ห้องสมุดคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์

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

นาย ธนุพงษ์ เงาจารุวงศ์

MR. THANUPONG NGAOJARUWONG

นาย มรกต ศรีอนุชาต

MR. MORAKOT SRIANUJATA

วัน เดือน ปี 11 ธ.ค. 2541

เลขทะเบียน 038886

เลขเรียกหนังสือ T A01ภ0 ธ 271 ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038886

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

THANUPONG NGAOJARUWONG
MORAKOT SRIANUJATA

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
1997

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์ CHARACTERISTIC ANALYSIS OF SUNLIGHT UNDER SHADOW THROUGH SOME OBJECTS BY COMPUTER SIMULATION		
ชื่อนักศึกษา	นายธนุพงษ์ เงาจารุวงศ์	**รหัสประจำตัว**	37014159
	นายมรกต ศรีอนุชาต	**รหัสประจำตัว**	37014337
หลักสูตร	วิศวกรรมศาสตรบัณฑิต	**สาขาวิชา**	วิศวกรรมเครื่องกล

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(อ.มณฑล ใจกุศล)

(อ.สกนธ์ คล่องบุญจิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การศึกษาลักษณะของเงาที่เกิดจากแสงแดดส่องโดยคอมพิวเตอร์

นักศึกษา นายธนุพงษ์ เงาจารุวงศ์

นายมรกต ศรีอนุชาต

อาจารย์ที่ปรึกษา อ.มณฑล ใจกุศล

อ.สกนธ์ คล่องบุญจิต

หลักสูตร วิศวกรรมศาสตร์บัณฑิต

สาขาวิชา วิศวกรรมเครื่องกล

พ.ศ. 2540

บทคัดย่อ

ในปัจจุบันนี้ ทั่วโลกได้มีการตื่นตัวในเรื่องของการประหยัดพลังงานเป็นอย่างมาก และการนำความรู้เกี่ยวกับธรรมชาติมาใช้ประโยชน์ในการประหยัดพลังงาน ก็เป็นวิธีหนึ่งซึ่งในต่างประเทศได้มีการพัฒนามาเป็นเวลานานแล้ว ถึงแม้ว่าผลที่ได้จะยังไม่เป็นที่น่าพอใจ แต่อย่างน้อยก็ถือว่าวิธีนี้เป็นแนวทางหนึ่ง ที่ช่วยในการประหยัดพลังงานได้ สำหรับในประเทศไทยนั้น วิธีการนี้ยังไม่เป็นวิธีที่ได้รับความนิยมมากนัก ดังนั้นจึงได้มีการนำความรู้ทางด้านนี้มาพัฒนา และประยุกต์ใช้ โดยโครงงานนี้จะเป็นการพัฒนาโปรแกรมคอมพิวเตอร์ที่ใช้ในการคำนวณหาตำแหน่งของดวงอาทิตย์ ในวันและเวลาที่ต้องการ เพื่อที่จะศึกษาลักษณะของเงาที่เกิดจากแสงแดดที่ส่องมา และจำลองลักษณะของเงาที่เกิดขึ้น พร้อมทั้งแสดงผลทางหน้าจอคอมพิวเตอร์ ซึ่งวิธีการนี้สามารถนำไปประยุกต์ใช้ เพื่อการพัฒนาการวางตำแหน่งของอาคาร หรือออกแบบสิ่งก่อสร้าง และอาจนำไปประยุกต์ใช้ในด้านอื่นได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Characteristic Analysis of Sunlight under Shadow through Some Object by Computer Simulation
Student	Mr.Thanupong Ngaojaruwong
	Mr.Morakot Srianujata
Thesis Advisor	Mr.Monthon Jaikusol
	Mr.Sakon Klongboonjit
Degree	Bachelor of Engineering in Mechanical Engineering
Year	1997

ABSTRACT

During this time, people all over the world become much more aware of the energy saving. One of many ways to save energy is to use the knowledge about the nature, which has been developed for quite a long time in other countries. Although the result isn't quite satisfactory, it at least is one of many ways that helps saving energy. For Thailand, very few groups of people are thinking about bringing the knowledge about the nature 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 in order to study the shape of the shadow of the sunlight and to make a model of its shapes which will be resulted on 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. พงษ์ศักดิ์ คำมูล คุณสุรีย์ รอดแจ่ม คุณวาสนา ม่วงโพธิ์ และคุณมณฑา เทียมเมือง ที่พร้อมคอยให้ความช่วยเหลือในด้านต่างๆ

ขอขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ช่วยเหลือ และให้คำแนะนำต่างๆ พร้อมทั้งช่วยตำหนิ ตักเตือนถึงข้อผิดพลาด จนสำเร็จสมบูรณ์ยิ่งขึ้น และยังให้กำลังใจต่อผู้วิจัย ทำให้การทำงานสนุก ไม่เครียดมาก จนสำเร็จได้

สุดท้ายขอขอบคุณภาควิชาวิศวกรรมเครื่องกล ที่ได้ให้ทุนสนับสนุนในการทำปริญญานิพนธ์ครั้งนี้

ธนุพงษ์ เงาจารุวงศ์

มรกต ศรีอนุชาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....V

สารบัญภาพ.....VI

บทที่ 1 บทนำ.....1

บทที่ 2 ทฤษฎีเบื้องต้นเกี่ยวกับการเกิดเงา.....2

2.1 ความสัมพันธ์ของตำแหน่งดวงอาทิตย์กับโลก.....2

2.2 การคำนวณในเรื่องเวลา.....9

2.3 เส้นทางการเดินของดวงอาทิตย์ (Sun Path Diagram).....11

2.3.1 การสร้างเส้นทางเดินของดวงอาทิตย์.....13

2.4 วิธีการและอุปกรณ์ป้องกันแสงแดด.....14

2.4.1การลดพื้นที่รับแสงแดด.....14

2.4.2 การออกแบบอุปกรณ์บังแดด.....14

2.4.3 วิธีการหาเงา.....15

บทที่ 3 แนวคิดและการทำงานของโปรแกรม.....16

บทที่ 4 เครื่องมืออุปกรณ์ การทดลอง และการคำนวณ.....21

บทที่ 5 เปรียบเทียบผลที่ได้จากการทดลองกับจากโปรแกรม.....23

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....29

ภาคผนวก Source code.....32

แผนภูมิอธิบายการทำงานของ Function ต่าง ๆ ในโปรแกรม.....108

Sun path Diagram สำหรับประเทศไทย.....123

บรรณานุกรม.....124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

4.1 ตารางแสดงการเปรียบเทียบผลของมุมที่วัดได้จริงกับที่ได้จากการคำนวณ.........22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงความสัมพันธ์ของการเคลื่อนที่ของโลกรอบดวงอาทิตย์...2

2.2 แสดงความสัมพันธ์และแสดงตำแหน่งของเส้นอาร์คติค (Arctic Circle) และเส้นแอนตาร์คติค (Antarctic Circle)...3

2.3 แสดงลักษณะของ Latitude L, Hour angle, Declination δ...4

2.4 แสดงค่าของมุม Declination δ ตลอดทั้งปี...5

2.5 แสดงเส้นทางเดินของดวงอาทิตย์ และมุม Zenith Z, Altitude angle α, Azimuth angle ϕ...7

2.6 แสดงมุมที่เกิดจากพื้นผิว ได้แก่มุม Tilt angle **S**, Surface Azimuth angle Ψ, Incedent angle **I**...7

2.7 แสดง Equation of Time ซึ่งเป็นฟังชั่น ของเวลาภายในหนึ่งปี...11

2.8 แสดงเส้นทางเคลื่อนที่พาดผ่านบนท้องฟ้า ของดวงอาทิตย์ในระหว่างปี...11

2.9 แสดง Sun Path Diagram ของประเทศไทย...12

2.10 แสดงมุม Profile angle β และความสัมพันธ์กับมุม Altitude angle α Azimuth angle ϕ และ Surface azimuth angle Ψ...15

3.1 แสดงลักษณะทั่วไปของตัวโปรแกรม...17

3.2 แสดงการป้อนค่าวันและเวลาที่ต้องการ...18

3.3 แสดงตัวอย่างลักษณะของเงาในวันที่ 2 พฤศจิกายน เวลา 10.15 น...18.

3.4 แสดงลักษณะเงาเมื่อมีการติดอุปกรณ์บังแดด...19

3.5 แสดงลักษณะเมื่อทำการเปลี่ยนแปลงรูปทรง...19

3.6 แสดงเมื่อรูปทรงที่เปลี่ยนไปได้ติดอุปกรณ์บังแดด...20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เนื่องจากในทุกวันนี้วิศวกร และสถาปนิก ส่วนใหญ่ในประเทศไทย ยังไม่ได้ให้ความสำคัญกับการศึกษาลักษณะของเงาที่เกิดขึ้น แต่จะให้ความสำคัญในความเชื่อเกี่ยวกับเรื่องของฮวงจุ้ยมากกว่า จึงทำให้การนำความรู้เรื่องนี้ไปใช้เพื่อประโยชน์ในการกันแดด และประหยัดพลังงานให้กับอาคารที่ไม่มีเครื่องปรับอากาศ และอาคารที่มีเครื่องปรับอากาศได้ไม่ดีเท่าที่ควร

สำหรับการที่จะออกแบบแผงกันแดด การตัดสินใจติดกระจกกรองแสงในด้านไหน การวางตำแหน่งต้นไม้ หรือการจะหันตัวอาคารไปในทิศทางใด อย่างไร จะต้องพิจารณาโดยดูจากทั้งทิศทางแดด และทิศทางลม ตำแหน่งและทางเดินของดวงอาทิตย์ โดยต้องพิจารณาว่าจะวางตำแหน่งอาคารอย่างไร จึงจะทำให้รังสีที่แผ่จากดวงอาทิตย์เข้าได้น้อยที่สุดในฤดูร้อน และให้ได้รับลมเต็มที่ โดยแผงกันแดดที่ออกแบบขึ้นนี้ จะช่วยลดแสงจ้าและความร้อนซึ่งอาจจะมาจากดวงอาทิตย์โดยตรง หรือสะท้อนจากพื้นดินและอาคารข้างเคียง สำหรับการออกแบบแผงบังแดดลักษณะต่าง ๆ จะให้ผลกับรูปลักษณะของตัวอาคารด้วย เช่นการเลือกใช้ชนิดของที่บังแดด คืออาจจะทำเป็นบานเกล็ดเล็กหรือใหญ่ หนา บาง แท่งตัน ตรงหรือ เอียง แยกจากตัวอาคาร หรือเป็นส่วนหนึ่งของตัวอาคาร ซึ่งนอกจากจะเป็นส่วนเสริมให้อาคารมีความงามเพิ่มขึ้นจนเป็นที่สะดุดตาของผู้ที่พบเห็นแล้ว ยังจะได้ประโยชน์จากการกันแดดกับการประหยัดพลังงานด้วย

