี้

$$Z + \alpha = \frac{\pi}{2} = 90° \qquad \textbf{(2-3)}$$

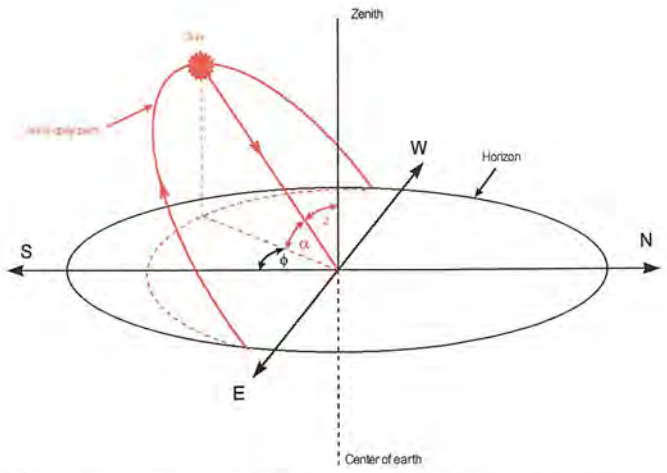
3. Azimuth angle ϕ คือ มุมของรังสีดวงอาทิตย์ที่วัดจากทิศใต้ไปทางทิศตะวันตกในระนาบแนวนอน จะมีค่าเป็นบวกสำหรับซีกโลกเหนือ ส่วนซีกโลกใต้วัดจากทิศเหนือไปทางทิศตะวันออกในระนาบแนวนอนจะมีค่าเป็นบวก หน่วยเป็นองศา

โดยมุมทั้ง 3 สามารถจัดให้อยู่ในรูปของมุมพื้นฐานโดยใช้หลักการของตรีโกณมิติ ซึ่งมีความสัมพันธ์กันดังนี้

$$\cos Z = \sin\alpha = \sin L \sin\delta + \cos L \cos\delta \cosh \qquad \textbf{(2-4)}$$

$$\sin\phi = \cos\delta \frac{\sinh}{\cos\alpha} \qquad \textbf{(2-5)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.4 แสดงเส้นทางเดินของดวงอาทิตย์ และมุม Zenith angle **Z**, Altitude angle α, Azimuth angle ϕ

สำหรับรังสีของดวงอาทิตย์บนพื้นผิวเอียง จำเป็นที่จะต้องกำหนดมุมมากกว่า 3 มุมให้สัมพันธ์กับพื้นผิวดังแสดงในภาพที่ 2.5 อันได้แก่

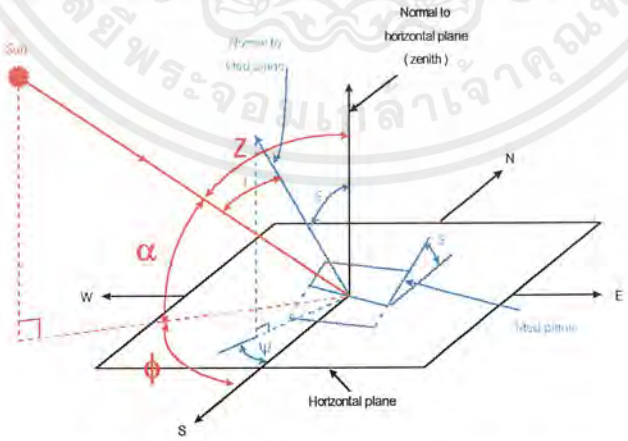
1. Tilt angle S คือ มุมระหว่างพื้นผิวกับระนาบแนวนอน

2. Surface azimuth angle ψ คือ มุมระหว่างเส้นตั้งฉากพื้นผิวกับทิศใต้วัดในแนวระดับโดยมีค่าเป็นบวกเมื่อวัดไปทางทิศตะวันตก

3. Incident angle I คือ มุมระหว่างรังสีดวงอาทิตย์กับเส้นตั้งฉากพื้นผิว ถ้าพื้นผิวกับระนาบแนวระดับวางทับกัน Incident angle **I** และ Zenith angle **Z** จะเท่ากัน

มุมที่กำหนดในภาพที่ 2.5 สามารถจัดให้อยู่ในรูปของมุมพื้นฐานที่ได้กำหนดไว้ในภาพที่ 2.3 ได้ดังนี้

$$\cos i = \sin L \sin\delta \cos s - \cos L \sin\delta \sin s \cos\psi + \cos L \cos\delta \cosh \cos s$$
$$+ \sin L \cos\delta \cosh \sin s \cos\psi + \cos\delta \sinh \sin s \sin\psi \qquad \textbf{(2-6)}$$



ภาพที่ 2.5 แสดงมุมที่เกิดจากพื้นผิว ได้แก่ Tilt angle **S**, Surface azimuth angle ψ และ Incident angle **I**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับพื้นผิวที่มีระนาบเดียวกับระนาบแนวระดับจะได้ $\mathbf{S} = 0$ และ $\mathbf{I} = \mathbf{Z}$ ฉะนั้น สมการที่ (2-6) จะลดรูปลง เหลือดังสมการที่ (2-4)

สำหรับพื้นผิวที่มีระนาบในแนวดิ่งจะได้ $\mathbf{S} = 90^\circ$ แทนลงในสมการที่ (2-6) ได้

$$\cos i = -\cos L \sin\delta \cos\psi + \sin L \cos\delta \cosh \cos\psi + \cos\delta \sinh \sin\psi \quad \textbf{(2-7)}$$

สำหรับพื้นผิวที่เอียงหันหน้าไปทางทิศใต้ในซีกโลกเหนือ จะทำให้ $\psi = 0^\circ$ สมการที่ (2-6) จะลดรูปลงเหลือ

$$\begin{aligned}\cos i &= \sin L \sin\delta \cos s - \cos L \sin\delta \sin s + \cos L \cos\delta \cosh \cos s \\ &\quad + \sin L \cos\delta \cosh \sin s \\ &= \sin(L-s)\sin\delta + \cos(L-s)\cos\delta \cosh \end{aligned} \quad \textbf{(2-8)}$$

สำหรับพื้นผิวที่เอียงหันหน้าไปทางทิศเหนือในซีกโลกใต้จะทำให้ $\psi = 180^\circ$ สมการที่ (2-6) จะลดรูปลงเหลือ

$$\cos i = \sin(L+s)\sin\delta + \cos(L+s)\cos\delta \cosh \quad \textbf{(2-9)}$$

บ่อยครั้งจำเป็นต้องหาเวลาที่ดวงอาทิตย์ขึ้นกับเวลาที่ดวงอาทิตย์ตก และความยาวของเวลาในแต่ละวัน โดยกำหนดให้ Hour Angle ตอนพระอาทิตย์ขึ้นให้เป็น h_s ($\alpha = 0$) จากสมการที่ (2-4) จะได้

$$\sin\alpha = \sin 0 = 0 = \sin L \sin\delta + \cos L \cos\delta \cosh_s$$

เพราะฉะนั้น

$$\cosh_s = -\tan L \tan\delta \quad \textbf{(2-10)}$$

h_s มีค่าเป็นบวกเมื่อพระอาทิตย์ตกและมีค่าเป็นศูนย์ ณ. Solar Noon ของแต่ละท้องถิ่น หรือ 15° ของลองติจูด (Longtitude) เท่ากับ 1 ชั่วโมง เวลาที่ดวงอาทิตย์ขึ้นและตกจากตำแหน่ง Solar Noon แต่ละท้องถิ่นสามารถหาได้จาก

$$\frac{1}{15}\cos^{-1}(-\tan L \tan\delta) \qquad \text{หน่วยเป็นชั่วโมง}$$

และความยาวในแต่ละวันหาได้จาก

$$\frac{2}{15}\cos^{-1}(-\tan L \tan\delta) \qquad \text{หน่วยเป็นชั่วโมง} \quad \textbf{(2-11)}$$

2.1.2 การคำนวณในเรื่องเวลา (Reckoning Of Time)

ความสัมพันธ์ระหว่าง Solar time กับ Standard time มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{Solar time} = \text{Standard time} + 4(\text{long}_{\text{standard}} - \text{long}_{\text{local}}) + \text{Equation of time} \qquad \textbf{(2-12)}$$

เมื่อ Longtitude correction term มีหน่วยเป็นนาที

$\text{Long}_{\text{standard}}$ เป็นค่า Standard meridian สำหรับโซนเวลา มีหน่วยเป็นองศา

$\text{Long}_{\text{local}}$ เป็นค่า Longtitude ของท้องถิ่น มีหน่วยเป็นองศาตะวันตก

Equation of time เป็นฟังก์ชันของเวลาใน 1 ปี มีหน่วยเป็นนาที และ สามารถหาได้จาก

$$\text{Equation of time} = 9.87\sin 2B - 7.53\cos B - 1.5\sin B \qquad \textbf{(2-13)}$$

เมื่อ $B = \frac{360}{364}(n-81)$

โดย n = จำนวนวันใน 1 ปี $(1 < n < 365)$

2.1.3 เส้นทางการเดินของดวงอาทิตย์ (Sun Path Diagram)

ระหว่างที่โลกกำลังหมุนรอบตัวเองและโคจรรอบดวงอาทิตย์ ตำแหน่งของดวงอาทิตย์บนท้องฟ้าจะเปลี่ยนแปลงตลอดวันตลอดทุกฤดู ดวงอาทิตย์จะขึ้นทางทิศตะวันออกโดยเอียงไปทางทิศใต้ในฤดูหนาว และจะเอียงไปทางทิศเหนือในฤดูร้อน ทางเดินของดวงอาทิตย์บนท้องฟ้าในฤดูร้อนจะสูงกว่าฤดูหนาว โดยเส้นทางการเดินของดวงอาทิตย์แสดงในภาพที่ 2.6 เมื่อตัวเลขในวงกลมแทน Solar time ใน 1 วัน ในภาพที่ 2.6 จะเห็นว่า Altitude angle α และ Azimuth angle ϕ มีค่า Solar time เท่ากับ 2 P.M.

ภาพที่ 2.6 แสดงเส้นทางเคลื่อนที่ของดวงอาทิตย์พาดผ่านบนท้องฟ้าในช่วง 1 ปี

เส้นทางเดินของดวงอาทิตย์ (Sun Path Diagram) จะมีลักษณะเป็นรูป 2 มิติในรูปวงกลม ค่า Altitude angle α จะวัดจากเส้นวงกลมที่มีขนาดเส้นผ่านศูนย์กลางต่าง ๆ กัน และ Azimuth angle ϕ จะวัดจากรอบ ๆ เส้นรอบวง ดังภาพที่ 2.7 ส่วนเวลาที่ใช้ใน Sun Path Diagram จะใช้ Solar time สำหรับ Solar Noon คือเวลาที่ดวงอาทิตย์ผ่านเส้นแบ่งเหนือใต้ของผู้สังเกต ซึ่งจะมีค่าแตกต่างจากเวลากลางวัน ส่วน Clock time จะเทียบกับ Eastern Standard Time ซึ่งใช้ค่าของ Solar Noon ณ. ตำแหน่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Longtitude = 150° E ที่เวลากลางวันของทุก ๆ วัน สำหรับผู้สังเกตที่อยู่นอกเส้น Meridian จะพบว่าค่าของ Solar time จะต่างจากค่าของ Clock time โดยส่วนที่แตกต่างเกิดจากค่าของ Solar Noon ที่ตำแหน่งนั้นเกิดคนละเวลากับ Solar Noon ที่ตำแหน่งบนเส้น Meridian

2.1.4 วิธีการหาเงา

ในการออกแบบเพื่อการประหยัดพลังงานหลักการคือ ต้องดูลักษณะการเกิดเงาของอุปกรณ์ที่กันแดด โดยในการหาเงาซึ่งเป็นฟังก์ชันของเวลาและวันใน 1 ปี

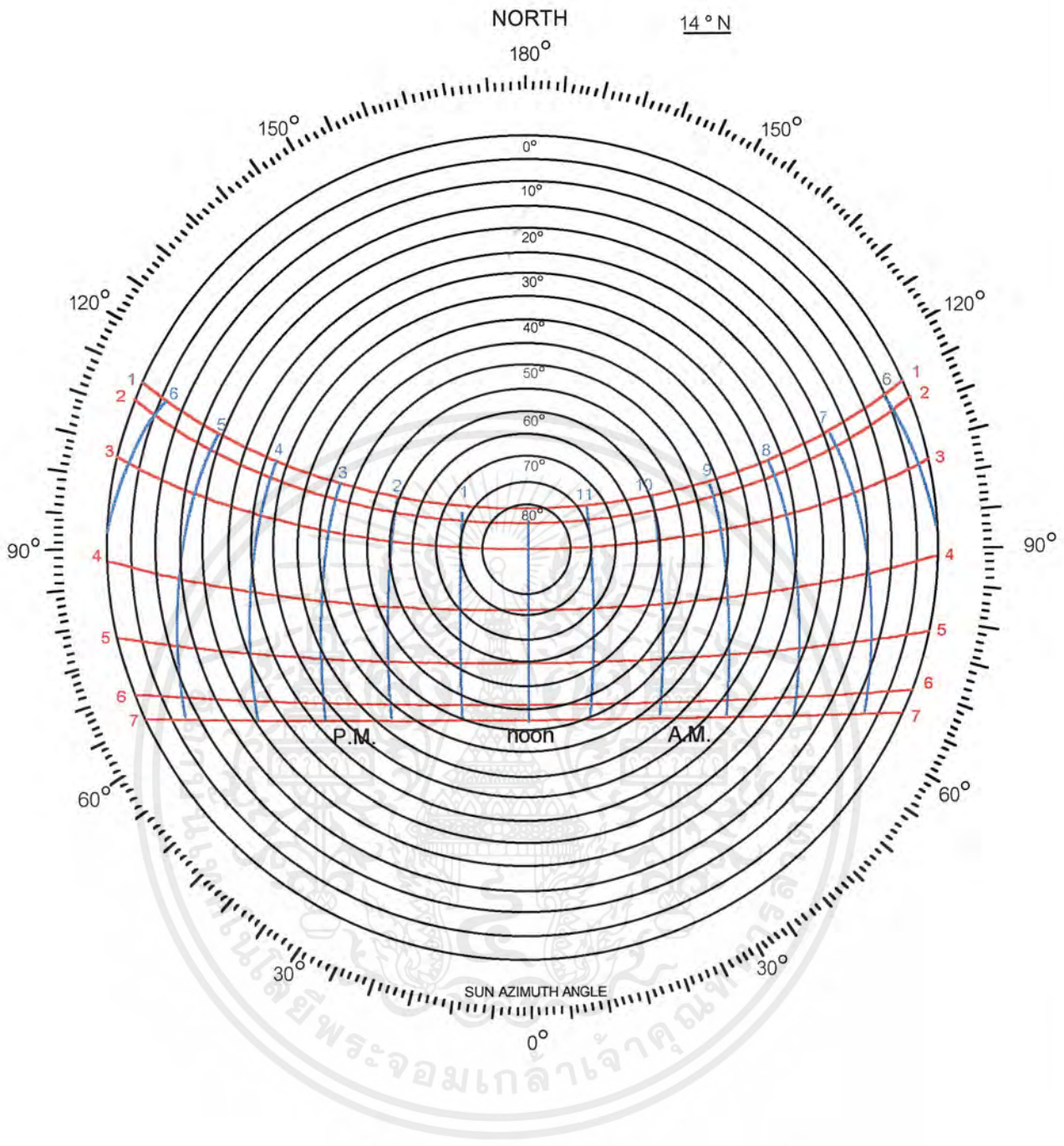
มุม Profile angle β คือ มุมทางตั้งในแนวระดับเดียวกับตำแหน่งของดวงอาทิตย์โดยจะกระทำในแนวแกนเดียวกับแนวแกนของอุปกรณ์บังแดด (ในเวลาเที่ยงวัน Profile angle จะเท่ากับ Altitude angle) จะอยู่ในเทอมของ Altitude angle α, Azimuth angle ϕ และ Surface azimuth angle ψ ดังสามารถแสดงในภาพที่ 2.8 ซึ่งจะได้ว่า

$$\tan\beta = \frac{\tan\alpha}{\cos(\phi - \psi)} \qquad \textbf{(2-14)}$$

เมื่ออุปกรณ์บังแดดหันหน้าไปทางทิศใต้ ($\psi = 0$) จะได้ว่า

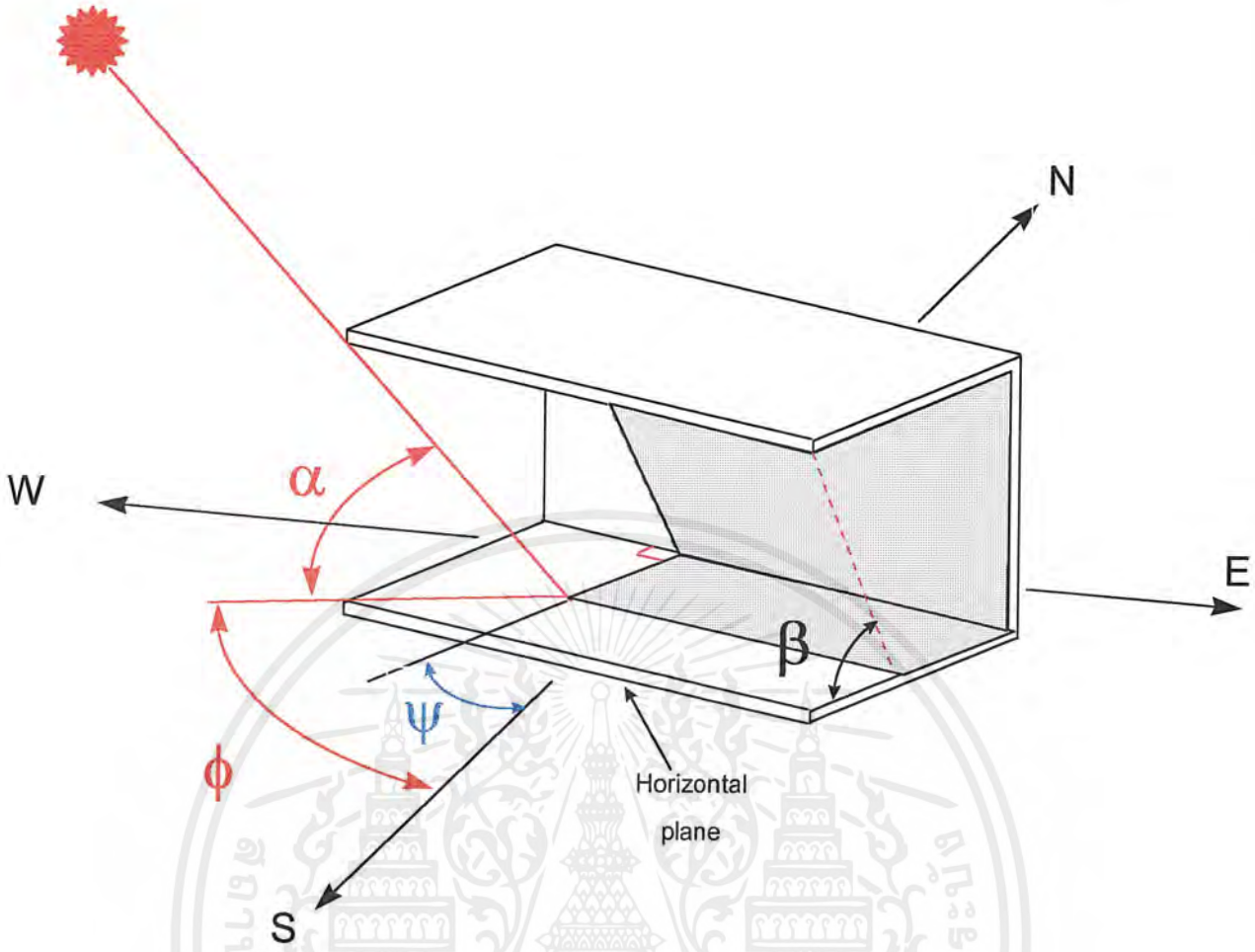
$$\tan\beta = \frac{\tan\alpha}{\cos\phi} \qquad \textbf{(2-15)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.7 แสดง Sun Path Diagram ของประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 แสดงมุม Profile angle β และความสัมพันธ์กับมุม Altitude angle α, Azimuth angle ϕ และ Surface azimuth angle ψ

2.2 การสร้างและการแปลงภาพแบบสองมิติ และสามมิติ

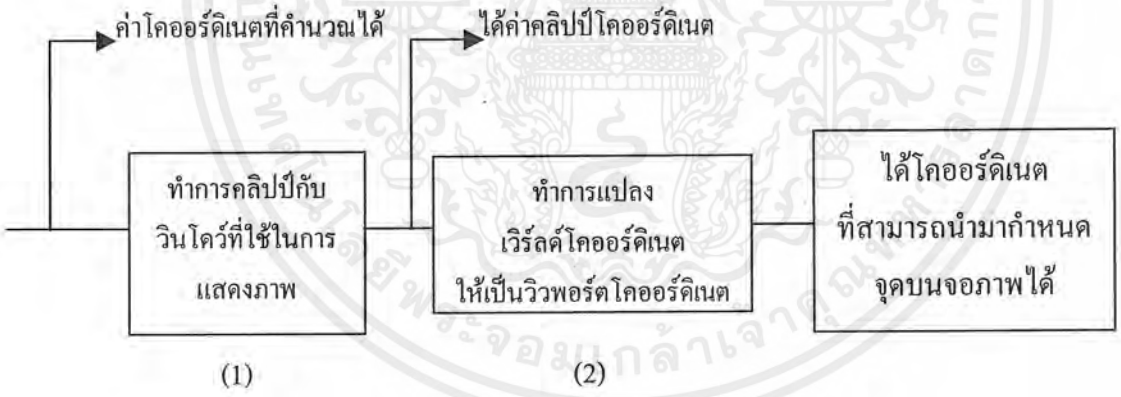
ในระบบคอมพิวเตอร์นั้น อาจจะกล่าวได้ว่าการสร้างรูปต่าง ๆ ก็คือการเรียกคำสั่ง (หรือโพรซีเดอร์) ที่ใช้สำหรับสร้างรูปนั้น ๆ ขึ้นมา หรือในกรณีของการสร้างรูปที่ซับซ้อนที่ไม่มีคำสั่งในการสร้างอยู่แล้ว เช่น รูปสามมิติ เราจะต้องใช้อัลกอริทึมสำหรับสร้างรูปนั้น ๆ โดยยังคงใช้คำสั่งสร้างรูปทรงพื้นฐาน เช่น จุด เส้นตรง เป็นพื้นฐานของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การสร้างรูปสองมิติ

ในการสร้างรูปสองมิตินั้น ขั้นตอนในการสร้างจะง่ายกว่าการสร้างภาพสามมิติเป็นอย่างมาก การคำนวณต่างๆ สามารถทำความเข้าใจได้อย่างตรงไปตรงมา เนื่องจากระบบการแสดงภาพเป็นระบบสองมิติอยู่แล้ว โดยโคออร์ดิเนตของตำแหน่งบนจอภาพประกอบขึ้นมาจาก ตำแหน่งตามแนวระดับ (x-ordinate) และตำแหน่งตามแนวตั้ง (y-ordinate)

ดังนั้นในการสร้างภาพต่างๆ ขึ้นมา เราสามารถกล่าวคร่าวๆ ได้ว่า เราสามารถกำหนดจุดต่างในภาพว่าอยู่ ณ ตำแหน่งใดบนจอภาพได้โดยตรง (หลังจากผ่านการแปลงจากโคออร์ดิเนตในวินโดว์ลงมายังวิวพอร์ตแล้ว) โดยไม่ต้องมีการนำโคออร์ดิเนตมาแปลงลดมิติเหมือนกับการแสดงภาพในระบบสามมิติ ตัวอย่างเช่น ในกรณีของเส้นตรง : กำหนดโคออร์ดิเนต X,Y ของจุดปลายทั้งสองด้านแล้วใช้คำสั่งเชื่อมระหว่างสองจุดนั้นๆ ในกรณีของรูปทรงอื่น : กำหนดโคออร์ดิเนตที่จำเป็นในการสร้างภาพแล้วใช้คำสั่งหรือ อัลกอริทึมในการสร้างรูปทรงนั้นๆ ซึ่งโดยทั่วไปแล้วขั้นตอนในการสร้างภาพสองมิตินั้นเริ่มด้วยการคำนวณหาโคออร์ดิเนตของส่วนประกอบของภาพในเวิร์ลด์โคออร์ดิเนต (หรือยูสเซอร์โคออร์ดิเนต) แล้วขั้นตอนต่อไปคือการคลิปปิ้งกับวินโดว์ที่ใช้ในการแสดงภาพ และขั้นตอนสุดท้ายคือ การแปลงโคออร์ดิเนตจากเวิร์ลด์โคออร์ดิเนตให้เป็นโคออร์ดิเนตที่อุปกรณ์แสดงภาพใช้ ดังแผนภูมิต่อไปนี้



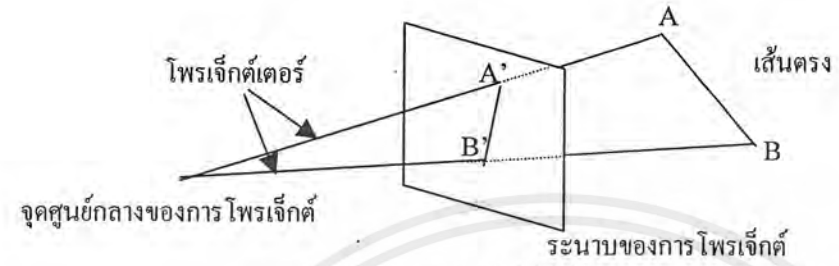
ภาพที่ 2.9 ขั้นตอนในการสร้างภาพสองมิติ

2.2.2 การสร้างรูปสามมิติ

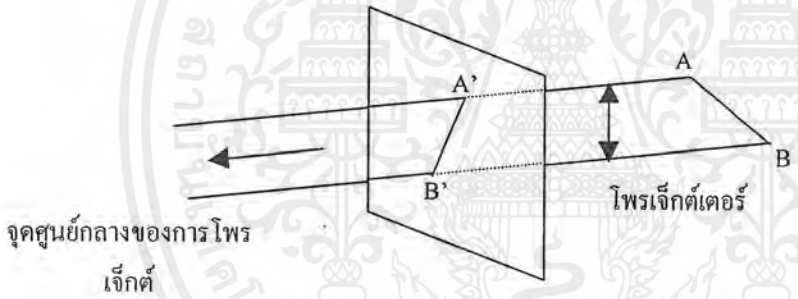
สำหรับการสร้างภาพสามมิตินั่นมีขั้นตอนในการสร้างที่ค่อนข้างซับซ้อน โดยเริ่มต้นเราต้องมีการกำหนดข้อมูลที่สำคัญต่างๆ ที่จำเป็นในการสร้างภาพสามมิติบนระบบแสดงภาพแบบสองมิติ ซึ่งได้แก่จุดศูนย์กลางของการโพรเจ็กต์ (center of projection) ระนาบของการโพรเจ็กต์ (projection) และลำแสง (projector) ดังภาพที่ 2.10 และ 2.11 ซึ่งสิ่งต่าง ๆ เหล่านี้เป็นสิ่งจำเป็นในการกำหนด (คำนวณหา) จุดที่ประกอบขึ้นมาเป็นภาพสามมิติบนระบบแสดงภาพแบบสองมิติ ดังตัวอย่างในภาพที่ 2.10 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 ซึ่งจุด A' และ B' คือจุดบนระนาบสองมิติของจุด A และ B ในระบบสามมิติ แต่ยังเป็นโชคดีอยู่อย่างหนึ่งว่าในการคำนวณนั้น เราไม่จำเป็นต้องคำนวณทุกๆ จุดที่ประกอบขึ้นเป็นภาพ เราเพียงแต่คำนวณเฉพาะจุดมุมก็เพียงพอ เนื่องจากความจริงที่ว่าโพรเจ็กต์ชันของเส้นตรงก็ยังเป็นเส้นตรงอยู่นั่นเอง แต่ถ้าเป็นเส้นโค้งหรือวงกลม เราก็จะต้องคำนวณทุกๆ จุดที่ประกอบขึ้นมาเป็นภาพนั้นๆ



ภาพที่ 2.10 เส้นตรง AB และ parallel projection A'B'



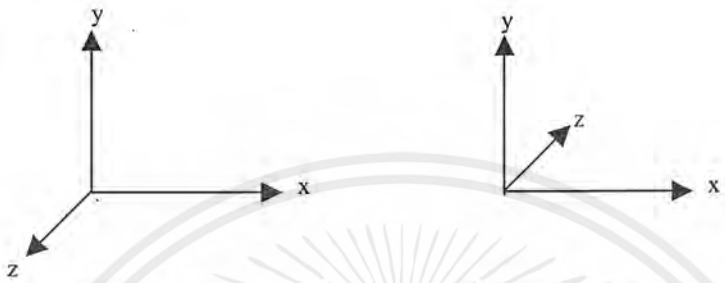
ภาพที่ 2.11 เส้นตรง AB และ parallel projection A'B'
(projector AA' และ BB' ขนานกัน)

จากภาพที่ 2.10 และ 2.11 นั้น ต้องทำความเข้าใจให้ได้ก่อนว่า รูปทั้งสองนั้นคือ วัตถุที่ลอยอยู่ในแผ่นกระดาษ โดยที่แผ่นกระดาษของเราในที่นี้ก็คือ projection plane ที่บรรจุเส้นตรง A'B' อยู่ หรือกล่าวอีกลักษณะหนึ่งก็คือว่ารูปทั้งสองเป็นรูปสามมิตินั่นเอง

ก่อนที่จะเริ่มการอธิบายต่อไป เราจะกำหนดสมมติฐานว่าระบบโคออร์ดิเนตสามมิติที่เราจะใช้ในการศึกษาเรื่องสามมิตินี้ จะใช้ระบบมือขวา (right-handed system) สำหรับกำหนดทิศทางซึ่งมีทิศทางของแกน X,Y และ Z ดังภาพที่ 2.12 (ก) สำหรับการแปลงสามมิติ เนื่องจากทำให้ผู้มองสามารถนึกถึงภาพสามมิติได้ง่าย และระบบโคออร์ดิเนตที่ใช้ในการแสดงภาพบนจอภาพ(ใช้ในการโพรเจ็กต์) เป็นระบบมือซ้าย (left-handed system) เนื่องจากระบบมือซ้ายจะมีความเป็นธรรมชาติมากกว่าในการแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพบนจอภาพ เพราะจอภาพมีลักษณะที่มีความลึก ดังภาพที่ 2.12 (ข) โดยที่ในการสร้างภาพสามมิติขึ้นมานั้น เราจะสร้างภาพขึ้นมาโดยใช้ระบบโคออร์ดิเนตมือขวาทำการคำนวณแปลงภาพต่างๆ แล้วเมื่อเราจะแสดงภาพลงบนจอภาพ เราก็จะทำการแปลงให้ภาพอยู่ในระบบโคออร์ดิเนตแบบมือซ้าย แล้วจึงทำการโพรเจ็กต์ไปยังระนาบของการโพรเจ็กต์ ซึ่งทำให้เราได้ศึกษาขั้นตอนในการแปลงระบบโคออร์ดิเนตไปในตัวด้วย



(ก) ระบบมือขวา (ข)ระบบมือซ้าย

ภาพที่ 2.12 แสดงระบบมือขวาและระบบมือซ้ายของระบบโคออร์ดิเนต

2.2.3 ขั้นตอนการสร้างภาพสามมิติในระบบแสดงผลแบบสองมิติ

เราจะมาศึกษากันถึงรายละเอียดในขั้นตอนที่ 1 และ 2 คือการคลิปป์กับวิวโวลุมและการโพรเจ็กต์ลงไปยังระนาบของการโพรเจ็กต์ เนื่องจากขั้นตอนที่เหลือนั้นเป็นขั้นตอนเดียวกับการแสดงภาพแบบสองมิติ

2.2.3.1 การคลิปป์กับวิวโวลุม

การคลิปป์กับวิวโวลุมนั้นก็มีลักษณะเดียวกับการคลิปป์กับวินโดว์ในระบบสองมิติ โดยมีข้อแตกต่างกันตรงที่ว่าการคลิปป์ในระบบสองมิตินั้นเราจะกำหนดพื้นที่ของการแสดงภาพ แล้วทำการคำนวณหาว่าภาพที่เราสร้างขึ้นมานั้นอยู่นอกพื้นที่ของการแสดงภาพหรือไม่ ถ้ามีส่วนใดส่วนหนึ่งนอกพื้นที่ เราก็ต้องทำการตัดส่วนของภาพนั้นออกไป แต่ในระบบสามมิตินั้นจะต้องกำหนดเป็นปริมาตรของการแสดงภาพ และเราจะต้องทำการคำนวณว่ามีส่วนใดของภาพของเราที่อยู่นอกปริมาตรที่กำหนดขึ้นหรือไม่

โดยการคำนวณหาจุดตัดของส่วนต่างๆ ของภาพของเรากับปริมาตรที่กำหนด ซึ่งการทำการคลิปป์กับวิวโวลุมนี้จะต้องใช้การคำนวณอย่างมาก เนื่องจากเราจะต้องหาว่าส่วนใด ๆ ของรูปทรงของเราอยู่นอกปริมาตรที่ใช้เป็นวิวโวลุม เราไม่สามารถนำเฉพาะส่วนใดส่วนหนึ่งของรูปทรงมาทำการคำนวณได้ ซึ่งไม่เหมือนกับที่เรามองภาพก็สามารถพิจารณาได้ว่าเราควรจะคำนวณการ คลิปป์กับส่วนใดบ้าง เพราะในระบบคอมพิวเตอร์ ข้อมูลมีรูปร่างหน้าตาเหมือนกัน ไม่มีข้อแตกต่างที่สามารถบอกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทันที ข้อมูลทั้งหมดจะต้องผ่านการเปรียบเทียบ (วิเคราะห์) ก่อนจึงจะสามารถนำข้อมูลที่ได้มาช่วยในการพิจารณาได้

2.2.3.2 การโพรเจ็กต์

โดยหลักการแล้ว การโพรเจ็กต์จะแปลงจุดที่อยู่ในระบบโคออร์เนตหนึ่งที่มีมิติ N ให้เป็นจุดในอีกระบบโคออร์ดิเนตที่มีมิติที่น้อยกว่า N สำหรับในกรณีเราจะศึกษากันนี้ การโพรเจ็กต์จะกระทำกันจากระบบโคออร์ดิเนตในระบบสองมิติ

การโพรเจ็กต์ของออบเจ็กต์ในระบบสามมิติ (ซึ่งในกรณีที่ใช้ศึกษานี้คือกลุ่มของจุดนั่นเอง) จะถูกกำหนดโดยลำแสง (projector) ที่เป็นเส้นตรงที่ออกมาจากจุดศูนย์กลางของการโพรเจ็กต์ (center of projection) ผ่านจุดต่างๆ บนออบเจ็กต์ที่เราต้องการทำโพรเจ็กต์ไปตัดกับระนาบของการโพรเจ็กต์เพื่อสร้างโพรเจ็กต์ชันขึ้นมา ดังในภาพที่ 2.10 และ 2.11 ซึ่งจากภาพทั้งสองนี้เป็นการแสดงการโพรเจ็กต์ในสองลักษณะของเส้นตรงเส้นเดียวกัน

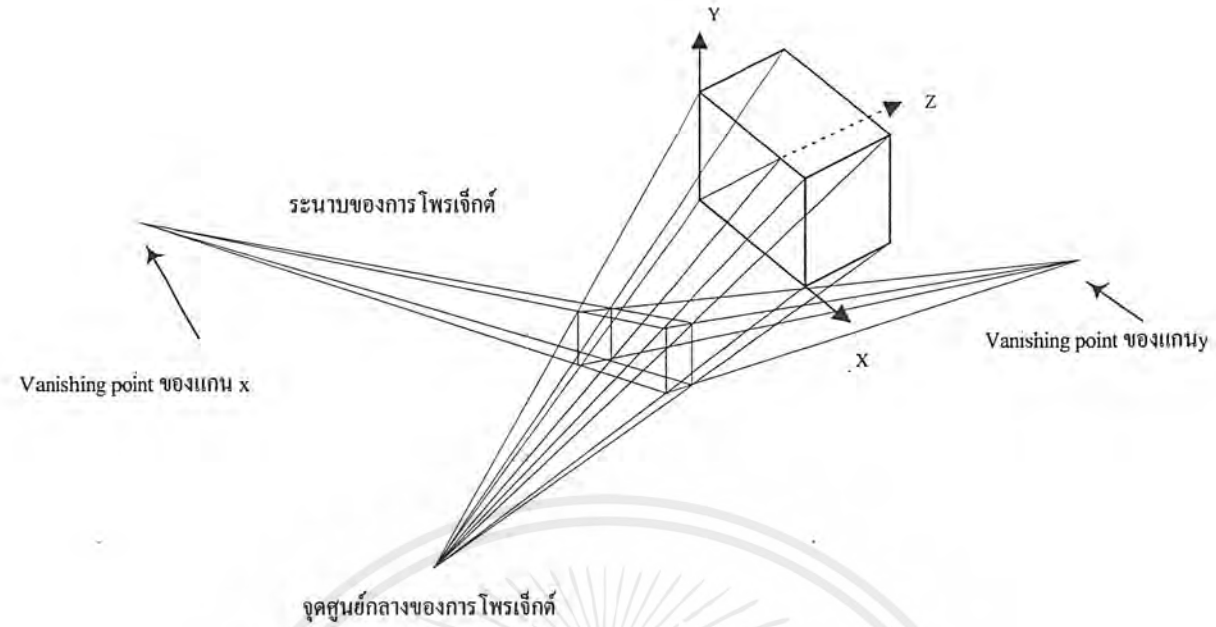
การทำโพรเจ็กต์ไปยังระนาบหนึ่งหรือที่เรียกว่า planer geometric projection นั้นยังแบ่งย่อยออกเป็น 2 ประเภทใหญ่ๆ คือ perspective projection และ parallel projection ซึ่งข้อแตกต่างของการโพรเจ็กต์ทั้งสองประเภทนี้อยู่ที่ความสัมพันธ์ระหว่างจุดศูนย์กลางของการโพรเจ็กต์กับระนาบของการโพรเจ็กต์ คือถ้าระยะจากจุดศูนย์กลางไปยังระนาบสามารถวัดได้ จะเป็นการโพรเจ็กต์ประเภท perspective แต่ถ้าระยะทางนั้นไม่สามารถกำหนดได้จะเป็นการโพรเจ็กต์ประเภท Parallel (ดูภาพที่ 2.10 และ 2.11 ประกอบ)

โดยที่ถ้าเราจะทำการโพรเจ็กต์แบบ Perspective เราจะต้องระบุจุดศูนย์กลางของการโพรเจ็กต์ลงไป ในขณะที่ถ้าเราจะทำการโพรเจ็กต์แบบ Pararllel เราจะต้องกำหนดทิศทางของการโพรเจ็กต์

เราจะมาศึกษาถึงข้อแตกต่างระหว่างการโพรเจ็กต์ประเภทต่างๆ ดังนี้

1. การโพรเจ็กต์แบบ Perspective โพรเจ็กต์ชันของกลุ่มของเส้นขนานที่จะถูกโพรเจ็กต์ที่ไม่ได้ขนานกับระนาบของการโพรเจ็กต์จะไปรวมกัน ณ จุดๆหนึ่งที่ถูกเรียกกันว่า vanishing point ดังภาพที่ 2.13 ซึ่งถ้ากลุ่มของเส้นขนานนี้ขนานกับแกน x,y และ z แกนใดแกนหนึ่ง เราจะเรียกว่า principal vanishing point ซึ่งจะมีได้มากที่สุดเพียง 3 จุดเท่านั้น ยกตัวอย่างเช่นถ้าระนาบของการโพรเจ็กต์ตัดตั้งฉากกับแกน Z แกน Z เท่านั้นที่จะมี principal vanishing point เพราะกลุ่มของเส้นตรงที่ขนานกับแกน x หรือแกน y จะขนานกับระนาบของการโพรเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.13 การโพรเจ็กต์แบบ perspective

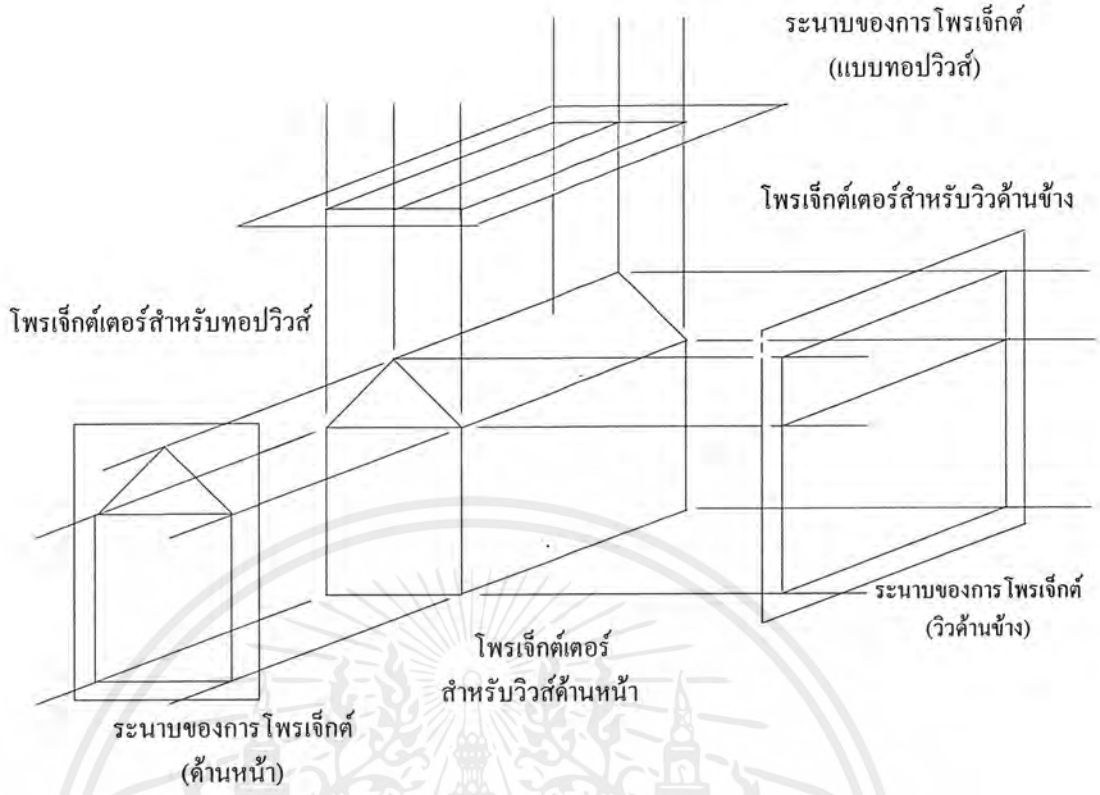
2. การโพรเจ็กต์แบบ Parallel แบ่งย่อยตามความสัมพันธ์ระหว่างทิศทางของการโพรเจ็กต์และนอร์มัลเวกเตอร์ (normal vector) ซึ่งหมายถึงเวกเตอร์ที่ตั้งฉากกับระนาบของการโพรเจ็กต์ดังนี้

2.1 orthographic parallel projection จะมีทิศทางของการโพรเจ็กต์และทิศทางของนอร์มัลเวกเตอร์ของการโพรเจ็กต์เป็นทิศทางเดียวกัน (ทิศทางของการโพรเจ็กต์ตั้งฉากกับระนาบของการโพรเจ็กต์) ชนิดของ orthographic projections ที่ใช้กันมากก็คือการโพรเจ็กต์ของ top front และ side projection ดังภาพที่ 2.14

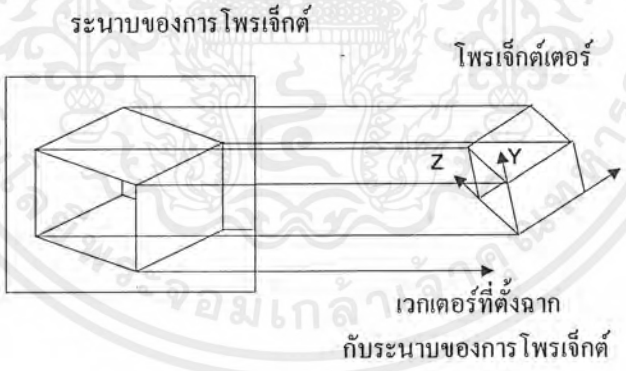
นอกจากนี้ก็ยังมี axonomitric orthographic projection ซึ่งมีคุณสมบัติคือการโพรเจ็กต์แบบนี้ทิศทางของระนาบของการโพรเจ็กต์จะไม่ตั้งฉากกับแกนหลักใดๆ(แกน X,Y และ Z) ซึ่งทำให้เราสามารถเห็นด้านต่างๆ ของออบเจ็กต์ได้มากกว่าหนึ่งด้านในเวลาเดียวกันคล้ายๆ กับโพรเจ็กต์ชันแบบ perspective นั่นเอง

โดยที่โพรเจ็กต์ชันแบบ Isometric ก็เป็นโพรเจ็กต์ชันแบบ Axonometric ประเภทที่ถูกเลือกใช้กันมาก ซึ่งมีคุณสมบัติเป็นนอร์มัลเวกเตอร์ของระนาบของการโพรเจ็กต์ที่ทำมุมกับแกน X,Y และ Z เท่าๆ กัน ดังตัวอย่างจากภาพที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.14 ตัวอย่างของ orthographic parallel projection

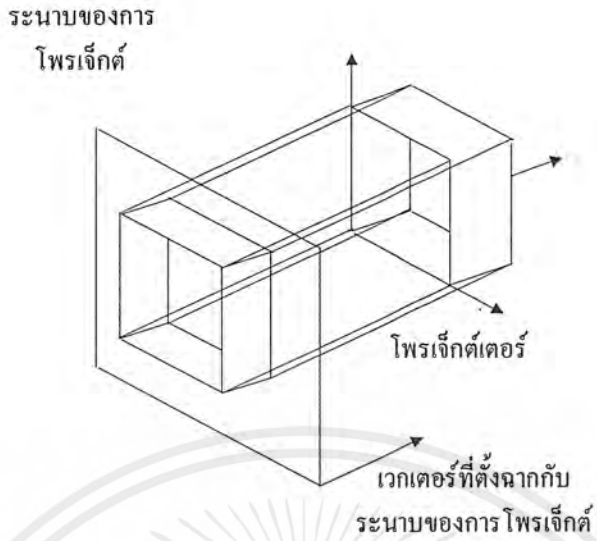


ภาพที่ 2.15 ตัวอย่างการโพรเจ็กต์แบบ Isometric

2.2 oblique prarllel เป็นชนิดที่ 2 ของ parallel projection ซึ่งมีคุณลักษณะที่สำคัญคือ ระนาบของการโพรเจ็กต์จะตั้งฉากกับแกนหลัก (X,Y และ Z) แกนใดแกนหนึ่ง ซึ่งจะทำให้เราสามารถวัดมุมและความยาวของการโพรเจ็กต์ของด้านของออบเจ็กต์ที่ขนานกับระนาบของการโพรเจ็กต์ ส่วนโพรเจ็กชันของด้านอื่นๆ นั้น เราสามารถวัดระยะตามแกน X,Y และ Z ได้เท่านั้น ดังภาพที่ 2.16

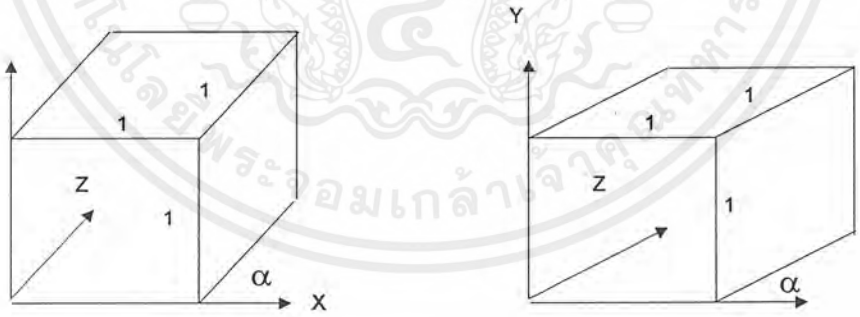
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ภาพที่ 2.16 ตัวอย่างการโพรเจ็กต์แบบ Oblique

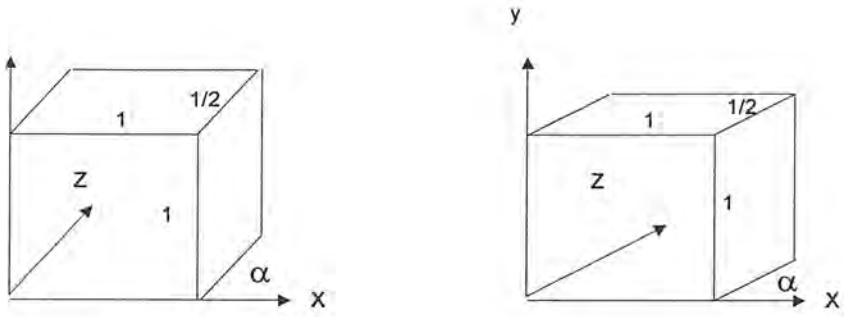
ชนิดของโพรเจ็กต์ชันแบบ Oblique ที่เป็นที่นิยมใช้กันมากคือ Cavalier และ Cabinet ซึ่งโพรเจ็กต์แบบ Cavalier นั้นจะมีทิศทางของการโพรเจ็กต์ ทำมุม 45 องศากับระนาบของการโพรเจ็กต์ ซึ่งจะทำให้โพรเจ็กต์ชันของเส้นตรงที่ตั้งฉากกับระนาบของการโพรเจ็กต์มีความยาวเท่ากับเส้นตรงนั้น



ภาพที่ 2.17 ตัวอย่างการโพรเจ็กต์แบบCavalier

ส่วนการโพรเจ็กต์แบบ Cabinet จะมีทิศทางของการโพรเจ็กต์ทำมุม arccot(1/2) องศากับระนาบของการโพรเจ็กต์ ซึ่งทำให้โพรเจ็กต์ชันของเส้นตรงที่ตั้งฉากกับระนาบของการโพรเจ็กต์มีความยาวครึ่งหนึ่งของเส้นตรงนั้น ซึ่งทำให้รูปที่ได้ดูเป็นธรรมชาติมากกว่าการโพรเจ็กต์ชันแบบ Cavalier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
33902



ภาพที่ 2.18 ตัวอย่างการโพรเจ็กต์แบบ Cabinet

2.2.4หลักพื้นฐานในการเขียนโปรแกรมเพื่อสร้างภาพสองมิติ

ดังที่กล่าวมาแล้วในตอนต้นของบทนี้ว่าหลักพื้นฐานในการเขียนโปรแกรมสำหรับสร้างภาพสองมิตินั้นไม่มีอะไรซับซ้อนเหมือนการสร้างภาพมิติ สิ่งที่สำคัญคือเราจะต้องกำหนดอัลกอริทึมที่เหมาะสมให้ได้ (ในกรณีที่เราต้องการเขียนโพรซีเดอร์สร้างภาพต่างๆ ขึ้นมาเองมิฉะนั้นเราก็สามารถใช้โพรซีเดอร์ที่มีมาแล้วได้เลย) ขั้นต่อไปคือทำการคลิปป์และแปลงโคออร์ดิเนตจากเวิร์ลด์โคออร์ดเนตไปยังวิวพอร์ตโคออร์ดิเนตดังที่ได้กล่าวมาแล้ว

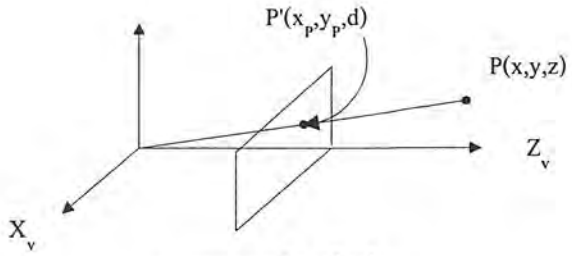
2.2.5หลักพื้นฐานในการเขียนโปรแกรมเพื่อสร้างภาพสามมิติ (อย่างง่าย)

คราวนี้เราจะมากล่าวกันถึงหลักในการเขียนโปรแกรมเพื่อสร้างภาพสามมิติ (บนระบบแสดงผลแบบสองมิติ) ซึ่งในที่นี้เราจะมาดูในรายละเอียดของหลักการเขียนโปรแกรมสำหรับสร้างภาพสามมิติ อย่างง่าย คือเราจะไม่กล่าวถึงการคลิปป์ด้วยวิวโวลุม และเราจะใช้ระนาบของการโพรเจ็กต์อยู่บน XY-plane (คือทุกจุดมีค่าZ เท่ากัน) เท่านั้น โดยสมมติฐานของเรานั้นก็คือ วัตถุที่เราจะใช้เป็นแบบในการสร้างภาพสามมิตินั้นมีขนาดเล็กกว่าวิวโวลุม(อยู่ภายใน)ที่เรากำหนดเสมอ

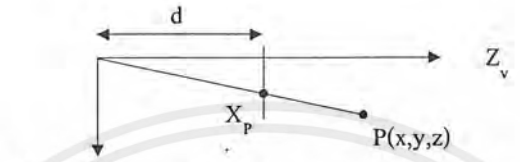
2.2.5.1การสร้างภาพสามมิติแบบ Perspective

ในการสร้างภาพสามมิติแบบ Perspective นั้น ขั้นตอนแรกก็คือการกำหนดโคออร์ดิเนตของจุด center of projevtion และการกำหนดระนาบของการโพรเจ็กต์ (ในที่นี้เราจะใช้ระนาบ xy ที่มีค่า Z=d (one point perspective projection) และใช้จุด center of projection อยู่ที่จุด (0,0,0) (เพื่อความว่างต่อการทำความเข้าใจ ซึ่งโดยทั่วไปแล้วไม่จำเป็น) จากนั้นเราก็จะต้องทำการคำนวณหาจุดตัดของโพรเจ็กต์เตอร์กับระนาบของการโพรเจ็กต์ของแต่ละจุดมุมของออบเจ็กต์ หลังจากได้จุดต่างๆ นี้แล้ว เราจึงสามารถออกคำสั่งเชื่อมต่อจุดที่เราคำนวณมาได้เพื่อสร้างภาพขึ้นมาอีกที่หนึ่ง (โดยนำเอาเฉพาะโคออร์ดิเนตในแกน X Y มาใช้เท่านี้ เนื่องจากค่า Z จะเท่ากันหมด คือเท่ากับ d)

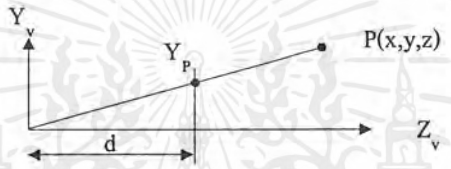
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ระนาบของการโพรเจ็กต์



เมื่อมองภาพด้านบน



เมื่อมองภาพด้านข้าง

ภาพที่ 2.19 การคำนวณหาจุดตัดของโพรเจ็กต์เตอร์และระนาบของการโพรเจ็กต์ของรูปสามมิติ แบบ Perspective

ในการคำนวณหาค่าโคออร์ดิเนตที่ได้จากการโพรเจ็กต์ (จุด P' จากการโพรเจ็กต์ของจุด P) ดังภาพข้างต้น เราสามารถใช้คุณสมบัติของสามเหลี่ยมคล้ายได้ดังนี้

$$x_P/d = x/z \text{ และ}$$

$$y_P/d = y/z$$

หรือ $x_P = d(x/z) = x(z/d)$

$y_P = d(y/z) = y(z/d)$

ซึ่งเราจะต้องนำสมการทั้งสองนี้ไปคำนวณกับจุดมุมทุกจุดของภาพของเรา แล้วเราจึงออกคำสั่งเชื่อมต่อจุดมุมต่างๆ ที่คำนวณได้โดยใช้คำสั่งในการลากเส้นสองมิติ คือ ใช้เฉพาะโคออร์ดิเนต X,Y เท่านั้น เพราะภาพสามมิติที่เราทำการโพรเจ็กต์มานี้อยู่ในระนาบ xy ที่มีค่า z=d

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

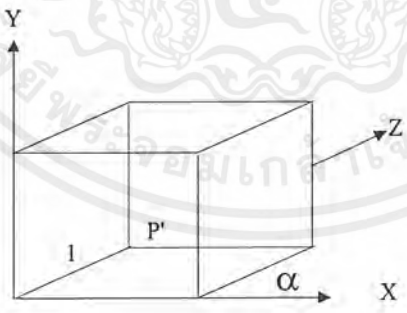
2.2.5.2 การสร้างภาพสามมิติแบบ Orthographic

การสร้างภาพสามมิติแบบ Orthographic ลงบนระนาบของการโพรเจ็กต์ที่มีค่า z=0 นั้นทิศทางของการโพรเจ็กต์จะเป็นทิศทางเดียวกับทิศทางตั้งฉากกับระนาบของการโพรเจ็กต์ ซึ่งในที่นี้ก็คือ แกน Z นั่นเอง ดังนั้นจุด P จะโพรเจ็กต์ไปยังจุดที่มีโคออร์ดิเนต x_P =x , y_P=y และ z_P=0 หรือก็คือแค่ตัดค่า z ออกไปนั่นเอง (ดูภาพที่ 2.14 ประกอบ) ซึ่งตามภาพที่ 2.14 เราก็จะได้ภาพด้านหน้า (front view) จากภาพบ้านสามมิติ และถ้าต้องการโพรเจ็กต์ลงบนระนาบที่มีค่า x=0 เรากำหนดให้ค่าของ x_p =0 หรือตัดค่า X ออกไป โดยนำเอาเฉพาะโคออร์ดิเนต Z,Y มาใช้ แต่เนื่องจากเราไม่มีแกนZY ในระบบสองมิติ เราจึงต้องแปลงโคออร์ดิเนตในแกน YZของเราให้เป็นโคออร์ดิเนตในแกน XY โดยให้ค้า z แทนค่า x และค่าของ y ยังคงเดิม แต่ถ้าสังเกตให้ดีจะเห็นได้ว่าภาพที่ได้จากการแปลงดังกล่าวเป็นภาพที่กลับซ้ายไปขวา เนื่องจากระบบที่เราใช้ในการคำนวณนี้เป็นระบบมือขวา ทำให้ค่า z ที่ชี้ออกนอกกระดาษมีค่าเป็นบวก เราจึงต้องทำการแปลงโดยคูณด้วย –1 เข้าไปกับค่า x ของทุกๆ จุดมุมของภาพที่ได้จากการโพรเจ็กต์ ซึ่งก็คือการแปลงจากระบบมือซ้ายนั้นเอง ซึ่งจะได้กล่าวในรายละเอียดอีกครั้งหนึ่ง

2.2.5.3 การสร้างภาพสามมิติแบบ Oblique

ในการคำนวณหาโพรเจ็กต์ชันแบบ Oblique ในที่นี้เราจะกำหนดให้ระนาบของการโพรเจ็กต์คือระนาบ XY ที่มีค่า z=0 ซึ่งมีทิศทางของการโพรเจ็กต์แทนด้วยเวกเตอร์ที่ผ่านจุดP ไปยังจุด P' ดังภาพที่ 2.20 โดยทิศทางของโพรเจ็กต์นี้ทำมุม β กับระนาบของการโพรเจ็กต์และทำมุม α กับแกน x

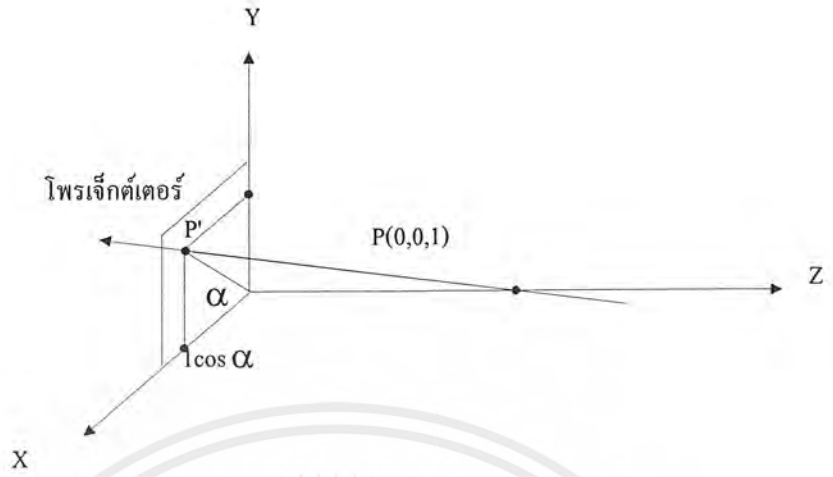
(ก)



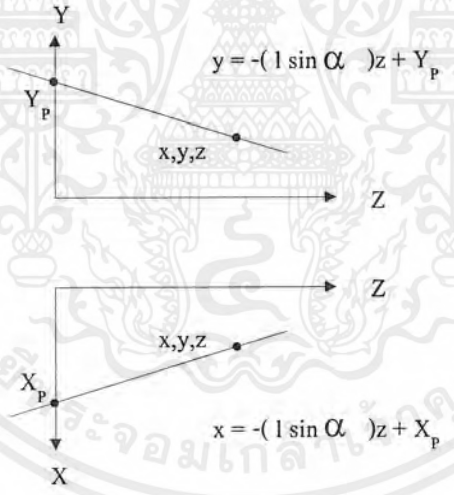
ภาพที่ 2.20 (ก) แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P(0,0,1) ไปยังจุด P' (1cosα, 1sinα,0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ข)



ภาพที่ 2.20 (ข) แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P(0,0,1) ไปยังจุด P' (1cosα, 1sinα,0)



ภาพที่ 2.21 แสดงการโพรเจ็กต์ชันแบบ Oblique Parallel จากจุด P ไปยังจุด P' (มองตามระนาบ YZ และ XZ)

จากภาพที่ 2.21 และสมการเส้นตรง เราจะได้ว่า

$x_P = z + z(l.\cos(\alpha)$ และ

$y_P = y + z(l.\sin(\alpha))$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 หลักพื้นฐานในการเขียนโปรแกรมเพื่อการแปลงภาพสองมิติ

ดังที่กล่าวมาแล้วว่าการแปลงภาพนั้นประกอบไปด้วย

1.การย้ายภาพ (tramslate)

2.การขยายหรือย่อภาพ (scale)

3.การหมุนภาพ (rotate)

โดยเราจะมาศึกษาวิธีการคำนวณเพื่อการแปลงภาพสองมิติกันก่อน

2.2.6.1 การย้ายภาพ

การย้ายภาพ (translate) ในที่นี้ก็คือการย้ายภาพจากตำแหน่งเดิมไปยังตำแหน่งใหม่ด้วยระยะทางที่เรากำหนด ซึ่งระยะทางนี้เราจะกำหนดเป็นสองส่วนด้วยกัน คือระยะทางตามแกน X และระนะทางตามแกน Y ดังสมการต่อไปนี้

$$x' = x+Dx$$

x คือ ค่า x-ordinate ของจุดเดิม

x'คือ ค่า x-ordinate ของจุดใหม่

Dx คือระยะทางตามแกน X

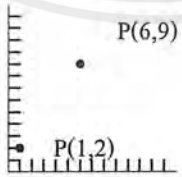
และ

$$y' = y + Dy$$

y คือ ค่า y-ordinate ของจุดเดิม

y' คือ ค่า y-ordinate ของจุดใหม่

Dy คือระยะทางตามแกน Y



ภาพที่ 2.22 การย้ายจุดจากจุด (1,2) ไปยังจุด (6,9) ได้ระยะทาง (5,7)

ถ้าเราเขียนในรูปของเมตริกจะได้

P = [x y] และ P' = [x' y'] โดย P คือจุดเดิม และ P' คือจุดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าเราให้	T = [Dx Dy]
ดังนั้นเราจะได้	[x' y'] = [x y] + [Dx Dy]
หรือ	P' = P+ T

เราสามารถย้ายภาพได้โดยใช้สมการดังกล่าวนี้กับจุดมุมทั้งหมดของภาพ โดยที่เราไม่จำเป็นจะต้องย้ายเส้นตรงทั้งเส้น เราสามารถย้ายเฉพาะจุดมุมเท่านั้น แล้วจึงใช้คำสั่งเชื่อมจุดมุมต่างๆ เข้าด้วยกันดังเดิม แต่สำหรับภาพที่ประกอบด้วยส่วนโค้งหรือพื้นที่ สิ่งที่สามารถทำได้ก็คือ เราจะต้องทำการย้ายจุดสำสำคัญที่ใช้ในการสร้างภาพ เช่นย้ายจุดศูนย์กลางของวงกลมสำหรับรูปที่ประกอบด้วยวงกลม แล้วจึงสร้างวงกลมขึ้น ณ ตำแหน่งใหม่ ซึ่งจะใช้เวลาน้อยกว่าการย้ายจุดบนเส้นรอบวงทั้งหมด

2.2.6.2 การขยายหรือย่อภาพ

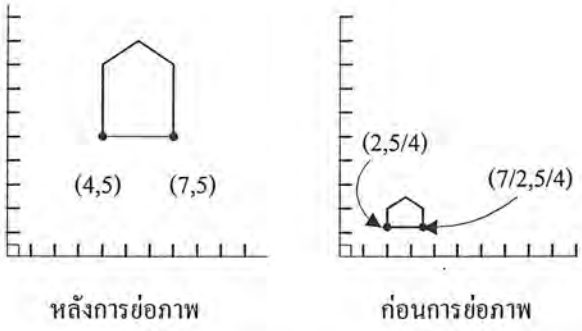
การขยายหรือย่อภาพ (scale) ในที่นี้ก็คือการทำการเปลี่ยนแปลงขนาดของภาพให้ใหญ่ขึ้นหรือเล็กลงตามค่าของสเกลแฟกเตอร์ที่เรากำหนด ตัวอย่างเช่น ถ้าเราต้องการขยายภาพให้มีขนาดใหญ่กว่าเดิมหนึ่งเท่า เราก็ต้องกำหนดให้ scale factor = 2 หรือถ้าเราต้องการย่อภาพให้มีขนาดเล็กลงกว่าเดิมหนึ่งเท่า เราก็ต้องกำหนดให้ scale factor = 0.5 เป็นต้น โดย scale factor นี้เราจะแบ่งออกเป็นสองส่วน คือ scale factor ตามแกน X และ scale factor ตามแกน Y ซึ่งถ้าเราต้องการขยายหรือย่อภาพให้มีการขยายหรือย่อในอัตราส่วนที่เท่ากันทั้งทางแกน X และแกน Y แล้ว เราก็ต้องกำหนดให้ค่าของ scale factor ทั้งทางแกน X และแกน Y เท่ากันนอกจากนี้แล้วถ้า scale factor ทางแกน X และแกน Y ไม่เท่ากันแล้ว จะมีผลให้รูปทรงของภาพเปลี่ยนแปลงไปด้วยในลักษณะเดียวกับการเปลี่ยนแปลง aspect ratio โดยในการขยายหรือย่อภาพนี้ เราจะกระทำกับจุดมุมของภาพลักษณะเดียวกับการย้ายภาพ

เราสามารถเขียนสมการในการขยายหรือย่อภาพได้ดังนี้

x' = x.Sx
- x คือ ค่า x-ordinate ของจุดเดิม
- x' คือ ค่า x-ordinate ของจุดใหม่
- Sx คือ scale factor ตามแกน X

y'= y.Sy
- y คือ ค่า y-ordinate ของจุดเดิม
- y' คือ ค่า y-ordinate ของจุดใหม่
- Sy คือ scale factor ตามแกน y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.23 แสดงการย่อภาพ

โดยถ้าเราเขียนให้อยู่ในรูปเมตริกลักษณะเดียวกับการย้ายภาพข้างต้น เราจะได้ดังนี้

ถ้ากำหนดให้ S คือ $\begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix}$

จากสูตรดังกล่าว เราจะได้ว่า

$$[x'\ y'] = [x\ y]\begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix}$$

หรือ P' = P.S

สมการดังกล่าวนี้เป็นการทำการขยายหรือย่อภาพ โดยใช้จุด (0,0) เป็นจุดเปรียบเทียบ ดังนั้น เมื่อเราทำการย่อหรือขยายขนาดแล้ว จะทำให้ระยะห่างของภาพที่ได้จากจุด (0,0) เปลี่ยนไปด้วยเนื่องจากการคำนวณนี้เป็นการคำนวณโดยการอาศัยจุด (0,0) เป็นหลักในการคำนวณ ดังภาพที่ 2.23 ถ้าเราต้องการคงตำแหน่งของภาพไว้ ณ ตำแหน่งเดิม เราจะต้องทำการขยาย หรือย่อภาพโดยอาศัยจุดศูนย์กลางของภาพเดิมเป็นจุดเปรียบเทียบ โดยเราจะต้องย้ายภาพมาให้จุดศูนย์กลางชองภาพที่เรากำหนดให้เป็นจุดเปรียบเทียบ อยู่ที่จุด (0,0) แล้วจึงทำการขยายหรือย่อภาพหลังจากนั้นเราจึงจะย้ายภาพกลับไปตำแหน่งเดิมโดยย้ายภาพให้จุดมุมที่ใช้เป็นจุดเปรียบเทียบกลับไปยังตำแหน่งเดิม

2.2.6.3 การหมุนภาพ

การหมุนภาพ (rotate) ในที่นี้ก็คือการทำการหมุนภาพและเปลี่ยนแปลงตำแหน่งของภาพไปตามมุมที่เรากำหนด โดยในการหมุนภาพนี้ เราจะหมุนจุดมุมของภาพในลักษณะเดียวกับการย้ายภาพและการขยายหรือย่อภาพ และเราจะกำหนดให้มุมเป็นบวกเมื่อเราหมุนทวนเข็มนาฬิกา ซึ่งเราสามารถเขียนสมการในการหมุนภาพได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

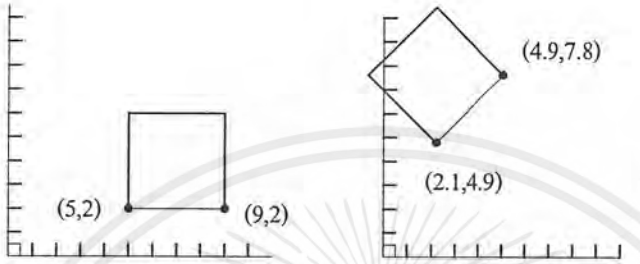
$x' = x.\cos(\theta) - y.\sin(\theta)$

$y' = x.\sin(\theta) + y.\cos(\theta)$

โดย (x,y) คือจุดเดิม

(x',y') คือจุดหลังจากการหมุน

และมุมที่ใช้ในการหมุนมีค่า = θ



ภาพที่ 2.24 แสดงลักษณะการหมุนภาพ

โดยถ้าเราเขียนให้อยู่ในรูปเมตริกลักษณะเดียวกับการย้ายภาพที่กล่าวมาแล้ว เราจะได้ดังนี้

$$[x'\ y'] = [x\ y]\begin{bmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{bmatrix}$$

หรือ $P' = P.R$

โดยการหมุนภาพในที่นี้เราจะใช้จุด (0,0) เป็นจุดศูนย์กลางของการหมุน ดังนั้นถ้าเราใช้สมการข้างต้นนี้ในการคำนวณภาพที่ไม่ได้มีจุดศูนย์กลางภาพอยู่ที่จุด (0,0) เราจะไม่ได้ภาพในลักษณะที่เราต้องการแน่ สิ่งที่เราจะต้องทำก็คือ ก่อนที่เราทำการหมุนภาพ เราจะต้องทำการย้ายภาพให้จุดศูนย์กลางของภาพมาอยู่ที่จุด (0,0) เสียก่อนแล้วจึงทำการหมุนภาพ แล้วย้ายภาพกลับไปอยู่ ณ ตำแหน่งเดิมอีกที่หนึ่ง

จากสมการต่างๆ ที่ได้กล่าวมาทั้งหมดในการแปลงภาพและเมตริกต่างๆ ที่ใช้สำหรับการแปลงภาพนั้น ในการเขียนโปรแกรมเพื่อแปลงภาพที่เราได้สร้างขึ้นมาแล้ว (ภาพที่กล่าวถึงนี้หมายถึงกลุ่มของจุดมุมของภาพ) เราก็จะต้องทำการเขียนเป็นส่วนของโปรแกรมที่นำจุดมุมแต่ละจุดมาคูณกับเมตริกที่ใช้ในการแปลงภาพที่เราต้องการ (หรือบวกในกรณีของการย้ายภาพ) ทีละเมตริกเพื่อให้ได้จุดมุมของภาพหลังจากการแปลงแล้ว ซึ่งค่อนข้างจะเสียเวลาในกรณีที่เราต้องทำการแปลงที่มากกว่าหนึ่งชนิดหรือหนึ่งครั้ง ในหัวข้อต่อไปเราจะมาเรียนวิธีที่ทำให้เราสามารถประหยัดการคำนวณได้ โดยการรวมเอา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตริกที่ใช้ในการแปลงเข้าด้วยกัน ซึ่งเราอาจจะนำเอาเมตริกในการทำการโพรเจ็กต์เข้าไปรวมด้วยก็ได้ ถ้าเราจะทำการโพรเจ็กต์หลังจากที่เราได้ทำการแปลงภาพแล้ว

2.2.6.4 ระบบโคออร์ดิเนตแบบ Homogeneous และการใช้เมตริกในการคำนวณ

ต่อไปเราจะทำความรู้จักกับระบบโคออร์ดิเนตแบบ Homogeneous เนื่องจากถ้าเรานำระบบนี้มาใช้งานแล้วเราก็จะส เมารถคำนวณหาเมตริกรวมของการแปลงทั้งหมด(การย้ายการย่อการขยายขนาด และการหมุน) ได้

ระบบโคออร์ดิเนตแบบ Homogeneous นี้ได้มีการเริ่มใช้ในเรขาคณิตก่อน แล้วจึงได้มีการนำมาใช้ในงานทางด้านกราฟิก โดยที่ในระบบนี้จุด(x,y) จะถูกเก็บไว้ในรูปแบบ(W.x, W.y, W) โดยที่ W คือ scale factor ซึ่งไม่เท่ากับ 0 ดังนี้ถ้าเรามีจุดที่มีโคออร์ดิเนตในระบบ Homogeneous เป็น (X,Y,W) เราจะสามารถหาค่าของ x,y ได้จาก

$$x = X/W$$

$$y = Y/W$$

ซึ่งในการใช้งานของเรานี้ เราจะกำหนดให้ W มีค่าเป็น 1 เสมอ ดังนั้นเราก็ไม่จำเป็น จะต้องคำนวณหาค่า x และ y จากโคออร์ดิเนตในโคออร์ดิเนตในโคออร์ดิเนตแบบ Homogeneous หรือเราอาจจะคิดอีกแง่หนึ่งก็ได้ว่า โคออร์ดิเนตของเรานี้เป็นจุดในสามมิติ โดยที่มีค่า z= 1 เสมอ นั่นก็คือเป็นจุดอยู่ในระนาบที่มีค่า z=1 นั่นเอง

เนื่องจากจุดๆ หนึ่งถูกเก็บไว้ในลักษณะของ (X,Y,W) ดังนั้นการเก็บข้อมูลของแต่ละจุดในรูปของเมตริกเราจะต้องเก็บไว้ในเมตริกขนาด 1x3 และเมตริกสำหรับใช้คำนวณในการแปลงทั้งสามประเภท จึงสามารถแทนด้วยเมตริกขนาด 3x3 ทั้งหมด ซึ่งเราจะได้คุณสมบัติเพิ่มเติมขึ้นมาจากระบบเดิมที่จุดๆ หนึ่งแทนด้วยเมตริก 1x2 และเมตริกการแปลงแบบเดิมคือ เราสามารถนำเมตริกทั้งหมดมาคูณกันเพื่อหาเมตริกสำหลับการแปลงรวมได้ นอกจากนี้เรายังสามารถนำเอาเมตริกในการโพรเจ็กต์และเมตริกของการแปลงระบบโคออร์ดิเนตมาคูณร่วมเข้าไปด้วยเพื่อหาผลลัพธ์สุทธิได้ ซึ่งข้อควรจำอย่างหนึ่งคือ ในการคูณเมตริกนั้น อันดับในการคูณไม่สามารถจะสลับกันได้ยกเว้นในบางกรณีเท่านั้น ดังนั้นในการหาเมตริกผลลัพธ์นั้น เราจะต้องคำนึงถึงลำดับของการคูณเป็นสำคัญด้วย โดยเมตริกการแปลงในระบบโคออร์ดิเนตแบบ Homogeneous เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การย้ายภาพ

$$[x'\ y'\ 1] = [x\ y\ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dx & Dy & 1 \end{bmatrix}$$

การขยายหรือย่อภาพ

$$[x'\ y'\ 1] = [x\ y\ 1] \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

การหมุนภาพด้วยมุม U

$$[x'\ y'\ 1] = [x\ y\ 1] \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ซึ่งถ้าเราต้องการแปลงภาพของเราโดยการย้ายภาพ การขยายภาพหรือย่อภาพ และการหมุนภาพอย่างหนึ่ง หรือทำการแปลงทั้งสามประเภท เราก็สามารถทำการคำนวณหาเมตริกรวมของการแปลงได้ โดยการนำเมตริกของการแปลงต่างๆ ที่เราต้องการใช้กับภาพมาคูณกันก่อน เพื่อให้ได้เมตริกของการแปลงทั้งหมด แล้วจึงเอาโคออร์ดิเนตของแต่ละจุดมาคูณกับเมตริกที่เราคำนวณได้

ข้อดีของวิธีนี้คือ ทำให้เราสามารถลดการคำนวณในการแปลงภาพไปได้อย่างมากเนื่องจากถ้าเราไม่ใช้ระบบโคออร์ดิเนตแบบ Homogeneous กับภาพของเรา เราจะต้องทำการคำนวณทีละขั้นกับทุกๆ จุดในภาพ แต่ถ้าเราใช้โคออร์ดิเนตแบบ Homogeneous แล้ว เราก็เพียงคำนวณหาเมตริกซึ่งใช้แทนการแปลงทั้งหมดที่ต้องการเพียงครั้งเดียวแล้วนำเมตริกที่คำนวณได้ไปคูณกับจุดแต่ละจุด ดังนั้นการคำนวณของแต่ละจุดจะเป็นเพียงการคูณระหว่างเมตริกขนาด 1x3 กับเมตริกขนาด 3x3 เพียงครั้งเดียว

2.2.7 หลักพื้นฐานในการเขียนแกรมเพื่อการแปลงภาพสามมิติ

ในระบบโคออร์ดิเนตแบบ Homogeneous จะมีลักษณะเดียวกับการแปลงภาพสองมิติโดยที่จุดๆหนึ่งจะมีโคออร์ดิเนตเป็น [XYZW] ซึ่งในการคำนวณเรากำหนดให้ W = 1 เราจะได้เมตริกในการแปลงภาพสามมิติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การย้ายภาพ

$$[x'\ y'\ z'\ 1] = [x\ y\ z\ 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Dx & Dy & Dz & 1 \end{bmatrix}$$

การขยายหรือย่อภาพ

$$[x'\ y'\ z'\ 1] = [x\ y\ z\ 1]\begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

การหมุนภาพ

การหมุนในระบบสามมิตินั้นสามารถแบ่งย่อยออกมาเป็นการหมุนแกน X,Y,Z เช่นเดียวกับโคออร์ดิเนตในระบบสามมิติ ซึ่งสามารถกำหนดให้เป็นโคออร์ดิเนตตามแกน x แกน y และแกนz ได้

หมุนรอบแกน z ด้วยมุม α

$$[x'\ y'\ z'\ 1] = [x\ y\ z\ 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

หมุมรอบแกน Y ด้วยมุม ϕ

$$[x'\ y'\ z'\ 1] = [x\ y\ z\ 1]\begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

สำหรับในกรณีที่เราต้องการแปลงภาพโดยการย้ายภาพขยายหรือย่อภาพหรือการหมุนภาพหรือทำการแปลงมากกว่าหนึ่งประเภทติดต่อกัน เราก็สามารถนำเมตริกในการแปลงของแต่ชะขั้นตอนมาทำการคูณกันในลักษณะเดียวกับการแปลงภาพสองมิติ โดยที่เราจะสามารถประหยัดการคำนวณไปได้เช่นเดียวกับการแปลงภาพสองมิติที่กล่าวมาแล้วข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7.1 การเขียนโปรแกรมเปลี่ยนระบบโคออร์ดิเนต

สำหรับการแปลงระบบโคออร์ดิเนตแบบ Homogeneous จากระบบมือขวาเป็นระบบมือซ้ายนั้น ถ้าเราสังเกตจากภาพที่ 2.12 เราจะเห็นว่าข้อแตกต่างระหว่างทั้งสองระบบก็คือ ค่าของ z ซึ่งจะมีเครื่องหมายตรงกันข้ามเท่านั้น ดังนั้นในการแปลงเราก็เพียงกำการคูณค่าของ z ด้วย –1 เท่านั้น หรือสามารถเขียนในรูปเมตริกได้ดังนี้

$$M_{rtol} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2.7.2 การเขียนโปรแกรมเพื่อทำการโพรเจ็กต์

ในระบบโคออร์ดิเนตแบบ Homogeneous จากวิธีการโพรเจ็กต์ที่เราได้กล่าวมาแล้วข้างต้นเราสามารถแทนการโพรเจ็กต์แบบ Perspective โดยมีระบบของการโพรเจ็กต์ตั้งฉากกับแกน z (ระบบ xy) ที่ระยะ d ดังกล่าวด้วยเมตริกต่อไปนี้

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/d \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

ซึ่งเมื่อนำเมตริกนี้มาคูณกับเมตริก [x y z 1] เราจะได้เมตริกผลลัพธ์คือ [x y z z/d] ซึ่งเราสามารถนำมาทำให้อยู่ในภาพที่สามารถนำไปใช้ได้ (w = 1) โดยนำ z/d หารทุกค่าก็จะได้ [x/(z/d) y/(z/d) d 1]

ในกรณีของการโพรเจ็กต์แบบ Orthographic ลงบนระนาบ z=0 โดยที่ทิศทางของการโพรเจ็กต์คือแกน z เมตริกของการโพรเจ็กต์คือ

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของการโพรเจ็กต์แบบ Oblique โดยระนาบของการโพรเจ็กต์คือระนาบ xy ที่ z=0 ทิศทางของการโพรเจ็กต์ทำมุม β กับระนาบของการโพรเจ็กต์ และทำมุม α กับแกน x เมตริกของการโพรเจ็กต์คือ

$$M_{ob} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l.\cos(\alpha) & l.\sin(\alpha) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2.8 สรุปขั้นตอนในการเขียนโปรแกรมเพื่อสร้างภาพสามมิติบนจอภาพแบบสองมิติ

จากเนื้อหาที่ได้กล่าวมาทั้งหมด สามารถสรุปเป็นขั้นตอนที่จะต้องกระทำในโปรแกรมเพื่อสร้างภาพสามมิติลนระบบแสดงภาพสองมิติได้ดังนี้

ขั้นที่ 1 กำหนดโคออร์ดิเนตของจุดมุมของภาพ (ในระบบมือขวา)