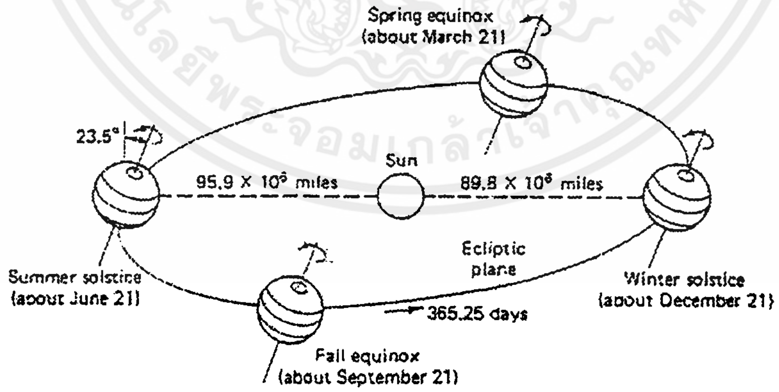
โดยจะทำการศึกษาสมการต่างๆ ที่ใช้ศึกษาถึงเส้นทางเดิน และหาตำแหน่งของดวงอาทิตย์ ในวันและเวลาต่างๆ ที่ต้องการ โดยจะบอกเป็นค่าของมุมต่างๆ เพื่อที่จะนำไปหาลักษณะของเงาที่เกิดขึ้นในวันและเวลานั้น เพื่อที่จะนำไปสร้างแบบจำลองของเงาด้วยคอมพิวเตอร์ โดยเงาและเส้นทางเดินของดวงอาทิตย์จะได้จากการคำนวณ โดยในขั้นต้นนี้เป็นส่วนเริ่มต้น ดังนั้นแบบจำลองของวัตถุจะเป็นแค่ กล่องสี่เหลี่ยมลูกบาศก์ และรูปทรงปิรามิด ทั้งนี้เพื่อเป็นแนวทางให้เกิดแนวความคิดที่จะมีการพัฒนา ให้เปลี่ยนเป็นรูปทรงอื่นๆ ซึ่งคล้ายวัตถุจริงได้ ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้นเกี่ยวกับการเกิดเงา

2.1 ความสัมพันธ์ของตำแหน่งดวงอาทิตย์กับโลก (Sun Positions in Relation To the Earth)

วงโคจรของโลกรอบดวงอาทิตย์เป็นรูปวงรีซึ่งมีระยะทางระหว่างจุดศูนย์กลางทั้งสอง ประมาณ 5 X 1.8^8 Km (9.3 X 10^7 miles) รูปที่ 2.1 แสดงความสัมพันธ์ของตำแหน่งดวงอาทิตย์กับโลกตลอดระยะเวลา 1 ปี ขณะที่โลกหมุนรอบตัวเองจะใช้เวลา 1 วัน และหมุนรอบดวงอาทิตย์จะใช้เวลา 1 ปี ส่วนดวงอาทิตย์จะหมุนรอบตัวเองจะใช้เวลาประมาณ 1 เดือน แกนหมุนของโลกจะเอียงทำมุม 23.5° กับแนวตั้งฉากของระนาบวงโคจรในการอ้างอิงระนาบของการโคจรทำได้โดย ถ้าซีกโลกเหนือเอียงไปทางดวงอาทิตย์ในฤดูร้อนและในทางกลับกันถ้าซีกโลกใต้เอียงไปทางดวงอาทิตย์จะเป็นฤดูหนาว ด้วยเหตุนี้จึงเกิดฤดูต่าง ๆ บนโลก

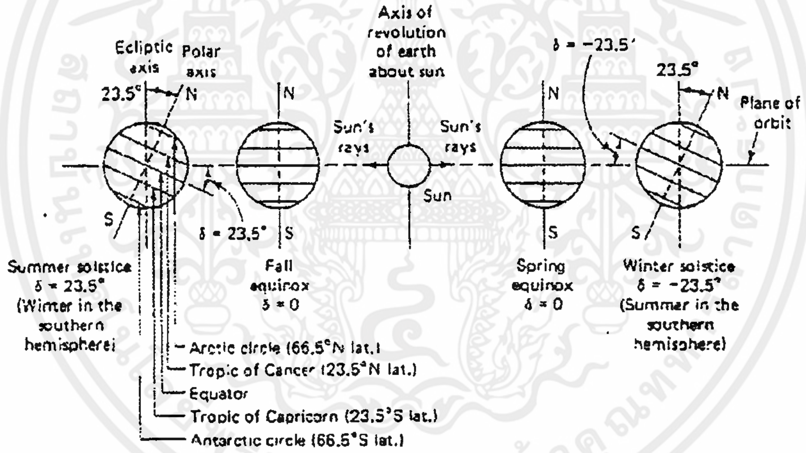


รูปที่ 2.1 แสดงความสัมพันธ์ของการเคลื่อนที่ของโลกรอบดวงอาทิตย์

เส้นอาร์คติค (Arctic Circle) **และเส้นแอนตาร์คติค** (Antarctic Circle) **เส้นทรอปิคออฟ-แคนเซอร์** (Tropic of Cancer) **และเส้นทรอปิคออฟแคพริคอน** (Tropic of Capricorn) **มีความ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมพันธ์กับดวงอาทิตย์ใน 4 ฤดูซึ่งแสดงดังรูปที่ 2.2 โดยในฤดูหนาวดวงอาทิตย์จะอยู่ห่างจากโลกที่สุดซึ่งอยู่ระมาณวันที่ 21 ของเดือนธันวาคม และเป็นช่วงที่ขั้วโลกเหนือจะเอียงหนีออกจากดวงอาทิตย์ 23.5° ด้วยเหตุนี้บริเวณเหนือเส้นอาร์คติค (Arctic Circle) จะมืดโดยสมบูรณ์ และบริเวณใต้เส้นแอนตาร์คติค (Antarctic Circle) จะสว่างโดยตลอด ส่วนในฤดูร้อนดวงอาทิตย์จะอยู่ห่างจากโลกที่สุดเช่นกันซึ่งจะอยู่ประมาณวันที่ 21 ของเดือนมิถุนายน ซึ่งขั้วโลกเหนือเอียงเข้าหาดวงอาทิตย์ 23 5° ด้วยเหตุนี้บริเวณเหนือเส้นอาร์คติค (Arctic Circle) จะสว่างตลอด และบริเวณใต้เส้นแอนตาร์คติค (Antarctic Circle) จะมืดโดยสมบูรณ์ ส่วนในฤดูใบไม้ร่วงจะมีกลางวันเท่ากับกลางคืนซึ่งจะอยู่ประมาณวันที่ 21 ของเดือนมีนาคม และวันที่ 21 ของเดือนกันยายน ตามลำดับ ซึ่งขั้วโลกเหนือและขั้วโลกใต้จะมีระยะห่างจากดวงอาทิตย์เท่ากัน ด้วยเหตุนี้ทุกแห่งบนโลกจะมีกลางวัน 12 ชั่วโมงและกลางคืน 12 ชั่วโมงเท่ากัน

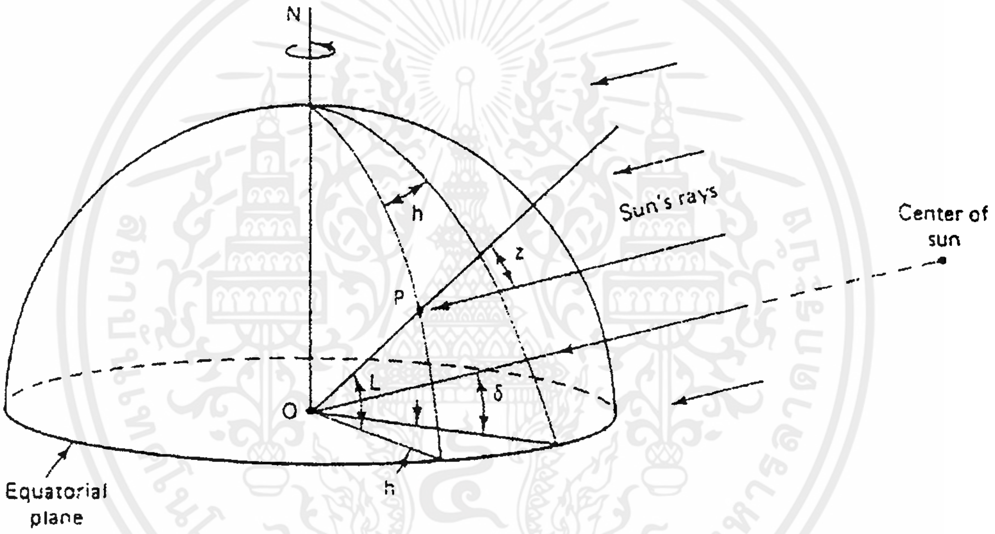


รูปที่ 2.2 แสดงความสัมพันธ์และแสดงตำแหน่งของเส้นอาร์คติค (Arctic Circle) และเส้นแอนตาร์คติค (Antarctic Circle)

ด้วยเหตุนี้รังสีที่ส่องมาบนโลกจึงเป็นฟังชันก์ทางเรขาคณิตเราจึงเลี่ยงไม่ได้ที่จะศึกษามุมพื้นฐานต่างๆ ที่เกิดขึ้นดังแสดงได้เป็นความสัมพันธ์ระหว่างดวงอาทิตย์กับผิวโลกดังรูปที่ 2.3 โดยกำหนดให้จุด P เป็นจุดที่อยู่บนผิวโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละติจูด (Latitude) L คือ มุมจากระนาบเส้นศูนย์สูตรถึงจุด P ละติจูด (Latitude) ที่อยู่เหนือเส้นศูนย์สูตรจะกำหนดให้มีค่าเป็นบวก จากรูปที่ 2.3 ละติจูดจะเป็นมุมระหว่างเส้น OP (โดยที่ O เป็นจุดศูนย์กลางของโลก) กับโพรเจคชั่นของเส้น OP บนระนาบเส้นศูนย์สูตรจากรูปที่ 2.1(b) จะพบว่าเส้นอาร์คติค (Arctic Circle) และเส้นแอนตาร์คติค (Antarctic Circle) จะมีค่าเท่ากับละติจูด 66.5° เหนือและใต้ตามลำดับ และเส้นทรอปิคออฟแคนเซอร์ (Tropic of Cancer) กับเส้นทรอปิคออฟแคพริคอน (Tropic of Capricorn) จะมีค่าเท่ากับละติจูด 23 5° เหนือและใต้ตามลำดับ



รูปที่ 2.3 แสดงลักษณะของ Latitude L, Hour angle, Declination δ

Declination δ คือ มุมที่รังสีของดวงอาทิตย์กระทำกับระนาบของเส้นศูนย์สูตร กำหนดให้มุม Declination δ ที่เอียงไปทางขั้วโลกเหนือมีค่าเป็นบวก จากรูปที่ 2.3 มุม Declination δ เป็นมุมระหว่างเส้นจากจุดศูนย์กลางโลก ถึงจุดศูนย์กลางของดวงอาทิตย์กระทำกับโพรเจคชั่นของเส้นนี้บนระนาบเส้นศูนย์สูตร จากรูปที่ 2.2 ค่าของมุม Declination จะมีค่าตั้งแต่ 0° ในฤดูใบไม้ผลิ (วันที่มีกลางวันเท่ากับกลางคืน) ถึง + 23.5° ในฤดูร้อน (วันที่ดวงอาทิตย์อยู่ไกล้โลกที่สุด) และจาก 0° ในฤดูใบไม้ร่วง (วันที่มีกลางวันเท่ากับกลางคืน) ถึง - 23.5° ในฤดูหนาว (วันที่ดวง-

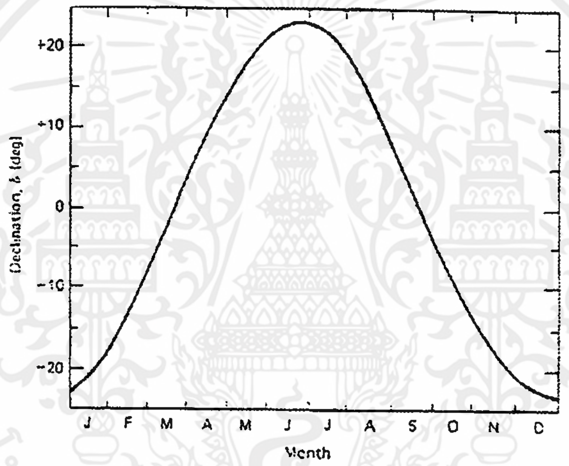
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาทิตย์อยู่ไกลจากโลกที่สุด) โดยที่ค่าการเปลี่ยนแปลงของมุม Declination ตลอดปีจะสามารถแสดงได้ดังในรูปที่ 2.4 และสูตรสำหรับการคำนวณคือ