ขั้นที่ 2 ทำการแปลง ย้าย ขยาย/ลดขนาด และการหมุน

ขั้นที่ 3 การแปลงโคออร์ดิเนตจากระบบมือขวาเป็นระบบมือซ้าย

ขั้นที่ 4 ทำการโพรเจ็กต์ลงบนระนาบของการโพรเจ็กต์

ขั้นที่ 5 นำโคออร์ดิเนต (x,y) ที่ได้ไปใช้ในคำสั่งของการสร้างรูป (โดยใช้คำสั่งต่างๆ ที่ใช้สำหรับสร้างภาพสองมิติ)

ซึ่งเมตริกที่ใช้ในขั้นตอนที่ 2, 3 และ 4 สามารถนำมาคูณกันเพื่อหาเมตริกที่ใช้แทนการปฏิบัติทั้งหมดได้ ถ้าเราใช้ระบบโคออร์ดิเนตแบบ Homogeneous (X, Y, Z, W) ดังที่ได้กล่าวมาแล้ว โดยที่สมมติฐานว่าวิวโวลุมของเรามีขนาดที่สามารถครอบคลุมรูปภาพได้ทั้งหมด (ไม่มีการคลิปปิ้ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

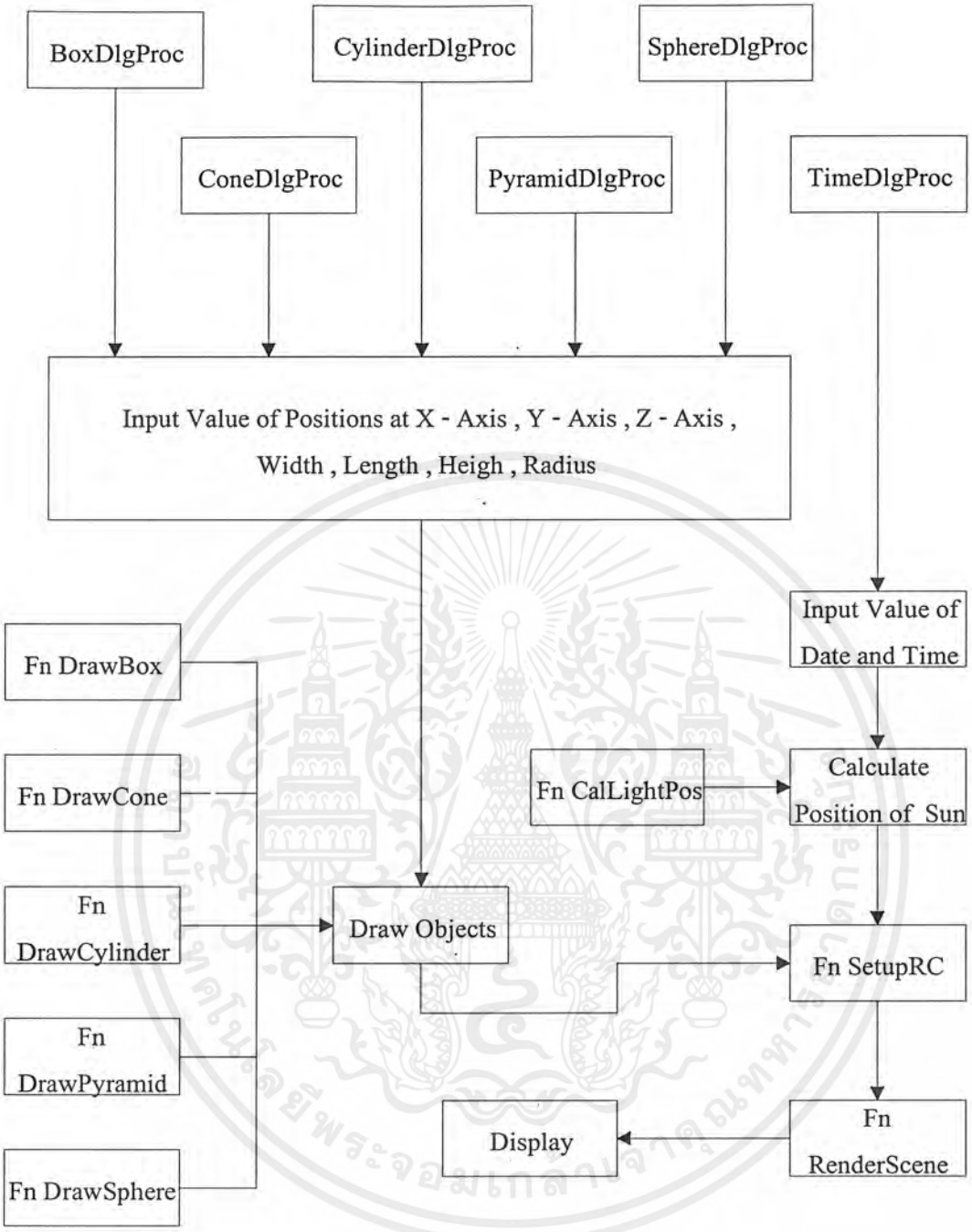
บทที่ 3
แนวคิดและขั้นตอนการทำงานของโปรแกรม

จากทฤษฎีต่าง ๆ ที่ได้ศึกษามาจากบทที่แล้วทำให้ทราบว่าการคำนวณสมการต่าง ๆ ที่เกี่ยวข้องกับการหาตำแหน่งของดวงอาทิตย์ขึ้นกับตัวแปรที่สำคัญคือ วันที่ เดือน และเวลาในช่วงที่ต้องการ ซึ่งจะมีการรับค่าเป็นวันที่ เดือน ชั่วโมง นาที โดยผู้วิจัยได้ใช้ภาษาคอมพิวเตอร์ Visual C + + Version 5 ในการเขียนโปรแกรม ทำงานบนสภาพแวดล้อมระบบปฏิบัติการวินโดว์ซึ่งเป็นระบบปฏิบัติการที่นิยมแพร่หลายและง่ายต่อการใช้งาน

ในการทำวิจัยผู้วิจัยได้สร้างโปรแกรม Solar.exe ขึ้น โดยโปรแกรมนี้จะสามารถป้อนวัตถุทรง 3 มิติง่าย ๆ คือ วัตถุรูปทรงกล่อง วัตถุทรงกรวย วัตถุทรงกระบอก วัตถุทรงปิรามิด และวัตถุทรงกลม ซึ่งวัตถุทรง 3 มิติง่าย ๆ เหล่านี้สามารถนำมาประกอบกันขึ้นเป็นวัตถุทรง 3 มิติที่ซับซ้อนได้และทำการคำนวณเพื่อสร้างเงา ซึ่งจากภาพที่ 3.1 ซึ่งแสดงแผนภาพการทำงานของโปรแกรมสามารถอธิบายได้ดังนี้

1. ทำการรับตำแหน่งในแกนต่าง ๆ และขนาดของวัตถุโดย
   - วัตถุรูปทรงกล่องจะใช้ฟังก์ชั่น BoxDlgProc
   - วัตถุทรงกรวยจะใช้ฟังก์ชั่น ConeDlgProc
   - วัตถุทรงกระบอกจะใช้ฟังก์ชั่น CylinderDlgProc
   - วัตถุทรงปิรามิดจะใช้ฟังก์ชั่น PyramidDlgProc
   - วัตถุทรงกลมจะใช้ฟังก์ชั่น SphereDlgProc
2. ทำการรับค่า วันที่ เดือน และเวลา ที่ต้องการแสดงผลโดยใช้ฟังก์ชัน TimeDlgProc
3. นำค่าที่ได้รับไปทำการคำนวณหามุมต่าง ๆ และค่าที่จำเป็นต้องใช้ในการแสดงผลโดยฟังก์ชัน CalLightPos
4. ค่าที่ได้จากฟังก์ชัน CalLightPos จะถูกส่งไปยังฟังก์ชัน SetupRC เพื่อเตรียมความพร้อมในการแสดงผล โดยฟังก์ชัน SetupRC นี้จะทำการเตรียมค่าอื่น ๆ ด้วย เช่น ค่าที่ได้จากการป้อนข้อมูลรูปทรง 3 มิติต่าง ๆ ลักษณะของแสงอาทิตย์ ลักษณะเงา สีของพื้นผิววัตถุ ฯลฯ
5. ค่าที่ได้จากฟังก์ชัน SetupRC จะถูกส่งไปยังฟังก์ชัน RenderScene เพื่อแสดงผลออกมาทางหน้าจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

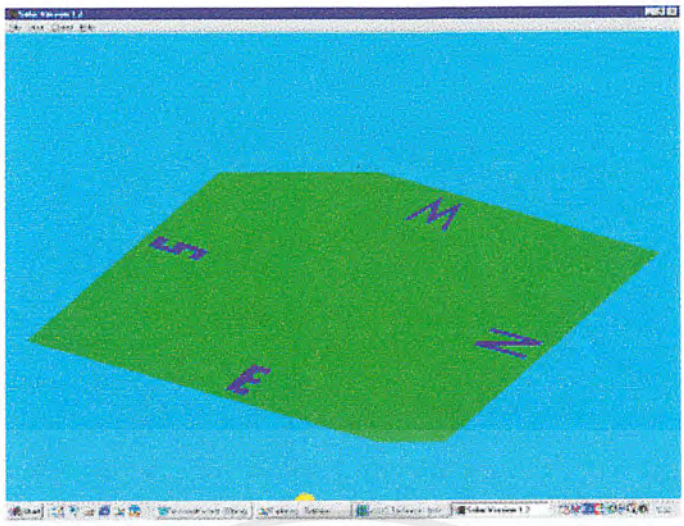


ภาพที่ 3.1 แผนภาพแสดงการทำงานของโปรแกรม Solar.exe

ลักษณะการทำงานของโปรแกรม Solar.exe

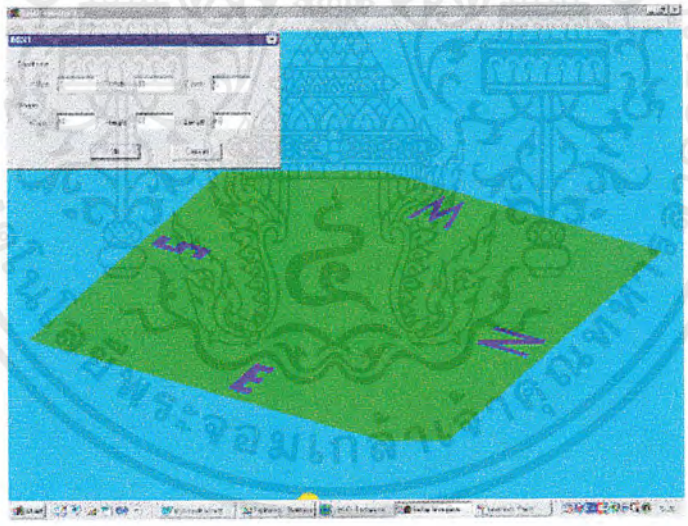
1. เมื่อเรียกโปรแกรม Solar.exe หน้าจอแสดงผลจะปรากฏภาพดังภาพที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 แสดงลักษณะของโปรแกรม Solar.exe

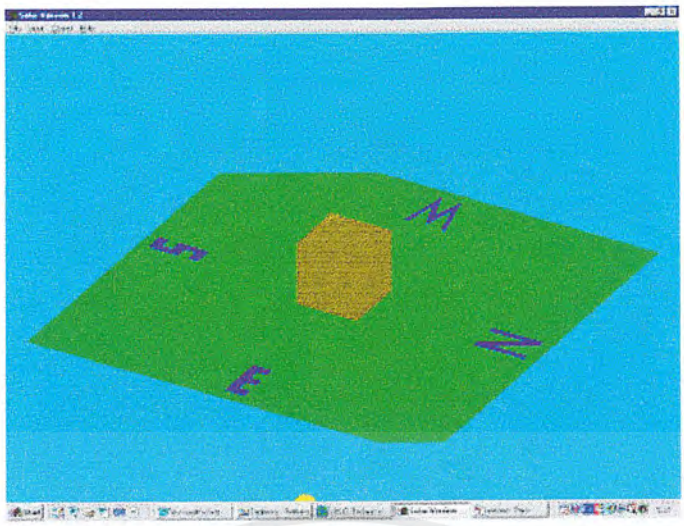
2. ใช้คำสั่ง Object - -> Box - -> Box1 จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดังภาพที่ 3.3 เพื่อป้อนข้อมูลตำแหน่งของวัตถุรูปทรงกล่อง ขนาดกว้าง ยาว และสูง



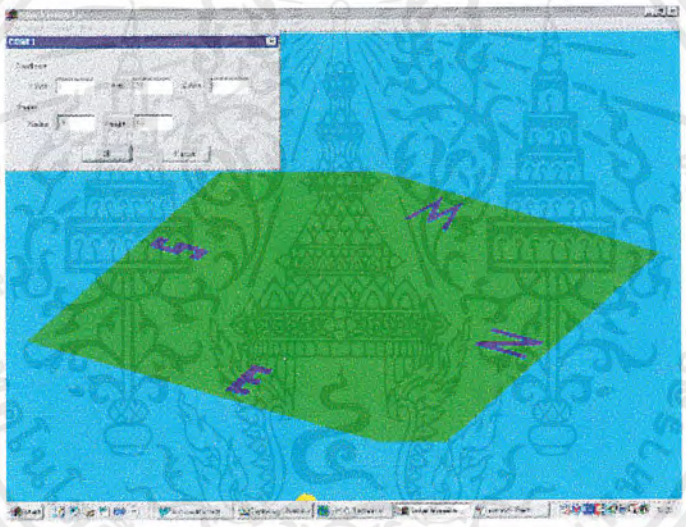
ภาพที่ 3.3 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุรูปทรงกล่อง

3. โปรแกรมจะทำการแสดงผลวัตถุรูปทรงกล่องที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป
4. ใช้คำสั่ง Object - -> Box - -> Box2 เพื่อสร้างวัตถุรูปทรงกล่องเพิ่มเติม
5. ใช้คำสั่ง Object - -> Cone - -> Cone1จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดังภาพที่ 3.5 เพื่อป้อนข้อมูลตำแหน่งของวัตถุทรงกรวย ขนาดรัศมี และความยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



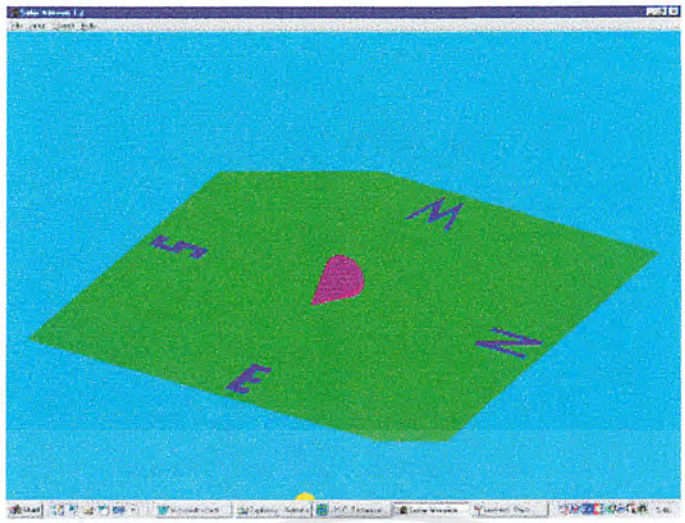
ภาพที่ 3.4 แสดงผลวัตถุรูปทรงกล่องที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป



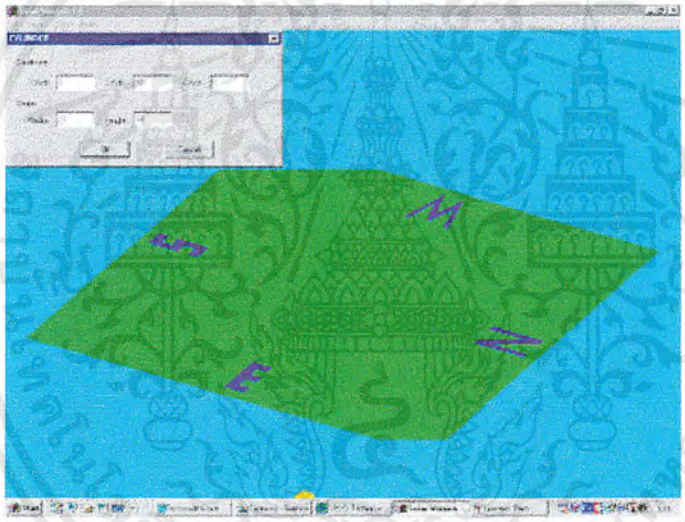
ภาพที่ 3.5 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงกรวย

6. โปรแกรมจะทำการแสดงผลวัตถุทรงกรวยที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป
7. ใช้คำสั่ง Object - -> Cone - -> Cone2 เพื่อสร้างวัตถุทรงกรวยเพิ่มเติม
8. ใช้คำสั่ง Object - -> Cylinder - -> Cylinder1 จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดัง ภาพที่ 3.7 เพื่อป้อนข้อมูลตำแหน่งของวัตถุทรงกระบอก ขนาดรัศมี และความสูง
9. โปรแกรมจะทำการแสดงผลวัตถุทรงกระบอกที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป
10. ใช้คำสั่ง Object - -> Cylinder - -> Cylinder2 เพื่อสร้างวัตถุทรงกระบอกเพิ่มเติม
11. ใช้คำสั่ง Object - -> Pyramid - -> Pyramid1 จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดัง ภาพที่ 3.9 เพื่อป้อนข้อมูลตำแหน่งของวัตถุทรงปิรามิด ขนาดความกว้างฐานและความสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



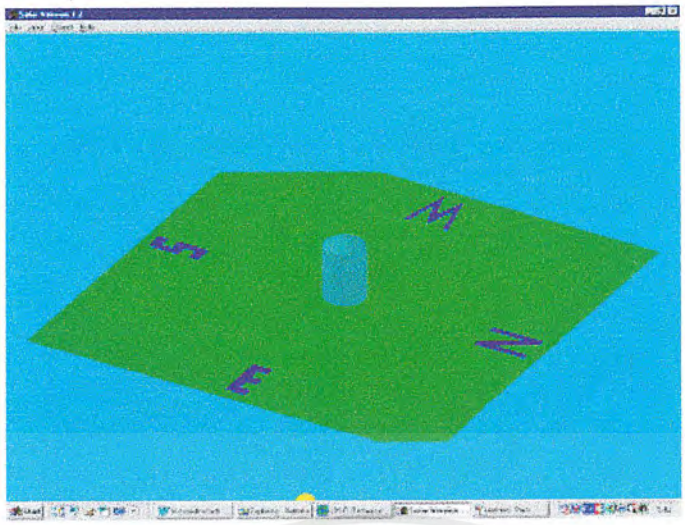
ภาพที่ 3.6 แสดงผลวัตถุทรงกรวยที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป



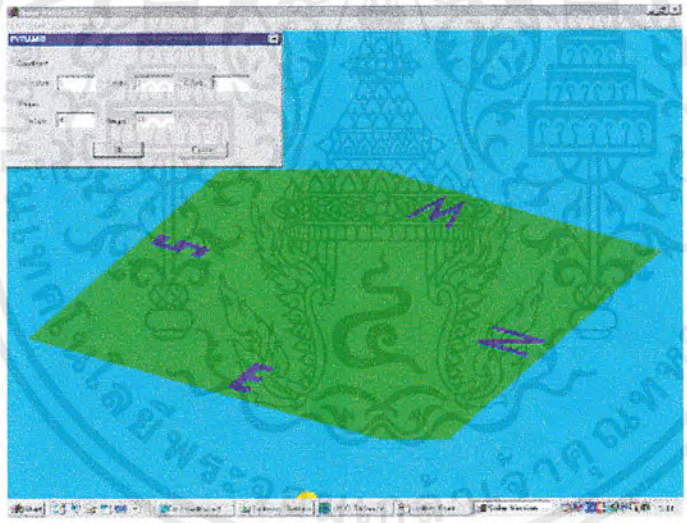
ภาพที่ 3.7 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงกระบอก

12. โปรแกรมจะทำการแสดงผลวัตถุทรงปิรามิดที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป
13. ใช้คำสั่ง Object - -> Pyramid - -> Pyramid2 เพื่อสร้างวัตถุทรงปิรามิดเพิ่มเติม
14. ใช้คำสั่ง Object - -> Sphere - -> Sphere1 จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดัง ภาพที่ 3.11 เพื่อป้อนข้อมูลตำแหน่งของวัตถุทรงกลม และขนาดรัศมี
15. โปรแกรมจะทำการแสดงผลวัตถุทรงกลมที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



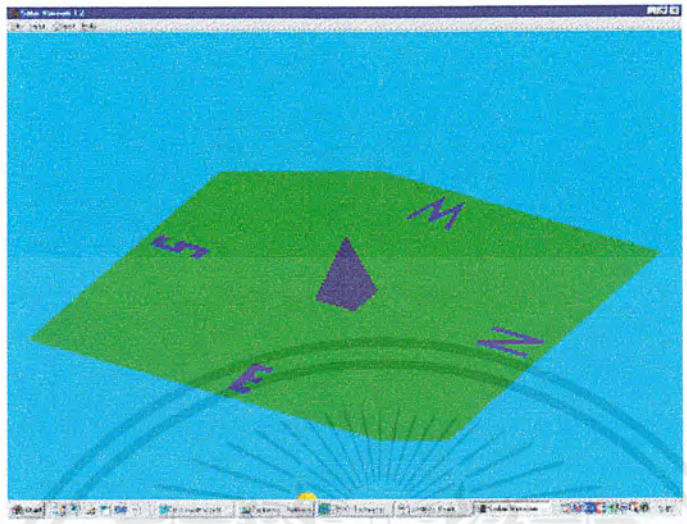
ภาพที่ 3.8 แสดงผลวัตถุทรงกระบอกที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป



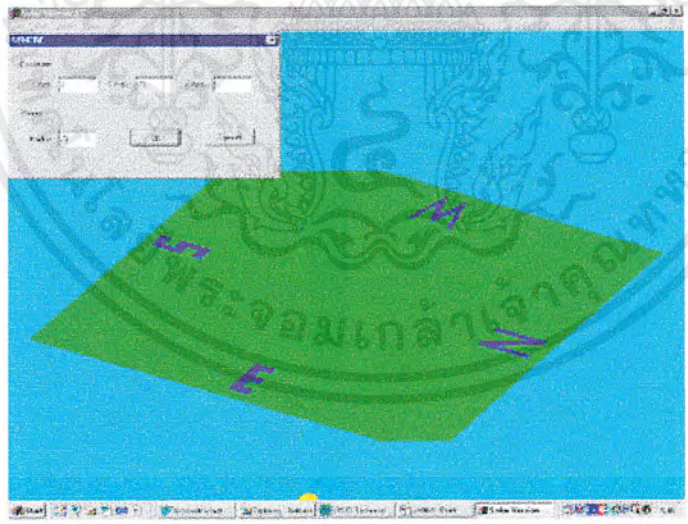
ภาพที่ 3.9 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุทรงปิรามิด

16. ใช้คำสั่ง Object - -> Sphere - -> Sphere2 เพื่อสร้างวัตถุทรงกลมเพิ่มเติม
17. ใช้คำสั่ง Input - -> Time จะปรากฏไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดังภาพที่ 3.13 เพื่อป้อนข้อมูล วัน และเวลา ที่ใช้ในการคำนวณตำแหน่งของดวงอาทิตย์ได้
18. โปรแกรมจะทำการแสดงผลวัตถุและลักษณะเงาของวัตถุ ในวัน และเวลาที่ป้อนเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

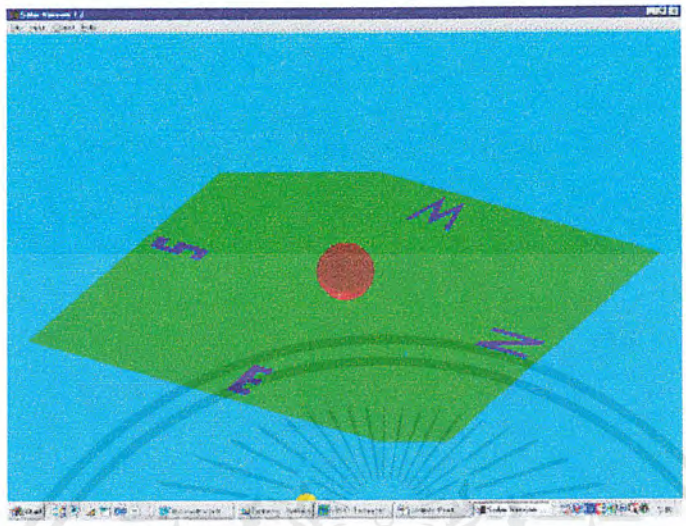


ภาพที่ 3.10 แสดงผลวัตถุทรงปิรามิดที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป

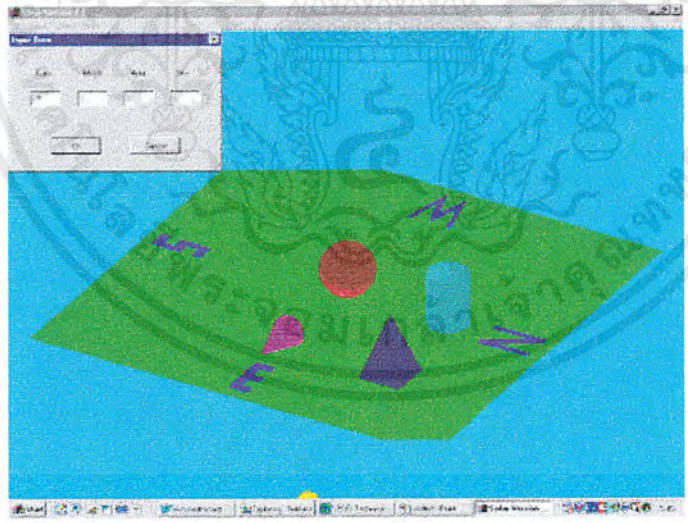


ภาพที่ 3.11 แสดงการป้อนข้อมูลตำแหน่งและขนาดรูปร่าง ของวัตถุรูปทรงกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

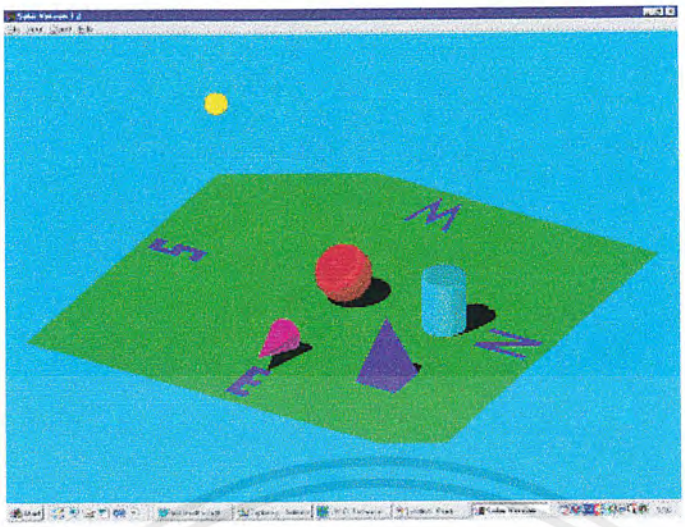


ภาพที่ 3.12 แสดงผลวัตถุทรงกลมที่ตำแหน่งและขนาดตามข้อมูลที่ป้อนเข้าไป



ภาพที่ 3.13 แสดงการป้อนข้อมูลวัน และเวลา ที่ใช้ในการคำนวณของดวงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.14 แสดงผลลักษณะเงาของวัตถุในวัน และเวลาที่ป้อนเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
เครื่องมืออุปกรณ์ การทดลอง และผลการทดลอง

4.1 เครื่องมือและอุปกรณ์

ในการวิจัยผู้วิจัยได้ทำการออกแบบโปรแกรมแสดงลักษณะของเงาที่เกิดจากรูปทรงสามมิติง่าย ๆ และเพื่อให้ทราบว่าการแสดงผลของโปรแกรมที่ออกแบบนั้นใกล้เคียงกับความเป็นจริงหรือไม่ จึงได้ทำการทดลองหาลักษณะการเกิดเงาจริงมาเปรียบเทียบกับผลที่ได้จากโปรแกรมโดยมีอุปกรณ์และเครื่องมือการทดลองดังนี้

1. วัตถุรูปทรงสามมิติ ง่าย ๆ
2. แบบจำลองบ้าน
3. แผ่นกระดาษแสดงทิศทาง
4. กล้องถ่ายรูป
5. ฟิล์มสี
6. เข็มทิศ

4.2 วิธีการทดลอง

1. นำวัตถุรูปทรงสามมิติต่าง ๆ ไปวางลงบนจุดกึ่งกลางของแผ่นกระดาษแสดงทิศทางในบริเวณกลางแจ้ง
2. ถ่ายรูปเงาที่เกิดจากวัตถุรูปทรงสามมิติต่าง ๆ
3. ทำซ้ำ 1. และ 2. ทุก ๆ 1 ชั่วโมง
4. นำรูปถ่ายที่ได้ไปเปรียบเทียบกับผลที่ได้จากโปรแกรมที่จำลองรูปขึ้นมา

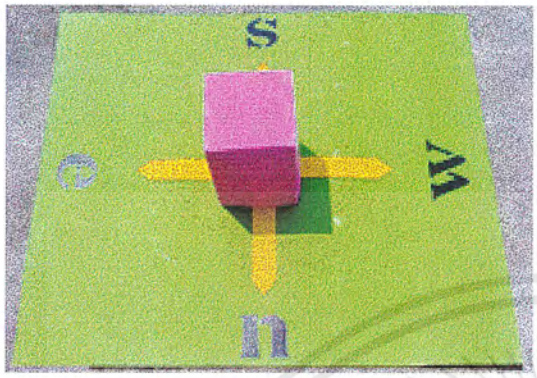
4.3 ผลการทดลอง

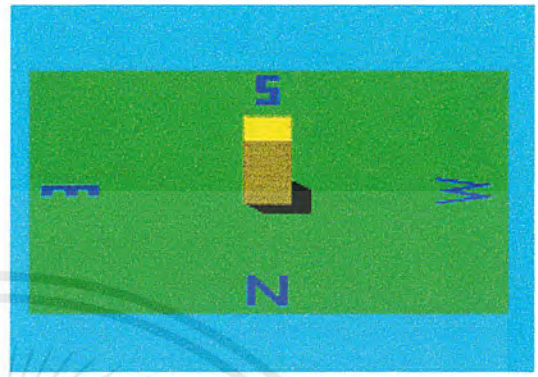
เพื่อทำการเปรียบเทียบผลที่ได้จากการทดลองกับผลที่ได้จากโปรแกรม ว่าจะได้ผลออกมาเป็นไปในแนวทางเดียวกันหรือไม่ ผู้วิจัยจึงทำการบันทึกรูปถ่ายเมื่อ วันที่ 21 มีนาคม พ.ศ. 2542 ตั้งแต่เวลา 11.00 – 15.00 นาฬิกา โดยแบ่งการบันทึกรูปถ่ายออกเป็นช่วง ๆ ช่วงละ 1 ชั่วโมง ซึ่งได้ผลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

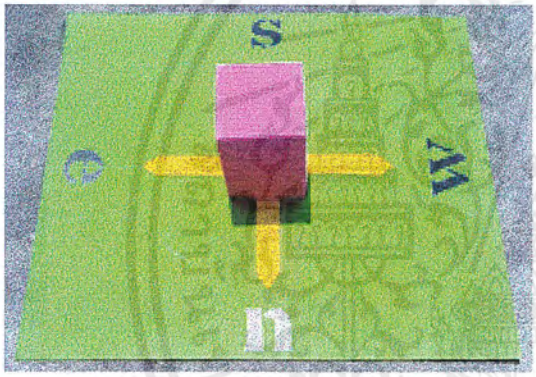
4.3.1 วัตถุรูปทรงกล่อง

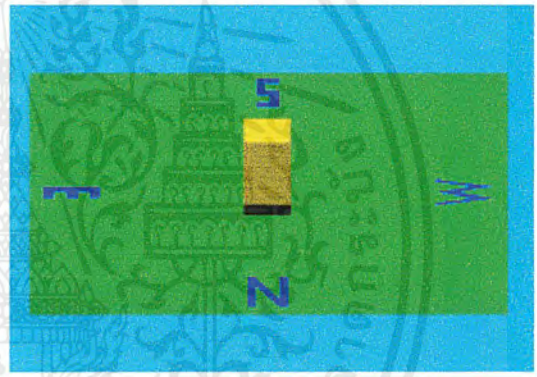
เวลา 11.00 น



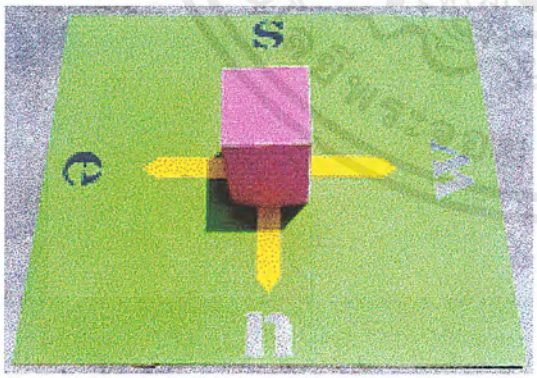


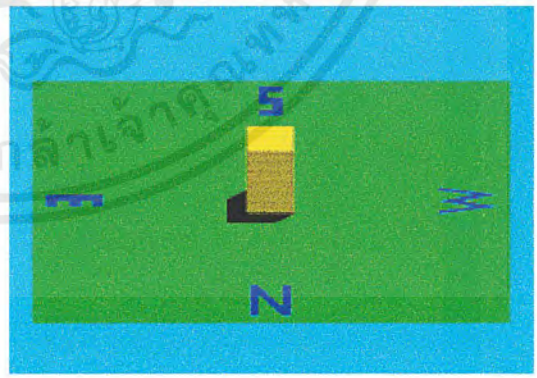
เวลา 12.00 น





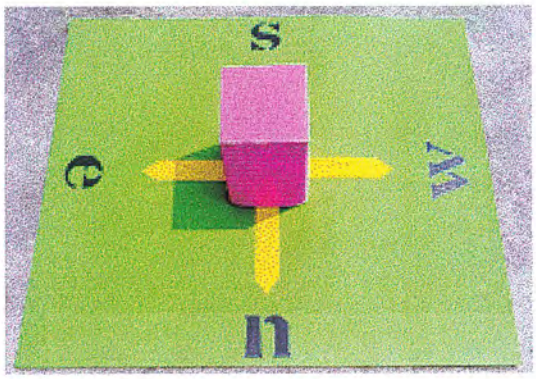
เวลา 13.00 น

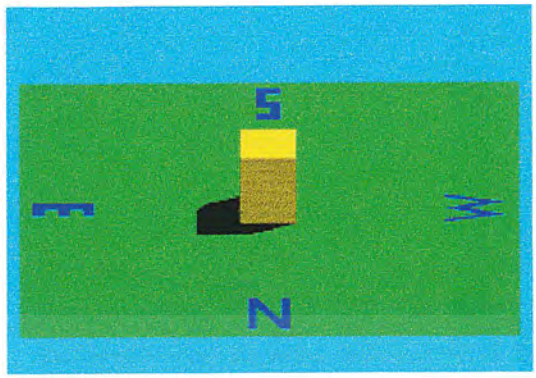




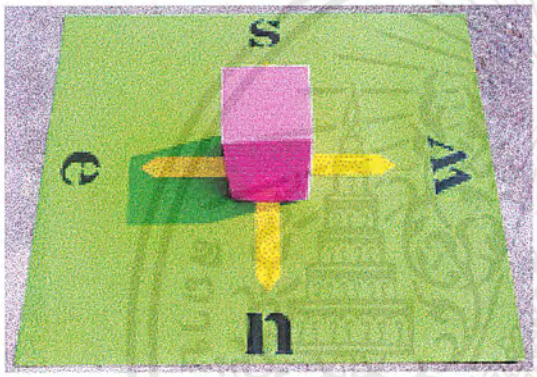
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

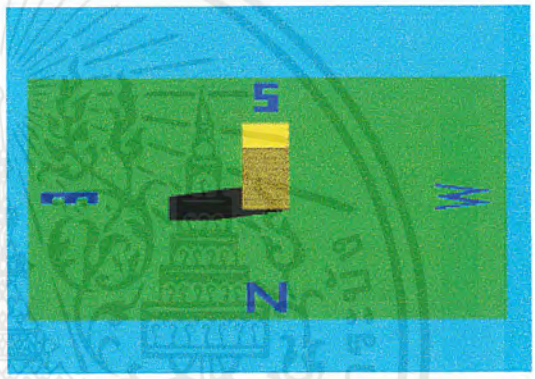
เวลา 14.00 น





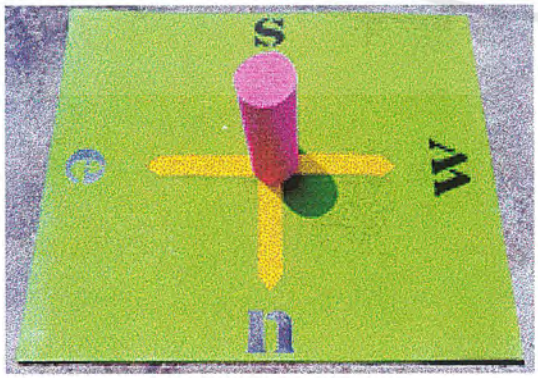
เวลา 15.00 น

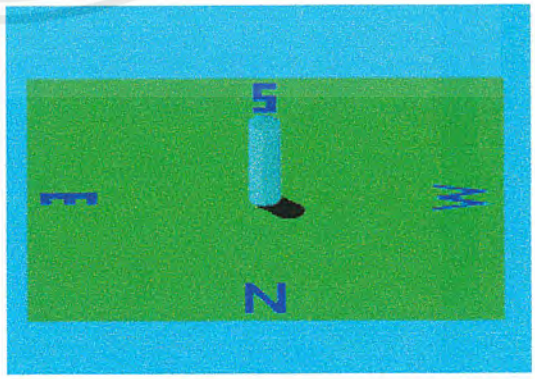




4.3.2 วัสดุรูปทรงกระบอก

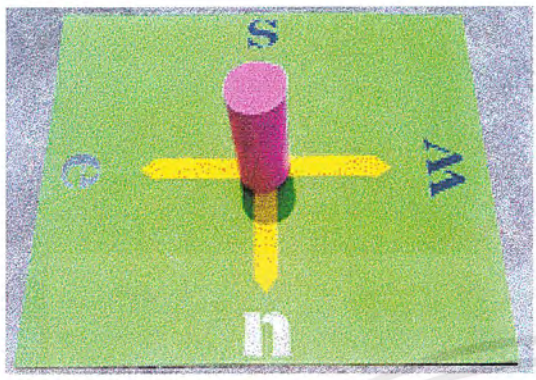
เวลา 11.00 น

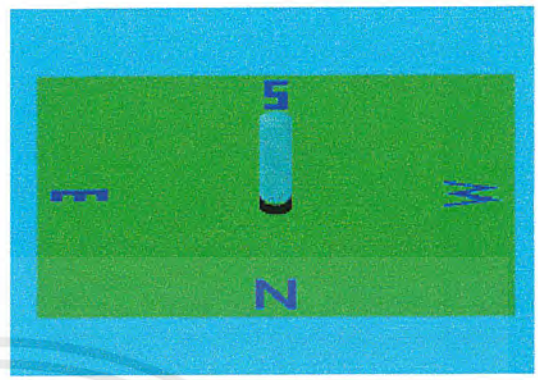




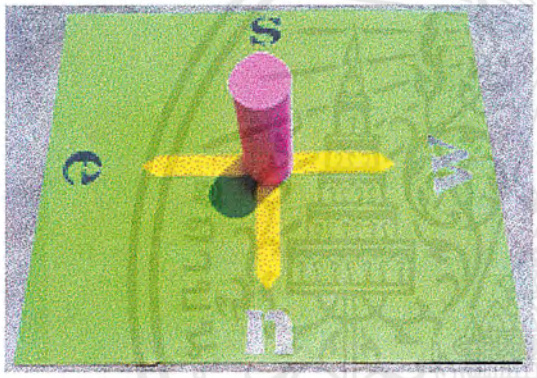
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