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

เมื่อ n เป็นจำนวนวันใน 1 ปี

ในการคำนวณเราจะใช้ค่า Declination คงที่ตลอดทั้งวัน



รูปที่ 2.4 แสดงค่าของมุม Declination δ ตลอดทั้งปี

Hour Angle h คือ มุมระหว่างจุดใด ๆ บนเส้นแวงกับทิศทางของดวงอาทิตย์บนโลกในรูปที่ 2.3 Hour angle ของจุด P คือ มุมที่วัดจากเส้นโพรเจคชั่น OP บนระนาบเส้นศูนย์สูตร กับเส้นโพรเจคชั่นของเส้นผ่านศูนย์กลางระหว่างโลกกับดวงอาทิตย์บนระนาบเส้นศูนย์สูตร โดยที่ ณ. Solar Noon ของแต่ละท้องถิ่นจะมีค่า Hour Angle เป็นศูนย์ และจากการที่โลกหมุนรอบตัวเองใช้เวลา 24 ชั่วโมง และหมุนไปเป็นมุมเท่ากับ 360° เพราะฉะนั้น 1 ชั่วโมงของโลกจะหมุนไปเป็นมุม 15° ของลองติจูด (Longtitude) ส่วนในตอนบ่ายเราจะกำหนดให้ค่า Hour Angle มีค่าเป็นบวก เช่น $h = -30^\circ$ จะได้เวลาเท่ากับ 10 A.M., $h = +30^\circ$ ก็จะได้เวลาเท่ากับ 2 P.M. ส่วนสูตรสำหรับการคำนวณคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เมื่อ ค่า + แสดงว่าเป็นตอนบ่าย และ ค่า - แสดงว่าเป็นตอนเช้า

ส่วนในการบอกตำแหน่งของดวงอาทิตย์จะต้องใช้มุมใหม่เพิ่มขึ้นอีก 3 มุมจาก 3 มุมพื้นฐานข้างต้นเป็นตัวบอกตำแหน่งของดวงอาทิตย์ และจะมีความสำคัญในการคำนวณด้วย ซึ่งแสดงในรูปที่ 2.5 โดยโลกจะหมุนไปทางทิศตะวันออก ส่วนดวงอาทิตย์จะปรากฏโดยการสังเกตุบนโลกจะเห็นว่าดวงอาทิตย์จะเคลื่อนที่บนท้องฟ้าในลักษณะเป็นเส้นโค้งโดยจะค่อย ๆ ขึ้น จากทิศตะวันออกจนกระทั่งถึงเส้น meridian ในตอนกลางวัน หลังจากนั้นจะคล้อยไป ทางทิศตะวัน-ตก

1. Zenith angle Z คือ มุมระหว่างรังสีดวงอาทิตย์กระทำในแนวดิ่ง หน่วยเป็นองศา

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

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

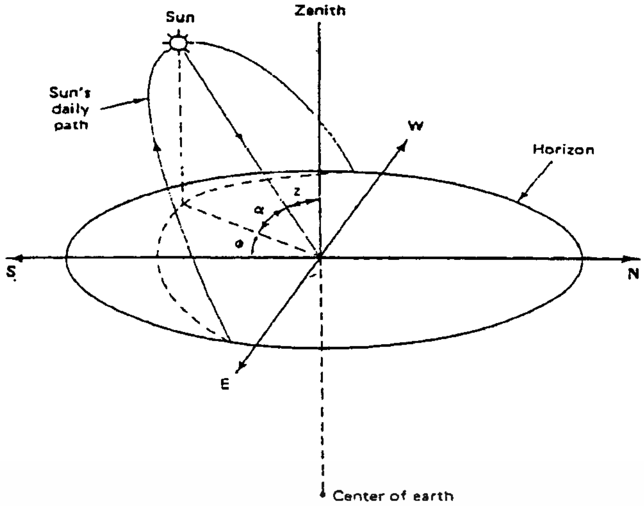
3. Azimuth angle ϕ คือ มุมของรังสี ดวงอาทิตย์ที่วัดจากทิศใต้ไปทางทิศตะวันตกในแนวระนาบแนวนอน จะมีค่าเป็นบวกสำหรับซีกโลกเหนือ ส่วนสำหรับซีกโลกใต้จะวัดจากทิศเหนือไปทางทิศตะวันออกในแนวระนาบแนวนอนมีค่าเป็นบวก หน่วยเป็นองศา

โดยมุมทั้ง 3 สามารถจัดให้อยู่ในรูปของมุมพื้นฐานโดยใช้หลักการของตรีโกณมิติ ซึ่งมีความสัมพันธ์กันดังนี้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



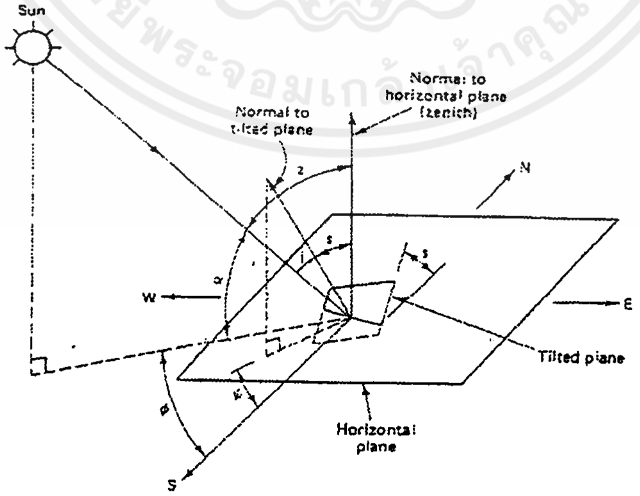
รูปที่ 2.5 แสดงเส้นทางเดินของดวงอาทิตย์ และมุม Zenith angle Z, Altitude angle α, Azimuth angle ϕ

สำหรับรังสีของดวงอาทิตย์บนพื้นผิวเอียง จำเป็นที่จะต้องกำหนดมุมมากกว่า 3 มุมให้สัมพันธ์กับพื้นผิวดังแสดงในรูปที่ 2.5 อันได้แก่

1. Tilt angle S คือ มุมระหว่างพื้นผิวกับระนาบแนวนอน

2. Surface angle ψ คือ มุมระหว่างเส้นตั้งฉากกับพื้นผิวกับทิศใต้วัดในแนวระดับโดยมีค่าเป็นบวกเมื่อวัดไปทางทิศตะวันตก

3. Incident angle I คือ มุมระหว่างรังสีดวงอาทิตย์กับเส้นตั้งฉากพื้นผิว ถ้าพื้นผิวกับระนาบแนวระดับวางทับกันจะได้ว่ามุม Incident angle I และมุม Zenith angle Z จะเท่ากัน



รูปที่ 2.6 แสดงมุมที่เกิดจากพื้นผิว ได้แก่มุม Tilt angle S, Surface Azimuth angle ψ, Incident angle I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มุมที่กำหนดในรูปที่ 2.5 สามารถให้อยู่ในรูปของมุมพื้นฐานที่ได้กำหนดไว้ในรูปที่ 2.2 ตามข้างล่างนี้

$$\cos i = \sin L \sin \delta \cos s - \cos L \sin \delta \sin s \cos \psi + \cos L \cos \delta \cosh \cos s$$

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

สำหรับพื้นผิวที่มีระนาบเดียวกับระนาบแนวระดับจะได้ว่า **S** = 0 และ **I** = **Z** ฉะนั้น สมการที่ (2-6) จะลดรูปลง เหลือดังสมการที่ (2-4)

สำหรับพื้นผิวที่มีระนาบ ๆ ในแนวดิ่งจะได้ว่า **S** = 90° แทนลงในสมการที่ (2-6)

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

สำหรับพื้นผิวที่เอียงหันหน้าไปทางทิศใต้ในซีกโลกเหนือ จะได้ว่า ψ = 0° จะทำให้สมการที่ (2-6) ลดรูปลงเหลือ

$$\cos i = \sin L \sin \delta \cos s - \cos L \sin \delta \sin s + \cos L \cos \delta \cosh \cos s$$

$$+ \sin L \cos \delta \cosh \sin s \qquad (2\text{-}8)$$

$$= \sin(L - s) \sin \delta + \cos(L - s) \cos \delta \cosh$$

สำหรับพื้นผิวที่เอียงหันหน้าไปทางทิศเหนือในซีกโลกใต้จะได้ว่า ψ = 180° สมการที่ (2-6) จะลดรูปลงเหลือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บ่อยครั้งจำเป็นต้องหาเวลาที่ดวงอาทิตย์ขึ้นกับเวลาที่ดวงอาทิตย์ตก และความยาวของเวลาในแต่ละวัน และกำหนดให้ Hour angle ตอนพระอาทิตย์ขึ้นให้เป็น h_s ($\alpha = 0$) โดยจากสมการที่ (2-4) จะได้ว่า

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

เพราะฉะนั้น

$$\cos h_s = -\tan L \tan\delta \qquad (2\text{-}10)$$

เมื่อ h_s เป็นบวกเมื่อพระอาทิตย์ตกและจะมีค่าเป็นศูนย์ ณ. Solar noon ของ แต่ละท้องถิ่น หรือ 15° ของลองติจูด (Longtitude) เท่ากับ 1 ชั่วโมง เวลาที่ดวงอาทิตย์ขึ้นและตกจากตำแหน่ง Solar noon แต่ละท้องถิ่นสามารถหาได้จาก

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

และความยาวในแต่ละวันหาได้จาก

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

2.2 <u>การคำนวณในเรื่องเวลา (Reckoning Of Time)</u>

ในการคำนวณเกี่ยวกับการแผ่รังสีของดวงอาทิตย์ (Solar radiation) เราจะใช้ Solar time เป็นตัวแสดงเวลาของวัน โดย Solar time เป็นพื้นฐานในการคำนวณหามุมการเคลื่อนที่ของดวงอาทิตย์ ที่พาดผ่านท้องฟ้า ส่วน Solar noon จะเป็นเวลาที่ดวงอาทิตย์พาดผ่านเส้น meridian ของผู้สังเกต โดยในแต่ละท้องถิ่นมักจะไม่เกิดขึ้นตรงตามเวลาในท้องถิ่นนั้นๆ ส่วนเวลามาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Standard Time) นั้นเราจะได้จากการเลือกเส้น meridian (เส้นที่ลากจากขั้วโลกเหนือไปยังขั้ว-โลกใต้) ที่ใกล้จุดศูนย์กลางของโซนเวลาที่ทุกๆ 15° ของ Longtitude และเนื่องจากในขณะที่ดวง-อาทิตย์เคลื่อนที่ไป 1° ของลองติจูด (Longtitude) จะใช้เวลาไป 4 นาทีเพราะฉะนั้นเทอมลองติจูดที่ถูกต้อง (Longtitude correction term) จะอยู่ในรูปของ $4(\text{long}_{\text{standard}} - \text{long}_{\text{local}})$ ซึ่งควรที่จะต้องเพิ่มเข้าไปในเวลามาตรฐาน (standard time) เพื่อที่จะได้เวลาที่ถูกต้อง ของแต่ละท้องถิ่น นอกจากนี้ในการเปลี่ยนแปลงความเร็วในการโคจรของโลกเป็นรูปวงรีรอบดวงอาทิตย์ ก็จะทำให้เวลาตอนที่ดวงอาทิตย์เคลื่อนที่พาดผ่านเส้น meridian มีการเปลี่ยนแปลงตลอดปี จึงจำเป็นที่จะต้องกำหนดแฟคเตอร์ซึ่งเรียกว่า equation of time ไปช่วยในการคำนวณ Solar time ฉะนั้นความสัมพันธ์ระหว่าง Solar time กับ Standard time มีดังนี้

$$\text{solar time} = \text{standard time} + 4(\text{long}_{\text{standard}} - \text{long}_{\text{local}}) + \text{equation of time} \quad (2\text{-}12)$$