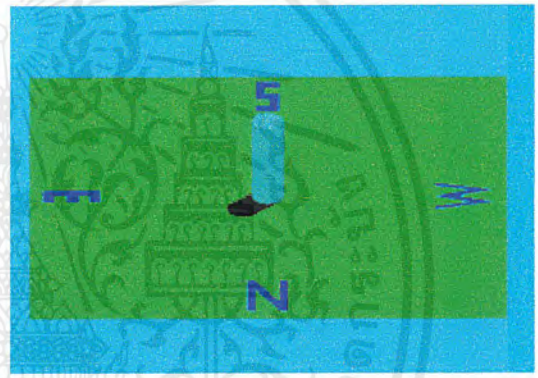
เวลา 12.00



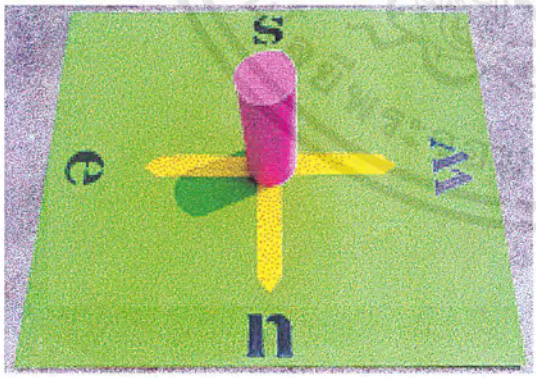


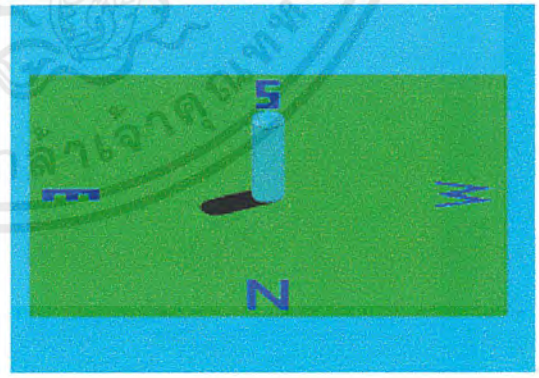
เวลา 13.00 น





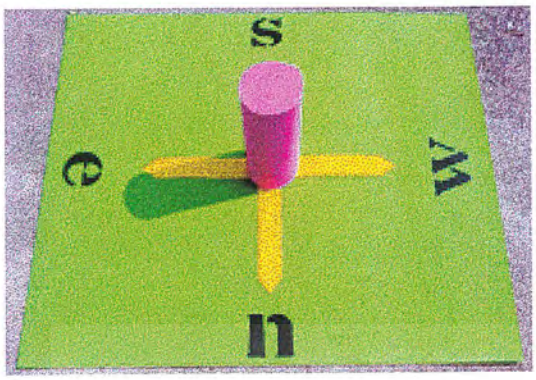
เวลา 14.00 น

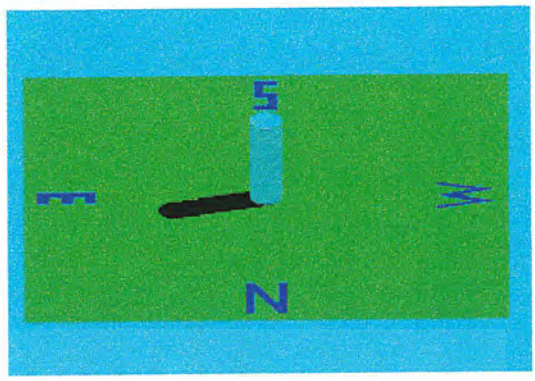




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

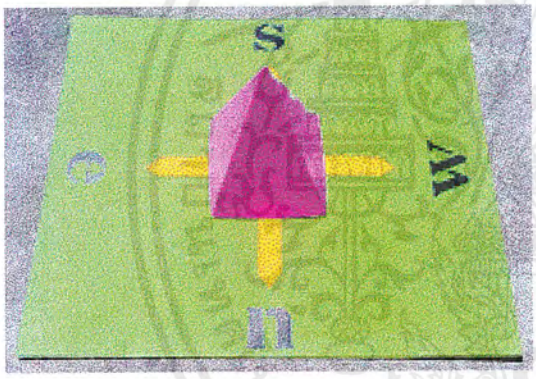
เวลา 15.00 น

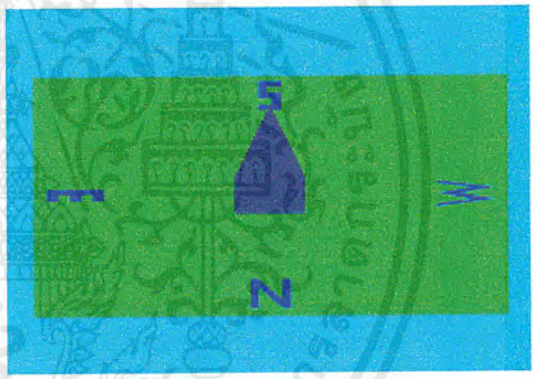




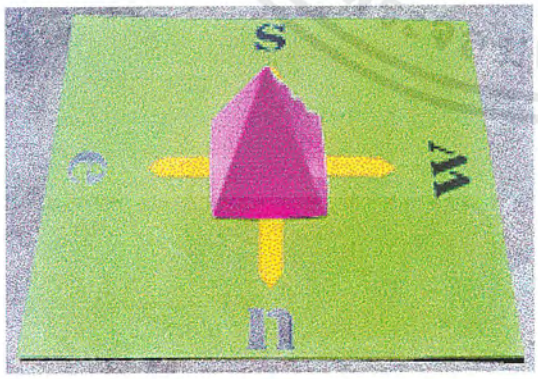
4.3.3 วัสดุรูปทรงปิรามิด

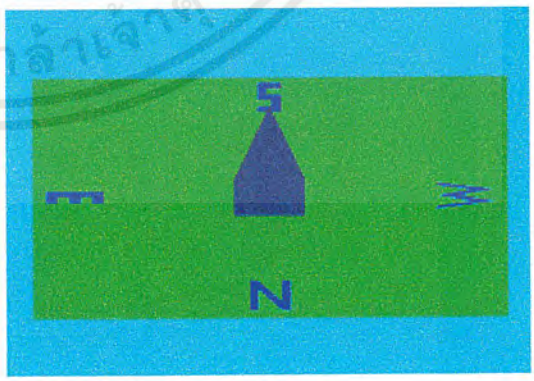
เวลา 11.00 น





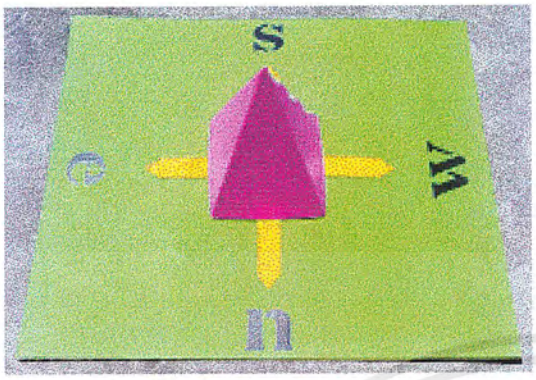
เวลา 12.00 น

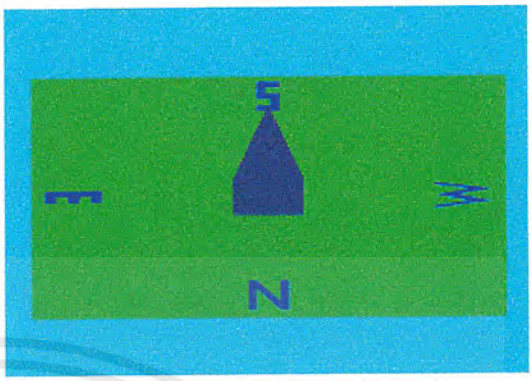




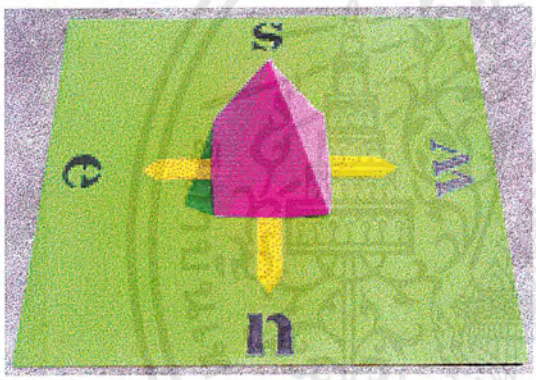
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

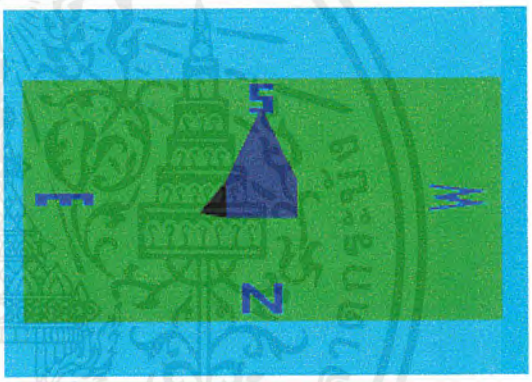
เวลา 13.00 น



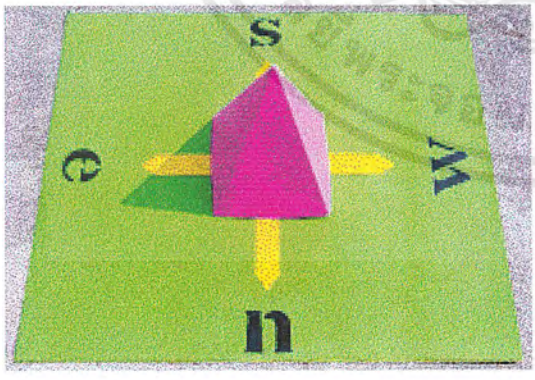


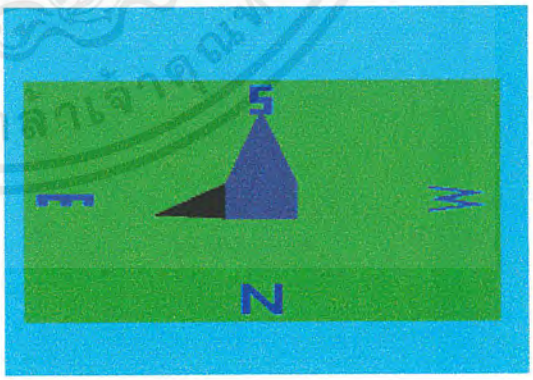
เวลา 14.00 น





เวลา 15.00 น

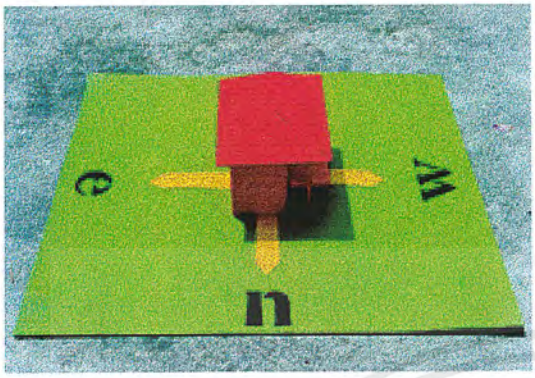


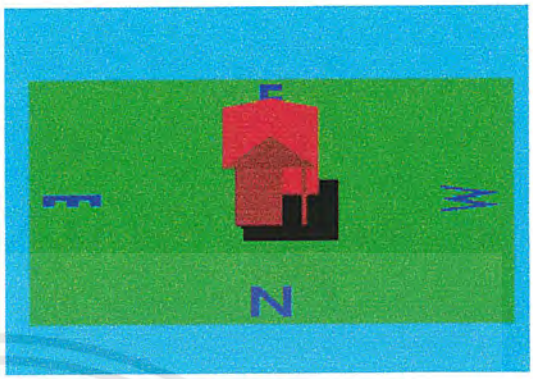


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

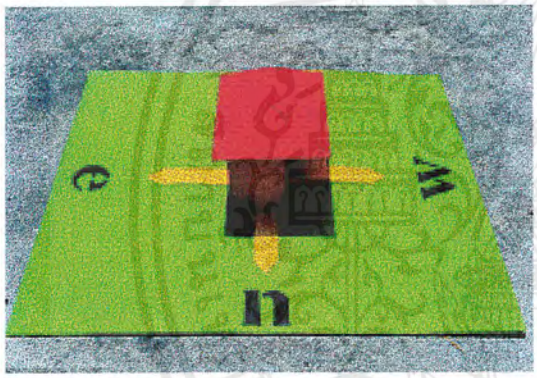
4.3.3 วัสดุรูปทรงบ้าน

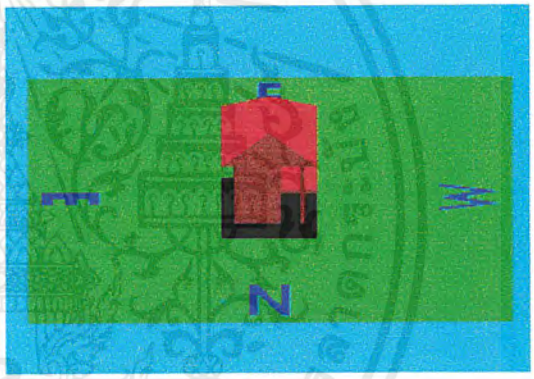
เวลา 11.00 น



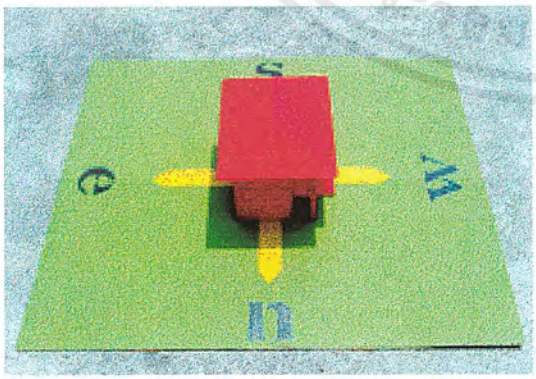


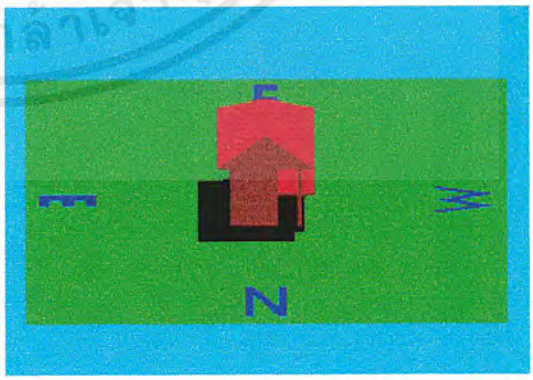
เวลา 12.00 น





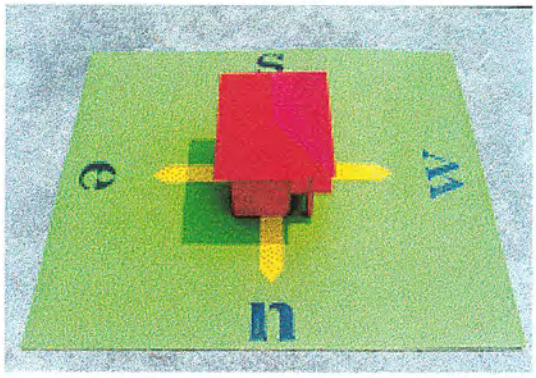
เวลา 13.00 น

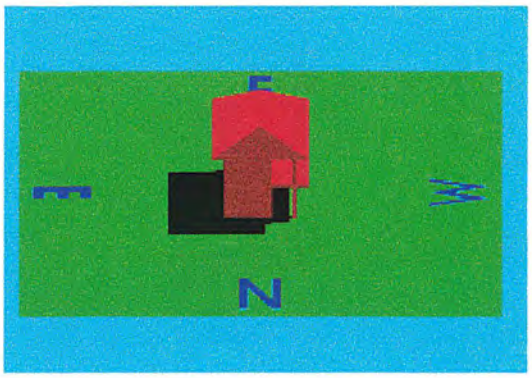




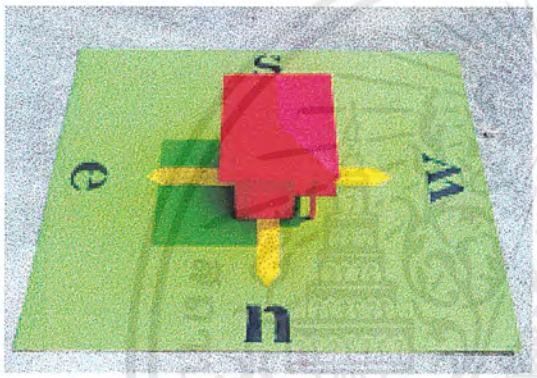
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

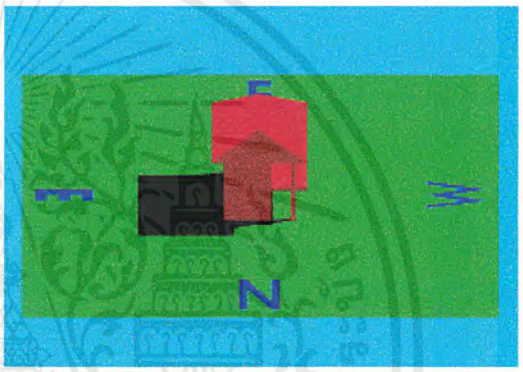
เวลา 14.00 น





เวลา 15.00 น





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย

จากการที่ได้ศึกษาถึงทฤษฎีเบื้องต้นเกี่ยวกับดวงอาทิตย์ การโคจรของโลกรอบดวงอาทิตย์ มุม และสมการที่เกี่ยวข้องในการเกิดเงาของวัตถุในเวลาที่ต้องการ การสร้างภาพสองมิติและสามมิติ เพื่อนำมาใช้ในการเขียนโปรแกรมจำลองลักษณะของเงาที่เกิดจากวัตถุทรงสามมิติซึ่งสามารถป้อนรูปทรงง่าย ๆ ได้ และนำผลของโปรแกรมที่ได้ไปเปรียบเทียบกับปรากฏการณ์ที่เกิดขึ้นจริงโดยการสร้างแบบจำลองและทำการบันทึกภาพ ณ เวลาต่าง ๆ

ดังนั้นจึงพอสรุปได้จากผลการทดลองที่เปรียบเทียบกับผลที่ได้จากโปรแกรมว่าโปรแกรมที่ได้พัฒนาให้สามารถป้อนรูปทรงต่าง ๆ สามารถแสดงผลออกมาได้ใกล้เคียงกับปรากฏการณ์จริง ซึ่งจะมีประโยชน์ในการทราบลักษณะเงาที่จะเกิดขึ้นได้ก่อนที่จะมีการสร้างอาคารทำให้สามารถออกแบบอาคารได้เหมาะสมกับการใช้งานของอาคารนั้น ๆ และ การต่อเติมอาคารเพื่อจุดประสงค์ในการบังแสงอาทิตย์ที่มารบกวนได้โดยช่วยให้เห็นลักษณะการบังแสงได้ดีที่สุด ซึ่งต่อไปอาจมรการพัฒนาโปรแกรมให้คำนวณหาระยะของเงาที่เกิดขึ้น และพลังงานความร้อนที่ได้รับ ณ ช่วงเวลานั้น ๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายการทำงานของฟังก์ชั่นต่าง ๆ

ฟังก์ชั่น	
1. AboutDlgProc	แสดงรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับโปรแกรม
2. BoxDlgProc	รับตำแหน่งในแกนต่าง ๆ ขนาดกว้าง ยาว และสูงของวัตถุรูปกล่องทางแป้นพิมพ์ แล้วนำไปสร้างเป็นภาพ 3 มิติ
3. CalcNormal	คำนวณหา Normal Vector
4. CalLightPos	คำนวณหามุมdeclination มุมaltitude มุมazimuth เพื่อหาตำแหน่งดวงอาทิตย์
5. ChangeSize	ป้องกันสัดส่วนของหน้าต่างแสดงผลเปลี่ยนแปลงเมื่อหน้าต่างแสดงผลเปลี่ยนขนาด
6. ConeDlgProc	รับตำแหน่งในแกนต่าง ๆ ขนาดรัศมี และความสูงของวัตถุทรงกรวยทางแป้นพิมพ์ แล้วนำไปสร้างเป็นภาพ 3 มิติ
7. CylinderDlgProc	รับตำแหน่งในแกนต่าง ๆ ขนาดรัศมี และความสูงของวัตถุทรงกระบอกทางแป้นพิมพ์ แล้วนำไปสร้างเป็นภาพ 3 มิติ
8. DrawBox	สร้างวัตถุรูปกล่อง
9. DrawCone	สร้างวัตถุทรงกรวย
10. DrawCylinder	สร้างวัตถุทรงกระบอก
11. DrawEast	สร้างตัวอักษร " E "
12. DrawHouseEast	สร้างวัตถุรูปทรงบ้านหันด้านหน้าไปทางทิศตะวันออก
13. DrawHouseNorth	สร้างวัตถุรูปทรงบ้านหันด้านหน้าไปทางทิศเหนือ
14. DrawHouseSouth	สร้างวัตถุรูปทรงบ้านหันด้านหน้าไปทางทิศใต้
15. DrawHouseWest	สร้างวัตถุรูปทรงบ้านหันด้านหน้าไปทางทิศตะวันตก
16. DrawNorth	สร้างตัวอักษร " N "
17. DrawPyramid	สร้างวัตถุรูปทรงปิรามิด
18. DrawSouth	สร้างตัวอักษร " S "
19. DrawSphere	สร้างวัตถุรูปทรงกลม
20. DrawWest	สร้างตัวอักษร " W "
21. HouseLocation	สร้างวัตถุรูปทรงบ้านหันด้านหน้าไปในทิศทางที่ได้รับการป้อนค่า
22. MakeShadowMatrix	คำนวณหาตำแหน่งแสดงเงา

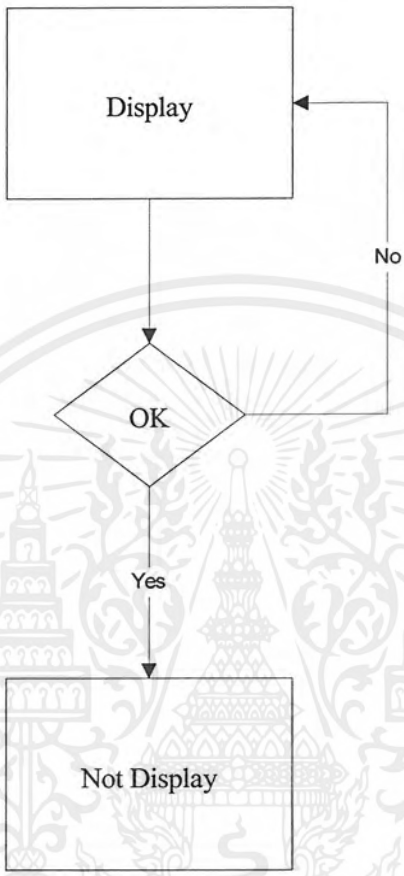
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายการทำงานของฟังก์ชั่นต่าง ๆ (ต่อ)

ฟังก์ชั่น	
23. PyramidDlgProc	รับตำแหน่งในแกนต่าง ๆ ขนาดความกว้างฐาน และความสูงของวัตถุทรงปิรามิดทางแป้นพิมพ์ แล้วนำไปสร้างเป็นภาพ 3 มิติ
24. ReduceToUnit	คำนวณหา Vector 1 หน่วย
25. RenderScene	สร้างรูปทรง 3 มิติต่าง ๆ และสร้างเงา
26. SetupRC	เตรียมข้อมูลต่าง ๆ ที่จำเป็นในการแสดงผลที่หน้าต่างแสดงผล
27. SphereDlgProc	รับตำแหน่งในแกนต่าง ๆ ขนาดรัศมีของวัตถุทรงกลมทางแป้นพิมพ์ แล้วนำไปสร้างเป็นภาพ 3 มิติ
28. TimeDlgProc	รับค่า วันที่ เดือน เวลา ทางแป้นพิมพ์ แล้วนำไปใช้ในการคำนวณ

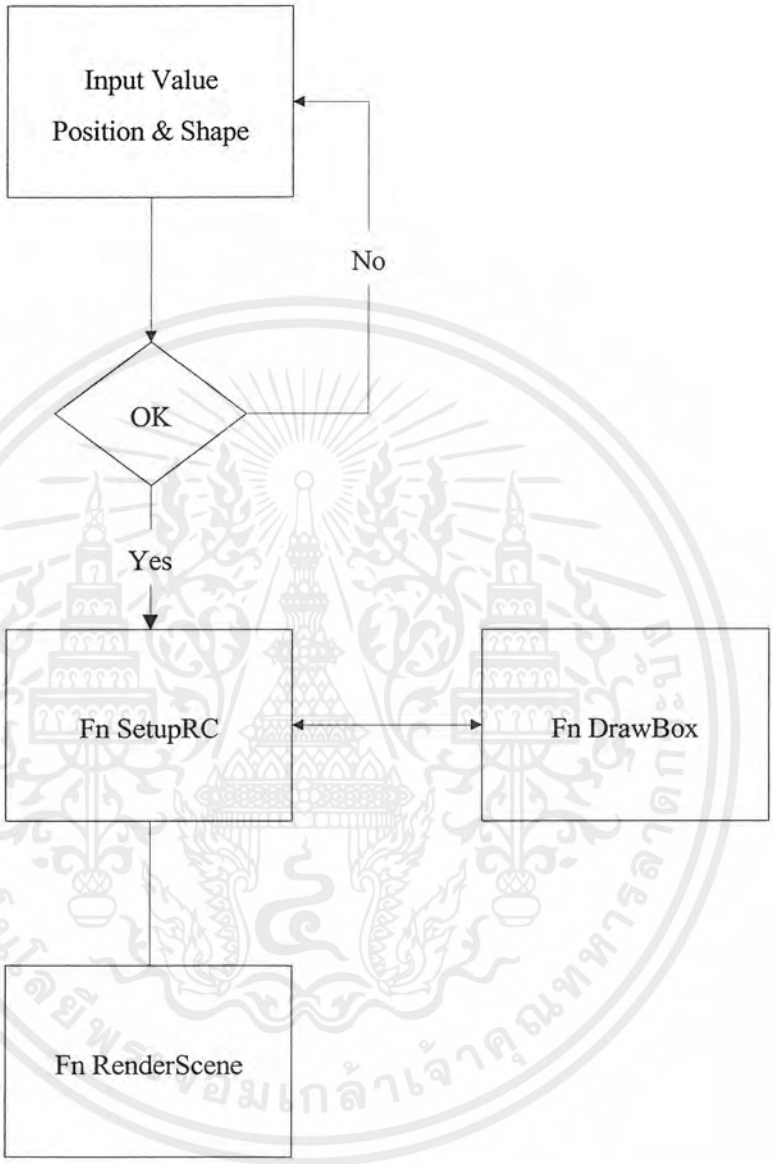
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function AboutDlgProc



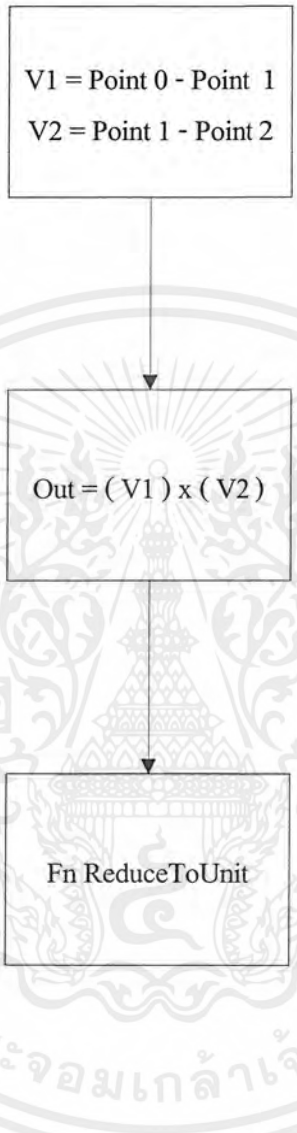
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function BoxDlgProc



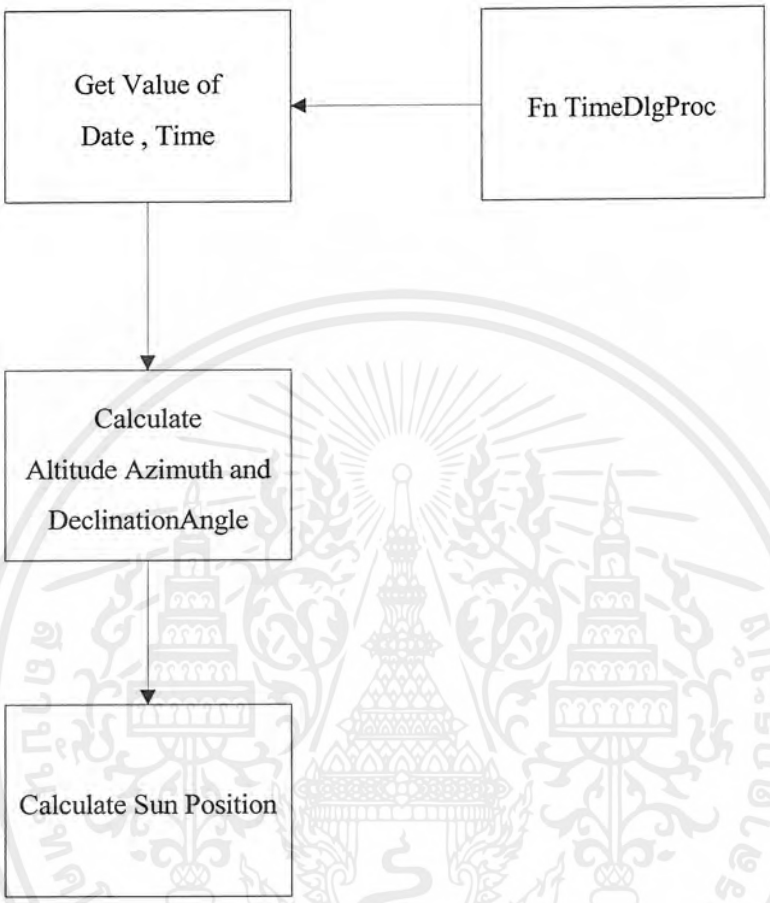
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function CalcNormal



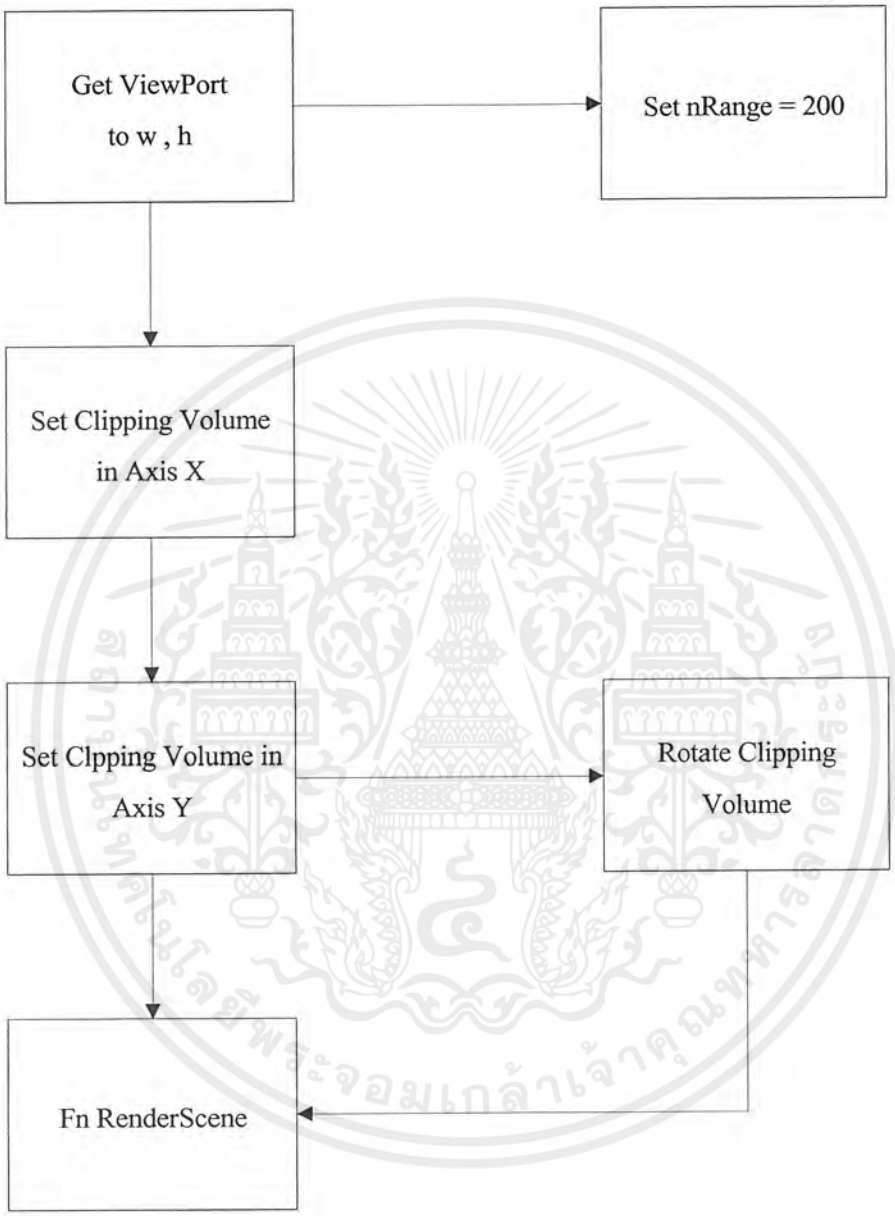
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function CalLightPos



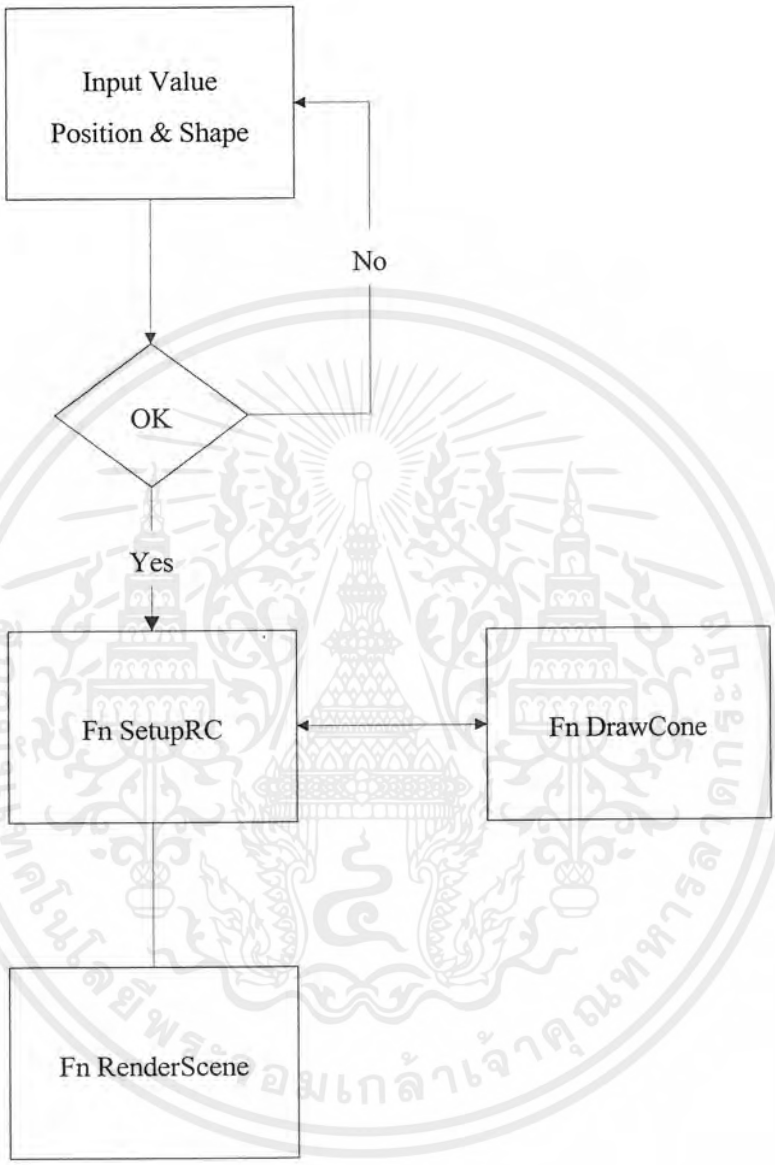
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ChangeSize



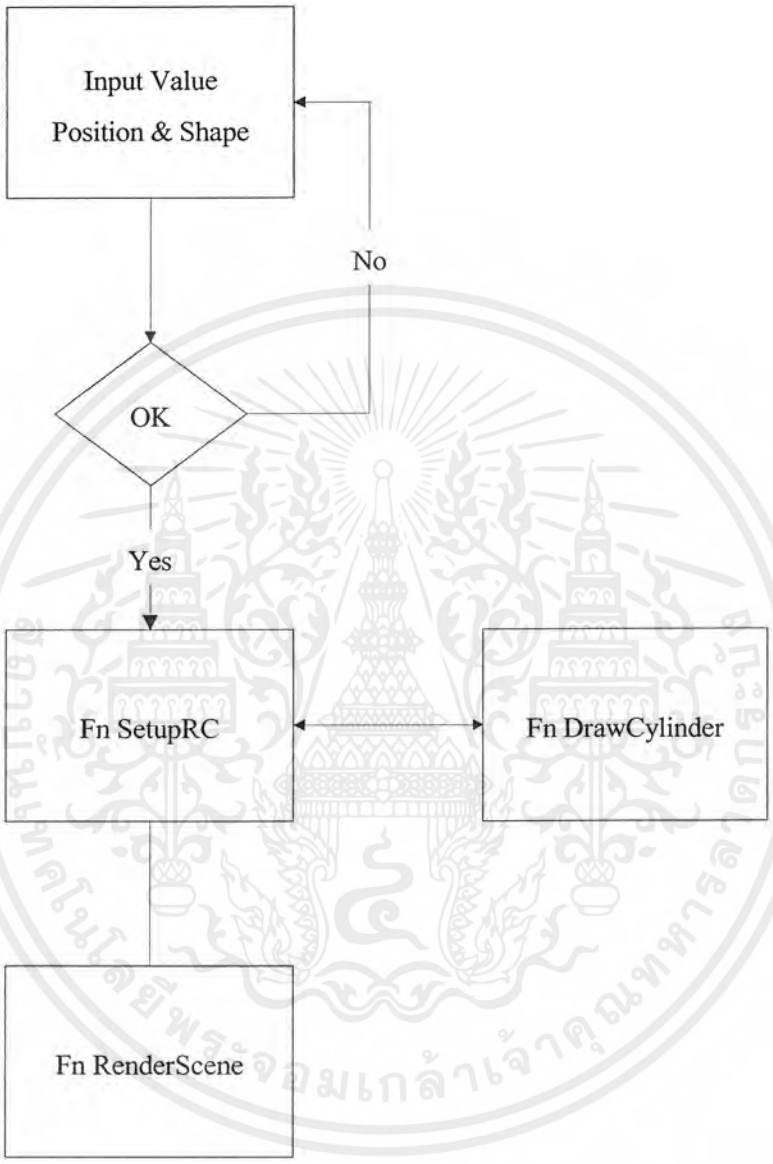
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ConeDlgProc



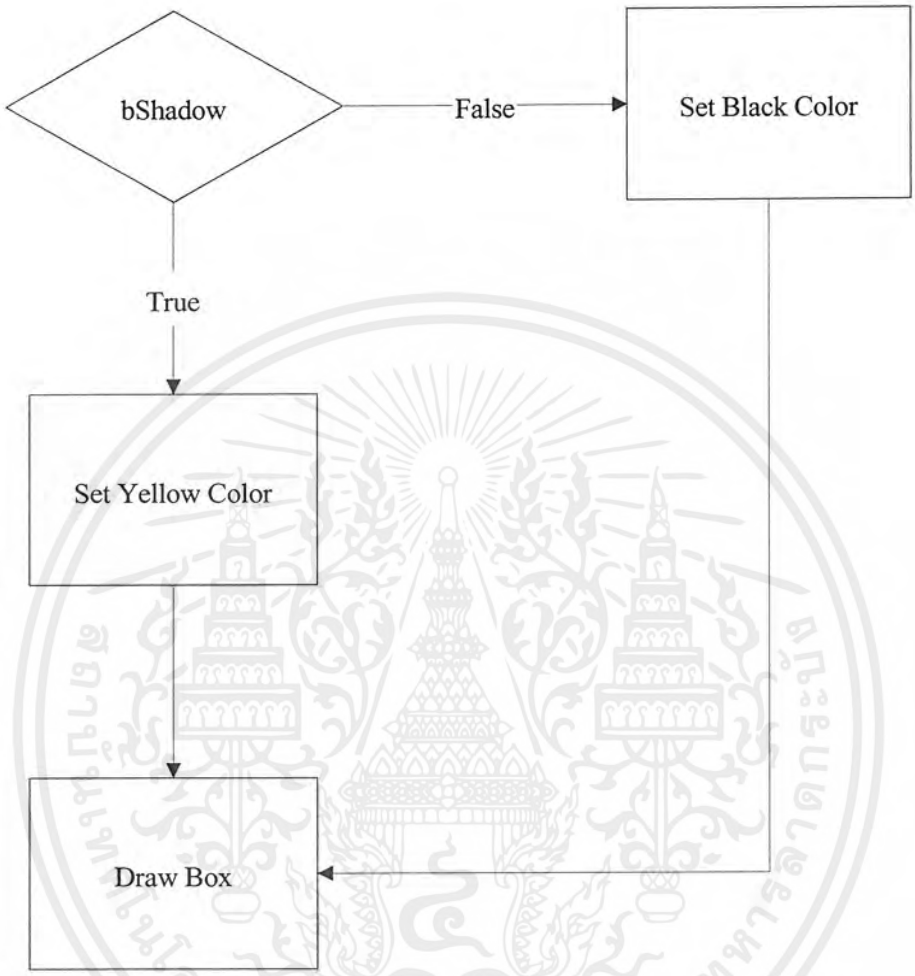
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function CylinderDlgProc



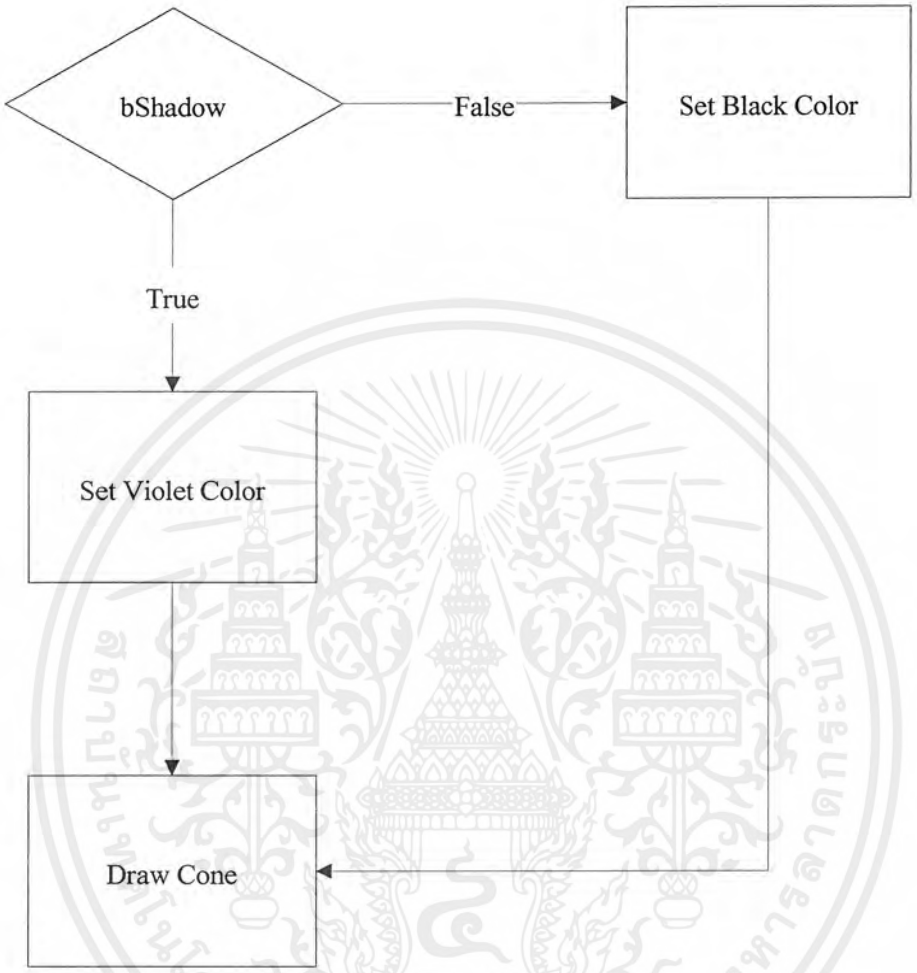
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawBox



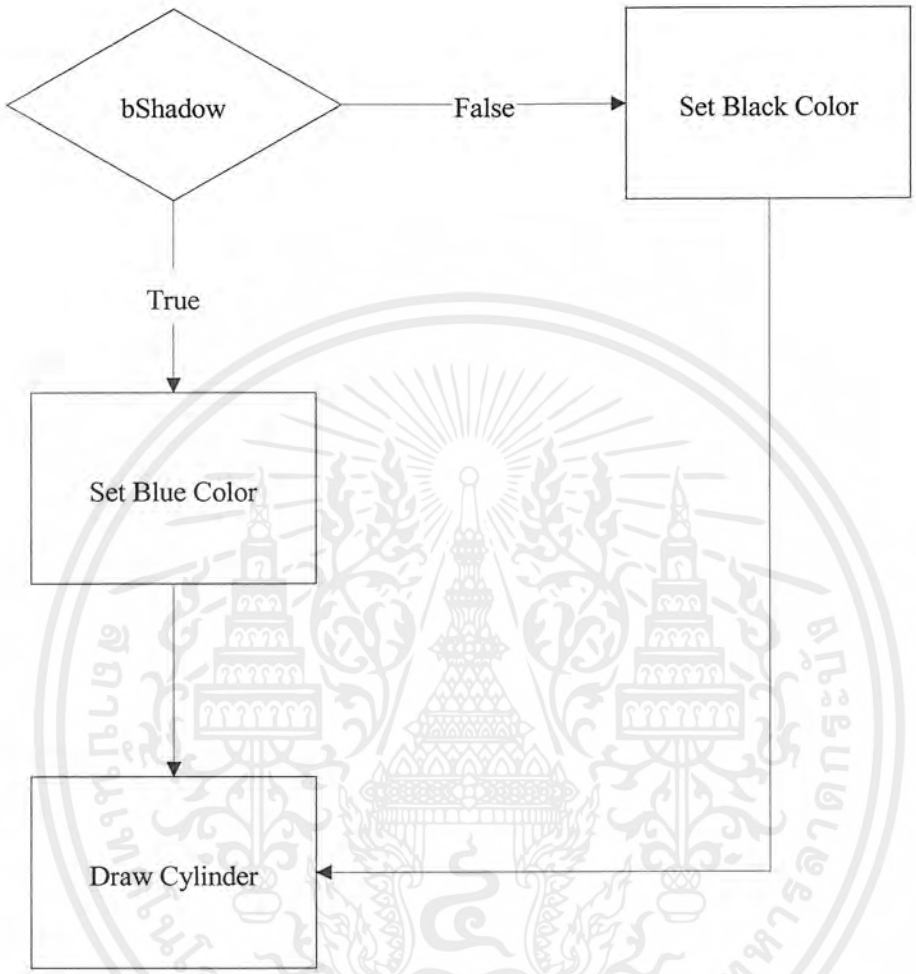
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawCone



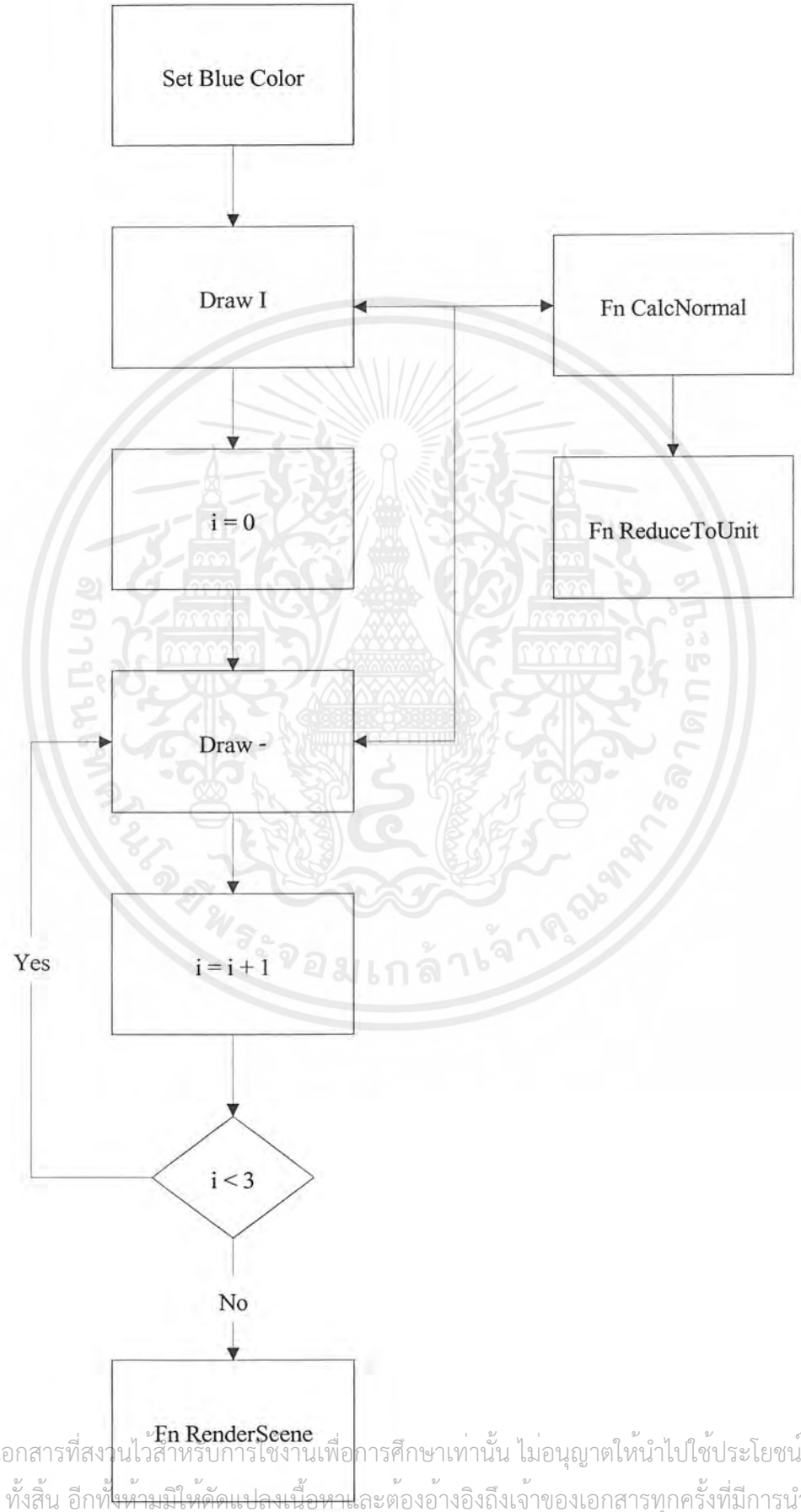
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawCylinder



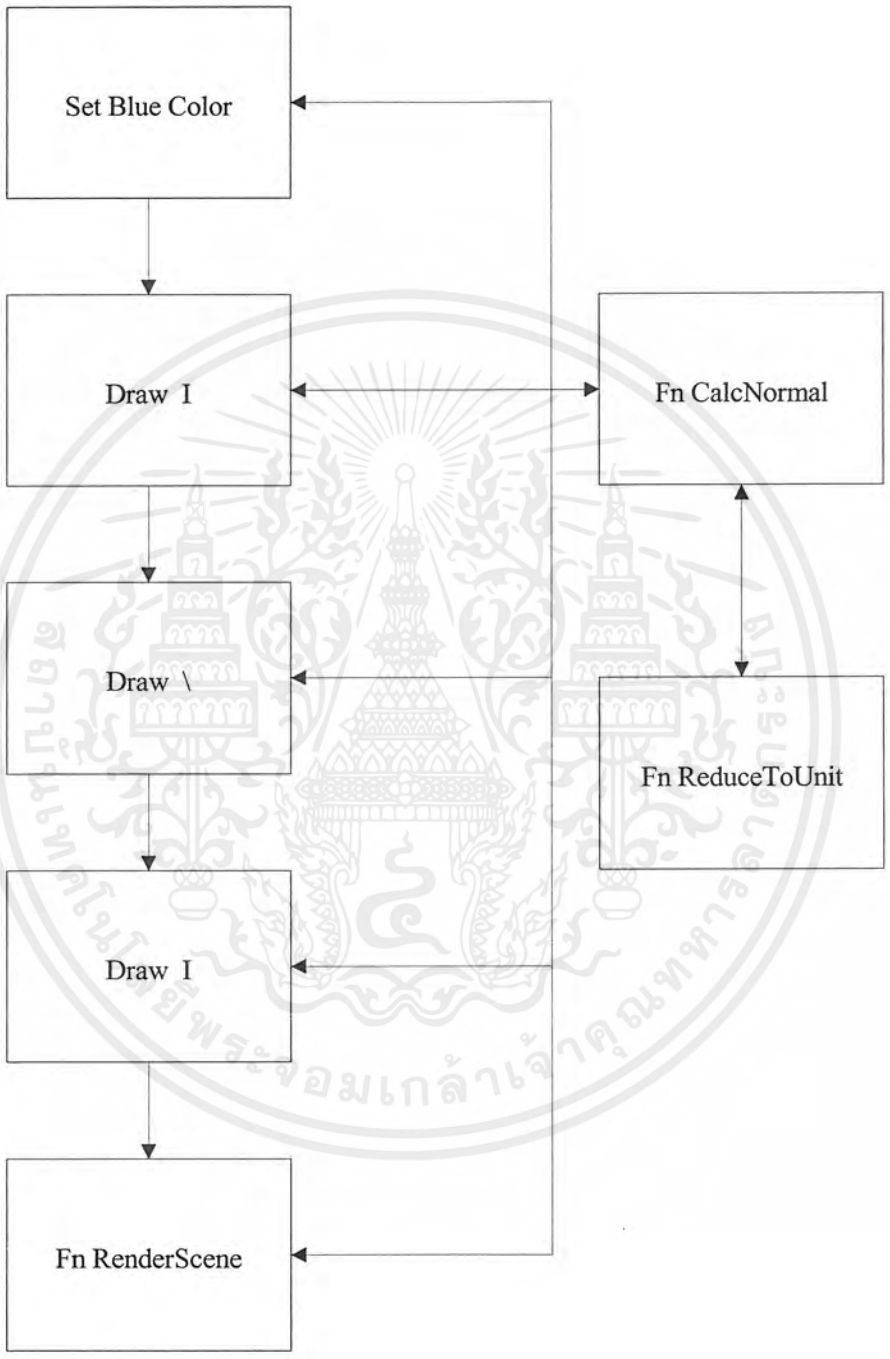
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawEast



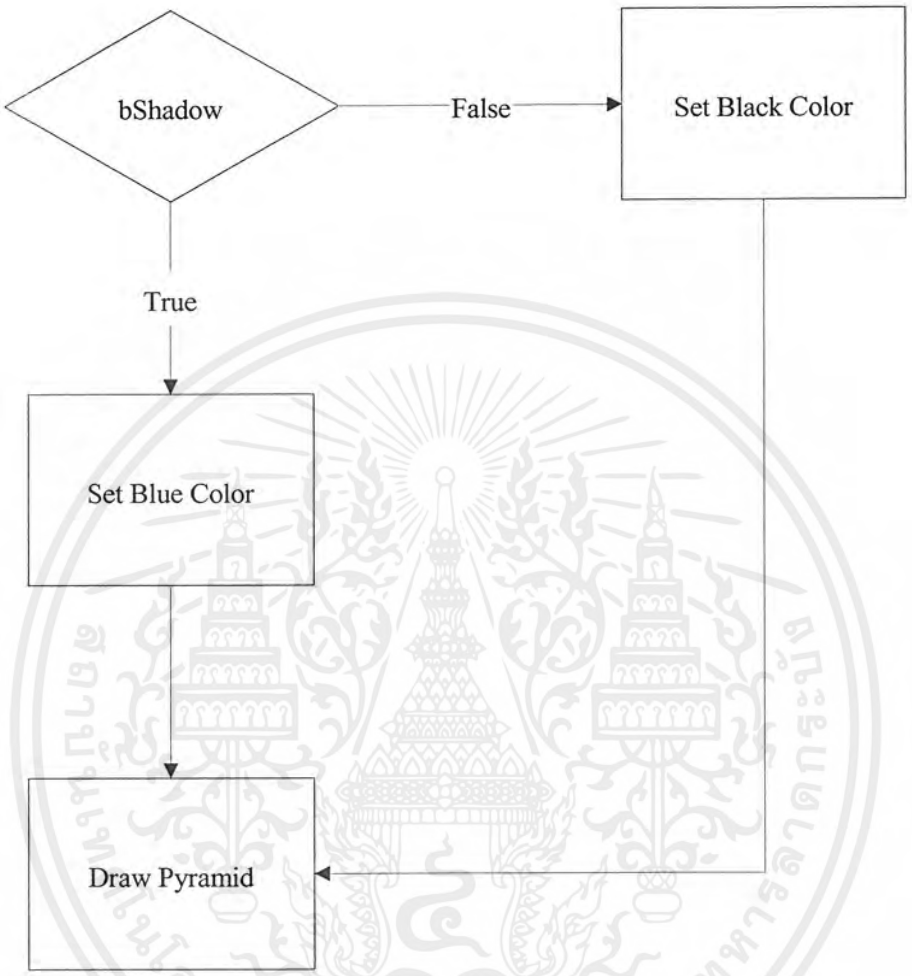
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawNorth



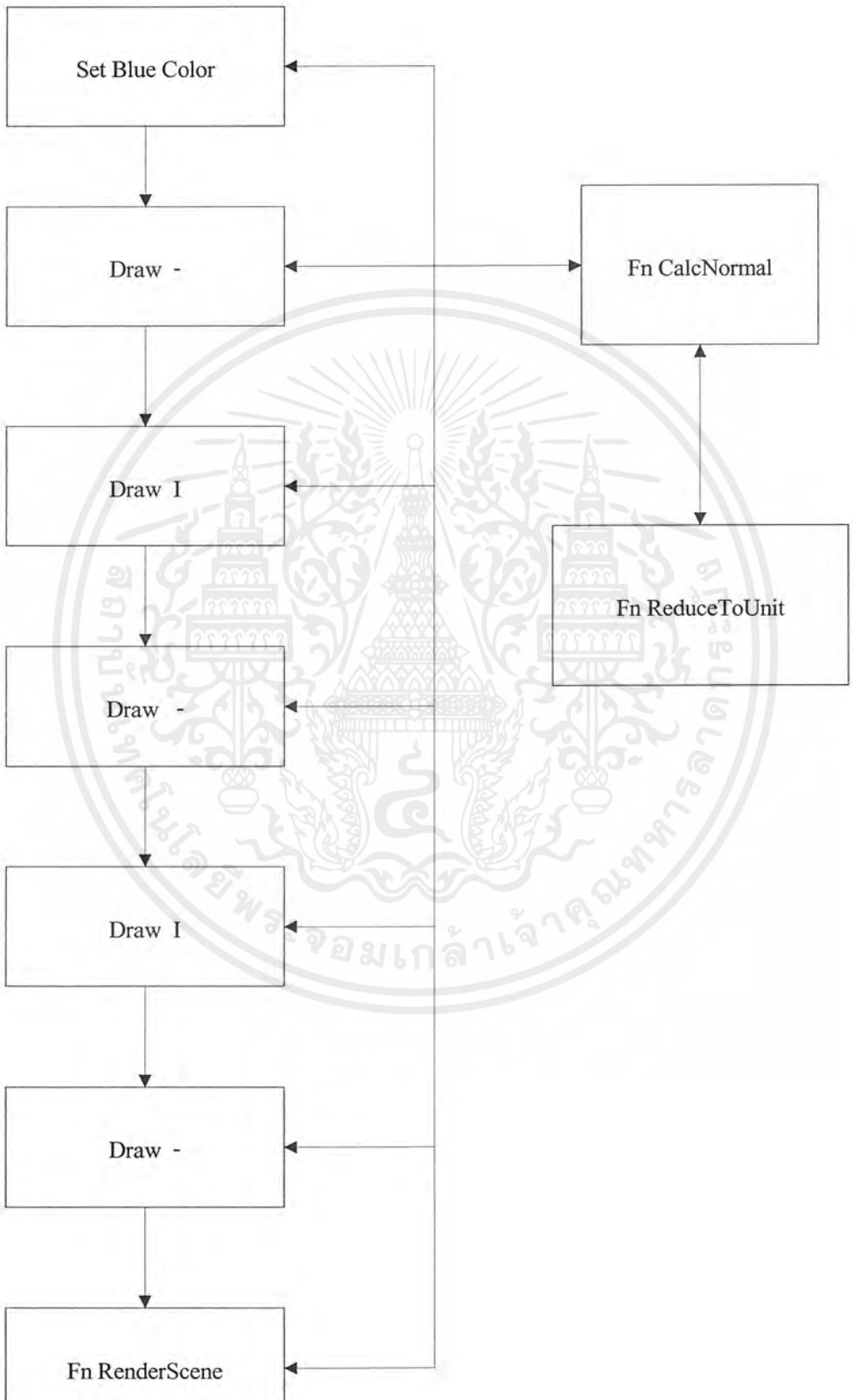
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawPyramid



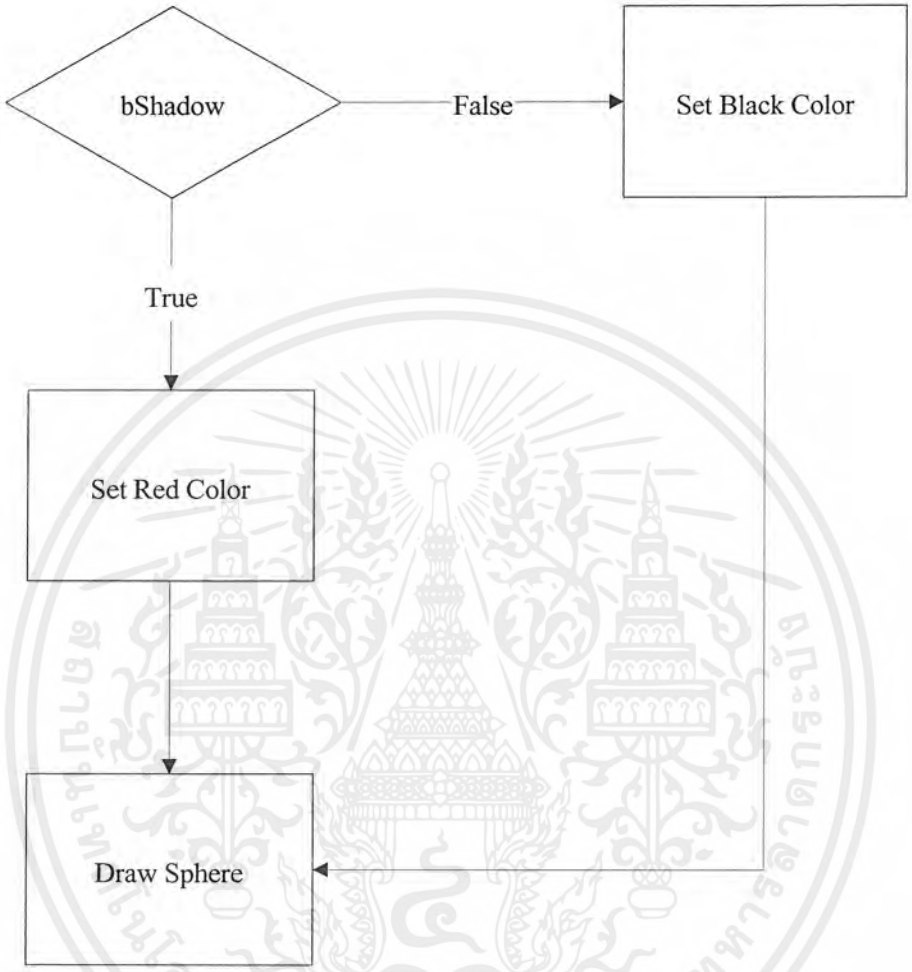
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawSouth



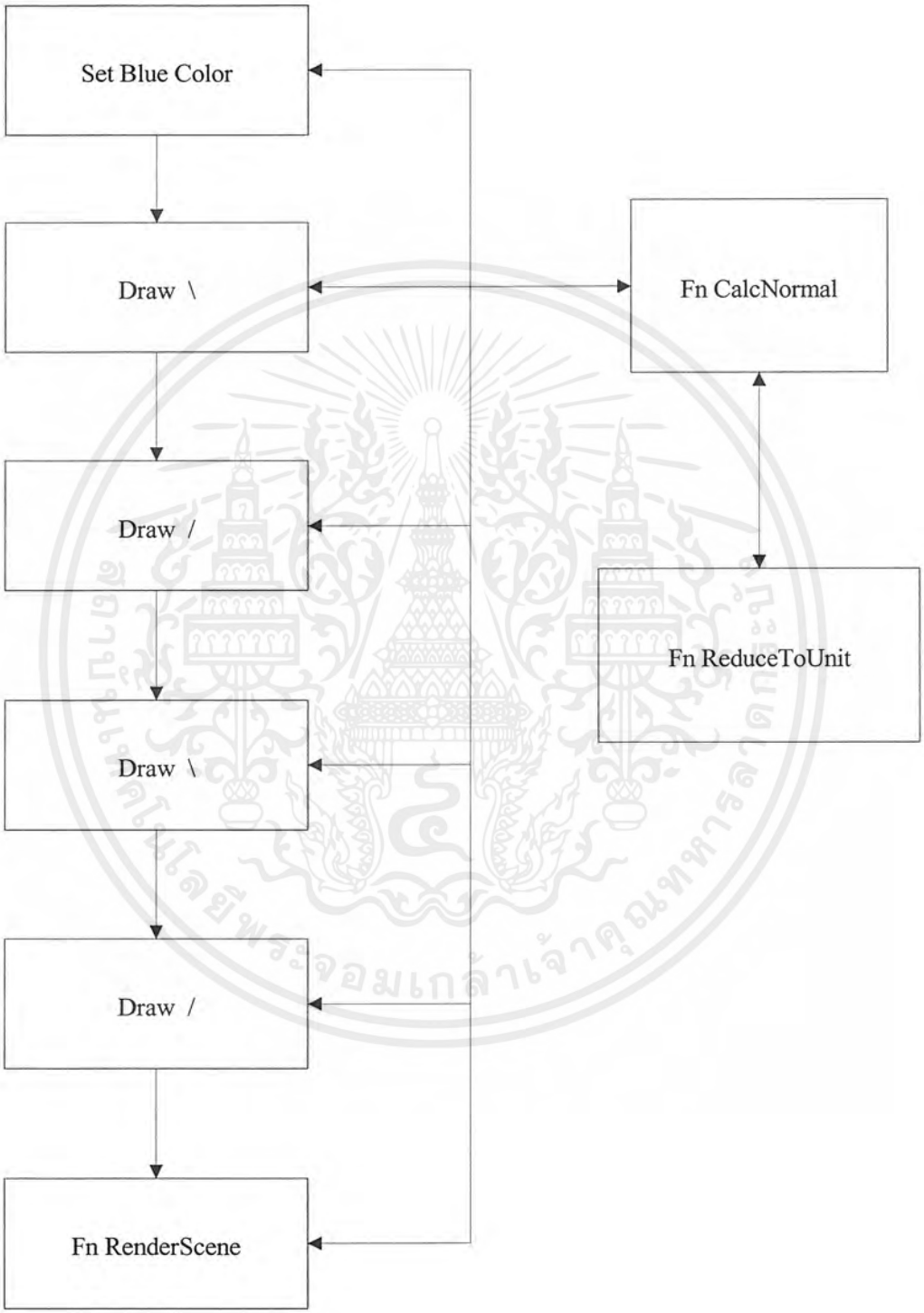
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawSphere



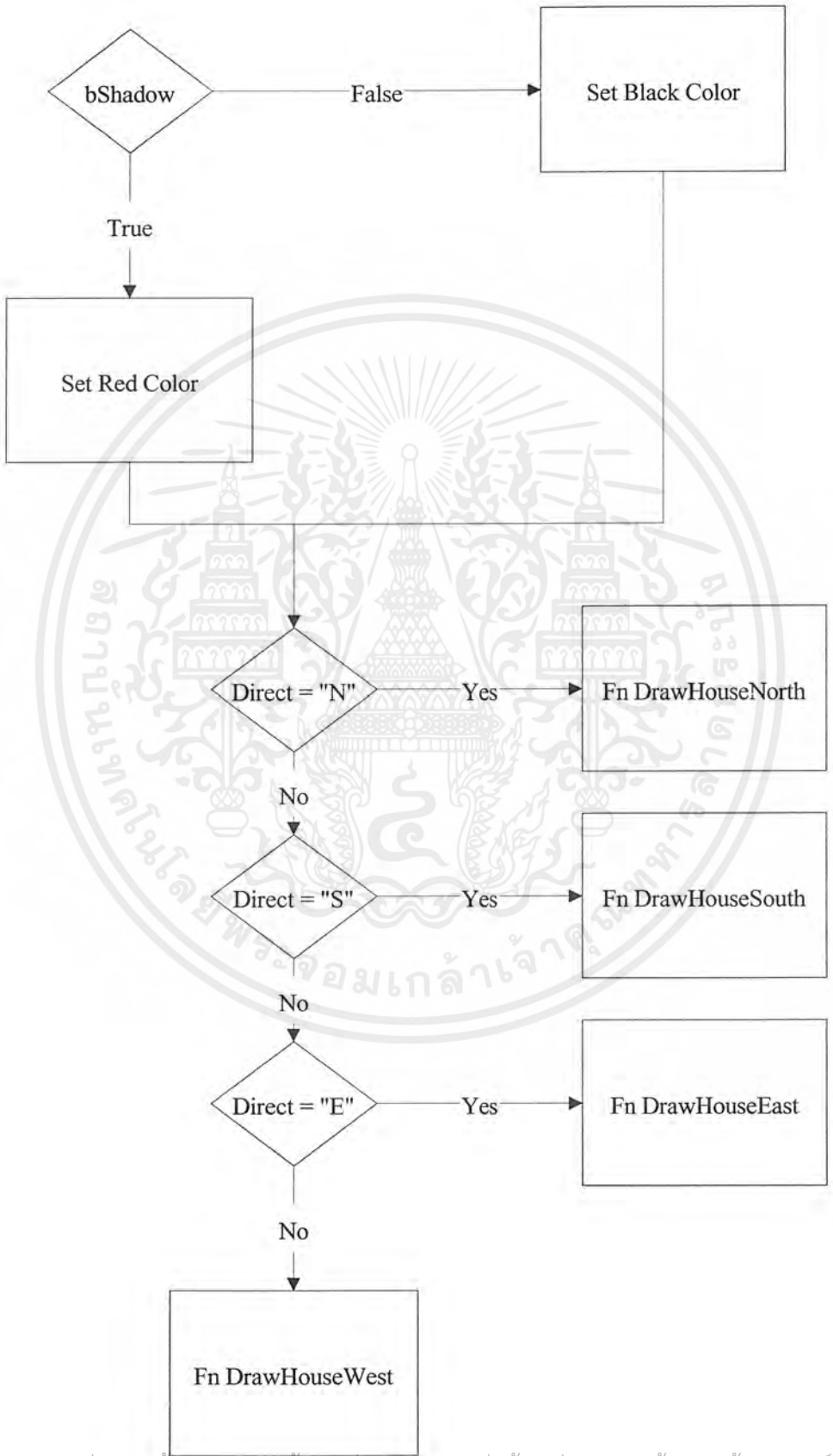
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawWest



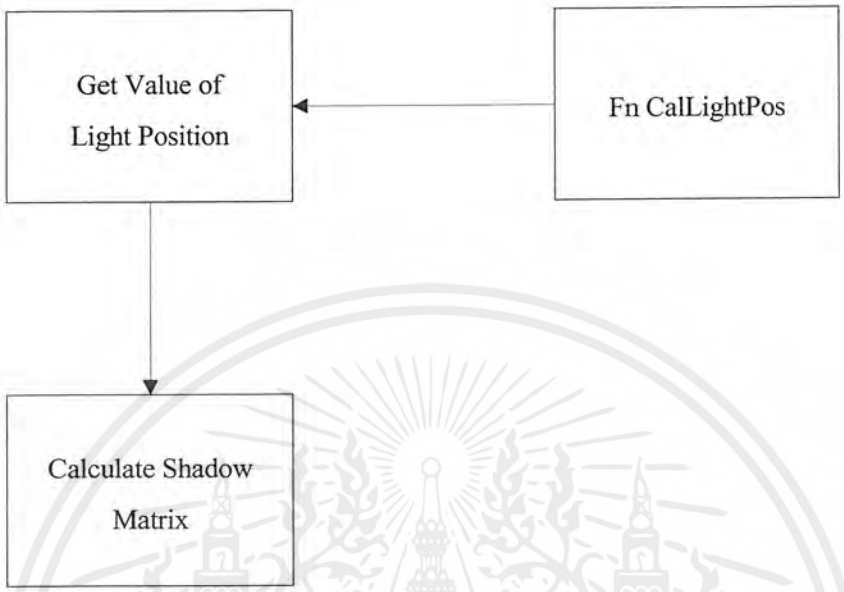
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function HouseLocation



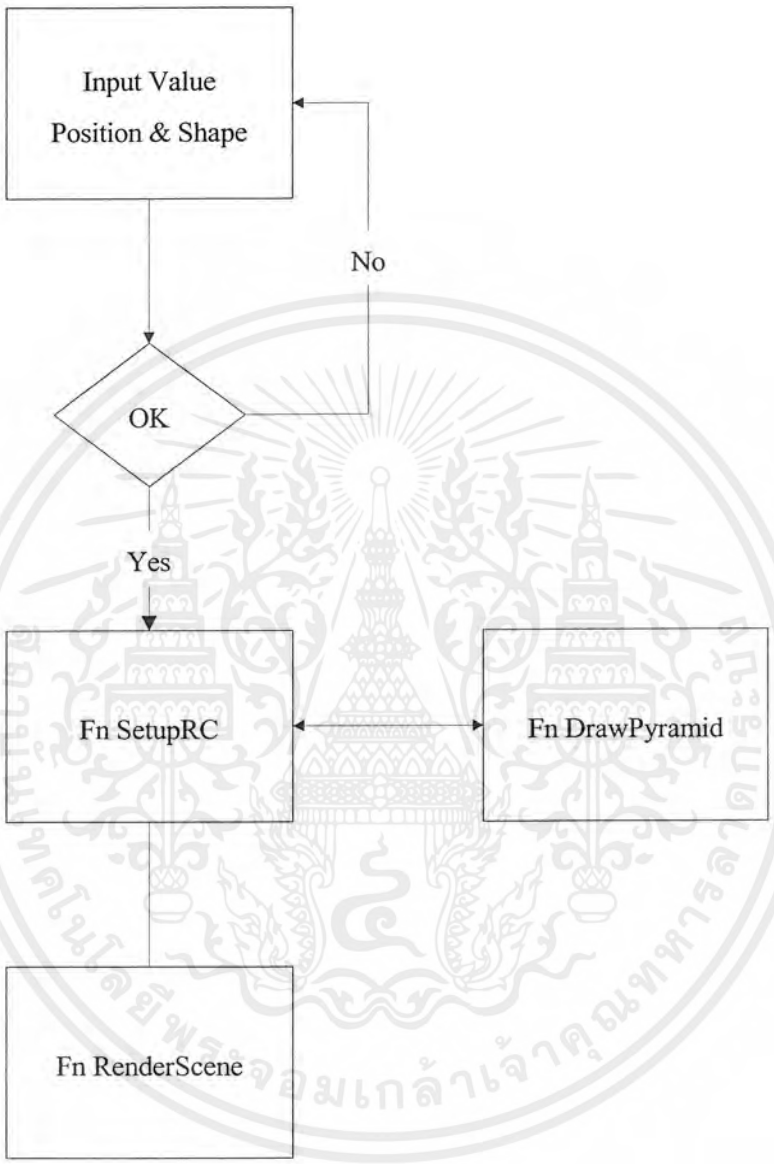
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function MakeShadowMatrix



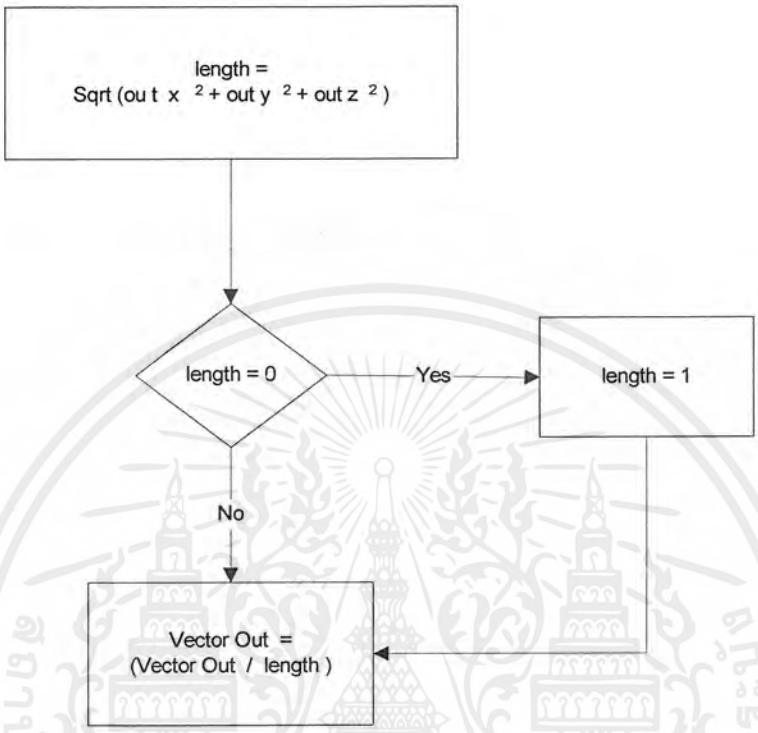
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function PyramidDlgProc



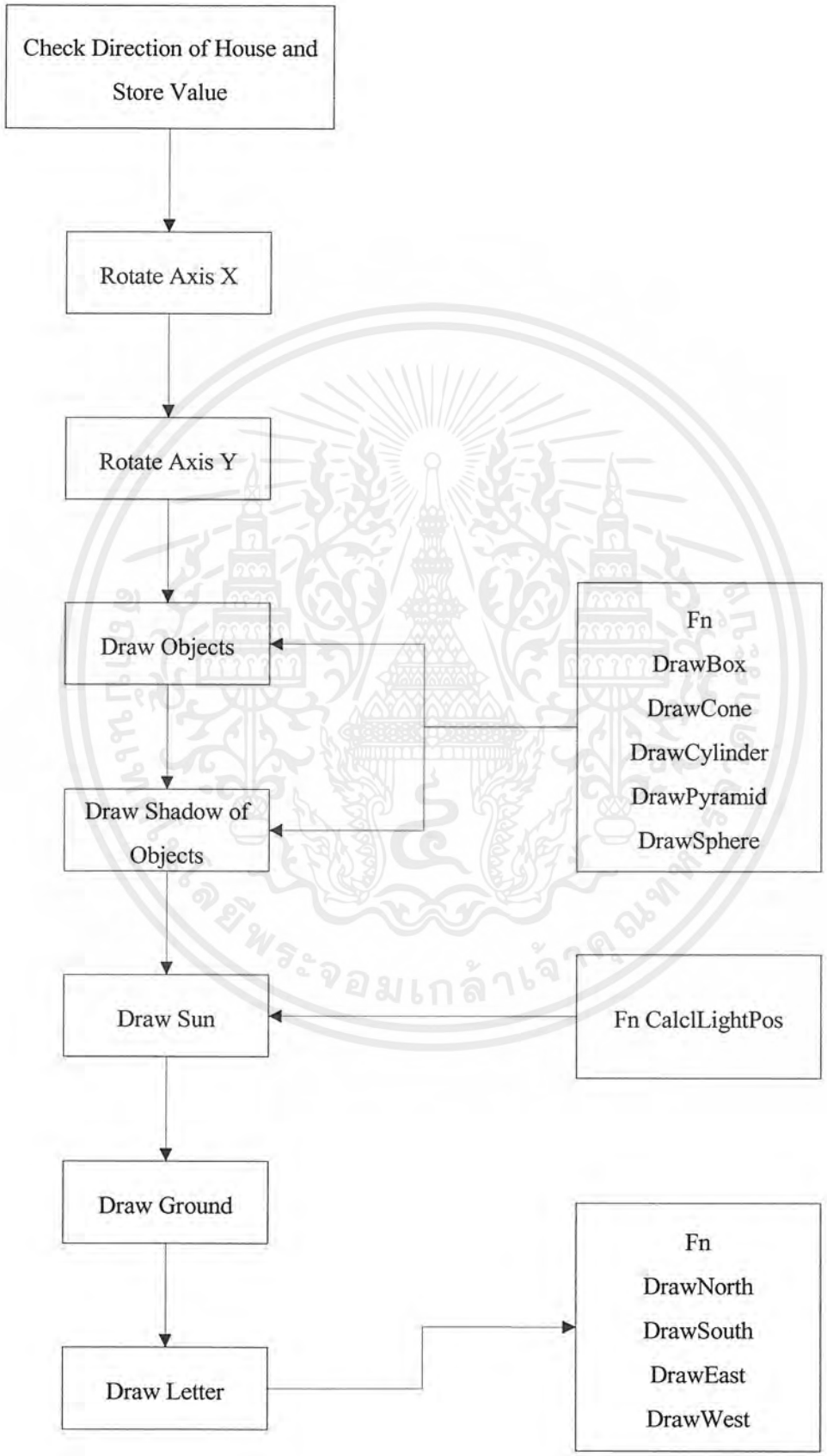
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ReduceToUnit



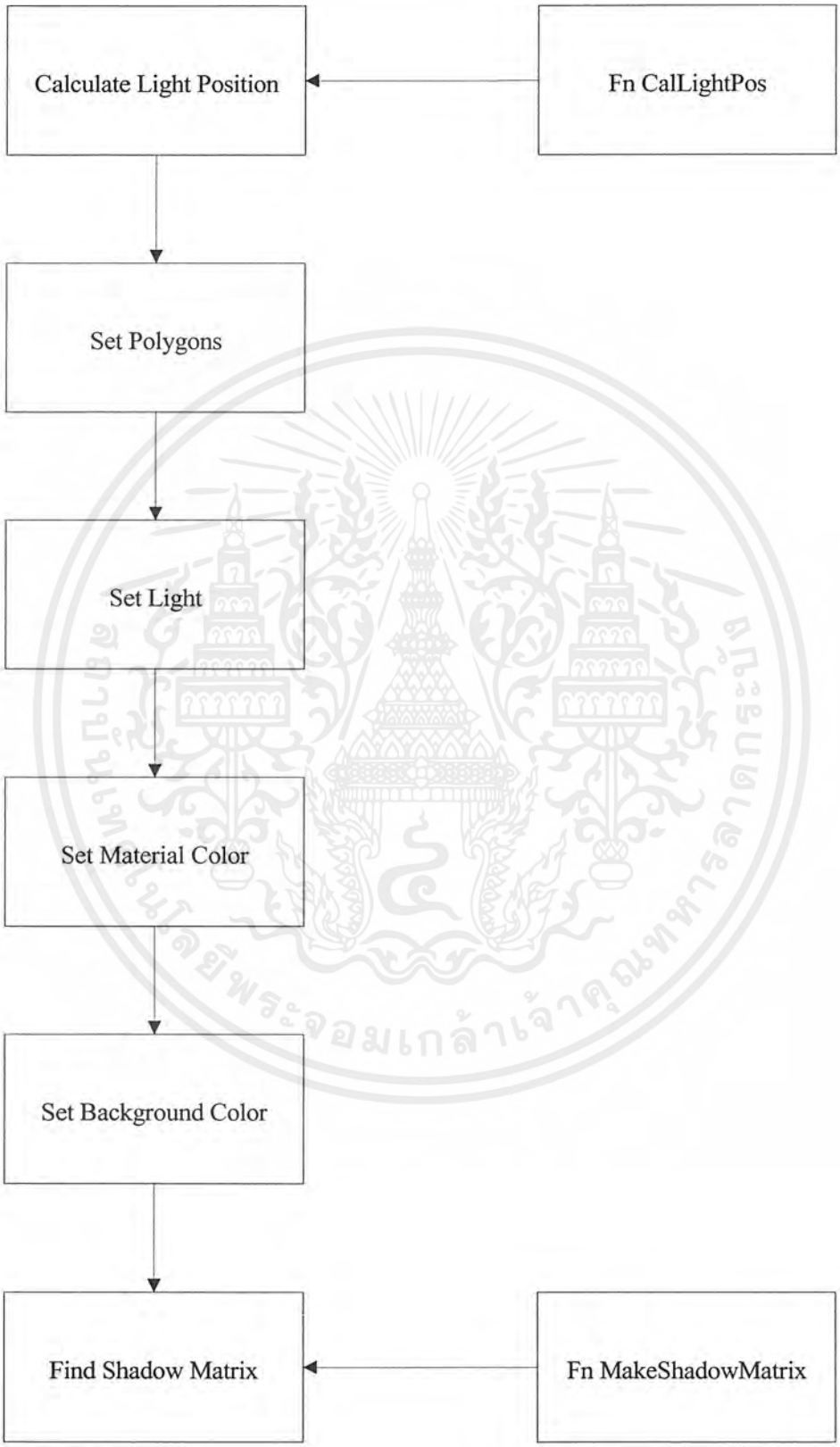
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function RenderScene