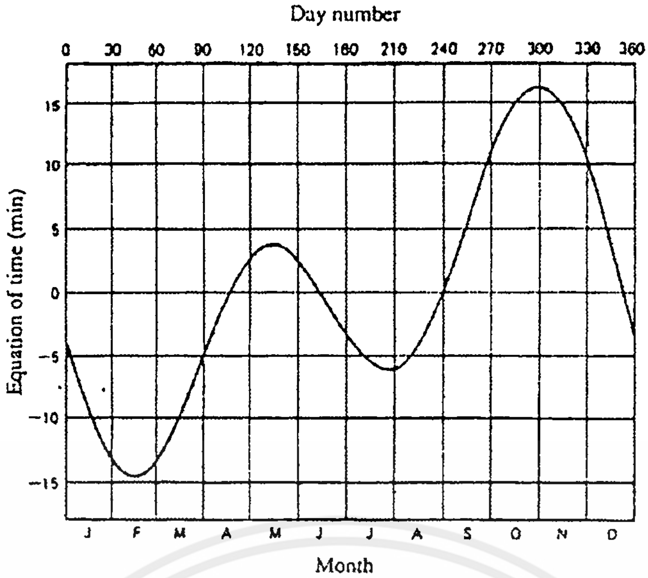
เมื่อ Longtitude correction term มีหน่วยเป็นนาที, $\text{Long}_{\text{standard}}$ เป็นค่า Standard meridian สำหรับโซนเวลา มีหน่วยเป็น องศา และ $\text{Long}_{\text{local}}$ เป็นค่า Longtitude ของท้องถิ่น มีหน่วยเป็นองศาตะวันตก ค่าของ equation of time เป็นฟังชันก์ของเวลาใน 1 ปี ซึ่งแสดงในรูปที่ 2.6 และมีหน่วยเป็นนาที และ สามารถหาได้จาก

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

เมื่อ $B = \frac{360}{364}(n-81)$

n = จำนวนวันใน 1 ปี ($1 < n < 365$)

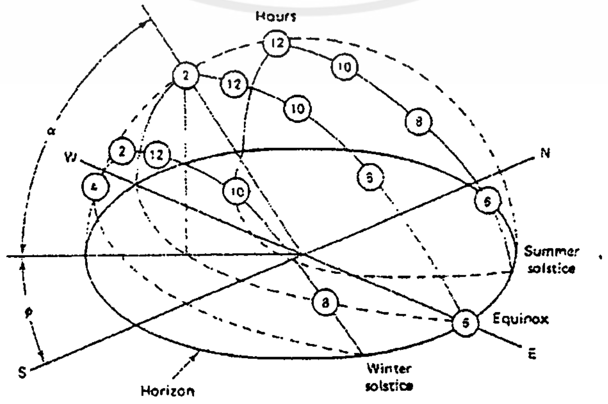
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดง Equation of Time ซึ่งเป็นฟังชั่นก์ของเวลาภายในหนึ่งปี

2.3 <u>เส้นทางการเดินของดวงอาทิตย์ (Sun Path Diagram)</u>

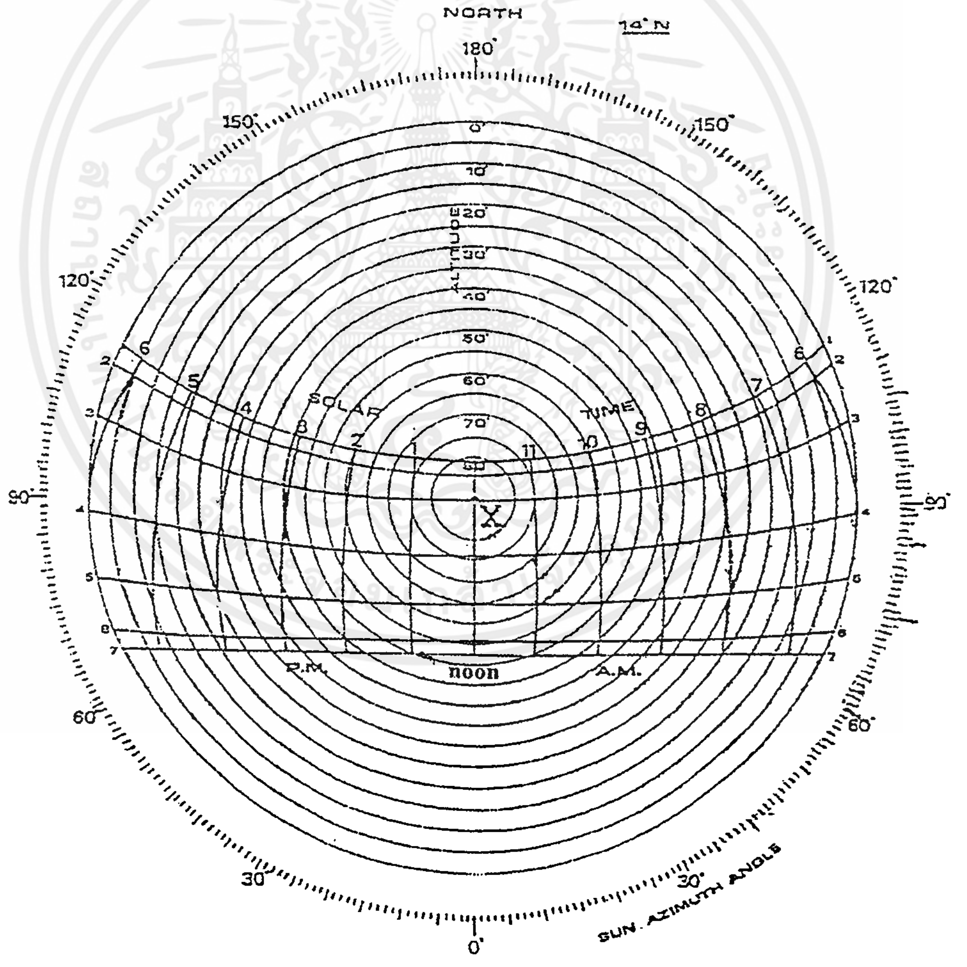
ระหว่างที่โลกกำลังหมุนรอบตัวเองและโคจรรอบดวงอาทิตย์ตำแหน่งของดวงอาทิตย์บนท้องฟ้าจะเปลี่ยนแปลงตลอดวันตลอดฤดู ดวงอาทิตย์จะขึ้นทางทิศตะวันออกโดยเอียงไปทางทิศใต้ ในฤดูหนาว และจะเอียงไปทางทิศเหนือในฤดูร้อน และทางเดินของดวงอาทิตย์บนท้องฟ้าในฤดูร้อนจะสูงกว่าฤดูหนาว เส้นทางการเดินของดวงอาทิตย์ แสดงในรูปที่ 2.7 เมื่อ ตัวเลขในวงกลมแทน Solar time ใน 1 วันในรูปที่ 2.7 จะเห็นว่า Altitude angle α และ Azimuth angle ϕ มีค่า Solar time เท่ากับ 2 P.M.



รูปที่ 2.8 แสดงเส้นทางเคลื่อนที่พาดผ่านบนท้องฟ้าของดวงอาทิตย์ในระหว่างปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นทางเดินของดวงอาทิตย์ (Sun Path Diagram) จะมีลักษณะเป็นรูป 2 มิติในรูปวงกลม ค่า Altitude angle α จะวัดด้วยเส้นวงกลม และ Azimuth angle ϕ จะวัดรอบ ๆ เส้นรอบวง ดังรูปที่ 2.9 ส่วนเวลาที่ใช้ใน Sun Path Diagram จะใช้ Solar time สำหรับ Solar noon คือเวลาที่ดวงอาทิตย์ผ่านเส้นแบ่งเส้นเหนือใต้ของผู้สังเกต ซึ่งจะมีค่าแตกต่างจากเวลากลางวัน ส่วน Clock time จะเทียบกับ Eastern Standard Time ซึ่งใช้ค่าของ Solar noon ณ .ตำแหน่งที่ Longtitude = 150° E ที่เวลากลางวันของทุก ๆ วัน สำหรับผู้สังเกตที่อยู่นอกเส้น Meridian จะพบว่าค่าของ Solar time จะต่างจากค่าของ Clock time โดยส่วนที่แตกต่างเกิดจากค่าของ Solar - noon ที่ตำแหน่งนั้น เกิดคนละเวลากับ Solar noon ที่ตำแหน่งบนเส้น Meridian



รูปที่ 2.9 แสดง Sun Path Diagram ของประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การสร้าง Sun Path Diagram

1. เขียนวงกลมแทนขนาดของ Altitude angle α จากสมการที่ (2-4) โดยเริ่มจากจุดกึ่งกลางเป็นตำแหน่ง ของมุม 90° ไล่ไปจนถึงเส้นนอกสุดที่ 0°

2. แบ่งเส้นรอบวงของวงกลมตามมุมออกเป็น 360° โดยเริ่มจากจุดด้านล่างให้เป็นตำแหน่งทิศใต้และเป็นมุม 0° แทนเป็นมุมของ Azimuth angle ϕ ซึ่งหาจากสมการที่ (2-5)

3. คำนวณหามุมตามข้อ 1 และ 2 ตั้งแต่ 6 โมงเช้า ถึง 6 โมงเย็นชั่วโมงละ 1 ครั้ง ในวันต่างๆดังนี้

วันที่ 21 มิถุนายน		
วันที่ 21 พฤษภาคม	ซึ่งเท่ากับ	วันที่ 21 กรกฎาคม
วันที่ 21 เมษายน	ซึ่งเท่ากับ	วันที่ 21 สิงหาคม
วันที่ 21 มีนาคม	ซึ่งเท่ากับ	วันที่ 21 กันยายน
วันที่ 21 กุมภาพันธ์	ซึ่งเท่ากับ	วันที่ 21 ตุลาคม
วันที่ 21 มกราคม	ซึ่งเท่ากับ	วันที่ 21 พฤศจิกายน
วันที่ 21 ธันวาคม		

ทั้งนี้จะใช้ค่า Latitude 14°N, Local Longtitude 100.5°E และ Standard Longtitude 105°E สำหรับกรุงเทพฯ

4. นำค่าที่ได้จากข้อที่ 3 มา plot ลงบน Chart วงกลม

5. ลาก Curve สำหรับจุดต่าง ๆ ในแต่ละวันให้อยู่บนเส้นตรงเดียวกัน (เส้นตามแนวขวาง) จะได้ทั้งหมด 7 เส้นและจุดต่างๆ ในเวลาเดียวกันให้อยู่บนเส้นตรงเดียวกัน (เส้นตามแนวดิ่ง) จะมีทั้งหมด 13 เส้น