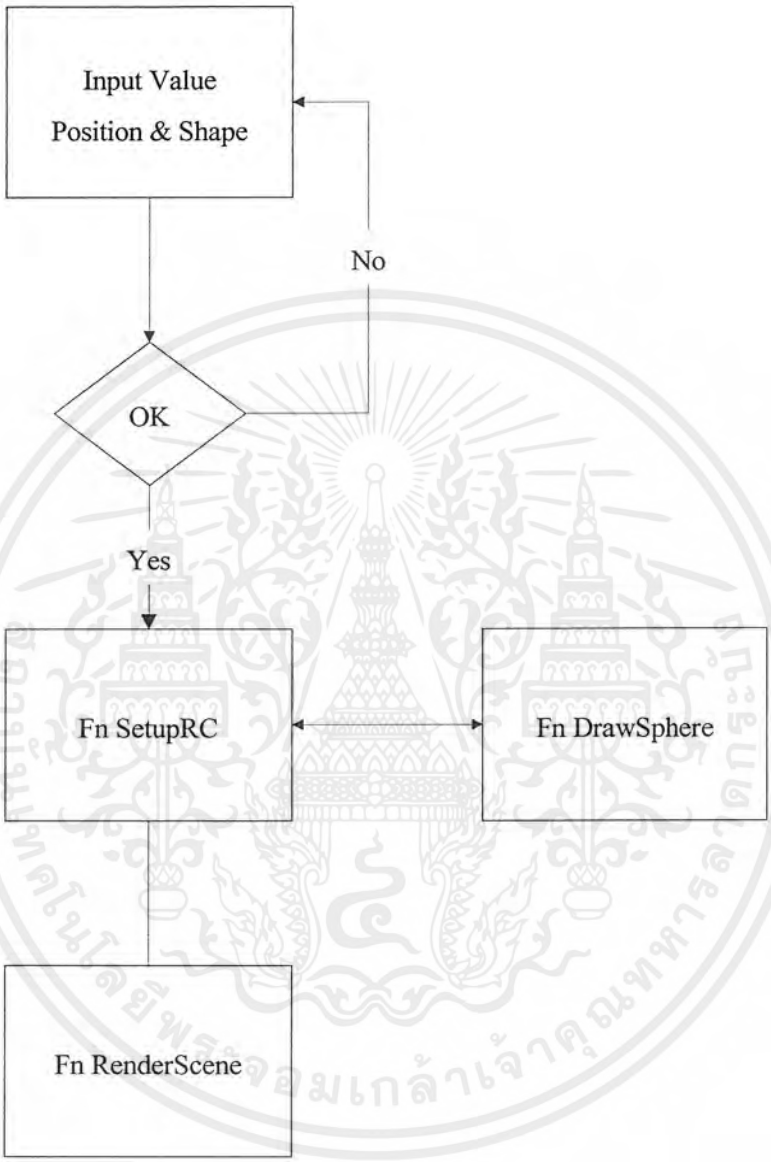
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function SetupRC



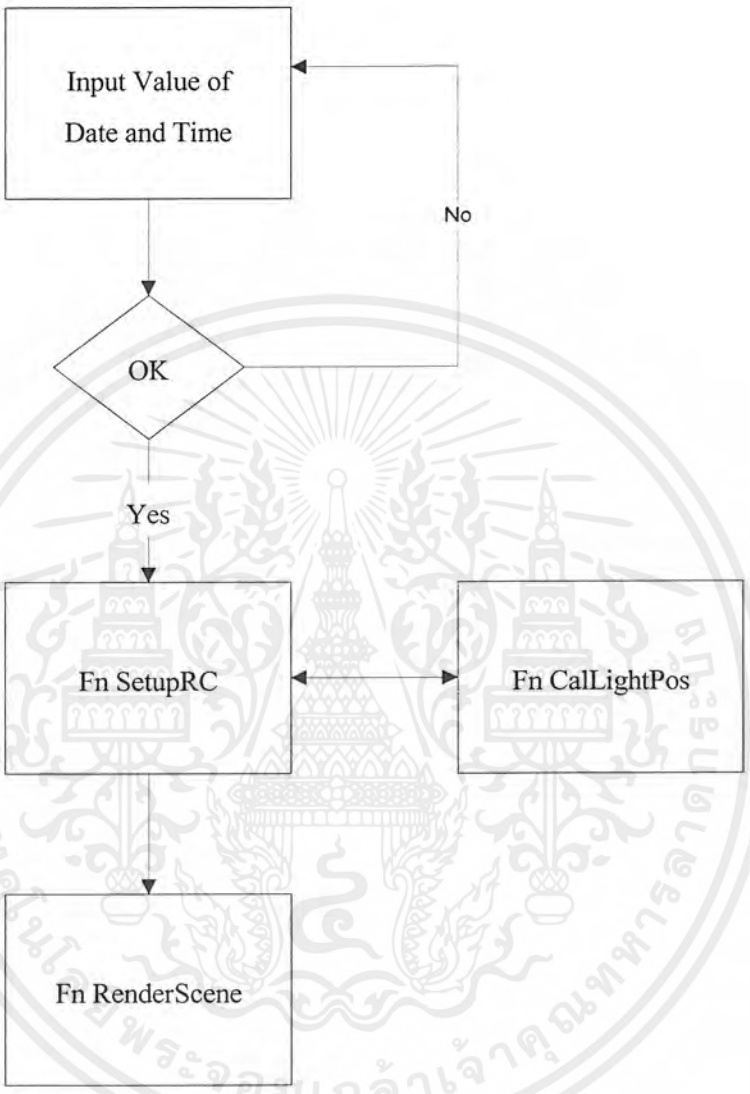
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function SphereDlgProc



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function TimeDlgProc



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Solar.c

```
#include <windows.h>
#include <gl\glu.h>
#include <gl\glaux.h>
#include <math.h>
#include "resource.h"

// Set color
#define glRGB(x, y, z)  glColor3ub((GLubyte)x, (GLubyte)y, (GLubyte)z)

// Define a constant for the value of PI
#define pi 3.1415f

// Color Palette handle
HPALETTE hPalette = NULL;

// Application name and instance storeage

static LPCTSTR lpszAppName = "Solar Version 1.2";
static HINSTANCE hInstance;

// Get value
unsigned int Day = 0;
unsigned int Month = 0 ;
unsigned int Hour = 0;
unsigned int Min = 0;

// Define variable for BoxX - Box10X
int BoxX=0,BoxY=0,BoxZ=0,BoxWidth=0,BoxHeight=0,BoxLength=0 ;
int Box2X=0,Box2Y=0,Box2Z=0,Box2Width=0,Box2Height=0,Box2Length=0;
int Box3X=0,Box3Y=0,Box3Z=0,Box3Width=0,Box3Height=0,Box3Length=0;
int Box4X=0,Box4Y=0,Box4Z=0,Box4Width=0,Box4Height=0,Box4Length=0;
int Box5X=0,Box5Y=0,Box5Z=0,Box5Width=0,Box5Height=0,Box5Length=0;
int Box6X=0,Box6Y=0,Box6Z=0,Box6Width=0,Box6Height=0,Box6Length=0;
int Box7X=0,Box7Y=0,Box7Z=0,Box7Width=0,Box7Height=0,Box7Length=0;
int Box8X=0,Box8Y=0,Box8Z=0,Box8Width=0,Box8Height=0,Box8Length=0;
int Box9X=0,Box9Y=0,Box9Z=0,Box9Width=0,Box9Height=0,Box9Length=0;
int Box10X=0,Box10Y=0,Box10Z=0,Box10Width=0,Box10Height=0,Box10Length=0;

// Define variable for ConeX - Cone10X
int ConeX=0,ConeY=0,ConeZ=0,ConeRadius=0,ConeHeight=0;
int Cone2X=0,Cone2Y=0,Cone2Z=0,Cone2Radius=0,Cone2Height=0;
int Cone3X=0,Cone3Y=0,Cone3Z=0,Cone3Radius=0,Cone3Height=0;
int Cone4X=0,Cone4Y=0,Cone4Z=0,Cone4Radius=0,Cone4Height=0;
int Cone5X=0,Cone5Y=0,Cone5Z=0,Cone5Radius=0,Cone5Height=0;
int Cone6X=0,Cone6Y=0,Cone6Z=0,Cone6Radius=0,Cone6Height=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int Cone7X=0,Cone7Y=0,Cone7Z=0,Cone7Radius=0,Cone7Height=0;
int Cone8X=0,Cone8Y=0,Cone8Z=0,Cone8Radius=0,Cone8Height=0;
int Cone9X=0,Cone9Y=0,Cone9Z=0,Cone9Radius=0,Cone9Height=0;
int Cone10X=0,Cone10Y=0,Cone10Z=0,Cone10Radius=0,Cone10Height=0;

// Define variable for CylinderX - Cylinder10X
int CylinderX=0,CylinderY=0,CylinderZ=0,CylinderRadius=0,CylinderHeight=0;
int Cylinder2X=0,Cylinder2Y=0,Cylinder2Z=0,Cylinder2Radius=0,Cylinder2Height=0;
int Cylinder3X=0,Cylinder3Y=0,Cylinder3Z=0,Cylinder3Radius=0,Cylinder3Height=0;
int Cylinder4X=0,Cylinder4Y=0,Cylinder4Z=0,Cylinder4Radius=0,Cylinder4Height=0;
int Cylinder5X=0,Cylinder5Y=0,Cylinder5Z=0,Cylinder5Radius=0,Cylinder5Height=0;
int Cylinder6X=0,Cylinder6Y=0,Cylinder6Z=0,Cylinder6Radius=0,Cylinder6Height=0;
int Cylinder7X=0,Cylinder7Y=0,Cylinder7Z=0,Cylinder7Radius=0,Cylinder7Height=0;
int Cylinder8X=0,Cylinder8Y=0,Cylinder8Z=0,Cylinder8Radius=0,Cylinder8Height=0;
int Cylinder9X=0,Cylinder9Y=0,Cylinder9Z=0,Cylinder9Radius=0,Cylinder9Height=0;
int Cylinder10X=0,Cylinder10Y=0,Cylinder10Z=0,Cylinder10Radius=0,Cylinder10Height=0;

// Define variable for PyramidX - Pyramid10X
int PyramidX=0,PyramidY=0,PyramidZ=0,PyramidWidth=0,PyramidHeight=0;
int Pyramid2X=0,Pyramid2Y=0,Pyramid2Z=0,Pyramid2Width=0,Pyramid2Height=0;
int Pyramid3X=0,Pyramid3Y=0,Pyramid3Z=0,Pyramid3Width=0,Pyramid3Height=0;
int Pyramid4X=0,Pyramid4Y=0,Pyramid4Z=0,Pyramid4Width=0,Pyramid4Height=0;
int Pyramid5X=0,Pyramid5Y=0,Pyramid5Z=0,Pyramid5Width=0,Pyramid5Height=0;
int Pyramid6X=0,Pyramid6Y=0,Pyramid6Z=0,Pyramid6Width=0,Pyramid6Height=0;
int Pyramid7X=0,Pyramid7Y=0,Pyramid7Z=0,Pyramid7Width=0,Pyramid7Height=0;
int Pyramid8X=0,Pyramid8Y=0,Pyramid8Z=0,Pyramid8Width=0,Pyramid8Height=0;
int Pyramid9X=0,Pyramid9Y=0,Pyramid9Z=0,Pyramid9Width=0,Pyramid9Height=0;
int Pyramid10X=0,Pyramid10Y=0,Pyramid10Z=0,Pyramid10Width=0,Pyramid10Height=0;

//  Define variable for SphereX - Sphere10X
int SphereX=0,SphereY=0,SphereZ=0,SphereRadius=0;
int Sphere2X=0,Sphere2Y=0,Sphere2Z=0,Sphere2Radius=0;
int Sphere3X=0,Sphere3Y=0,Sphere3Z=0,Sphere3Radius=0;
int Sphere4X=0,Sphere4Y=0,Sphere4Z=0,Sphere4Radius=0;
int Sphere5X=0,Sphere5Y=0,Sphere5Z=0,Sphere5Radius=0;
int Sphere6X=0,Sphere6Y=0,Sphere6Z=0,Sphere6Radius=0;
int Sphere7X=0,Sphere7Y=0,Sphere7Z=0,Sphere7Radius=0;
int Sphere8X=0,Sphere8Y=0,Sphere8Z=0,Sphere8Radius=0;
int Sphere9X=0,Sphere9Y=0,Sphere9Z=0,Sphere9Radius=0;
int Sphere10X=0,Sphere10Y=0,Sphere10Z=0,Sphere10Radius=0;

// Set direction of house's direction
BOOL North, South ,West ,East;

// Set ground location on axis Y
GLfloat y = -30.0f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Rotation
static GLfloat xRot = 0.0f;
static GLfloat yRot = 0.0f;

// Light values and coordinates
GLfloat  ambientLight[] = { 0.3f, 0.3f, 0.3f, 1.0f };
GLfloat  diffuseLight[] = { 0.7f, 0.7f, 0.7f, 1.0f };
GLfloat  specular[] = { 1.0f, 1.0f, 1.0f, 1.0f};
GLfloat  specref[] =  { 1.0f, 1.0f, 1.0f, 1.0f };
GLfloat    lightPos[4];

//Set direction
char direction = 'o';

// Transformation matrix to project shadow
GLfloat shadowMat[4][4], shadowMatDirect[4][4];

// Declaration for Window procedure
LRESULT CALLBACK WndProc(   HWND hWnd,UINT message,
                            WPARAM wParam,LPARAM lParam);

// Dialog Procedure for Time box
BOOL APIENTRY TimeDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);

// Dialog Procedure for Box box - Box10 box
BOOL APIENTRY BoxDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box2DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box3DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box4DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box5DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box6DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box7DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box8DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box9DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Box10DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);

// Dialog Procedure for Cone box - Cone10 box
BOOL APIENTRY ConeDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone2DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone3DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone4DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone5DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone6DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone7DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone8DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone9DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cone10DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Dialog Procedure for Cylinder box - Cylinder10 box
BOOL APIENTRY CylinderDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder2DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder3DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder4DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder5DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder6DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder7DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder8DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder9DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Cylinder10DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);

// Dialog Procedure for Pyramid box - Pyramid10 box
BOOL APIENTRY PyramidDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid2DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid3DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid4DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid5DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid6DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid7DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid8DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid9DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Pyramid10DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);

// Dialog Procedure for Sphere box - Sphere10 box
BOOL APIENTRY SphereDlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere2DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere3DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere4DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere5DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere6DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere7DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere8DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere9DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);
BOOL APIENTRY Sphere10DlgProc (HWND hDlg,UINT message,UINT wParam,LONG lParam);

// Dialog procedure for about box
BOOL APIENTRY AboutDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam);

// Set Pixel Format function - forward declaration
void SetDCPixelFormat(HDC hDC);

// Calculate Light Position
void CalLightPos(GLfloat light[4])
{
    int month1[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int date, month, i, check,n ,total_min;
float Latitude, n1, time;
float declination, hourangle, altitude, azimuth, radian;

//set date month time
date = Day;
month = Month;
time = (float)(Hour + (Min/60.0f));
total_min = (Hour*60) + Min;
n=0;

// result is radian
Latitude = 14.0f*pi/180.0f;
for(i=1;i<month;i++)
    n+= month1[i];
n = n+date;
check=n;
radian = (284.0f+n) * 2.0f * pi / 365.0f;

// result is degree
declination = 23.45f*(float)sin(radian);

// degree to radian
declination = declination * pi/180.0f;
n1 = 0.25f * (time-12.0f) * 60.0f;

// degree to radian
hourangle = n1 * pi / 180.0f;

// result is radian
altitude = (float)asin( ( sin(Latitude) * sin(declination) ) + ( cos(Latitude) * cos(declination) * cos
(hourangle) ) );
azimuth = (float)asin( cos(declination) * ( sin(hourangle) / cos(altitude) ) );

// radian to degree
altitude = altitude*180.0f/(pi);
azimuth = azimuth*180.0f/(pi);

// set Positive angle
// if ((check>=81.0f) && (check<=263.0f))
//   azimuth = 180.0f - azimuth;

//if (check>263)
    //if ((time>=1.0f ) && (time<=12.0f))
    //   azimuth = 360.0f + azimuth;
if ((total_min) > 720.0)
    azimuth =azimuth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* #######  (01JAN-21MAR) & (21SEP-31DEC) ####### */
else if ( (check<81.0)||(check>263.0) )
        azimuth = 360.0f + azimuth;

/* ############### (24APR-18AUG) ################ */
else if ( (check>113.0)&&(check<231.0) )
        azimuth = 180.0f-azimuth;

/* ######### (22MAR-23APR) & (19AUG-20SEP) ######### */
else if ( (check>=81.0)&&(check<=113.0) || (check>=231.0)&&(check<=263.0) )
        {
        switch(check)
            {
            case 81:
                if (total_min>361)
                    azimuth = 360.0f+azimuth;
                else
                    azimuth = 180.0f-azimuth;
                break;

            case 82:
                if (total_min>368)
                    azimuth = 360.0f+azimuth;
                else
                    azimuth = 180.0f-azimuth;
                break;

            case 83:
                if (total_min > 374)
                    azimuth = 360.0f+azimuth;
                else
                    azimuth = 180.0f-azimuth;
                break;

            case 84:
                if (total_min > 381)
                    azimuth = 360.0f+azimuth;
                else
                    azimuth = 180.0f-azimuth;
                break;

            case 85:
                if (total_min > 387)
                    azimuth = 360.0f+azimuth;
                else
                    azimuth = 180.0f-azimuth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

    case 86:
        if (total_min > 394)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 87:
        if (total_min > 400)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 88:
    if (total_min>407)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 89:
    if (total_min > 413)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 90:
    if (total_min > 420)
        azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 91:
    if (total_min > 427)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 92:
    if (total_min > 433)
            azimuth = 360.0f+azimuth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
            azimuth = 180.0f-azimuth;
        break;

    case 93:
    if (total_min > 440)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 94:
    if (total_min > 447)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 95:
    if (total_min > 454)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 96:
    if (total_min > 461)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 97:
    if (total_min > 468)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 98:
    if (total_min > 475)
            azimuth = 360+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case 99:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (total_min > 482)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 100:
if (total_min > 489)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 101:
if (total_min > 497)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 102:
if (total_min > 504)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 103:
if (total_min > 512)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 104:
if (total_min > 520)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case 105:
if (total_min > 528)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 106:
if (total_min > 536)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (107):
if (total_min > 545)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (108):
if (total_min > 554)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (109):
if (total_min > 563)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (110):
if (total_min > 573)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (111):
if (total_min > 583)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (112):
if (total_min > 593)
        azimuth = 360.0f+azimuth;
    else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            azimuth = 180.0f-azimuth;
        break;

    case (113):
    if (total_min > 605)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (263):
        if (total_min>361)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (262):
        if (total_min>368)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (261):
    if (total_min > 374)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (260):
    if (total_min > 381)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (259):
    if (total_min > 387)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (258):
    if (total_min > 394)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (257):
    if (total_min > 400)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (256):
    if (total_min>407)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (255):
    if (total_min > 413)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (254):
    if (total_min > 420)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (253):
    if (total_min > 427)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (252):
    if (total_min > 433)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case (251):
if (total_min > 440)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (250):
if (total_min > 447)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (249):
if (total_min > 454)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (248):
if (total_min > 461)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (247):
if (total_min > 468)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (246):
if (total_min > 475)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (245):
if (total_min > 482)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    break;

case (244):
if (total_min > 489)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (243):
if (total_min > 497)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (242):
if (total_min > 504)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (241):
if (total_min > 512)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (240):
if (total_min > 520)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (239):
if (total_min > 528)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

case (238):
if (total_min > 536)
        azimuth = 360.0f+azimuth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
            azimuth = 180.0f-azimuth;
        break;

    case (237):
    if (total_min > 545)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (236):
    if (total_min > 554)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (235):
    if (total_min > 563)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (234):
    if (total_min > 573)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (233):
    if (total_min > 583)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (232):
    if (total_min > 593)
            azimuth = 360.0f+azimuth;
        else
            azimuth = 180.0f-azimuth;
        break;

    case (231):
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (total_min > 605)
                azimuth = 360.0f+azimuth;
            else
                azimuth = 180.0f-azimuth;
            break;

        } /* ### END SWITCH ### */
    }

    // degree to radian
    altitude = altitude*pi/180.0f;
    azimuth = azimuth*pi/180.0f;
    light[0] = -250.0f * (float)(cos(altitude) * cos(azimuth));
    light[1] = 250.0f * (float)sin(altitude);
    light[2] = -250.0f * (float)(cos(altitude) * sin(azimuth));
    light[3] = 0.0f;
}

// Reduces a normal vector specified as a set of three coordinates,
// to a unit normal vector of length one.
void ReduceToUnit(float vector[3])
{
    float length;

    // Calculate the length of the vector
    length = (float)sqrt((vector[0]*vector[0]) +
                         (vector[1]*vector[1]) +
                         (vector[2]*vector[2]));

    // Keep the program from blowing up by providing an exceptable
    // value for vectors that may calculated too close to zero.
    if(length == 0.0f)
        length = 1.0f;

    // Dividing each element by the length will result in a
    // unit normal vector.
    vector[0] /= length;
    vector[1] /= length;
    vector[2] /= length;
}

// Points p1, p2, & p3 specified in counter clock-wise order
void calcNormal(float v[3][3], float out[3])
{
    float v1[3],v2[3];
    static const int x = 0;
    static const int y = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    static const int z = 2;

    // Calculate two vectors from the three points
    v1[x] = v[0][x] - v[1][x];
    v1[y] = v[0][y] - v[1][y];
    v1[z] = v[0][z] - v[1][z];
    v2[x] = v[1][x] - v[2][x];
    v2[y] = v[1][y] - v[2][y];
    v2[z] = v[1][z] - v[2][z];

    // Take the cross product of the two vectors to get
    // the normal vector which will be stored in out
    out[x] = v1[y]*v2[z] - v1[z]*v2[y];
    out[y] = v1[z]*v2[x] - v1[x]*v2[z];
    out[z] = v1[x]*v2[y] - v1[y]*v2[x];

    // Normalize the vector (shorten length to one)
    ReduceToUnit(out);
}

// Creates a shadow projection matrix out of the plane equation
// coefficients and the position of the light. The return value is stored
// in destMat[][]
void MakeShadowMatrix(GLfloat points[3][3], GLfloat lightPos[4], GLfloat destMat[4][4])
{
    GLfloat planeCoeff[4];
    GLfloat dot;

    // Find the plane equation coefficients
    // Find the first three coefficients the same way we
    // find a normal.
    calcNormal(points,planeCoeff);

    // Find the last coefficient by back substitutions
    planeCoeff[3] = - (
        (planeCoeff[0]*points[2][0]) + (planeCoeff[1]*points[2][1]) +
        (planeCoeff[2]*points[2][2]));

    // Dot product of plane and light position
    dot = planeCoeff[0] * lightPos[0] +
          planeCoeff[1] * lightPos[1] +
          planeCoeff[2] * lightPos[2] +
          planeCoeff[3] * lightPos[3];

    // Now do the projection
    // First column
   destMat[0][0] = dot - lightPos[0] * planeCoeff[0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    destMat[1][0] = 0.0f - lightPos[0] * planeCoeff[1];
    destMat[2][0] = 0.0f - lightPos[0] * planeCoeff[2];
    destMat[3][0] = 0.0f - lightPos[0] * planeCoeff[3];

    // Second column
    destMat[0][1] = 0.0f - lightPos[1] * planeCoeff[0];
    destMat[1][1] = dot - lightPos[1] * planeCoeff[1];
    destMat[2][1] = 0.0f - lightPos[1] * planeCoeff[2];
    destMat[3][1] = 0.0f - lightPos[1] * planeCoeff[3];

    // Third Column
    destMat[0][2] = 0.0f - lightPos[2] * planeCoeff[0];
    destMat[1][2] = 0.0f - lightPos[2] * planeCoeff[1];
    destMat[2][2] = dot - lightPos[2] * planeCoeff[2];
    destMat[3][2] = 0.0f - lightPos[2] * planeCoeff[3];

    // Fourth Column
    destMat[0][3] = 0.0f - lightPos[3] * planeCoeff[0];
    destMat[1][3] = 0.0f - lightPos[3] * planeCoeff[1];
    destMat[2][3] = 0.0f - lightPos[3] * planeCoeff[2];
    destMat[3][3] = dot - lightPos[3] * planeCoeff[3];
}

// Change viewing volume and viewport.  Called when window is resized
void ChangeSize(GLsizei w, GLsizei h)
{
    GLfloat nRange = 200.0f;

    // Prevent a divide by zero
    if(h == 0)
        h = 1;

    // Set Viewport to window dimensions
    glViewport(0, 0, w, h);

    // Reset projection matrix stack
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Establish clipping volume (left, right, bottom, top, near, far)
    if (w <= h)
        glOrtho (-nRange, nRange, -nRange*h/w, nRange*h/w, -nRange, nRange);
    else
        glOrtho (-nRange*w/h, nRange*w/h, -nRange, nRange, -nRange, nRange);

    // Reset Model view matrix stack
    glMatrixMode(GL_MODELVIEW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glLoadIdentity();

    // Change view angle
    glRotatef(30.0f, 1.0f, 0.0f, 0.0f);
    glRotatef(330.0f, 0.0f, 1.0f, 0.0f);
}

// This function does any needed initialization on the rendering
// context.
void SetupRC()
{
    // Any three points on the ground (counter clockwise order)
    GLfloat points[3][3] = {{ -30.0f, y+0.01f, -20.0f },{ -30.0f, y+0.01f, 20.0f },{ 40.0f, y+0.01f, 20.0f }};

    // Any three point on HouseNorth wall
    GLfloat pointsN[3][3] = {{ 10.01f, 30.0f, 40.0f },{ 10.01f, 0.0f, 40.0f },{ 10.01f, 0.0f, -40.0f }};

    // Any three point on HouseSouth wall
    GLfloat pointsS[3][3] = {{ -10.01f, 30.0f, 40.0f },{ -10.01f, 30.0f, 0.0f },{ -10.01f, 0.0f, 0.0f }};

    // Any three point on HouseEast wall
    GLfloat pointsE[3][3] = {{ 40.0f, 30.0f, 10.01f },{ -40.0f, 30.0f, 10.01f },{ -40.0f, -30.0f, 10.01f }};

    // Any three point on HouseWest wall
    GLfloat pointsW[3][3] = {{ 40.0f, 30.0f, -10.01f },{ 40.0f, -30.0f, -10.01f },{ 0.0f, -30.0f, -10.01f }};

    // Calculate light position
    CalLightPos(lightPos);

    glEnable(GL_DEPTH_TEST);// Hidden surface removal
    glFrontFace(GL_CCW);        // Counter clock-wise polygons face out

    glEnable(GL_LIGHTING);          // Enable lighting

    // Setup and enable light 0
    glLightfv(GL_LIGHT0,GL_AMBIENT,ambientLight);
    glLightfv(GL_LIGHT0,GL_DIFFUSE,diffuseLight);
    glLightfv(GL_LIGHT0,GL_SPECULAR,specular);
    glLightfv(GL_LIGHT0,GL_POSITION,lightPos);
    glEnable(GL_LIGHT0);

    glEnable(GL_COLOR_MATERIAL);    // Enable Material color tracking

    // Front material ambient and diffuse colors track glColor
    glColorMaterial(GL_FRONT,GL_AMBIENT_AND_DIFFUSE);

    // All materials hereafter have full specular reflectivity
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // with a high shine
    glMaterialfv(GL_FRONT, GL_SPECULAR,specref);
    glMateriali(GL_FRONT,GL_SHININESS,128);

    // Black background
    glClearColor(0.0f, 1.0f, 1.0f, 1.0f );

    // Calculate projection matrix to draw shadow on the ground
    MakeShadowMatrix(points, lightPos, shadowMat);
    switch(direction)
        {
        case 'N':
            // Calculate projection matrix to draw shadow on the Direct cube
            MakeShadowMatrix(pointsN, lightPos, shadowMatDirect);
        break;

        case 'S':
            // Calculate projection matrix to draw shadow on the Direct cube
            MakeShadowMatrix(pointsS, lightPos, shadowMatDirect);
        break;

        case 'E':
            // Calculate projection matrix to draw shadow on the Direct cube
            MakeShadowMatrix(pointsE, lightPos, shadowMatDirect);
        break;

        case 'W':
            // Calculate projection matrix to draw shadow on the Direct cube
            MakeShadowMatrix(pointsW, lightPos, shadowMatDirect);
        break;
        default:
        return;
        }
}

// Function DrawBox - DrawBox10
void DrawBox(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(BoxX , BoxY, BoxZ);
    auxSolidBox(BoxWidth,BoxHeight,BoxLength);
    glTranslatef(-BoxX , -BoxY, -BoxZ);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}
void DrawBox2(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box2X , Box2Y, Box2Z);
    auxSolidBox(Box2Width,Box2Height,Box2Length);
    glTranslatef(-Box2X , -Box2Y, -Box2Z);
    }
}
void DrawBox3(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box3X , Box3Y, Box3Z);
    auxSolidBox(Box3Width,Box3Height,Box3Length);
    glTranslatef(-Box3X , -Box3Y, -Box3Z);
    }
}
void DrawBox4(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box4X , Box4Y, Box4Z);
    auxSolidBox(Box4Width,Box4Height,Box4Length);
    glTranslatef(-Box4X , -Box4Y, -Box4Z);
    }
}
void DrawBox5(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box5X , Box5Y, Box5Z);
    auxSolidBox(Box5Width,Box5Height,Box5Length);
    glTranslatef(-Box5X , -Box5Y, -Box5Z);
    }
}
void DrawBox6(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box6X , Box6Y, Box6Z);
    auxSolidBox(Box6Width,Box6Height,Box6Length);
    glTranslatef(-Box6X , -Box6Y, -Box6Z);
    }
}
void DrawBox7(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box7X , Box7Y, Box7Z);
    auxSolidBox(Box7Width,Box7Height,Box7Length);
    glTranslatef(-Box7X , -Box7Y, -Box7Z);
    }
}
void DrawBox8(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box8X , Box8Y, Box8Z);
    auxSolidBox(Box8Width,Box8Height,Box8Length);
    glTranslatef(-Box8X , -Box8Y, -Box8Z);
    }
}
void DrawBox9(BOOL bShadow)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box9X , Box9Y, Box9Z);
    auxSolidBox(Box9Width,Box9Height,Box9Length);
    glTranslatef(-Box9X , -Box9Y, -Box9Z);
    }
}
void DrawBox10(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 255, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Box10X , Box10Y, Box10Z);
    auxSolidBox(Box10Width,Box10Height,Box10Length);
    glTranslatef(-Box10X , -Box10Y, -Box10Z);
    }
}

// Function DrawCone - DrawCone10
void DrawCone(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(ConeX , ConeY, ConeZ);
    auxSolidCone(ConeRadius,ConeHeight);
    glTranslatef(-ConeX , -ConeY, -ConeZ);
    }
}
void DrawCone2(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
    glTranslatef(Cone2X , Cone2Y, Cone2Z);
    auxSolidCone(Cone2Radius,Cone2Height);
    glTranslatef(-Cone2X , -Cone2Y, -Cone2Z);
    }
}
void DrawCone3(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone3X , Cone3Y, Cone3Z);
    auxSolidCone(Cone3Radius,Cone3Height);
    glTranslatef(-Cone3X , -Cone3Y, -Cone3Z);
    }
}
void DrawCone4(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone4X , Cone4Y, Cone4Z);
    auxSolidCone(Cone4Radius,Cone4Height);
    glTranslatef(-Cone4X , -Cone4Y, -Cone4Z);
    }
}
void DrawCone5(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone5X , Cone5Y, Cone5Z);
    auxSolidCone(Cone5Radius,Cone5Height);
    glTranslatef(-Cone5X , -Cone5Y, -Cone5Z);
    }
}
void DrawCone6(BOOL bShadow)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone6X , Cone6Y, Cone6Z);
    auxSolidCone(Cone6Radius,Cone6Height);
    glTranslatef(-Cone6X , -Cone6Y, -Cone6Z);
    }
}
void DrawCone7(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone7X , Cone7Y, Cone7Z);
    auxSolidCone(Cone7Radius,Cone7Height);
    glTranslatef(-Cone7X , -Cone7Y, -Cone7Z);
    }
}
void DrawCone8(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone8X , Cone8Y, Cone8Z);
    auxSolidCone(Cone8Radius,Cone8Height);
    glTranslatef(-Cone8X , -Cone8Y, -Cone8Z);
    }
}
void DrawCone9(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone9X , Cone9Y, Cone9Z);
    auxSolidCone(Cone9Radius,Cone9Height);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glTranslatef(-Cone9X , -Cone9Y, -Cone9Z);
    }
}
void DrawCone10(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cone10X , Cone10Y, Cone10Z);
    auxSolidCone(Cone10Radius,Cone10Height);
    glTranslatef(-ConeX , -Cone10Y, -Cone10Z);
    }
}

// Function DrawCylinder - DrawCylinder10
void DrawCylinder(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(CylinderX , CylinderY, CylinderZ);
    auxSolidCylinder(CylinderRadius,CylinderHeight);
    glTranslatef(-CylinderX , -CylinderY, -CylinderZ);
    }
}
void DrawCylinder2(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder2X , Cylinder2Y, Cylinder2Z);
    auxSolidCylinder(Cylinder2Radius,Cylinder2Height);
    glTranslatef(-Cylinder2X , -Cylinder2Y, -Cylinder2Z);
    }
}
void DrawCylinder3(BOOL bShadow)
{
    if(!bShadow)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder3X , Cylinder3Y, Cylinder3Z);
    auxSolidCylinder(Cylinder3Radius,Cylinder3Height);
    glTranslatef(-Cylinder3X , -Cylinder3Y, -Cylinder3Z);
    }
}
void DrawCylinder4(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder4X , Cylinder4Y, Cylinder4Z);
    auxSolidCylinder(Cylinder4Radius,Cylinder4Height);
    glTranslatef(-Cylinder4X , -Cylinder4Y, -Cylinder4Z);
    }
}
void DrawCylinder5(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder5X , Cylinder5Y, Cylinder5Z);
    auxSolidCylinder(Cylinder5Radius,Cylinder5Height);
    glTranslatef(-Cylinder5X , -Cylinder5Y, -Cylinder5Z);
    }
}
void DrawCylinder6(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder6X , Cylinder6Y, Cylinder6Z);
    auxSolidCylinder(Cylinder6Radius,Cylinder6Height);
    glTranslatef(-Cylinder6X , -Cylinder6Y, -Cylinder6Z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}
void DrawCylinder7(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder7X , Cylinder7Y, Cylinder7Z);
    auxSolidCylinder(Cylinder7Radius,Cylinder7Height);
    glTranslatef(-Cylinder7X , -Cylinder7Y, -Cylinder7Z);
    }
}
void DrawCylinder8(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder8X , Cylinder8Y, Cylinder8Z);
    auxSolidCylinder(Cylinder8Radius,Cylinder8Height);
    glTranslatef(-Cylinder8X , -Cylinder8Y, -Cylinder8Z);
    }
}
void DrawCylinder9(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder9X , Cylinder9Y, Cylinder9Z);
    auxSolidCylinder(Cylinder9Radius,Cylinder9Height);
    glTranslatef(-Cylinder9X , -Cylinder9Y, -Cylinder9Z);
    }
}
void DrawCylinder10(BOOL bShadow)
{
    if(!bShadow)
        glRGB(0, 255, 255);
    else
        glRGB(0,0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
    glTranslatef(Cylinder10X , Cylinder10Y, Cylinder10Z);
    auxSolidCylinder(Cylinder10Radius,Cylinder10Height);
    glTranslatef(-Cylinder10X , -Cylinder10Y, -Cylinder10Z);
    }
}

// Function DrawPyramid - DrawPyramid10
void DrawPyramid(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(PyramidX,PyramidY+PyramidHeight,PyramidZ);
        glVertex3f(PyramidX+PyramidWidth/2,PyramidY,PyramidZ+PyramidWidth/2);
        glVertex3f(PyramidX+PyramidWidth/2,PyramidY,PyramidZ-PyramidWidth/2);
        glVertex3f(PyramidX-PyramidWidth/2,PyramidY,PyramidZ-PyramidWidth/2);
        glVertex3f(PyramidX-PyramidWidth/2,PyramidY,PyramidZ+PyramidWidth/2);
        glVertex3f(PyramidX+PyramidWidth/2,PyramidY,PyramidZ+PyramidWidth/2);
    }
    glEnd();
}
void DrawPyramid2(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid2X,Pyramid2Y+Pyramid2Height,Pyramid2Z);
        glVertex3f(Pyramid2X+Pyramid2Width/2,Pyramid2Y,Pyramid2Z+Pyramid2Width/2);
        glVertex3f(Pyramid2X+Pyramid2Width/2,Pyramid2Y,Pyramid2Z-Pyramid2Width/2);
        glVertex3f(Pyramid2X-Pyramid2Width/2,Pyramid2Y,Pyramid2Z-Pyramid2Width/2);
        glVertex3f(Pyramid2X-Pyramid2Width/2,Pyramid2Y,Pyramid2Z+Pyramid2Width/2);
        glVertex3f(Pyramid2X+Pyramid2Width/2,Pyramid2Y,Pyramid2Z+Pyramid2Width/2);
    }
    glEnd();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void DrawPyramid3(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid3X,Pyramid3Y+Pyramid3Height,Pyramid3Z);
        glVertex3f(Pyramid3X+Pyramid3Width/2,Pyramid3Y,Pyramid3Z+Pyramid3Width/2);
        glVertex3f(Pyramid3X+Pyramid3Width/2,Pyramid3Y,Pyramid3Z-Pyramid3Width/2);
        glVertex3f(Pyramid3X-Pyramid3Width/2,Pyramid3Y,Pyramid3Z-Pyramid3Width/2);
        glVertex3f(Pyramid3X-Pyramid3Width/2,Pyramid3Y,Pyramid3Z+Pyramid3Width/2);
        glVertex3f(Pyramid3X+Pyramid3Width/2,Pyramid3Y,Pyramid3Z+Pyramid3Width/2);
    }
    glEnd();
}
void DrawPyramid4(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid4X,Pyramid4Y+Pyramid4Height,Pyramid4Z);
        glVertex3f(Pyramid4X+Pyramid4Width/2,Pyramid4Y,Pyramid4Z+Pyramid4Width/2);
        glVertex3f(Pyramid4X+Pyramid4Width/2,Pyramid4Y,Pyramid4Z-Pyramid4Width/2);
        glVertex3f(Pyramid4X-Pyramid4Width/2,Pyramid4Y,Pyramid4Z-Pyramid4Width/2);
        glVertex3f(Pyramid4X-Pyramid4Width/2,Pyramid4Y,Pyramid4Z+Pyramid4Width/2);
        glVertex3f(Pyramid4X+Pyramid4Width/2,Pyramid4Y,Pyramid4Z+Pyramid4Width/2);
    }
    glEnd();
}
void DrawPyramid5(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        glVertex3f(Pyramid5X,Pyramid5Y+Pyramid5Height,Pyramid5Z);
        glVertex3f(Pyramid5X+Pyramid5Width/2,Pyramid5Y,Pyramid5Z+Pyramid5Width/2);
        glVertex3f(Pyramid5X+Pyramid5Width/2,Pyramid5Y,Pyramid5Z-Pyramid5Width/2);
        glVertex3f(Pyramid5X-Pyramid5Width/2,Pyramid5Y,Pyramid5Z-Pyramid5Width/2);
        glVertex3f(Pyramid5X-Pyramid5Width/2,Pyramid5Y,Pyramid5Z+Pyramid5Width/2);
        glVertex3f(Pyramid5X+Pyramid5Width/2,Pyramid5Y,Pyramid5Z+Pyramid5Width/2);
    }
    glEnd();
}
void DrawPyramid6(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid6X,Pyramid6Y+Pyramid6Height,Pyramid6Z);
        glVertex3f(Pyramid6X+Pyramid6Width/2,Pyramid6Y,Pyramid6Z+Pyramid6Width/2);
        glVertex3f(Pyramid6X+Pyramid6Width/2,Pyramid6Y,Pyramid6Z-Pyramid6Width/2);
        glVertex3f(Pyramid6X-Pyramid6Width/2,Pyramid6Y,Pyramid6Z-Pyramid6Width/2);
        glVertex3f(Pyramid6X-Pyramid6Width/2,Pyramid6Y,Pyramid6Z+Pyramid6Width/2);
        glVertex3f(Pyramid6X+Pyramid6Width/2,Pyramid6Y,Pyramid6Z+Pyramid6Width/2);
    }
    glEnd();
}
void DrawPyramid7(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid7X,Pyramid7Y+Pyramid7Height,Pyramid7Z);
        glVertex3f(Pyramid7X+Pyramid7Width/2,Pyramid7Y,Pyramid7Z+Pyramid7Width/2);
        glVertex3f(Pyramid7X+Pyramid7Width/2,Pyramid7Y,Pyramid7Z-Pyramid7Width/2);
        glVertex3f(Pyramid7X-Pyramid7Width/2,Pyramid7Y,Pyramid7Z-Pyramid7Width/2);
        glVertex3f(Pyramid7X-Pyramid7Width/2,Pyramid7Y,Pyramid7Z+Pyramid7Width/2);
        glVertex3f(Pyramid7X+Pyramid7Width/2,Pyramid7Y,Pyramid7Z+Pyramid7Width/2);
    }
    glEnd();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void DrawPyramid8(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid8X,Pyramid8Y+Pyramid8Height,Pyramid8Z);
        glVertex3f(Pyramid8X+Pyramid8Width/2,Pyramid8Y,Pyramid8Z+Pyramid8Width/2);
        glVertex3f(Pyramid8X+Pyramid8Width/2,Pyramid8Y,Pyramid8Z-Pyramid8Width/2);
        glVertex3f(Pyramid8X-Pyramid8Width/2,Pyramid8Y,Pyramid8Z-Pyramid8Width/2);
        glVertex3f(Pyramid8X-Pyramid8Width/2,Pyramid8Y,Pyramid8Z+Pyramid8Width/2);
        glVertex3f(Pyramid8X+Pyramid8Width/2,Pyramid8Y,Pyramid8Z+Pyramid8Width/2);
    }
    glEnd();
}
void DrawPyramid9(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid9X,Pyramid9Y+Pyramid9Height,Pyramid9Z);
        glVertex3f(Pyramid9X+Pyramid9Width/2,Pyramid9Y,Pyramid9Z+Pyramid9Width/2);
        glVertex3f(Pyramid9X+Pyramid9Width/2,Pyramid9Y,Pyramid9Z-Pyramid9Width/2);
        glVertex3f(Pyramid9X-Pyramid9Width/2,Pyramid9Y,Pyramid9Z-Pyramid9Width/2);
        glVertex3f(Pyramid9X-Pyramid9Width/2,Pyramid9Y,Pyramid9Z+Pyramid9Width/2);
        glVertex3f(Pyramid9X+Pyramid9Width/2,Pyramid9Y,Pyramid9Z+Pyramid9Width/2);
    }
    glEnd();
}
void DrawPyramid10(BOOL bShadow)
{
    // Set to black for the shadow once
    if(!bShadow)
        glRGB(0, 0, 255);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glBegin(GL_TRIANGLE_FAN);
    {
        glVertex3f(Pyramid10X,Pyramid10Y+Pyramid10Height,Pyramid10Z);
        glVertex3f(Pyramid10X+Pyramid10Width/2,Pyramid10Y,Pyramid10Z+Pyramid10Width/2);
        glVertex3f(Pyramid10X+Pyramid10Width/2,Pyramid10Y,Pyramid10Z-Pyramid10Width/2);
        glVertex3f(Pyramid10X-Pyramid10Width/2,Pyramid10Y,Pyramid10Z-Pyramid10Width/2);
        glVertex3f(Pyramid10X-Pyramid10Width/2,Pyramid10Y,Pyramid10Z+Pyramid10Width/2);
        glVertex3f(Pyramid10X+Pyramid10Width/2,Pyramid10Y,Pyramid10Z+Pyramid10Width/2);
    }
    glEnd();
}

// Function DrawSphere - DrawSphere10
void DrawSphere(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(SphereX , SphereY, SphereZ);
        auxSolidSphere(SphereRadius);
        glTranslatef(-SphereX , -SphereY, -SphereZ);
    }
}
void DrawSphere2(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere2X , Sphere2Y, Sphere2Z);
        auxSolidSphere(Sphere2Radius);
        glTranslatef(-Sphere2X , -Sphere2Y, -Sphere2Z);
    }
}
void DrawSphere3(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glTranslatef(Sphere3X , Sphere3Y, Sphere3Z);
        auxSolidSphere(Sphere3Radius);
        glTranslatef(-Sphere3X , -Sphere3Y, -Sphere3Z);
    }
}
void DrawSphere4(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere4X , Sphere4Y, Sphere4Z);
        auxSolidSphere(Sphere4Radius);
        glTranslatef(-Sphere4X , -Sphere4Y, -Sphere4Z);
    }
}
void DrawSphere5(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere5X , Sphere5Y, Sphere5Z);
        auxSolidSphere(Sphere5Radius);
        glTranslatef(-Sphere5X , -Sphere5Y, -Sphere5Z);
    }
}
void DrawSphere6(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere6X , Sphere6Y, Sphere6Z);
        auxSolidSphere(Sphere6Radius);
        glTranslatef(-Sphere6X , -Sphere6Y, -Sphere6Z);
    }
}
void DrawSphere7(BOOL bShadow)
{
    if(!bShadow)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere7X , Sphere7Y, Sphere7Z);
        auxSolidSphere(Sphere7Radius);
        glTranslatef(-Sphere7X , -Sphere7Y, -Sphere7Z);
    }
}
void DrawSphere8(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere8X , Sphere8Y, Sphere8Z);
        auxSolidSphere(Sphere8Radius);
        glTranslatef(-Sphere8X , -Sphere8Y, -Sphere8Z);
    }
}
void DrawSphere9(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere9X , Sphere9Y, Sphere9Z);
        auxSolidSphere(Sphere9Radius);
        glTranslatef(-Sphere9X , -Sphere9Y, -Sphere9Z);
    }
}
void DrawSphere10(BOOL bShadow)
{
    if(!bShadow)
        glRGB(255, 0, 0);
    else
        glRGB(0,0,0);
    glTranslatef(0.0f , y+0.1f, 0.0f);
    {
        glTranslatef(Sphere10X , Sphere10Y, Sphere10Z);
        auxSolidSphere(Sphere10Radius);
        glTranslatef(-Sphere10X , -Sphere10Y, -Sphere10Z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}

// This function just specifically draws the House
void DrawHouseNorth()
{
    float normal[3];  // Storeage for calculated surface normal
    // Call only once for all remaining points
    glBegin(GL_QUADS);
        {
        // Verticies for Front panel
        float v[4][3] =    {{ -30.0f, 30.0f, 40.0f},{ -30.0f, -30.0f, 40.0f},
                           { 10.0f, -30.0f, 40.0f},{10.0f,30.0f,40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]); glVertex3fv(v[1]);glVertex3fv(v[2]); glVertex3fv(v[3]);
        //glEnd();
        }
        {
        // Verticies for Back panel
        float v[4][3] =    {{ -30.0f, 30.0f, -40.0f},{ 10.0f, 30.0f, -40.0f},
                           { 10.0f, -30.0f, -40.0f},{-30.0f,-30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]); glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
        {
        // Verticies for Top panel
        float v[4][3] =    {{ 40.0f, 30.0f, 50.0f},{ 40.0f, 30.0f, -50.0f},
                           { -40.0f, 30.0f, -50.0f},{-40.0f,30.0f,50.0f}};
            // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
////////////////////////////////////////////////////////
        //post1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
// Verticies for Left panel of post1
float v[4][3] =    {{ 30.0f, -30.0f, 40.0f}, { 30.0f, -30.0f, 36.0f},
                   { 30.0f, 30.0f, 36.0f},{30.0f,30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel of post1
float v[4][3] =    {{ 26.0f, 30.0f, 40.0f},{ 26.0f, 30.0f, 36.0f},
                   { 26.0f, -30.0f, 36.0f},{26.0f,-30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel of post1
float v[4][3] =    {{ 26.0f, 30.0f, 40.0f},{ 26.0f, -30.0f, 40.0f},
                   { 30.0f, -30.0f, 40.0f},{30.0f,30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Back panel of post1
   float v[4][3] =    {{ 26.0f, 30.0f, 36.0f},{ 30.0f, 30.0f, 36.0f},
                      { 30.0f, -30.0f, 36.0f},{26.0f,-30.0f,36.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
////////////////////////////////////////////////////
    //post2
    {
    // Verticies for Left panel of post2
    float v[4][3] =    {{ 30.0f, -30.0f, -36.0f},{ 30.0f, -30.0f, -40.0f},
                       { 30.0f, 30.0f, -40.0f},{30.0f,30.0f,-36.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Right panel of post2
    float v[4][3] =    {{ 26.0f, 30.0f, -36.0f}, { 26.0f, 30.0f, -40.0f},
                       { 26.0f, -30.0f, -40.0f},{26.0f,-30.0f,-36.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Front panel of post2
    float v[4][3] =    {{ 26.0f, 30.0f, -36.0f},{ 26.0f, -30.0f, -36.0f},
                       { 30.0f, -30.0f, -36.0f},{30.0f,30.0f,-36.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Back panel of post2
       float v[4][3] =    {{ 26.0f, 30.0f, -40.0f},{ 30.0f, 30.0f, -40.0f},
                          { 30.0f, -30.0f, -40.0f},{26.0f,-30.0f,-40.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//glBegin(GL_QUADS);
    glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////////////////////
{
// Verticies for Right panel
float v[4][3] =    {{ -30.0f, 30.0f, 40.0f},{ -30.0f, 30.0f, -40.0f},
                    { -30.0f, -30.0f, -40.0f},{-30.0f,-30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for scond floor panel
float v[4][3] =    {{ 40.0f, -2.0f, 40.0f},{ 40.0f, -2.0f, -40.0f},
                    { -30.0f, -2.0f, -40.0f},{-30.0f,-2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for scond floor panel
float v[4][3] =    {{ 40.0f, 2.0f, 40.0f},{ 40.0f, 2.0f, -40.0f},
                    { -30.0f, 2.0f, -40.0f},{-30.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel in 2nd floor
float v[4][3] =    {{ 40.0f, -2.0f, 40.0f},{ 40.0f, -2.0f, -40.0f},
                    { 40.0f, 2.0f, -40.0f},{40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel in 2nd floor
float v[4][3] =   {{ 10.0f, 2.0f, 40.0f},{ 10.0f, -2.0f, 40.0f},
                  { 40.0f, -2.0f, 40.0f},{40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Back panel
float v[4][3] =   {{ 10.0f, 2.0f, -40.0f},{ 40.0f, 2.0f, -40.0f},
                  { 40.0f, -2.0f, -40.0f},{10.0f,-2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel
float v[4][3] =   {{ 10.0f, -30.0f, 40.0f}, { 10.0f, -30.0f, -40.0f},
                  { 10.0f, 30.0f, -40.0f},{10.0f,30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel
float v[4][3] =   {{ -30.0f, 30.0f, 40.0f},{ -30.0f, 30.0f, -40.0f},
                  { -30.0f, -30.0f, -40.0f},{-30.0f,-30.0f,40.0f}};
// Calculate the normal for the plane
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////// House Roof ///////////
{
// Verticies for East panel
float v[4][3] =   {{ -40.0f, 30.0f, -50.0f},{ -40.0f, 30.0f, 50.0f},
                  { 0.0f, 50.0f, 30.0f}, {.0f,50.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
}
{
// Verticies for Eest panel
float v[4][3] =   {{ -40.0f, 30.0f, -50.0f},{ 0.0f, 50.0f, -30.0f},
                  { 0.0f, 50.0f, -30.0f},{40.0f,30.0f,-50.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
}
{
// Verticies for South panel
float v[4][3] =   {{ -40.0f, 30.0f, 50.0f},{ 40.0f, 30.0f, 50.0f},
                  { 0.0f, 50.0f, 30.0f}, {0.0f,50.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
}
{
// Verticies for West panel
float v[4][3] =   {{ 40.0f, 30.0f, -50.0f}, { 0.0f, 50.0f, -30.0f},
                  { 0.0f, 50.0f, 30.0f}, {40.0f,30.0f,50.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
   glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    // Done drawing points
    glEnd();

}

// Draw House East
void DrawHouseEast()
{
    float normal[3];  // Storeage for calculated surface normal
    // Call only once for all remaining points
    glBegin(GL_QUADS);
        {
        // Verticies for Front panel
        float v[4][3] =    {{ -40.0f, 30.0f, 10.0f}, { -40.0f, -30.0f, 10.0f},
                           { -40.0f, -30.0f, -30.0f},{-40.0f,30.0f,-30.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
        {
        // Verticies for Back panel
        float v[4][3] =    {{ 40.0f, 30.0f, -30.0f},{ 40.0f, 30.0f, 10.0f},
                           { 40.0f, -30.0f, 10.0f},{40.0f,-30.0f,-30.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
        {
        // Verticies for Top panel
        float v[4][3] =    {{ -50.0f, 30.0f, 40.0f}, { 50.0f, 30.0f, 40.0f},
                           { 50.0f, 30.0f, -40.0f},{-50.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
////////////////////////////////////////////////////
    //post1
    {
    // Verticies for Left panel of post1
    float v[4][3] =    {{ -40.0f, -30.0f, 30.0f},{ -36.0f, -30.0f, 30.0f},
                       { -36.0f, 30.0f, 30.0f},{-40.0f,30.0f,30.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Right panel of post1
    float v[4][3] =    {{ -40.0f, 30.0f, 26.0f}, { -36.0f, 30.0f, 26.0f},
                       { -36.0f, -30.0f, 26.0f}, {-40.0f,-30.0f,26.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);glVertex3fv(v[0]); glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Front panel of post1
    float v[4][3] =    {{ -40.0f, 30.0f, 26.0f},{ -40.0f, -30.0f, 26.0f},
                       { -40.0f, -30.0f, 30.0f},{-40.0f,30.0f,30.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
       glNormal3fv(normal);
       glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Back panel of post1
       float v[4][3] =    {{ -36.0f, 30.0f, 26.0f},{ -36.0f, 30.0f, 30.0f},
                          { -36.0f, -30.0f, 30.0f},{-36.0f,-30.0f,26.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////
//post2
{
// Verticies for Left panel of post2
float v[4][3] =    {{ 36.0f, -30.0f, 30.0f},{ 40.0f, -30.0f, 30.0f},
                    { 40.0f, 30.0f, 30.0f},{36.0f,30.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel of post2
float v[4][3] =    {{ 36.0f, 30.0f, 26.0f},{ 40.0f, 30.0f,  26.0f},
                    { 40.0f, -30.0f, 26.0f},{36.0f,-30.0f,26.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel of post2
float v[4][3] =    {{ 36.0f, 30.0f, 26.0f},{ 36.0f, -30.0f, 26.0f},
                    { 36.0f, -30.0f, 30.0f},{36.0f,30.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
// Verticies for Back panel of post2
    float v[4][3] =    {{ 40.0f, 30.0f, 26.0f},{ 40.0f, 30.0f, 30.0f},
                       { 40.0f, -30.0f, 30.0f},{40.0f,-30.0f,26.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);  glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////////
{
// Verticies for Right panel
float v[4][3] =    {{ -40.0f, 30.0f, -30.0f},{ 40.0f, 30.0f, -30.0f},
                   { 40.0f, -30.0f, -30.0f}, {-40.0f,-30.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for scond floor panel
float v[4][3] =    {{ -40.0f, 2.0f, 40.0f},{ 40.0f, 2.0f, 40.0f},
                   { 40.0f, 2.0f, -30.0f},{-40.0f,2.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel in 2nd floor
float v[4][3] =    {{ -40.0f, -2.0f, 40.0f},{ 40.0f, -2.0f, 40.0f},
                   { 40.0f, 2.0f, 40.0f}, {-40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel in 2nd floor
float v[4][3] =    {{ 40.0f, -2.0f, 40.0f},{ 40.0f, -2.0f, 10.0f},
                   { 40.0f, 2.0f, 10.0f}, {40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel in 2nd floor
float v[4][3] =    {{ -40.0f, 2.0f, 10.0f},{ -40.0f, -2.0f, 10.0f},
                   { -40.0f, -2.0f, 40.0f},{-40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel
float v[4][3] =    {{ -40.0f, -30.0f, 10.0f},{ 40.0f, -30.0f, 10.0f},
                   { 40.0f, 30.0f, 10.0f},{-40.0f,30.0f,10.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////// House Roof ////////////////////////
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Verticies for Eest panel
        float v[4][3] =    {{ -50.0f, 30.0f, 40.0f}, { 50.0f, 30.0f, 40.0f},
                           { 30.0f, 50.0f, 0.0f}, {-30.0f,50.0f,0.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for Eest panel
        float v[4][3] =    {{ -50.0f, 30.0f, 40.0f}, { -30.0f, 50.0f, 0.0f},
                           { -30.0f, 50.0f, 0.0f},{-50.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for South panel
        float v[4][3] =    {{ 50.0f, 30.0f, -40.0f}, { 50.0f, 30.0f, 40.0f},
                           { 30.0f, 50.0f, 0.0f}, {30.0f,50.0f,0.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for West panel
        float v[4][3] =    {{ -50.0f, 30.0f, -40.0f},{ -30.0f, 50.0f, 0.0f},
                           { 30.0f, 50.0f, 0.0f}, {50.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
    glEnd();
}

// DrawHouse South
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void DrawHouseSouth()
{
    float normal[3];  // Storeage for calculated surface normal
    // Call only once for all remaining points
    glBegin(GL_QUADS);
        {
        // Verticies for Front panel
        float v[4][3] =    {{ -10.0f, 30.0f, -40.0f},{ -10.0f, -30.0f, -40.0f},
                           { 30.0f, -30.0f, -40.0f}, {30.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
        {
        // Verticies for Back panel
        float v[4][3] =    {{ 30.0f, 30.0f, 40.0f},{ -10.0f, 30.0f, 40.0f},
                           { -10.0f, -30.0f, 40.0f},{30.0f,-30.0f,40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
 ////////////////////////////////////////////////////
        //post1
        {
        // Verticies for Left panel of post1
        float v[4][3] =    {{ -30.0f, -30.0f, -40.0f},{ -30.0f, -30.0f, -36.0f},
                           { -30.0f, 30.0f, -36.0f},{-30.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        //glEnd();
        }
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Verticies for Right panel of post1
float v[4][3] =    {{ -26.0f, 30.0f, -40.0f},{ -26.0f, 30.0f,-36.0f},
                    { -26.0f, -30.0f, -36.0f},{-26.0f,-30.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel of post1
float v[4][3] =    {{ -26.0f, 30.0f, -40.0f},{ -26.0f, -30.0f, -40.0f},
                    { -30.0f, -30.0f,-40.0f}, {-30.0f,30.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Back panel of post1
    float v[4][3] =    {{ -26.0f, 30.0f, -36.0f},{ -30.0f, 30.0f, -36.0f},
                        { -30.0f, -30.0f, -36.0f},{-26.0f,-30.0f,-36.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////
//post2
{
// Verticies for Left panel of post2
float v[4][3] =    {{ -30.0f, -30.0f, 36.0f},{ -30.0f, -30.0f, 40.0f},
                    { -30.0f, 30.0f, 40.0f},{-30.0f,30.0f,36.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel of post2
float v[4][3] =    {{ -26.0f, 30.0f, 36.0f}, { -26.0f, 30.0f,  40.0f},
                   { -26.0f, -30.0f, 40.0f},{-26.0f,-30.0f,36.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel of post2
float v[4][3] =    {{ -26.0f, 30.0f, 36.0f}, { -26.0f, -30.0f, 36.0f},
                   { -30.0f, -30.0f, 36.0f},{-30.0f,30.0f,36.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Back panel of post2
float v[4][3] =    {{ -26.0f, 30.0f, 40.0f}, { -30.0f, 30.0f, 40.0f},
                   { -30.0f, -30.0f, 40.0f}, {-26.0f,-30.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////////
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Verticies for Right panel
float v[4][3] =    {{ 30.0f, 30.0f, -40.0f}, { 30.0f, 30.0f, 40.0f},
                   { 30.0f, -30.0f, 40.0f},{30.0f,-30.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for scond floor panel
float v[4][3] =    {{ -40.0f, 2.0f, -40.0f},{ -40.0f, 2.0f, 40.0f},
                   { 30.0f, 2.0f, 40.0f},{30.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel in 2nd floor
float v[4][3] =    {{ -40.0f, -2.0f, -40.0f},{ -40.0f, -2.0f, 40.0f},
                   { -40.0f, 2.0f, 40.0f},{-40.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel in 2nd floor
float v[4][3] =    {{ -40.0f, -2.0f, 40.0f},{ -10.0f, -2.0f, 40.0f},
                   { -10.0f, 2.0f, 40.0f},{-40.0f,2.0f,40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel in 2nd floor
float v[4][3] =    {{ -10.0f, 2.0f, -40.0f},{ -10.0f, -2.0f, -40.0f},
                   { -40.0f, -2.0f, -40.0f},{-40.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel
float v[4][3] =    {{ -10.0f, -30.0f, -40.0f},{ -10.0f, -30.0f, 40.0f},
                   { -10.0f, 30.0f, 40.0f},{-10.0f,30.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////// House Roof ////////////////////////
{
// Verticies for Eest panel
float v[4][3] =    {{ -40.0f, 30.0f, -50.0f},{ -40.0f, 30.0f, 50.0f},
                   { 0.0f, 50.0f, 30.0f}, {.0f,50.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
}
{
// Verticies for Eest panel
float v[4][3] =    {{ -40.0f, 30.0f, -50.0f},{ 0.0f, 50.0f, -30.0f},
                   { 0.0f, 50.0f, -30.0f},{40.0f,30.0f,-50.0f}};
// Calculate the normal for the plane
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for South panel
        float v[4][3] =    {{ -40.0f, 30.0f, 50.0f}, { 40.0f, 30.0f, 50.0f},
                           { 0.0f, 50.0f, 30.0f}, {0.0f,50.0f,30.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for West panel
        float v[4][3] =    {{ 40.0f, 30.0f, -50.0f}, { 0.0f, 50.0f, -30.0f},
                           { 0.0f, 50.0f, 30.0f}, {40.0f,30.0f,50.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
    glEnd();
    }

// Draw House West
void DrawHouseWest()
{
    float normal[3];  // Storeage for calculated surface normal
    // Call only once for all remaining points
    glBegin(GL_QUADS);
        {
        // Verticies for Right panel
        float v[4][3] =    {{ 40.0f, -30.0f, -10.0f},{ 40.0f, 30.0f, -10.0f},
                           { 40.0f, 30.0f, 30.0f},{40.0f,-30.0f,30.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
        //glBegin(GL_QUADS);
            glNormal3fv(normal);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel
float v[4][3] =    {{ -40.0f, 30.0f, 30.0f}, { -40.0f, 30.0f, -10.0f},
                   { -40.0f, -30.0f, -10.0f},{-40.0f,-30.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////
//post1
{
// Verticies for Front panel of post1
float v[4][3] =    {{ 40.0f, -30.0f, -30.0f},{ 36.0f, -30.0f, -30.0f},
                   { 36.0f, 30.0f, -30.0f},{40.0f,30.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Back panel of post1
float v[4][3] =    {{ 40.0f, 30.0f, -26.0f},{ 36.0f, 30.0f,-26.0f},
                   { 36.0f, -30.0f, -26.0f},{40.0f,-30.0f,-26.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel of post1
float v[4][3] =    {{ 40.0f, 30.0f, -26.0f}, { 40.0f, -30.0f, -26.0f},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    { 40.0f, -30.0f,-30.0f},{40.0f,30.0f,-30.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
        glNormal3fv(normal);
        glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Left panel of post1
    float v[4][3] =    {{ 36.0f, 30.0f, -26.0f},{ 36.0f, 30.0f, -30.0f},
                       { 36.0f, -30.0f, -30.0f},{36.0f,-30.0f,-26.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
        glNormal3fv(normal);
        glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
////////////////////////////////////////////////////////
    //post2
    {
    // Verticies for Front panel of post2
    float v[4][3] =    {{ -36.0f, -30.0f, -30.0f},{ -40.0f, -30.0f, -30.0f},
                       { -40.0f, 30.0f, -30.0f},{-36.0f,30.0f,-30.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
        glNormal3fv(normal);
        glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
    //glEnd();
    }
    {
    // Verticies for Back panel of post2
    float v[4][3] =    {{ -36.0f, 30.0f, -26.0f},{ -40.0f, 30.0f,  -26.0f},
                       { -40.0f, -30.0f, -26.0f},{-36.0f,-30.0f,-26.0f}};
    // Calculate the normal for the plane
    calcNormal(v,normal);
    // Draw the quadrangle using the plane normal
    // for all the vertices
    //glBegin(GL_QUADS);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]); glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel of post2
float v[4][3] =    {{ -36.0f, 30.0f, -26.0f},{ -36.0f, -30.0f, -26.0f},
                   { -36.0f, -30.0f, -30.0f},{-36.0f,30.0f,-30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel of post2
float v[4][3] =    {{ -40.0f, 30.0f, -26.0f},{ -40.0f, 30.0f, -30.0f},
                   { -40.0f, -30.0f, -30.0f},{-40.0f,-30.0f,-26.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////////////////////////////////////////
{
// Verticies for Back panel
float v[4][3] =    {{ 40.0f, 30.0f, 30.0f},{ -40.0f, 30.0f, 30.0f},
                   { -40.0f, -30.0f, 30.0f},{40.0f,-30.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for scond floor panel
float v[4][3] =    {{ 40.0f, 2.0f, -40.0f},{ -40.0f, 2.0f, -40.0f},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                { -40.0f, 2.0f, 30.0f},{40.0f,2.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Front panel in 2nd floor
float v[4][3] =    {{ 40.0f, -2.0f, -40.0f},{ -40.0f, -2.0f, -40.0f},
                { -40.0f, 2.0f, -40.0f},{40.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Left panel in 2nd floor
float v[4][3] =    {{ -40.0f, -2.0f, -40.0f},{ -40.0f, -2.0f, -10.0f},
                { -40.0f, 2.0f, -10.0f},{-40.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Right panel in 2nd floor
float v[4][3] =    {{ 40.0f, 2.0f, -10.0f},{ 40.0f, -2.0f, -10.0f},
                { 40.0f, -2.0f, -40.0f},{40.0f,2.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//glEnd();
}
{
// Verticies for Front panel
float v[4][3] =    {{ 40.0f, 30.0f, -10.0f}, { 40.0f, -30.0f, -10.0f},
                   { -40.0f, -30.0f, -10.0f},{-40.0f,30.0f,-10.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);

// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
{
// Verticies for Top panel
float v[4][3] =    {{ 40.0f, 30.0f, 30.0f},{ 40.0f, 30.0f, -10.0f},
                   { -40.0f, 30.0f, -10.0f},{-40.0f,30.0f,30.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
//glBegin(GL_QUADS);
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
//glEnd();
}
////////////////// House Roof ////////////////////////
{
// Verticies for Back panel
float v[4][3] =    {{ -50.0f, 30.0f, 40.0f}, { 50.0f, 30.0f, 40.0f},
                   { 30.0f, 50.0f, 0.0f}, {-30.0f,50.0f,0.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
// Draw the quadrangle using the plane normal
// for all the vertices
    glNormal3fv(normal);
    glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
}
{
// Verticies for Left panel
float v[4][3] =    {{ -50.0f, 30.0f, 40.0f}, { -30.0f, 50.0f, 0.0f},
                   { -30.0f, 50.0f, 0.0f},{-50.0f,30.0f,-40.0f}};
// Calculate the normal for the plane
calcNormal(v,normal);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for Right panel
        float v[4][3] =    {{ 50.0f, 30.0f, -40.0f}, { 50.0f, 30.0f, 40.0f},
                           { 30.0f, 50.0f, 0.0f}, {30.0f,50.0f,0.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
        {
        // Verticies for Front panel
        float v[4][3] =    {{ -50.0f, 30.0f, -40.0f},{ -30.0f, 50.0f, 0.0f},
                           { 30.0f, 50.0f, 0.0f}, {50.0f,30.0f,-40.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
  glEnd();
}

// Draw HOUSE LOCATION
void HouseLocation(BOOL bShadow, char direct)
{
    if (!bShadow)
        // set colour is Red
        glRGB(255,0,0);
    else
        // set colour is Black
        glRGB(0,0,0);

    //Check direction
    switch(direct)
        {
            // North
            case 'N':DrawHouseNorth();
            break;
            // South
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case 'S':DrawHouseSouth();
            break;
            //East
            case 'E':DrawHouseEast();
            break;
            //West
            case 'W':DrawHouseWest();
            break;
            default:
            return;
        }
}

// Draw Letter
void DrawEast(void)
{
    int i;
    float z=0.0f,normal[3];
    // set colour is Blue
    glRGB(0,0,255);
    // Draw the East
    glBegin(GL_QUADS);
        {
        // Verticies for Front panel
        float v[4][3] =    {{-10.f ,y+1.0f , 190.0f},{ -0.0f ,y+1.0f , 190.0f},
                           { -0.0f ,y+1.0f , 140.0f},{ -10.f ,y+1.0f , 140.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
        }
    for (i=0; i<3; i++)
    {
        // Verticies for Front panel
        float v[4][3] =    {{10.0f ,y+1.0f , 150.0f+z},{ 10.0f ,y+1.0f , 140.0f+z},
                           { -0.0f ,y+1.0f , 140.0f+z}, {  -0.0f ,y+1.0f , 150.0f+z}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);z+=20.0f;
    }
    glEnd();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void DrawNorth(void)
{
   float z=0.0f,normal[3];
   // set colour is Blue
   glRGB(0,0,255);
   // Draw the North
   glBegin(GL_QUADS);
      {
      // Verticies for Front panel
      float v[4][3] =    {{190.0f ,y+1.0f , 10.0f},{ 140.0f ,y+1.0f , 10.0f},
                         { 140.0f ,y+1.0f , 17.0f},{ 190.0f ,y+1.0f , 17.0f}};
      // Calculate the normal for the plane
      calcNormal(v,normal);
      // Draw the quadrangle using the plane normal
      // for all the vertices
         glNormal3fv(normal);
         glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
      }
      {
      // Verticies for Front panel
      float v[4][3] =    {{190.0f ,y+1.0f , -10.0f},{ 175.0f ,y+1.0f , -10.0f},
                         { 140.0f ,y+1.0f , 10.0f},{ 155.0f ,y+1.0f , 10.0f}};
      // Calculate the normal for the plane
      calcNormal(v,normal);
      // Draw the quadrangle using the plane normal
      // for all the vertices
         glNormal3fv(normal);
         glVertex3fv(v[0]);glVertex3fv(v[1]);glVertex3fv(v[2]);glVertex3fv(v[3]);
      }
      {
      // Verticies for Front panel
      float v[4][3] =    {{190.0f ,y+1.0f , -10.0f},{ 190.0f ,y+1.0f , -17.0f},
                         { 140.0f ,y+1.0f , -17.0f},{ 140.0f ,y+1.0f , -10.0f}};
      // Calculate the normal for the plane
      calcNormal(v,normal);
      // Draw the quadrangle using the plane normal
      // for all the vertices
         glNormal3fv(normal);
         glVertex3fv(v[0]); glVertex3fv(v[1]);  glVertex3fv(v[2]); glVertex3fv(v[3]);
      }
   glEnd();
}

void DrawWest(void)
{
   float z=0.0f,normal[3];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // set colour is Blue
    glRGB(0,0,255);
    // Draw the West
    glBegin(GL_QUADS);
        {
        // Verticies for Front panel
        float v[4][3] =    {{10.0f ,y+1.0f , -190.0f}, { 10.0f ,y+1.0f , -170.0f},
                           { 17.0f ,y+1.0f , -140.0f},{ 22.0f ,y+1.0f , -140.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
        {
        float v[4][3] =    {{10.0f ,y+1.0f , -170.0f},{ 10.0f ,y+1.0f , -190.0f},
                           { 0.0f ,y+1.0f , -160.0f},{ 0.0f ,y+1.0f , -140.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
        {
        float v[4][3] =    {{-10.0f ,y+1.0f , -170.0f}, {  0.0f ,y+1.0f , -140.0f},
                           {  0.0f ,y+1.0f , -160.0f},{ -10.0f ,y+1.0f , -190.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
        {
        float v[4][3] =    {{-10.0f ,y+1.0f , -190.0f}, { -22.0f ,y+1.0f , -140.0f},
                           { -17.0f ,y+1.0f , -140.0f},{ -10.0f ,y+1.0f , -170.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
    glEnd();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void DrawSouth(void)
{
    int i;
    float z=0.0f,normal[3];
    // set colour is Blue
    glRGB(0,0,255);
    // Draw the South
    glBegin(GL_QUADS):
        {
        // Verticies for Front panel
        float v[4][3] =    {{-150.0f ,y+1.0f ,-2.0f},{ -150.0f ,y+1.0f ,-10.0f},
                           { -160.0f ,y+1.0f ,-10.0f},{ -160.0f ,y+1.0f ,-2.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
        {
        // Verticies for Front panel
        float v[4][3] =    {{-170.0f ,y+1.0f ,2.0f},{ -180.0f ,y+1.0f ,2.0f},
                           { -180.0f ,y+1.0f ,10.0f},{ -170.0f ,y+1.0f ,10.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]);
        }
        for (i=0; i<3; i++)
        {
        // Verticies for Front panel
        float v[4][3] =    {{-190.0f+z ,y+1.0f , -10.0f},{ -190.0f+z ,y+1.0f , +10.0f},
                           { -180.0f+z ,y+1.0f , +10.0f},{ -180.0f+z ,y+1.0f , -10.0f}};
        // Calculate the normal for the plane
        calcNormal(v,normal);
        // Draw the quadrangle using the plane normal
        // for all the vertices
            glNormal3fv(normal);
            glVertex3fv(v[0]); glVertex3fv(v[1]); glVertex3fv(v[2]); glVertex3fv(v[3]); z+=20.0f;
        }
    glEnd();
}

// Called to draw scene
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void RenderScene(void)
{
    char direct;
    if (North)
    {
        direct = 'N'; direction = direct;
    }
    if (South)
    {
        direct = 'S'; direction = direct;
    }
    if (West)
    {
        direct = 'W'; direction = direct;
    }
    if (East)
    {
        direct = 'E'; direction = direct;
    }

    // Clear the window with current clearing color
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    // Save the matrix state and do the rotations
    glMatrixMode(GL_MODELVIEW);

    // Save matrix state and do the rotation
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the House
    HouseLocation(FALSE,direct);
    // Restore transformations
    glPopMatrix();

// Draw BOX1 - BOX10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the BOX1
    DrawBox(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the BOX2
    DrawBox2(FALSE);
    glPopMatrix();

    glPushMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX3
DrawBox3(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX4
DrawBox4(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX5
DrawBox5(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX6
DrawBox6(FALSE);
glPopMatrix();

glPushMatrix();  glRotatef(xRot, 1.0f, 0.0f, 0.0f);
glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX7
DrawBox7(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX8
DrawBox8(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX9
DrawBox9(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the BOX10
DrawBox10(FALSE);
glPopMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw CONE1 - CONE10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE1
    DrawCone(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE2
    DrawCone2(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE3
    DrawCone3(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE4
    DrawCone4(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE5
    DrawCone5(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE6
    DrawCone6(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE7
    DrawCone7(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE8
    DrawCone8(FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE9
    DrawCone9(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CONE10
    DrawCone10(FALSE);
    glPopMatrix();

// Draw CYLINDER1 - CYLINDER10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER1
    DrawCylinder(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER2
    DrawCylinder2(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER3