6. สำหรับการอ่านค่าต่าง ๆ จาก Diagram ให้ประมาณจากค่าวันและเวลาที่กำหนดเอาไว้โดยถือว่า ตำแหน่งของผู้สังเกตอยู่ที่มุม Altitude 90° ในเวลาเที่ยงของวันที่ดวงอาทิตย์อยู่ตรงหัวพอดี

หมายเหตุ สำหรับกรุงเทพฯ ดวงอาทิตย์จะอยู่ตรงศีรษะพอดี ในวันที่ 21 เมษายน และ 21 สิงหาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 วิธีการและอุปกรณ์ป้องกันแสงแดด

2.4.1การลดพื้นที่รับแสงแดด

1. จากธรรมชาติ เช่นใช้ต้นไม้อาคารข้างเคียง เป็นที่บังแสงแดด

2. จากรูปทรงอาคารของอาคารโดยเอาส่วนหนึ่งของอาคารบังแดดให้ส่วนอื่น เช่นหลังคาจั่วที่รับแดดเพียงครึ่งเดียว

3. ออกแบบบริเวณโดยรอบ และเหนือหน้าต่าง เช่นการทำหลังคายื่นยาวออกมานอกอาคาร การทำระเบียง การทำกันสาด

4. ชนิดของหน้าต่าง เช่น การทำหน้าต่างบานเกล็ดไม้ทั้งชนิดติดตาย และปรับได้การทำหน้าต่างบานเลื่อน และพับได้

5. การบังแดดด้านในหน้าต่าง เช่น ม่าน มู่ลี่

6. การจัดภายในเพื่อลดการสะท้อน แสงทำให้สบายตา เช่น การใช้สี

2.4.2 การออกแบบอุปกรณ์บังแดด

ก่อนอื่นจะต้องเลือกวันและเดือนที่ต้องการกันแดด ซึ่งจะขึ้นอยู่กับความร้อนหนาวในฤดูต่าง ๆ และสภาพลมฟ้าอากาศในขอบเขตที่ให้ความสบาย (Comfort zone) ในเวลาที่มีความร้อนมากเกินขีด Comfort zone เราจะต้องหาทางบังแดด ในเขตอบอุ่นอาจจะออกแบบภายหลังวันที่ 1 เมษายน ส่วนในประเทศแถบเหนือขึ้นไปมาก อาจจะออกแบบหลังวันที่ 12 พฤษภาคม ซึ่งจะกันแดดได้ถึงวันที่ 11 กรกฎาคม ในเขตเมืองหนาวนั้นส่วนมากจะใช้ต้นไม้ซึ่งมีใบปกคลุมเต็ม ช่วยบังแดดในฤดูร้อนแต่ผลัดใบในฤดูหนาวซึ่งเป็นฤดูที่ไม่ต้องการร่มเงา เป็นเครื่องบังแดดโดยธรรมชาติ แต่วิธีนี้สามารถใช้กับอาคารที่สูงไม่เกิน 2 ชั้น

สำหรับประเทศไทยโดยเฉพาะที่กรุงเทพฯ จะต้องการบังแดดตลอดทั้งปี ถ้าสังเกตดูจะพบว่าทางด้านทิศใต้ แสงแดดจะเข้าเป็นมุมต่ำสุดในเดือนธันวาคม เราจึงเลือกเวลาออกแบบในเดือนธันวาคม สำหรับประเทศที่อยู่ทางซีกโลกใต้โดยมากใช้เดือนมิถุนายน

เมื่อเราได้วันที่และเดือนที่จะต้องป้องกันแดดแล้ว เราอาจจะเลือกเวลาที่จะออกแบบใหม่โดยดูจากเวลาที่จะใช้สอยอาคาร ซึ่งอาคารต่างชนิดกัน ก็จะมีเวลาใช้แตกต่างกัน เช่นโรงเรียนอาจจะต้องการกันแดดตั้งแต่ 8.00 น.ถึง 15.00 น. ตามเวลาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 วิธีการหาเงา

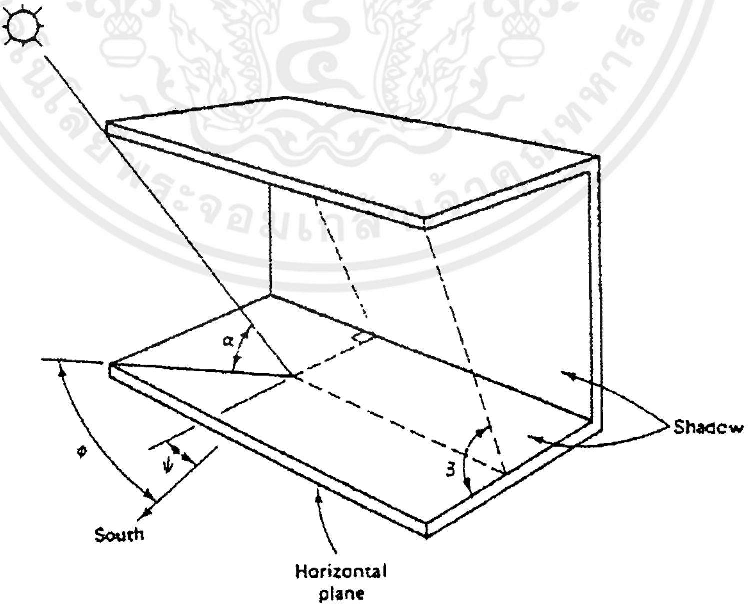
ในการออกแบบเพื่อการประหยัดพลังงาน หลักการคือ ต้องดูลักษณะการเกิดเงาของอุปกรณ์ที่บังแดด โดยในการหาเงาซึ่งเป็นฟังชั่นก์ของเวลาและวันใน 1 ปี

มุม Profile angle β คือ มุมทางตั้งในแนวระดับเดียวกับตำแหน่งดวงอาทิตย์โดยจะกระทำในแนวแกนเดียวกับแนวแกนของอุปกรณ์บังแดด (ในเวลาเที่ยงวัน Profile angle จะเท่ากับ Altitude angle) จะอยู่ในเทอมของ Altitude angle α, Azimuth angle ϕ และ Surface azimuth angle ψ ดังสามารถแสดงในรูปที่ 2.9 เพราะฉะนั้นจะมีสมการเป็น

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

เมื่ออุปกรณ์บังแดดหันหน้าไปทางทิศใต้ ($\psi = 0$) จะได้ว่า

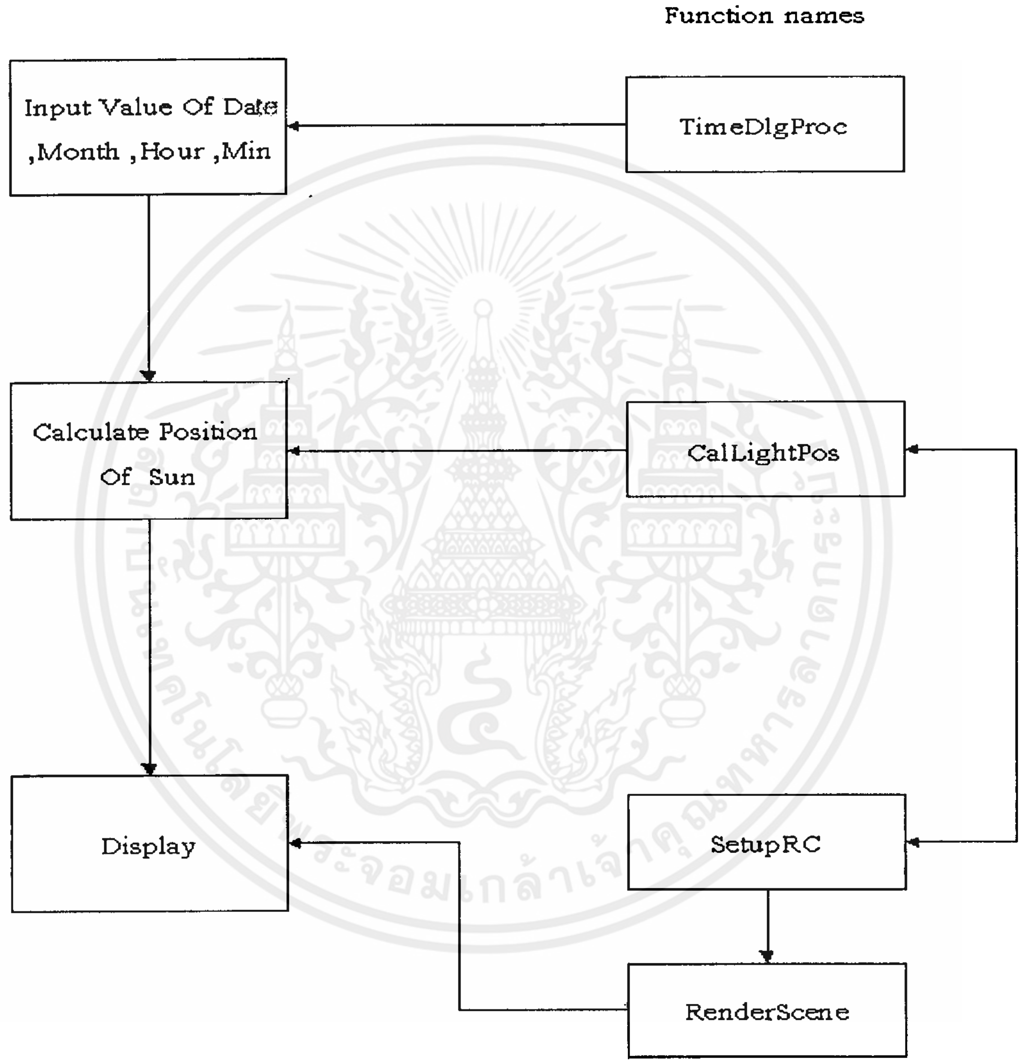
$$\tan\beta = \frac{\tan\alpha}{\cos\phi} \qquad (2\text{-}15)$$



รูปที่ 2.10 แสดงมุม Profile angle β และความสัมพันธ์กับมุม Altitude angle α, Azimuth angle ϕ และ Surface azimuth angle ψ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
แนวคิดและการทำงานของโปรแกรม



จากที่ได้กล่าวมา ว่าเราต้องการสร้างแบบจำลองของเงาที่เกิดจากแสงแดดส่อง ในวันและเวลาที่เราต้องการ เพราะฉะนั้น ข้อมูลที่เราจำเป็นต้องใช้ก็จะมี วันที่ เดือนและเวลาที่ต้องการแสดงผล โดยสามารถป้อนได้ละเอียดถึงหน่วยนาที โดยทางผู้วิจัยได้ใช้ภาษา Visual C++ version 5 ซึ่งโปรแกรมที่ทำงานบนสภาพแวดล้อมวินโดว์ในการเขียน โดยทางผู้วิจัยได้ทำการแสดงรายละเอียด และขั้นตอนทำงานของฟังก์ชันย่อยต่าง ๆ ไว้ในส่วนของภาคผนวกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากแผนภาพการทำงานข้างต้นพอจะอธิบายการทำงานของโปรแกรมได้ดังนี้

1. ทำการรับค่า วัน เดือน และเวลาที่ต้องการในการแสดงผล โดยฟังก์ชัน TimeDlgProc
2. นำค่าที่ได้ไปทำการคำนวณหามุมต่าง ๆ ที่จำเป็นต้องใช้ในการแสดงผล โดยฟังก์ชัน CalLightPos
3. ฟังก์ชัน CalLightPos จะทำการคำนวณหาตำแหน่งของดวงอาทิตย์ และจะส่งค่าที่ได้ไปยังฟังก์ชัน SetupRC เพื่อเตรียมความพร้อมสำหรับ ปัจจัยต่าง ๆ ที่ใช้ในการแสดงผลเช่น ลักษณะของแสง เงา สีของพื้นผิววัตถุ และลักษณะของเงาที่จะเกิดขึ้น
4. จากนั้นฟังก์ชัน SetupRC จะส่งค่าต่อไปยัง ฟังก์ชัน RenderScene เพื่อใช้สร้าง รูปต่าง ๆ แสงและเงาออกมาทางหน้าจอคอมพิวเตอร์

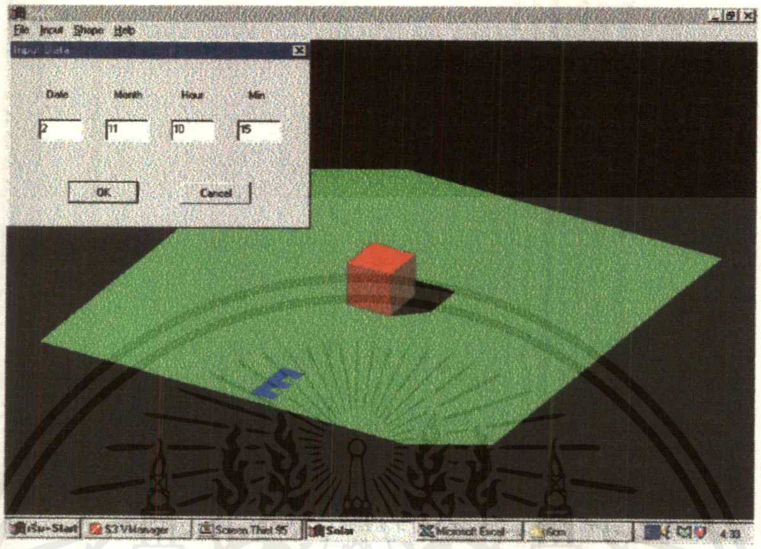
<u>จากโปรแกรมที่ได้เขียนขึ้น จะมีลักษณะและการใช้งานดังนี้</u>

1.เมื่อเรียกใช้งานโปรแกรม Solar.exe ก็จะมีลักษณะดังรูป

รูปที่ 3.1 แสดงลักษณะทั่วไปของตัวโปรแกรม

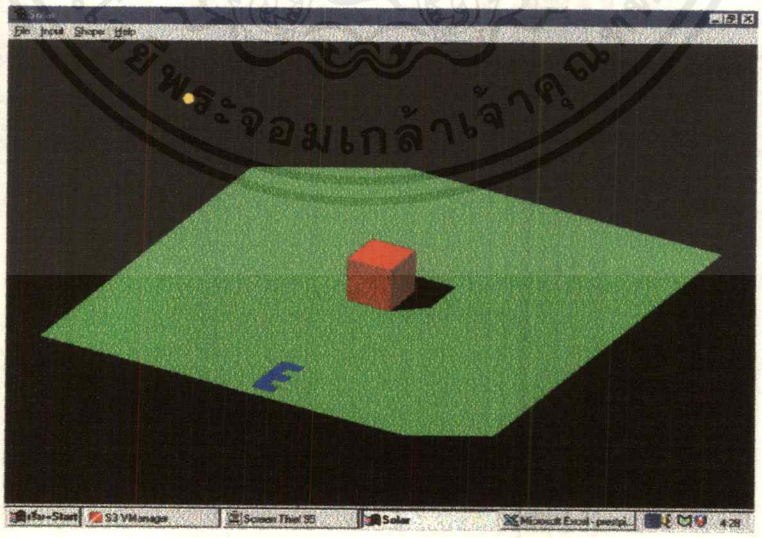
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้คำสั่ง Input / Time ก็จะมีไดอะลอกบ๊อกซ์ (Dialog box) ขึ้นดังรูปที่ 3.2 ให้ทำการป้อนข้อมูลต่าง ๆ



รูปที่ 3.2 แสดงการป้อนค่าวันและเวลาที่ต้องการ

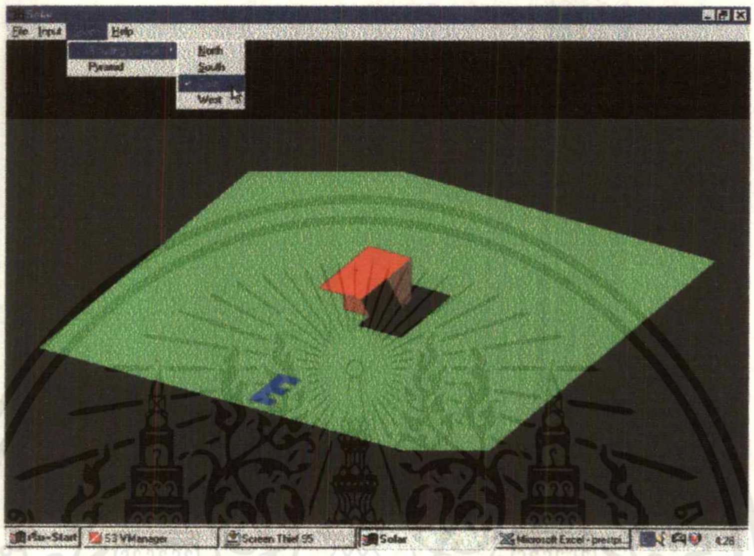
3. หลังจากคลิกที่ ปุ่ม O.k. โปรแกรมก็จะทำการแสดงผลลักษณะของเงาในวัน และเวลาตามที่ผู้ใช้ได้ป้อนไว้ ดังรูปที่ 3.3



รูปที่ 3.3 แสดงตัวอย่างลักษณะของเงาในวันที่ 2 พฤศจิกายน เวลา 10.15 น.

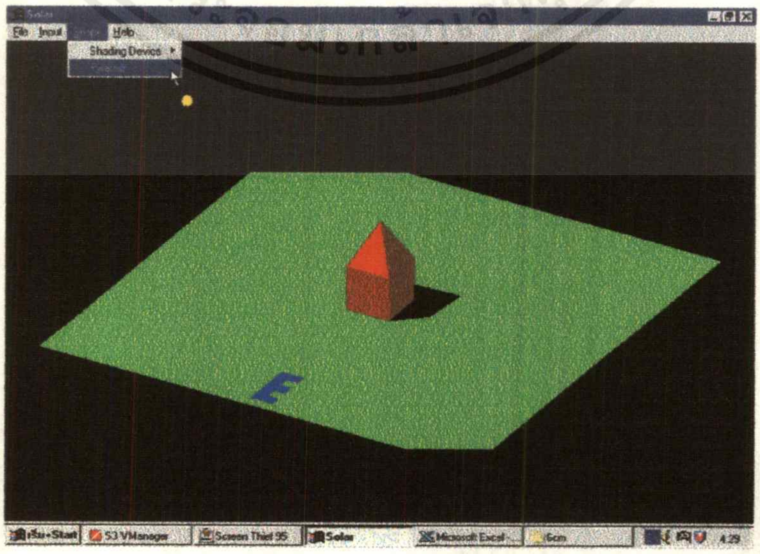
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นอกจากนี้เรายังสามารถทดสอบการบังแดด ของแผงบังแดดที่ติดกับตัววัตถุที่เราสมมุติขึ้นได้ด้วยโดยใช้คำสั่ง Shape / Shading Device / N, E, S, W โดยเราสามารถเลือกได้ครั้งละ 1 ด้านเท่านั้น ถ้าจะต้องการเปลี่ยนด้าน จะต้องทำการคลิกที่ด้านเก่าอีกครั้งเพื่อนำออกก่อน



รูปที่ 3.4 แสดงลักษณะเงาเมื่อมีการติดอุปกรณ์บังแดด

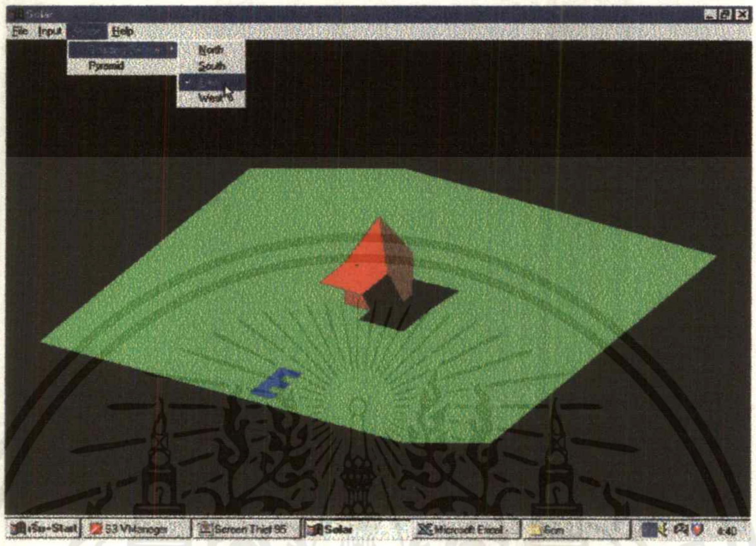
5. และยังสามารถเปลี่ยนรูปร่างของวัตถุที่เราสมมุติขึ้นมาเพื่อที่จะดูลักษณะและทิศทางของเงาที่จะเกิด ไปเป็นรูปทรงปิรามิดบนทรงสี่เหลี่ยมได้



รูปที่ 3.5 แสดงลักษณะเมื่อทำการเปลี่ยนแปลงรูปทรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038886

6. หลังจากที่มีการเพิ่มเติมปิรามิดไปบนวัตถุเริ่มต้นแล้ว ก็ยังสามารถเพิ่มแผงบังแดดเพื่อที่จะดูลักษณะของเงาที่เกิดขึ้นได้อีก



รูปที่ 3.6 แสดงเมื่อรูปทรงที่เปลี่ยนไปได้ติดอุปกรณ์บังแดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
เครื่องมืออุปกรณ์และการทดลอง

เครื่องมือและอุปกรณ์

1. กล่องสี่เหลี่ยมลูกบาศก์
2. กล้องถ่ายรูป
3. แผ่นกระดาษแสดงทิศทาง
4. เข็มทิศ

วิธีการทดลอง

1. นำกล่องไปวางบนแผ่นกระดาษแสดงทิศทางในที่กลางแจ้งโดยให้กล่องหันหน้าไปทางทิศใต้

2. ถ่ายรูปเงาที่เกิดจากกล่อง โดยแบ่งเป็นช่วงๆ ละ 15 นาที และทำการวัดค่าต่าง ๆ ที่จำเป็นต้องใช้ในการหาตำแหน่งของดวงอาทิตย์

3. นำค่าที่วัดไปคำนวณหา มุม Azimuth ϕ และมุม Altitude α แล้วนำไปเปรียบเทียบกับการคำนวณจากสูตรที่มี

4. พร้อมทั้งนำรูปที่ถ่ายได้ไปเปรียบเทียบกับผลที่ได้จากโปรแกรมที่จำลองรูปขึ้นมา

วิธีการคำนวณ

1. หามุม Declination ,δ เป็นอันดับแรก ซึ่งจะใช้การหาจากรูปที่ 2.4 หรือว่าจากสมการที่ (2-1) ก็ได้

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

เมื่อ n เป็นจำนวนวันใน 1 ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คำนวณหา Hour angle ,**h** จากสมการที่ (2-2)