    DrawCylinder3(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER4
    DrawCylinder4(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER5
    DrawCylinder5(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // draw the CYLINDER6
    DrawCylinder6(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER7
    DrawCylinder7(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER8
    DrawCylinder8(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER9
    DrawCylinder9(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the CYLINDER10
    DrawCylinder10(FALSE);
    glPopMatrix();

// Draw PYRAMID1 - PYRAMID10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid1
    DrawPyramid(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid2
    DrawPyramid2(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid3
    DrawPyramid3(FALSE);
    glPopMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid4
    DrawPyramid4(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid5
    DrawPyramid5(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid6
    DrawPyramid6(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid7
    DrawPyramid7(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid8
    DrawPyramid8(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid9
    DrawPyramid9(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    //Draw Pyramid10
    DrawPyramid10(FALSE);
    glPopMatrix();

// Draw SPHERE1 - SPHERE10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the SPHERE
    DrawSphere(FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere2(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere3(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere4(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere5(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere6(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere7(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
DrawSphere8(FALSE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
// draw the SPHERE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DrawSphere9(FALSE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // draw the SPHERE
    DrawSphere10(FALSE);
    glPopMatrix();

    // Get ready to draw the shadow and the ground
    // First disable lighting and save the projection state
    glPushAttrib(GL_LIGHTING_BIT);
    glDisable(GL_LIGHTING);
    // Save matrix state and do the rotation
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // Multiply by shadow projection matrix
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the House
    HouseLocation(TRUE,direct);
    // Restore transformations
    glPopMatrix();

// Draw Shadow BOX1 - BOX10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the BOX1
    DrawBox(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the BOX2
    DrawBox2(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the BOX3
    DrawBox3(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX4
DrawBox4(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX5
DrawBox5(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX6
DrawBox6(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX7
DrawBox7(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX8
DrawBox8(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX9
DrawBox9(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the BOX10
DrawBox10(TRUE);
glPopMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw Shadow CONE1 - CONE10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE1
    DrawCone(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE2
    DrawCone2(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE3
    DrawCone3(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE4
    DrawCone4(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE5
    DrawCone5(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE6
    DrawCone6(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DrawCone7(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE8
    DrawCone8(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE9
    DrawCone9(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the CONE10
    DrawCone10(TRUE);
    glPopMatrix();

// Draw Shadow CYLINDER1 - CYLINDER10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the cylinder1
    DrawCylinder(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the cylinder2
    DrawCylinder2(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the cylinder3
    DrawCylinder3(TRUE);
    glPopMatrix();

    glPushMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder4
DrawCylinder4(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder5
DrawCylinder5(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder6
DrawCylinder6(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder7
DrawCylinder7(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder8
DrawCylinder8(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder9
DrawCylinder9(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the cylinder10
DrawCylinder10(TRUE);
glPopMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw Shadow PYRAMID1 - PYRAMID10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid1
    DrawPyramid(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid2
    DrawPyramid2(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid3
    DrawPyramid3(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid4
    DrawPyramid4(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid5
    DrawPyramid5(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid6
    DrawPyramid6(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //Draw Pyramid7
    DrawPyramid7(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid8
    DrawPyramid8(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid9
    DrawPyramid9(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    //Draw Pyramid10
    DrawPyramid10(TRUE);
    glPopMatrix();

// Draw Shadow SPHERE1 - SPHERE10
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the sphere1
    DrawSphere(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the sphere2
    DrawSphere2(TRUE);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glMultMatrixf((GLfloat *)shadowMat);
    // draw the sphere3
    DrawSphere3(TRUE);
    glPopMatrix();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere4
DrawSphere4(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere5
DrawSphere5(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere6
DrawSphere6(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere7
DrawSphere7(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere8
DrawSphere8(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere9
DrawSphere9(TRUE);
glPopMatrix();

glPushMatrix();
glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glMultMatrixf((GLfloat *)shadowMat);
// draw the sphere10
DrawSphere10(TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glPopMatrix();

// Draw the light source
    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glTranslatef(lightPos[0],lightPos[1], lightPos[2]);
    glRGB(255,255,0);
    auxSolidSphere(10.0f);
    glPopMatrix();

    glPushMatrix();
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);   glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    // Draw the ground, we do manual shading to a darker green
    // in the background to give the illusion of depth
    glBegin(GL_QUADS);
        glRGB(0,255,0);
        glVertex3f(200.0f, y, -200.0f);      glVertex3f(-200.0f, y, -200.0f);
        glVertex3f(-200.0f, y, 200.0f);      glVertex3f(200.0f, y, 200.0f);
    glEnd();

    // Draw Letter
    DrawEast();
    DrawNorth();
    DrawWest();
    DrawSouth();
    glPopMatrix();

    // Restore lighting state variables
    glPopAttrib();

    // Flush drawing commands
    glFlush();
}

// If necessary, creates a 3-3-2 palette for the device context listed.
HPALETTE GetOpenGLPalette(HDC hDC)
{
    HPALETTE hRetPal = NULL;// Handle to palette to be created
    PIXELFORMATDESCRIPTOR pfd; // Pixel Format Descriptor
    LOGPALETTE *pPal;          // Pointer to memory for logical palette
    int nPixelFormat;          // Pixel format index
    int nColors;               // Number of entries in palette
    int i;                     // Counting variable
    BYTE RedRange,GreenRange,BlueRange;
                               // Range for each color entry (7,7,and 3)

    // Get the pixel format index and retrieve the pixel format description
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nPixelFormat = GetPixelFormat(hDC);
DescribePixelFormat(hDC, nPixelFormat, sizeof(PIXELFORMATDESCRIPTOR), &pfd);

// Does this pixel format require a palette?  If not, do not create a
// palette and just return NULL
if(!(pfd.dwFlags & PFD_NEED_PALETTE))
    return NULL;

// Number of entries in palette.  8 bits yeilds 256 entries
nColors = 1 << pfd.cColorBits;

// Allocate space for a logical palette structure plus all the palette entries
pPal = (LOGPALETTE*)malloc(sizeof(LOGPALETTE) +nColors*sizeof(PALETTEENTRY));

// Fill in palette header
pPal->palVersion = 0x300;      // Windows 3.0
pPal->palNumEntries = nColors; // table size

// Build mask of all 1's.  This creates a number represented by having
// the low order x bits set, where x = pfd.cRedBits, pfd.cGreenBits, and
// pfd.cBlueBits.
RedRange = (1 << pfd.cRedBits) -1;
GreenRange = (1 << pfd.cGreenBits) - 1;
BlueRange = (1 << pfd.cBlueBits) -1;

// Loop through all the palette entries
for(i = 0; i < nColors; i++)
    {
    // Fill in the 8-bit equivalents for each component
    pPal->palPalEntry[i].peRed = (i >> pfd.cRedShift) & RedRange;
    pPal->palPalEntry[i].peRed = (unsigned char)(
        (double) pPal->palPalEntry[i].peRed * 255.0 / RedRange);

    pPal->palPalEntry[i].peGreen = (i >> pfd.cGreenShift) & GreenRange;
    pPal->palPalEntry[i].peGreen = (unsigned char)(
        (double)pPal->palPalEntry[i].peGreen * 255.0 / GreenRange);

    pPal->palPalEntry[i].peBlue = (i >> pfd.cBlueShift) & BlueRange;
    pPal->palPalEntry[i].peBlue = (unsigned char)(
        (double)pPal->palPalEntry[i].peBlue * 255.0 / BlueRange);

    pPal->palPalEntry[i].peFlags = (unsigned char) NULL;
    }

// Create the palette
hRetPal = CreatePalette(pPal);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // Go ahead and select and realize the palette for this device context
    SelectPalette(hDC,hRetPal,FALSE);
    RealizePalette(hDC);

    // Free the memory used for the logical palette structure
    free(pPal);

    // Return the handle to the new palette
    return hRetPal;
}


// Select the pixel format for a given device context
void SetDCPixelFormat(HDC hDC)
{
    int nPixelFormat;

    static PIXELFORMATDESCRIPTOR pfd = {
        sizeof(PIXELFORMATDESCRIPTOR),  // Size of this structure
        1,                              // Version of this structure
        PFD_DRAW_TO_WINDOW |            // Draw to Window (not to bitmap)
        PFD_SUPPORT_OPENGL |            // Support OpenGL calls in window
        PFD_DOUBLEBUFFER,               // Double buffered
        PFD_TYPE_RGBA,                  // RGBA Color mode
        24,                             // Want 24bit color
        0,0,0,0,0,0,                    // Not used to select mode
        0,0,                            // Not used to select mode
        0,0,0,0,0,                      // Not used to select mode
        32,                             // Size of depth buffer
        0,                              // Not used to select mode
        0,                              // Not used to select mode
        PFD_MAIN_PLANE,                 // Draw in main plane
        0,                              // Not used to select mode
        0,0,0 };                        // Not used to select mode

    // Choose a pixel format that best matches that described in pfd
    nPixelFormat = ChoosePixelFormat(hDC, &pfd);

    // Set the pixel format for the device context
    SetPixelFormat(hDC, nPixelFormat, &pfd);
}

// Entry point of all Windows programs
int APIENTRY WinMain( HINSTANCE      hInst,
                      HINSTANCE      hPrevInstance,
                      LPSTR          lpCmdLine,
                      int            nCmdShow)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    MSG         msg;        // Windows message structure
    WNDCLASS    wc;         // Windows class structure
    HWND        hWnd;       // Storeage for window handle

    hInstance = hInst;

    // Register Window style
    wc.style            = CS_HREDRAW | CS_VREDRAW ;
    wc.lpfnWndProc      = (WNDPROC) WndProc;
    wc.cbClsExtra       = 0;
    wc.cbWndExtra       = 0;
    wc.hInstance        = hInstance;
    wc.hIcon            = NULL;
    wc.hCursor          = LoadCursor(NULL, IDC_ARROW);

    // No need for background brush for OpenGL window
    wc.hbrBackground    = NULL;

    wc.lpszMenuName     = MAKEINTRESOURCE(IDR_MENU);
    wc.lpszClassName    = lpszAppName;

    // Register the window class
    if(RegisterClass(&wc) == 0)
        return FALSE;

    // Create the main application window
    hWnd = CreateWindow( lpszAppName,  lpszAppName,

                // OpenGL requires WS_CLIPCHILDREN and WS_CLIPSIBLINGS
                WS_OVERLAPPEDWINDOW | WS_CLIPCHILDREN | WS_CLIPSIBLINGS,

                // Window position and size
                CW_USEDEFAULT, 0, CW_USEDEFAULT, 0,
                //50, 50,
                //400, 400,
                NULL, NULL, hInstance, NULL);

    // If window was not created, quit
    if(hWnd == NULL)
        return FALSE;

    // Display the window
    ShowWindow(hWnd,SW_SHOW);
    UpdateWindow(hWnd);

    // Process application messages until the application closes
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while( GetMessage(&msg, NULL, 0, 0))
        {
        TranslateMessage(&msg);      DispatchMessage(&msg);
        }
    return msg.wParam;
}

// Window procedure, handles all messages for this program
LRESULT CALLBACK WndProc(   HWND hWnd,UINT message,
                            WPARAM wParam,LPARAM lParam)
{
    static HGLRC hRC;              // Permenant Rendering context
    static HDC hDC;               // Private GDI Device context
    switch (message)
        {
        // Window creation, setup for OpenGL
        case WM_CREATE:
            // Store the device context
            hDC = GetDC(hWnd);
            // Select the pixel format
            SetDCPixelFormat(hDC);
            // Create palette if needed
            hPalette = GetOpenGLPalette(hDC);
            // Create the rendering context and make it current
            hRC = wglCreateContext(hDC);
            wglMakeCurrent(hDC, hRC);
            // Call OpenGL setup code
            SetupRC();
            // Turn off all effects by default
            North = FALSE;
            South = FALSE;
            West = FALSE;
            East = FALSE;
            break;

        // Window is being destroyed, cleanup
        case WM_DESTROY:
            // Deselect the current rendering context and delete it
            wglMakeCurrent(hDC,NULL);
            wglDeleteContext(hRC);
            ReleaseDC(hWnd,hDC);
            if(hPalette != NULL)
                DeleteObject(hPalette);
            // Tell the application to terminate after the window
            // is gone.
            PostQuitMessage(0);
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Window is resized.
case WM_SIZE:
    // Call our function which modifies the clipping
    // volume and viewport
    ChangeSize(LOWORD(lParam), HIWORD(lParam));
    break;

// The painting function.  This message sent by Windows
// whenever the screen needs updating.
case WM_PAINT:
    {
    // Call OpenGL setup code
    SetupRC();
    // Call OpenGL drawing code
    RenderScene();
    SwapBuffers(hDC);
    // Validate the newly painted client area
    ValidateRect(hWnd,NULL);
    }
    break;

// Key press, check for arrow keys to do cube rotation.
case WM_KEYDOWN:
    {
    if(wParam == VK_UP)          xRot-= 5.0f;
    if(wParam == VK_DOWN)        xRot += 5.0f;
    if(wParam == VK_LEFT)        yRot -= 5.0f;
    if(wParam == VK_RIGHT)       yRot += 5.0f;
    if(xRot > 356.0f)            xRot = 0.0f;
    if(xRot < -1.0f)             xRot = 355.0f;
    if(yRot > 356.0f)            yRot = 0.0f;
    if(yRot < -1.0f)             yRot = 355.0f;
    // Keep rotations within 360 degrees
    xRot = (float)((const int)xRot % 360);
    yRot = (float)((const int)yRot % 360);
    InvalidateRect(hWnd,NULL,FALSE);
    }
    break;

// Windows is telling the application that it may modify
// the system palette.  This message in essance asks the
// application for a new palette.
case WM_QUERYNEWPALETTE:
    // If the palette was created.
    if(hPalette)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int nRet;
        // Selects the palette into the current device context
        SelectPalette(hDC, hPalette, FALSE);
        // Map entries from the currently selected palette to
        // the system palette.  The return value is the number
        // of palette entries modified.
        nRet = RealizePalette(hDC);
        // Repaint, forces remap of palette in current window
        InvalidateRect(hWnd,NULL,FALSE);
        return nRet;
        }
    break;

// This window may set the palette, even though it is not the
// currently active window.
case WM_PALETTECHANGED:
    // Don't do anything if the palette does not exist, or if
    // this is the window that changed the palette.
    if((hPalette != NULL) && ((HWND)wParam != hWnd))
        {
        // Select the palette into the device context
        SelectPalette(hDC,hPalette,FALSE);
        // Map entries to system palette
        RealizePalette(hDC);
        // Remap the current colors to the newly realized palette
        UpdateColors(hDC);
        return 0;
        }
    break;

// A menu command
case WM_COMMAND:
    {
    HANDLE hMenu = GetMenu(hWnd);
    switch(LOWORD(wParam))
        {
        // Exit the program
        case ID_FILE_EXIT:
            DestroyWindow(hWnd);
            break;
        // Put the day ,month ,time
        case ID_INPUT_TIME:
            DialogBox (hInstance,
                MAKEINTRESOURCE(IDD_DIALOG_TIME),
                hWnd,
                TimeDlgProc);
            InvalidateRect(hWnd,NULL,FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                UpdateWindow(hWnd);
                break;
            // Turn on Draw House on north
            case ID_OBJECT_HOUSE_NORTH:
                North = !North;
                if(North)
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_NORTH,MF_BYCOMMAND |
MF_CHECKED);
                else
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_NORTH,MF_BYCOMMAND |
MF_UNCHECKED);
                InvalidateRect(hWnd,NULL,FALSE);
                break;
            // Turn on Draw shading device on south
            case ID_OBJECT_HOUSE_SOUTH:
                South = !South;
                if(South)
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_SOUTH,MF_BYCOMMAND |
MF_CHECKED);
                else
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_SOUTH,MF_BYCOMMAND |
MF_UNCHECKED);
                InvalidateRect(hWnd,NULL,FALSE);
                break;
            // Turn on Draw shading device on WEST
            case ID_OBJECT_HOUSE_WEST:
                West = !West;
                if(West)
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_WEST,MF_BYCOMMAND |
MF_CHECKED);
                else
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_WEST,MF_BYCOMMAND |
MF_UNCHECKED);
                InvalidateRect(hWnd,NULL,FALSE);
                break;
            // Turn on Draw shading device on EAST
            case ID_OBJECT_HOUSE_EAST:
                East = !East;
                if(East)
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_EAST,MF_BYCOMMAND |
MF_CHECKED);
                else
                    CheckMenuItem(hMenu,ID_OBJECT_HOUSE_EAST,MF_BYCOMMAND |
MF_UNCHECKED);
                InvalidateRect(hWnd,NULL,FALSE);
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// BOX1 - BOX10
case ID_OBJECT_BOX:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX),
        hWnd,
        BoxDlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_BOX2:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX2),
        hWnd,
        Box2DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_BOX3:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX3),
        hWnd,
        Box3DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_BOX4:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX4),
        hWnd,
        Box4DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_BOX5:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX5),
        hWnd,
        Box5DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_BOX6:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_BOX6),
        hWnd,
        Box6DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_BOX7:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_BOX7),
            hWnd,
            Box7DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_BOX8:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_BOX8),
            hWnd,
            Box8DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_BOX9:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_BOX9),
            hWnd,
            Box9DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_BOX10:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_BOX10),
            hWnd,
            Box10DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;

    // CONE1 - CONE10
    case ID_OBJECT_CONE:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE),
            hWnd,
            ConeDlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE2:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE2),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            hWnd,
            Cone2DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE3:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE3),
            hWnd,
            Cone3DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE4:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE4),
            hWnd,
            Cone4DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE5:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE5),
            hWnd,
            Cone5DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE6:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE6),
            hWnd,
            Cone6DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE7:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE7),
            hWnd,
            Cone7DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE8:
        DialogBox (hInstance,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MAKEINTRESOURCE(IDD_DIALOG_CONE8),
            hWnd,
            Cone8DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE9:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE9),
            hWnd,
            Cone9DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CONE10:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CONE10),
            hWnd,
            Cone10DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;

    // CYLINDER1 - CYLINDER10
    case ID_OBJECT_CYLINDER:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER),
            hWnd,
            CylinderDlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER2:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER2),
            hWnd,
            Cylinder2DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER3:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER3),
            hWnd,
            Cylinder3DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    case ID_OBJECT_CYLINDER4:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER4),
            hWnd,
            Cylinder4DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER5:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER5),
            hWnd,
            Cylinder5DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER6:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER6),
            hWnd,
            Cylinder6DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER7:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER7),
            hWnd,
            Cylinder7DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER8:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER8),
            hWnd,
            Cylinder8DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER9:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER9),
            hWnd,
            Cylinder9DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_CYLINDER10:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_CYLINDER10),
            hWnd,
            Cylinder10DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;

    // PYRAMID1 - PYRAMID10
    case ID_OBJECT_PYRAMID:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID),
            hWnd,
            PyramidDlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID2:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID2),
            hWnd,
            Pyramid2DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID3:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID3),
            hWnd,
            Pyramid3DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID4:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID4),
            hWnd,
            Pyramid4DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID5:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID5),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            hWnd,
            Pyramid5DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID6:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID6),
            hWnd,
            Pyramid6DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID7:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID7),
            hWnd,
            Pyramid7DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID8:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID8),
            hWnd,
            Pyramid8DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID9:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID9),
            hWnd,
            Pyramid9DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;
    case ID_OBJECT_PYRAMID10:
        DialogBox (hInstance,
            MAKEINTRESOURCE(IDD_DIALOG_PYRAMID10),
            hWnd,
            Pyramid10DlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;

    // SPHERE1 - SPHERE10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case ID_OBJECT_SPHERE:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE),
        hWnd,
        SphereDlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_SPHERE2:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE2),
        hWnd,
        Sphere2DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_SPHERE3:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE3),
        hWnd,
        Sphere3DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_SPHERE4:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE4),
        hWnd,
        Sphere4DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_SPHERE5:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE5),
        hWnd,
        Sphere5DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
    break;
case ID_OBJECT_SPHERE6:
    DialogBox (hInstance,
        MAKEINTRESOURCE(IDD_DIALOG_SPHERE6),
        hWnd,
        Sphere6DlgProc);
    InvalidateRect(hWnd,NULL,FALSE);
    UpdateWindow(hWnd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
        case ID_OBJECT_SPHERE7:
            DialogBox (hInstance,
                MAKEINTRESOURCE(IDD_DIALOG_SPHERE7),
                hWnd,
                Sphere7DlgProc);
            InvalidateRect(hWnd,NULL,FALSE);
            UpdateWindow(hWnd);
            break;
        case ID_OBJECT_SPHERE8:
            DialogBox (hInstance,
                MAKEINTRESOURCE(IDD_DIALOG_SPHERE8),
                hWnd,
                Sphere8DlgProc);
            InvalidateRect(hWnd,NULL,FALSE);
            UpdateWindow(hWnd);
            break;
        case ID_OBJECT_SPHERE9:
            DialogBox (hInstance,MAKEINTRESOURCE(IDD_DIALOG_SPHERE9),
                hWnd,Sphere9DlgProc);
            InvalidateRect(hWnd,NULL,FALSE);
            UpdateWindow(hWnd);
            break;
        case ID_OBJECT_SPHERE10:
            DialogBox (hInstance,
                MAKEINTRESOURCE(IDD_DIALOG_SPHERE10),
                hWnd,Sphere10DlgProc);
            InvalidateRect(hWnd,NULL,FALSE);
            UpdateWindow(hWnd);
            break;

        // Display the about box
        case ID_HELP_ABOUT:
            DialogBox (hInstance,
                MAKEINTRESOURCE(IDD_DIALOG_ABOUT),
                hWnd,
                AboutDlgProc);
            break;
        }
    }
    break;


default:   // Passes it on if unproccessed
   return (DefWindowProc(hWnd, message, wParam, lParam));

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return (0L);
}


// Dialog Time procedure.
BOOL APIENTRY TimeDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{
    switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return FALSE;
            break;

         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Day = GetDlgItemInt(hDlg,IDC_EDIT_DAY,NULL,0);
                    Month = GetDlgItemInt(hDlg,IDC_EDIT_MONTH,NULL,0);
                    Hour = GetDlgItemInt(hDlg,IDC_EDIT_HOUR,NULL,0);
                    Min = GetDlgItemInt(hDlg,IDC_EDIT_MIN,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }

    return FALSE;
}

// Dialog Box procedure.
BOOL APIENTRY BoxDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {   switch (wParam)
            {
            case IDOK :
                {
                BoxX = GetDlgItemInt(hDlg,IDC_EDIT_BOX_X,NULL,0);
                BoxY = GetDlgItemInt(hDlg,IDC_EDIT_BOX_Y,NULL,0);
                BoxZ = GetDlgItemInt(hDlg,IDC_EDIT_BOX_Z,NULL,0);
                BoxWidth = GetDlgItemInt(hDlg,IDC_EDIT_BOX_WIDTH,NULL,0);
                BoxHeight = GetDlgItemInt(hDlg,IDC_EDIT_BOX_HEIGHT,NULL,0);
                BoxLength = GetDlgItemInt(hDlg,IDC_EDIT_BOX_LENGTH,NULL,0);

                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Box2 procedure.
BOOL APIENTRY Box2DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Box2X = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_X,NULL,0);
                    Box2Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_Y,NULL,0);
                    Box2Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_Z,NULL,0);
                    Box2Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_WIDTH,NULL,0);
                    Box2Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_HEIGHT,NULL,0);
                    Box2Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX2_LENGTH,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Box3 procedure.
BOOL APIENTRY Box3DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                Box3X = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_X,NULL,0);
                Box3Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_Y,NULL,0);
                Box3Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_Z,NULL,0);
                Box3Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_WIDTH,NULL,0);
                Box3Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_HEIGHT,NULL,0);
                Box3Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX3_LENGTH,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
             case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
             }
      }
          break;
    }
    return FALSE;
}

// Dialog Box4 procedure.
BOOL APIENTRY Box4DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Box4X = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_X,NULL,0);
                    Box4Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_Y,NULL,0);
                    Box4Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_Z,NULL,0);
                    Box4Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_WIDTH,NULL,0);
                    Box4Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_HEIGHT,NULL,0);
                    Box4Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX4_LENGTH,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Box5 procedure.
BOOL APIENTRY Box5DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Box5X = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_X,NULL,0);
                    Box5Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_Y,NULL,0);
                    Box5Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_Z,NULL,0);
                    Box5Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_WIDTH,NULL,0);
                    Box5Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_HEIGHT,NULL,0);
                    Box5Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX5_LENGTH,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Box6 procedure.
BOOL APIENTRY Box6DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Box6X = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_X,NULL,0);
                    Box6Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_Y,NULL,0);
                    Box6Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_Z,NULL,0);
                    Box6Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_WIDTH,NULL,0);
                    Box6Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_HEIGHT,NULL,0);
                    Box6Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX6_LENGTH,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Box7 procedure.
BOOL APIENTRY Box7DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {  switch (wParam)
               {
               case IDOK :
                  {
                  Box7X = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_X,NULL,0);
                  Box7Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_Y,NULL,0);
                  Box7Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_Z,NULL,0);
                  Box7Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_WIDTH,NULL,0);
                  Box7Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_HEIGHT,NULL,0);
                  Box7Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX7_LENGTH,NULL,0);
                  EndDialog(hDlg,TRUE);
                  }
                  break;
               case IDCANCEL:
                  EndDialog(hDlg,FALSE);
                  break;
               }
            }
            break;
    }
    return FALSE;
}

// Dialog Box8 procedure.
BOOL APIENTRY Box8DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {  switch (wParam)
               {
               case IDOK :
                  {
                  Box8X = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_X,NULL,0);
                  Box8Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_Y,NULL,0);
                  Box8Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_Z,NULL,0);
                  Box8Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_WIDTH,NULL,0);
                  Box8Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_HEIGHT,NULL,0);
                  Box8Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX8_LENGTH,NULL,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Box9 procedure.
BOOL APIENTRY Box9DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
          {
          case IDOK :
              {
              Box9X = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_X,NULL,0);
              Box9Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_Y,NULL,0);
              Box9Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_Z,NULL,0);
              Box9Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_WIDTH,NULL,0);
              Box9Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_HEIGHT,NULL,0);
              Box9Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX9_LENGTH,NULL,0);
              EndDialog(hDlg,TRUE);
              }
              break;
          case IDCANCEL:
              EndDialog(hDlg,FALSE);
              break;
          }
      }
          break;
    }
    return FALSE;
}

// Dialog Box10 procedure.
BOOL APIENTRY Box10DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                Box10X = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_X,NULL,0);
                Box10Y = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_Y,NULL,0);
                Box10Z = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_Z,NULL,0);
                Box10Width = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_WIDTH,NULL,0);
                Box10Height = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_HEIGHT,NULL,0);
                Box10Length = GetDlgItemInt(hDlg,IDC_EDIT_BOX10_LENGTH,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
             case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
             }
      }
          break;
   }
   return FALSE;
}

// Dialog Cone procedure.
BOOL APIENTRY ConeDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                ConeX = GetDlgItemInt(hDlg,IDC_EDIT_CONE_X,NULL,0);
                ConeY = GetDlgItemInt(hDlg,IDC_EDIT_CONE_Y,NULL,0);
                ConeZ = GetDlgItemInt(hDlg,IDC_EDIT_CONE_Z,NULL,0);
                ConeRadius = GetDlgItemInt(hDlg,IDC_EDIT_CONE_RADIUS,NULL,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ConeHeight = GetDlgItemInt(hDlg,IDC_EDIT_CONE_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cone2 procedure.
BOOL APIENTRY Cone2DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cone2X = GetDlgItemInt(hDlg,IDC_EDIT_CONE2_X,NULL,0);
                    Cone2Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE2_Y,NULL,0);
                    Cone2Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE2_Z,NULL,0);
                    Cone2Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE2_RADIUS,NULL,0);
                    Cone2Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE2_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
        }
            break;
    }
    return FALSE;
}

// Dialog Cone3 procedure.
BOOL APIENTRY Cone3DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
           return (TRUE);
           break;
       // Process command messages
      case WM_COMMAND:
           {  switch (wParam)
              {
              case IDOK :
                 {
                 Cone3X = GetDlgItemInt(hDlg,IDC_EDIT_CONE3_X,NULL,0);
                 Cone3Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE3_Y,NULL,0);
                 Cone3Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE3_Z,NULL,0);
                 Cone3Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE3_RADIUS,NULL,0);
                 Cone3Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE3_HEIGHT,NULL,0);
                 EndDialog(hDlg,TRUE);
                 }
                 break;
              case IDCANCEL:
                 EndDialog(hDlg,FALSE);
                 break;
              }
      }
           break;
   }
   return FALSE;
}

// Dialog Cone4 procedure.
BOOL APIENTRY Cone4DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
           return (TRUE);
           break;
       // Process command messages
      case WM_COMMAND:
           {  switch (wParam)
              {
              case IDOK :
                 {
                 Cone4X = GetDlgItemInt(hDlg,IDC_EDIT_CONE4_X,NULL,0);
                 Cone4Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE4_Y,NULL,0);
                 Cone4Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE4_Z,NULL,0);
                 Cone4Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE4_RADIUS,NULL,0);
                 Cone4Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE4_HEIGHT,NULL,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
        break;
    }
    return FALSE;
}

// Dialog Cone5 procedure.
BOOL APIENTRY Cone5DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
            {
            case IDOK :
              {
              Cone5X = GetDlgItemInt(hDlg,IDC_EDIT_CONE5_X,NULL,0);
              Cone5Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE5_Y,NULL,0);
              Cone5Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE5_Z,NULL,0);
              Cone5Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE5_RADIUS,NULL,0);
              Cone5Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE5_HEIGHT,NULL,0);
              EndDialog(hDlg,TRUE);
              }
              break;
            case IDCANCEL:
              EndDialog(hDlg,FALSE);
              break;
            }
        }
          break;
    }
    return FALSE;
}

// Dialog Cone6 procedure.
BOOL APIENTRY Cone6DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cone6X = GetDlgItemInt(hDlg,IDC_EDIT_CONE6_X,NULL,0);
                    Cone6Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE6_Y,NULL,0);
                    Cone6Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE6_Z,NULL,0);
                    Cone6Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE6_RADIUS,NULL,0);
                    Cone6Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE6_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
        }
            break;
    }
    return FALSE;
}

// Dialog Cone7 procedure.
BOOL APIENTRY Cone7DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cone7X = GetDlgItemInt(hDlg,IDC_EDIT_CONE7_X,NULL,0);
                    Cone7Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE7_Y,NULL,0);
                    Cone7Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE7_Z,NULL,0);
                    Cone7Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE7_RADIUS,NULL,0);
                    Cone7Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE7_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cone8 procedure.
BOOL APIENTRY Cone8DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
            return (TRUE);
            break;
       // Process command messages
      case WM_COMMAND:
            {  switch (wParam)
            {
            case IDOK :
              {
              Cone8X = GetDlgItemInt(hDlg,IDC_EDIT_CONE8_X,NULL,0);
              Cone8Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE8_Y,NULL,0);
              Cone8Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE8_Z,NULL,0);
              Cone8Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE8_RADIUS,NULL,0);
              Cone8Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE8_HEIGHT,NULL,0);
              EndDialog(hDlg,TRUE);
              }
              break;
            case IDCANCEL:
              EndDialog(hDlg,FALSE);
              break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cone9 procedure.
BOOL APIENTRY Cone9DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cone9X = GetDlgItemInt(hDlg,IDC_EDIT_CONE9_X,NULL,0);
                    Cone9Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE9_Y,NULL,0);
                    Cone9Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE9_Z,NULL,0);
                    Cone9Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE9_RADIUS,NULL,0);
                    Cone9Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE9_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Cone10 procedure.
BOOL APIENTRY Cone10DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cone10X = GetDlgItemInt(hDlg,IDC_EDIT_CONE10_X,NULL,0);
                    Cone10Y = GetDlgItemInt(hDlg,IDC_EDIT_CONE10_Y,NULL,0);
                    Cone10Z = GetDlgItemInt(hDlg,IDC_EDIT_CONE10_Z,NULL,0);
                    Cone10Radius = GetDlgItemInt(hDlg,IDC_EDIT_CONE10_RADIUS,NULL,0);
                    Cone10Height = GetDlgItemInt(hDlg,IDC_EDIT_CONE10_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cylinder procedure.
BOOL APIENTRY CylinderDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
            {
            case IDOK :
                {
                CylinderX = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER_X,NULL,0);
                CylinderY = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER_Y,NULL,0);
                CylinderZ = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER_Z,NULL,0);
                CylinderRadius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER_RADIUS,NULL,0);
                CylinderHeight = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
          break;
    }
    return FALSE;
}

// Dialog Cylinder2 procedure.
BOOL APIENTRY Cylinder2DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder2X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER2_X,NULL,0);
                    Cylinder2Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER2_Y,NULL,0);
                    Cylinder2Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER2_Z,NULL,0);
                    Cylinder2Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER2_RADIUS,NULL,0);
                    Cylinder2Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER2_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Cylinder3 procedure.
BOOL APIENTRY Cylinder3DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder3X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER3_X,NULL,0);
                    Cylinder3Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER3_Y,NULL,0);
                    Cylinder3Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER3_Z,NULL,0);
                    Cylinder3Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER3_RADIUS,NULL,0);
                    Cylinder3Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER3_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cylinder4 procedure.
BOOL APIENTRY Cylinder4DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
            {
            case IDOK :
                {
                Cylinder4X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER4_X,NULL,0);
                Cylinder4Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER4_Y,NULL,0);
                Cylinder4Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER4_Z,NULL,0);
                Cylinder4Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER4_RADIUS,NULL,0);
                Cylinder4Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER4_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cylinder5 procedure.
BOOL APIENTRY Cylinder5DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder5X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER5_X,NULL,0);
                    Cylinder5Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER5_Y,NULL,0);
                    Cylinder5Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER5_Z,NULL,0);
                    Cylinder5Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER5_RADIUS,NULL,0);
                    Cylinder5Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER5_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Cylinder6 procedure.
BOOL APIENTRY Cylinder6DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder6X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER6_X,NULL,0);
                    Cylinder6Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER6_Y,NULL,0);
                    Cylinder6Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER6_Z,NULL,0);
                    Cylinder6Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER6_RADIUS,NULL,0);
                    Cylinder6Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER6_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EndDialog(hDlg,FALSE);
                break;
            }
        }
        break;
    }
    return FALSE;
}

// Dialog Cylinder7 procedure.
BOOL APIENTRY Cylinder7DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                Cylinder7X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER7_X,NULL,0);
                Cylinder7Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER7_Y,NULL,0);
                Cylinder7Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER7_Z,NULL,0);
                Cylinder7Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER7_RADIUS,NULL,0);
                Cylinder7Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER7_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
             case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
             }
          }
          break;
    }
    return FALSE;
}

// Dialog Cylinder8 procedure.
BOOL APIENTRY Cylinder8DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder8X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER8_X,NULL,0);
                    Cylinder8Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER8_Y,NULL,0);
                    Cylinder8Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER8_Z,NULL,0);
                    Cylinder8Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER8_RADIUS,NULL,0);
                    Cylinder8Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER8_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
        }
            break;
    }
    return FALSE;
}

// Dialog Cylinder9 procedure.
BOOL APIENTRY Cylinder9DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Cylinder9X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER9_X,NULL,0);
                    Cylinder9Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER9_Y,NULL,0);
                    Cylinder9Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER9_Z,NULL,0);
                    Cylinder9Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER9_RADIUS,NULL,0);
                    Cylinder9Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER9_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Cylinder10 procedure.
BOOL APIENTRY Cylinder10DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
       case WM_COMMAND:
            {  switch (wParam)
               {
               case IDOK :
                  {
                  Cylinder10X = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER10_X,NULL,0);
                  Cylinder10Y = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER10_Y,NULL,0);
                  Cylinder10Z = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER10_Z,NULL,0);
                  Cylinder10Radius = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER10_RADIUS,NULL,0);
                  Cylinder10Height = GetDlgItemInt(hDlg,IDC_EDIT_CYLINDER10_HEIGHT,NULL,0);
                  EndDialog(hDlg,TRUE);
                  }
                  break;
               case IDCANCEL:
                  EndDialog(hDlg,FALSE);
                  break;
               }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid procedure.
BOOL APIENTRY PyramidDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    PyramidX = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID_X,NULL,0);
                    PyramidY = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID_Y,NULL,0);
                    PyramidZ = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID_Z,NULL,0);
                    PyramidWidth = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID_WIDTH,NULL,0);
                    PyramidHeight = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Pyramid2 procedure.
BOOL APIENTRY Pyramid2DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Pyramid2X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID2_X,NULL,0);
                    Pyramid2Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID2_Y,NULL,0);
                    Pyramid2Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID2_Z,NULL,0);
                    Pyramid2Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID2_WIDTH,NULL,0);
                    Pyramid2Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID2_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid3 procedure.
BOOL APIENTRY Pyramid3DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Pyramid3X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID3_X,NULL,0);
                    Pyramid3Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID3_Y,NULL,0);
                    Pyramid3Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID3_Z,NULL,0);
                    Pyramid3Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID3_WIDTH,NULL,0);
                    Pyramid3Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID3_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid4 procedure.
BOOL APIENTRY Pyramid4DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {   switch (wParam)
            {
            case IDOK :
                {
                Pyramid4X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID4_X,NULL,0);
                Pyramid4Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID4_Y,NULL,0);
                Pyramid4Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID4_Z,NULL,0);
                Pyramid4Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID4_WIDTH,NULL,0);
                Pyramid4Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID4_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid5 procedure.
BOOL APIENTRY Pyramid5DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Pyramid5X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID5_X,NULL,0);
                    Pyramid5Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID5_Y,NULL,0);
                    Pyramid5Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID5_Z,NULL,0);
                    Pyramid5Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID5_WIDTH,NULL,0);
                    Pyramid5Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID5_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid6 procedure.
BOOL APIENTRY Pyramid6DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
       case WM_COMMAND:
            {  switch (wParam)
               {
               case IDOK :
                  {
                  Pyramid6X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID6_X,NULL,0);
                  Pyramid6Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID6_Y,NULL,0);
                  Pyramid6Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID6_Z,NULL,0);
                  Pyramid6Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID6_WIDTH,NULL,0);
                  Pyramid6Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID6_HEIGHT,NULL,0);
                  EndDialog(hDlg,TRUE);
                  }
                  break;
               case IDCANCEL:
                  EndDialog(hDlg,FALSE);
                  break;
               }
       }
            break;
    }
    return FALSE;
}

// Dialog Pyramid7 procedure.
BOOL APIENTRY Pyramid7DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
       case WM_COMMAND:
            {  switch (wParam)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
            case IDOK :
                {
                Pyramid7X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID7_X,NULL,0);
                Pyramid7Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID7_Y,NULL,0);
                Pyramid7Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID7_Z,NULL,0);
                Pyramid7Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID7_WIDTH,NULL,0);
                Pyramid7Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID7_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid8 procedure.
BOOL APIENTRY Pyramid8DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
       case WM_COMMAND:
            {  switch (wParam)
                {
                case IDOK :
                    {
                    Pyramid8X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID8_X,NULL,0);
                    Pyramid8Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID8_Y,NULL,0);
                    Pyramid8Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID8_Z,NULL,0);
                    Pyramid8Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID8_WIDTH,NULL,0);
                    Pyramid8Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID8_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
        }
        return FALSE;
}

// Dialog Pyramid9 procedure.
BOOL APIENTRY Pyramid9DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                    Pyramid9X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID9_X,NULL,0);
                    Pyramid9Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID9_Y,NULL,0);
                    Pyramid9Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID9_Z,NULL,0);
                    Pyramid9Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID9_WIDTH,NULL,0);
                    Pyramid9Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID9_HEIGHT,NULL,0);
                    EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
        }
            break;
    }
    return FALSE;
}

// Dialog Pyramid10 procedure.
BOOL APIENTRY Pyramid10DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{   switch (message)
    {   // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
         // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case IDOK :
                {
                Pyramid10X = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID10_X,NULL,0);
                Pyramid10Y = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID10_Y,NULL,0);
                Pyramid10Z = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID10_Z,NULL,0);
                Pyramid10Width = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID10_WIDTH,NULL,0);
                Pyramid10Height = GetDlgItemInt(hDlg,IDC_EDIT_PYRAMID10_HEIGHT,NULL,0);
                EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
        break;
    }
    return FALSE;
}

// Dialog Sphere procedure.
BOOL APIENTRY SphereDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
            {
            case IDOK :
                {
                    SphereX = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE_X,NULL,0);
                    SphereY = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE_Y,NULL,0);
                    SphereZ = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE_Z,NULL,0);
                    SphereRadius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE_RADIUS,NULL,0);
                    EndDialog(hDlg,TRUE);
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
          break;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return FALSE;
}

// Dialog Sphere2 procedure.
BOOL APIENTRY Sphere2DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
           return (TRUE);
           break;
       // Process command messages
      case WM_COMMAND:
           {  switch (wParam)
              {
              case IDOK :
                 {
                    Sphere2X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE2_X,NULL,0);
                    Sphere2Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE2_Y,NULL,0);
                    Sphere2Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE2_Z,NULL,0);
                    Sphere2Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE2_RADIUS,NULL,0);
                    EndDialog(hDlg,TRUE);
                 }
                 break;
              case IDCANCEL:
                 EndDialog(hDlg,FALSE);
                 break;
              }
      }
           break;
   }
   return FALSE;
}

// Dialog Sphere3 procedure.
BOOL APIENTRY Sphere3DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
           return (TRUE);
           break;
       // Process command messages
      case WM_COMMAND:
           {  switch (wParam)
              {
              case IDOK :
                 {
                    Sphere3X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE3_X,NULL,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Sphere3Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE3_Y,NULL,0);
                Sphere3Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE3_Z,NULL,0);
                Sphere3Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE3_RADIUS,NULL,0);
                EndDialog(hDlg,TRUE);
            }
            break;
        case IDCANCEL:
            EndDialog(hDlg,FALSE);
            break;
        }
    }
        break;
  }
  return FALSE;
}

// Dialog Sphere4 procedure.
BOOL APIENTRY Sphere4DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
  {  // Initialize the dialog box
    case WM_INITDIALOG:
        return (TRUE);
        break;
     // Process command messages
    case WM_COMMAND:
        {  switch (wParam)
        {
        case IDOK :
            {
                Sphere4X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE4_X,NULL,0);
                Sphere4Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE4_Y,NULL,0);
                Sphere4Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE4_Z,NULL,0);
                Sphere4Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE4_RADIUS,NULL,0);
                EndDialog(hDlg,TRUE);
            }
            break;
        case IDCANCEL:
            EndDialog(hDlg,FALSE);
            break;
        }
    }
        break;
  }
  return FALSE;
}

// Dialog Sphere5 procedure.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BOOL APIENTRY Sphere5DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                   Sphere5X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE5_X,NULL,0);
                   Sphere5Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE5_Y,NULL,0);
                   Sphere5Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE5_Z,NULL,0);
                   Sphere5Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE5_RADIUS,NULL,0);
                   EndDialog(hDlg,TRUE);
                }
                break;
             case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
             }
      }
          break;
   }
   return FALSE;
}

// Dialog Sphere6 procedure.
BOOL APIENTRY Sphere6DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
   {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
             {
             case IDOK :
                {
                   Sphere6X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE6_X,NULL,0);
                   Sphere6Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE6_Y,NULL,0);
                   Sphere6Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE6_Z,NULL,0);
                   Sphere6Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE6_RADIUS,NULL,0);
                   EndDialog(hDlg,TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                break;
            case IDCANCEL:
                EndDialog(hDlg,FALSE);
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Sphere7 procedure.
BOOL APIENTRY Sphere7DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
          return (TRUE);
          break;
       // Process command messages
      case WM_COMMAND:
          {  switch (wParam)
            {
            case IDOK :
              {
                Sphere7X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE7_X,NULL,0);
                Sphere7Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE7_Y,NULL,0);
                Sphere7Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE7_Z,NULL,0);
                Sphere7Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE7_RADIUS,NULL,0);
                EndDialog(hDlg,TRUE);
              }
              break;
            case IDCANCEL:
              EndDialog(hDlg,FALSE);
              break;
            }
        }
          break;
    }
    return FALSE;
}

// Dialog Sphere8 procedure.
BOOL APIENTRY Sphere8DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
      case WM_INITDIALOG:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                        Sphere8X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE8_X,NULL,0);
                        Sphere8Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE8_Y,NULL,0);
                        Sphere8Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE8_Z,NULL,0);
                        Sphere8Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE8_RADIUS,NULL,0);
                        EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
                    break;
                }
            }
            break;
    }
    return FALSE;
}

// Dialog Sphere9 procedure.
BOOL APIENTRY Sphere9DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
        case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
        case WM_COMMAND:
            {   switch (wParam)
                {
                case IDOK :
                    {
                        Sphere9X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE9_X,NULL,0);
                        Sphere9Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE9_Y,NULL,0);
                        Sphere9Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE9_Z,NULL,0);
                        Sphere9Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE9_RADIUS,NULL,0);
                        EndDialog(hDlg,TRUE);
                    }
                    break;
                case IDCANCEL:
                    EndDialog(hDlg,FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            }
        }
            break;
    }
    return FALSE;
}

// Dialog Sphere10 procedure.
BOOL APIENTRY Sphere10DlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            return (TRUE);
            break;
        // Process command messages
       case WM_COMMAND:
            {  switch (wParam)
               {
               case IDOK :
                  {
                     Sphere10X = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE10_X,NULL,0);
                     Sphere10Y = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE10_Y,NULL,0);
                     Sphere10Z = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE10_Z,NULL,0);
                     Sphere10Radius = GetDlgItemInt(hDlg,IDC_EDIT_SPHERE10_RADIUS,NULL,0);
                     EndDialog(hDlg,TRUE);
                  }
                  break;
               case IDCANCEL:
                  EndDialog(hDlg,FALSE);
                  break;
               }
        }
            break;
    }
    return FALSE;
}

// Dialog About procedure.
BOOL APIENTRY AboutDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{  switch (message)
    {  // Initialize the dialog box
       case WM_INITDIALOG:
            {
            return (TRUE);
            }
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Process command messages
        case WM_COMMAND:
            {   // Validate and Make the changes
                if(LOWORD(wParam) == IDOK)
                    EndDialog(hDlg,TRUE);
            }
            break;
        // Closed from sysbox
        case WM_CLOSE:
            EndDialog(hDlg,TRUE);
            break;
    }
    return FALSE;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Ted J. Jansen, "Solar Engineering Technology", New Jersy: Prentice - Hall, 1985.
2. S.V. Szokolay, "Solar Ennergy and Building", 2nd .Ed.
3. Nathan and Ori Gurewich, "Visual C++ 5 in 21 Days", 4th.Ed., Sams Publishing,1997.
4. Richard S. Wright JR., Michael Sweet, "OpenGL Superbible", Waite Group Press™
5. ริชาร์ด เอส. ไวเนอร์ และ เลวิส เจ. พินสัน, "ภาษา C ++ เพื่อการใช้งาน", เรียบเรียงโดย ปรีดา ลิ้มถวิล
6. ธันวา ศรีประโมง, "ภาษาซี สำหรับวิศวกรรม" , พิมพ์ครั้งที่ 4 ,โครงการตำราวิชาการ มหาวิทยาลัยเทคโนโลยีมหานคร
7. ธนุพงษ์ เงาจารุวงศ์ , มรกต ศรีอนุชาต , "การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์" , ปริญญานิพนธ์ สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้