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

เมื่อ ค่า + แสดงว่าเป็นตอนบ่าย และ ค่า - แสดงว่าเป็นตอนเช้า

3. คำนวณหา Altitude angle ,α จากสมการที่ (2-4)

$$\cos Z = \sin\alpha = \sin L \sin\delta + \cos L \cos\delta \cosh$$

4. คำนวณหา Azimuth angle ,ϕ จากสมการที่ (2-3)

$$\sin\phi = \cos\delta \frac{\sinh}{\cos\alpha}$$

จากนั้นนำค่าที่คำนวณได้จากการวัดจริงไปเปรียบเทียบกับจากการคำนวณ ก็จะได้ดังตารางที่ 4.1

เวลา		ค่าที่ได้จากการวัดจริง		ค่าที่ได้จากการคำนวณ	
		Azimuth ϕ	Altitude α	Azimuth ϕ	Altitude α
9.00	AM	-61.46	38.27	-61.69	38.25
10.00	AM	-50.26	49.33	-50.02	50.35
11.00	AM	-23.08	60.85	-30.28	59.86
12.00	AM	0	64.44	0	63.78
1.00	PM	23.05	60.73	30.28	59.86
2.00	PM	50.17	50.27	50.02	50.35
3.00	PM	61.42	38.29	61.69	38.25

ตารางที่ 4.1 แสดงการเปรียบเทียบผลของมุมที่วัดได้จริงกับที่ได้จากการคำนวณ

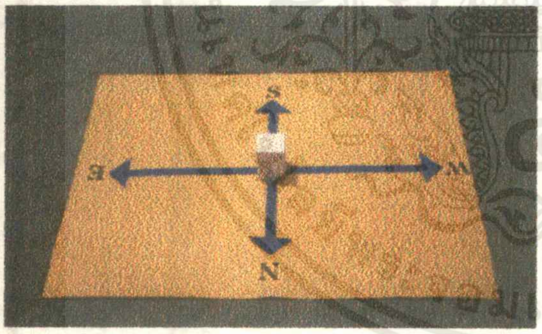
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
เปรียบเทียบผลที่ได้จากโปรแกรมกับจากการทดลอง

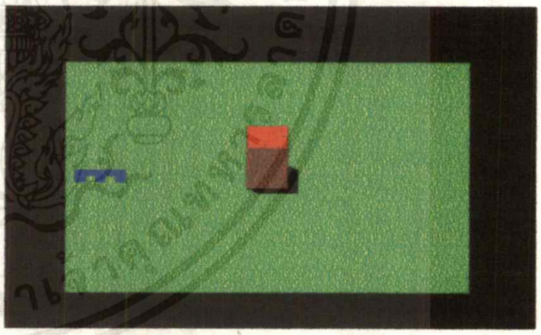
เพื่อทำการเปรียบเทียบผลที่ได้จากการทดลอง กับผลที่ได้จากโปรแกรม ว่าจะออกมาไปในแนวทางเดียวกันกับรูปถ่ายที่ได้จากการทดลอง หรือไม่ โดยค่าที่ได้จะได้ทางผู้วิจัยได้ทำการบันทึกไว้ ณ. วันที่ 27 มีนาคม พ.ศ. 2541 เวลาตั้งแต่ 11.00 – 15.00 โดยทำการเก็บภาพทุก ๆ 15 นาที ซึ่งผลที่ได้ก็ค่อนข้างจะใกล้เคียงกับของจริงมากซึ่งจะเป็นดังนี้

<u>ผลที่ได้จากการทดลองจริง</u>

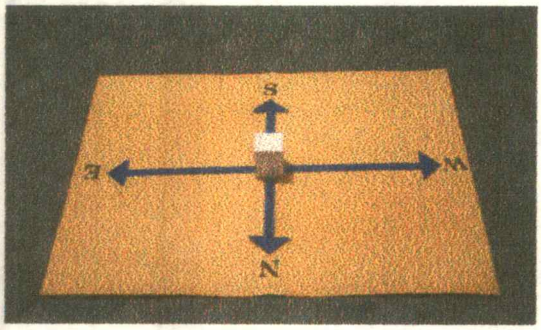
<u>ผลที่ได้จากทางโปรแกรม</u>



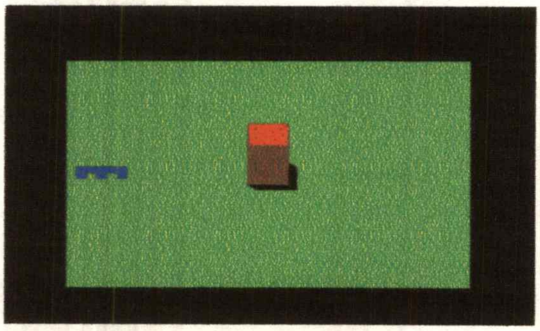
เวลา 11.00 น.



เวลา 11.00 น.



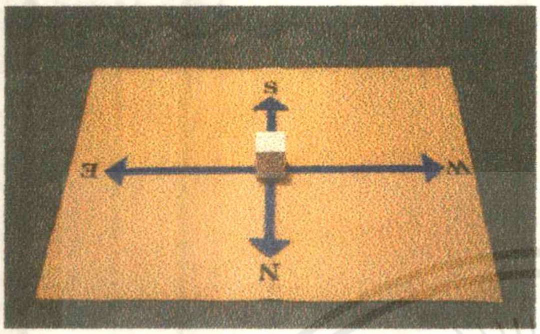
เวลา 11.15 น.

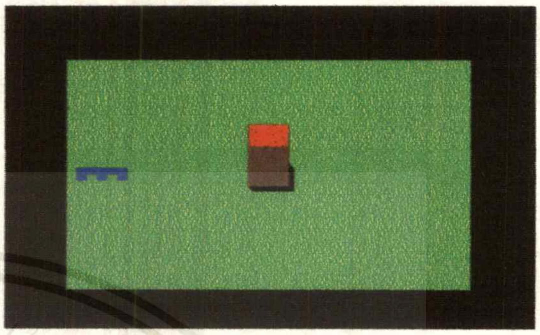


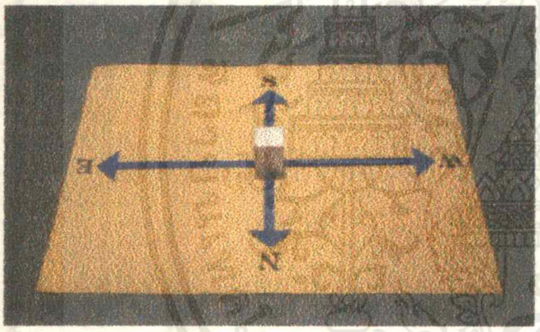
เวลา 11.15 น.

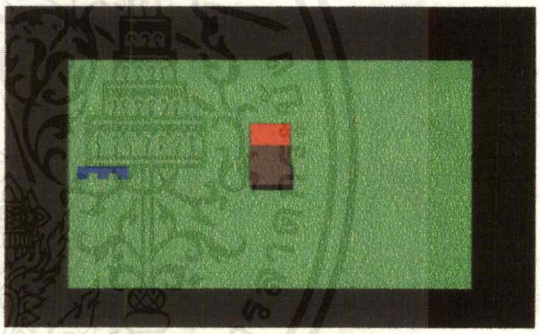
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

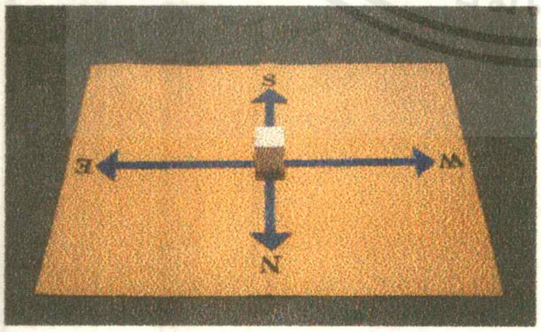
<u>ผลที่ได้จากการทดลองจริง</u> | <u>ผลที่ได้จากทางโปรแกรม</u>

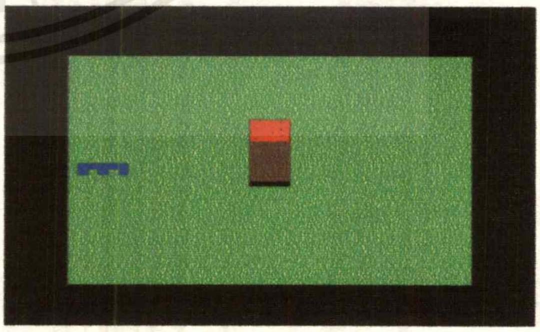

เวลา 11.30 น.


เวลา 11.30 น.


เวลา 11.45 น.

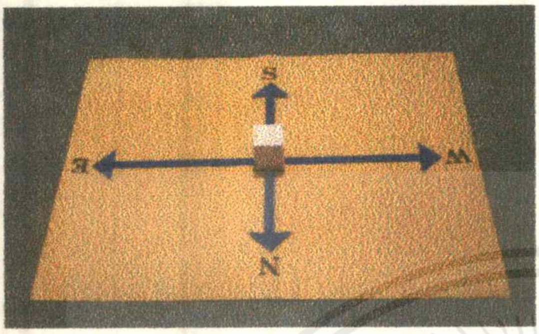

เวลา 11.45 น.


เวลา 12.00 น.

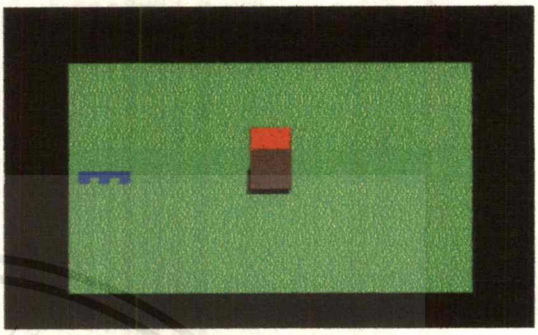

เวลา 12.00 น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

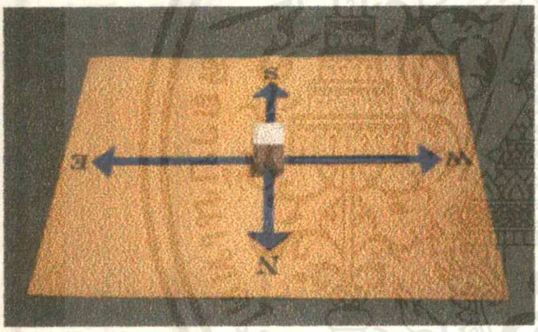
<u>ผลที่ได้จากการทดลองจริง</u> | **<u>ผลที่ได้จากทางโปรแกรม</u>**



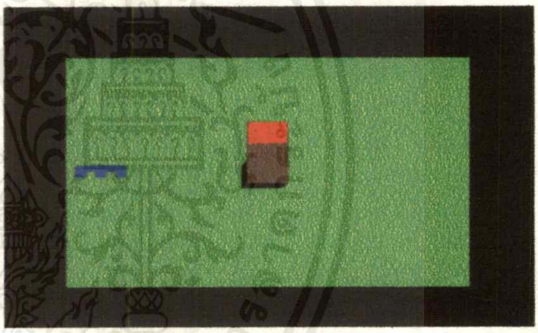
เวลา 12.15 น.



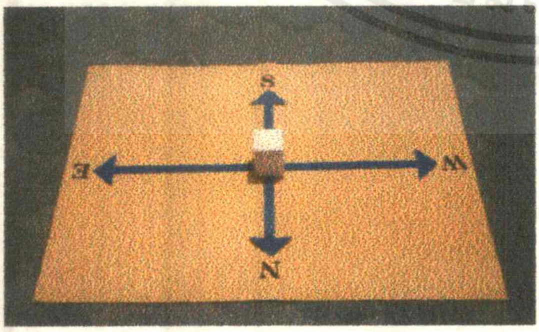
เวลา 12.15 น.



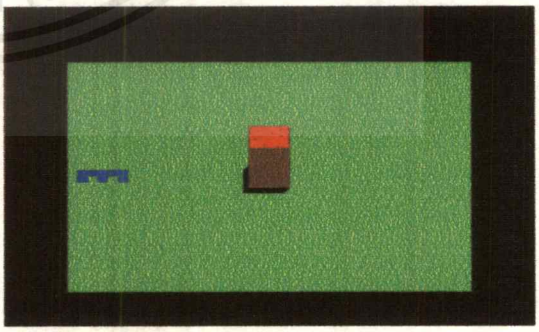
เวลา 12.30 น.



เวลา 12.30 น.



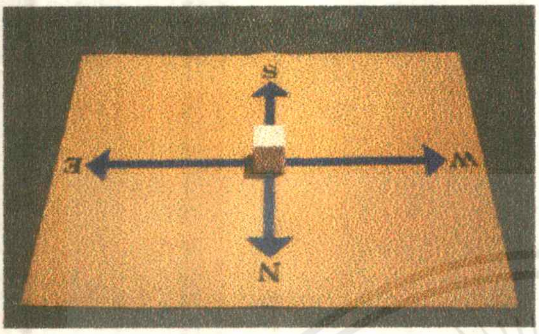
เวลา 12.45 น.



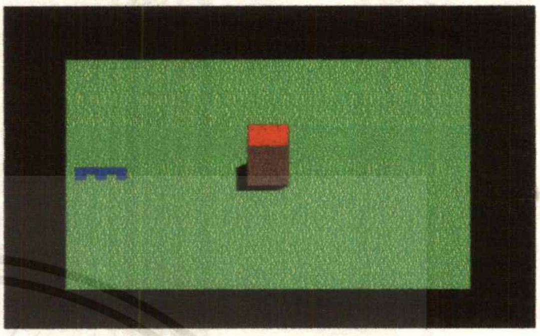
เวลา 12.45 น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

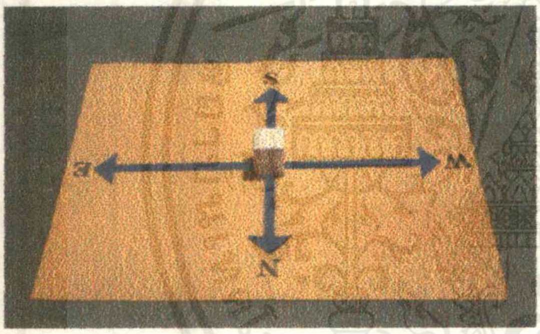
ผลที่ได้จากการทดลองจริง | **ผลที่ได้จากทางโปรแกรม**



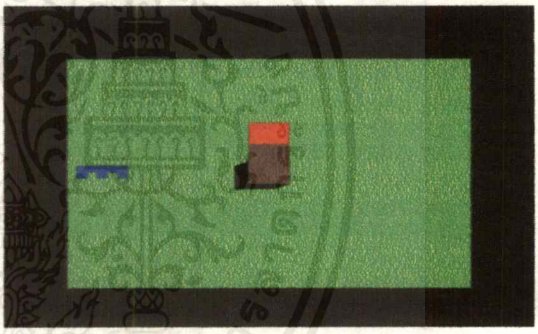
เวลา 13.00 น.



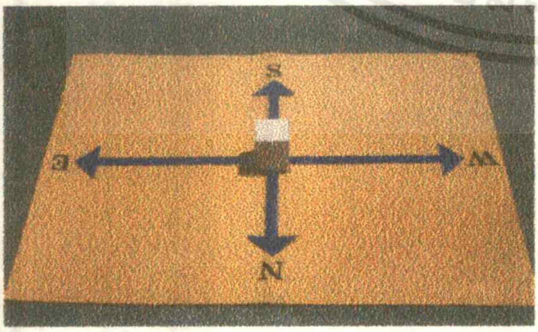
เวลา 13.00 น.



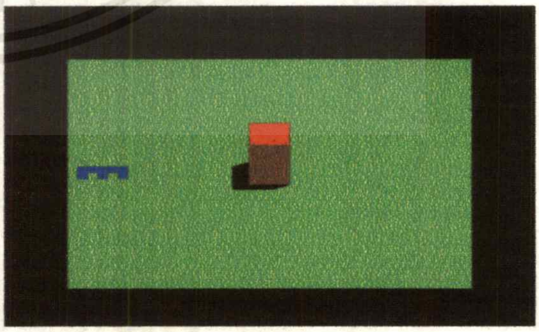
เวลา 13.15 น.



เวลา 13.15 น.



เวลา 13.30 น.

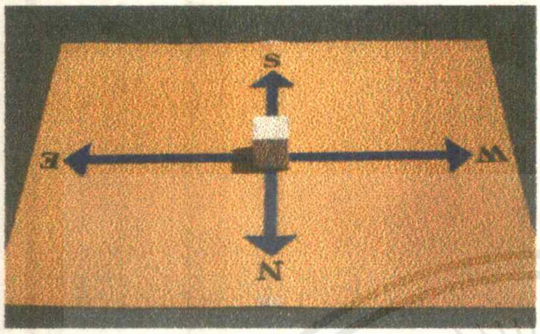


เวลา 13.30 น.

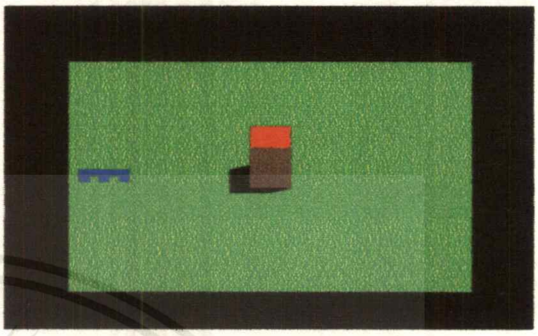
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>**ผลที่ได้จากการทดลองจริง**</u>

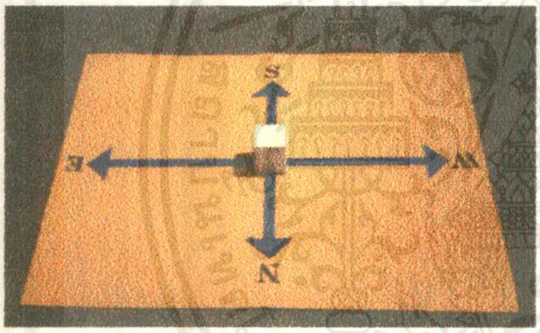
ผลที่ได้จากทางโปรแกรม



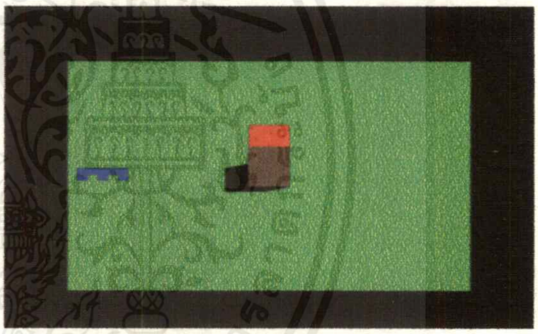
เวลา 13.45 น.



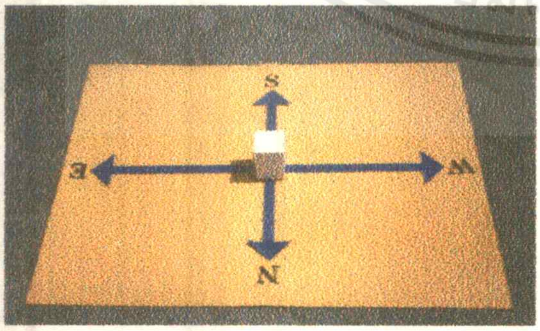
เวลา 13.45 น.



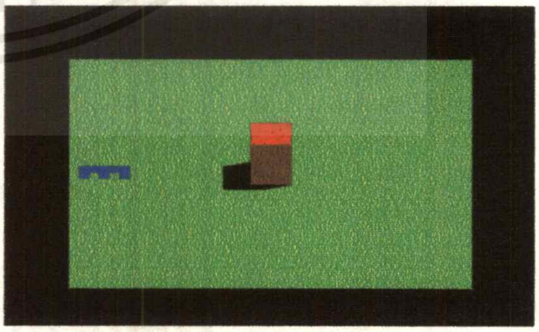
เวลา 14.00 น.



เวลา 14.00 น.



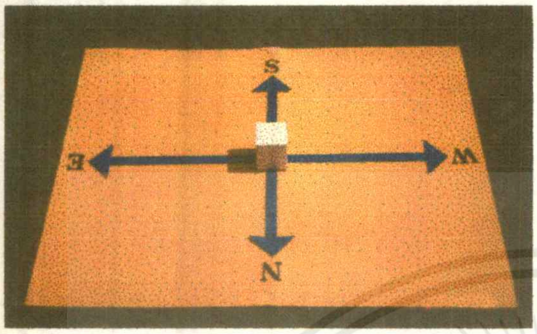
เวลา 14.15 น.



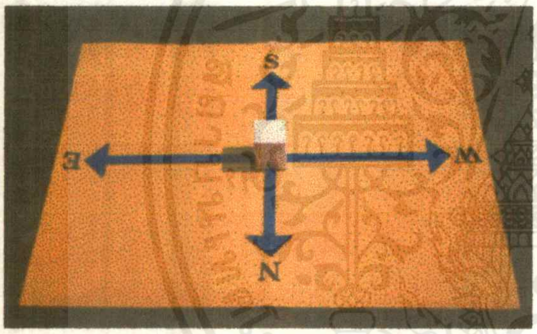
เวลา 14.15 น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

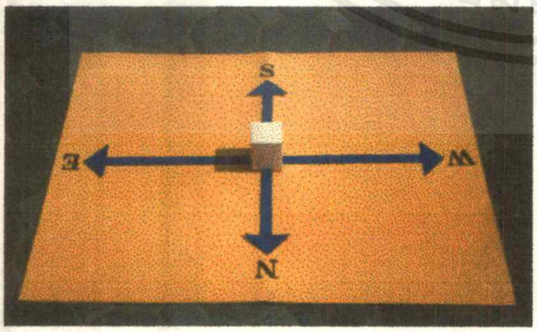
<u>ผลที่ได้จากการทดลองจริง</u>



เวลา 14.30 น.

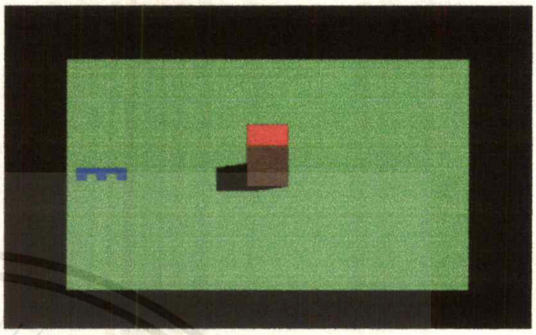


เวลา 14.45 น.

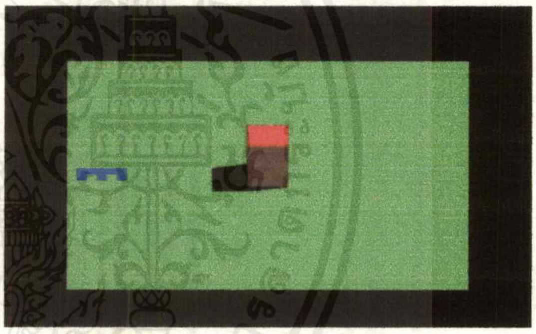


เวลา 15.00 น.

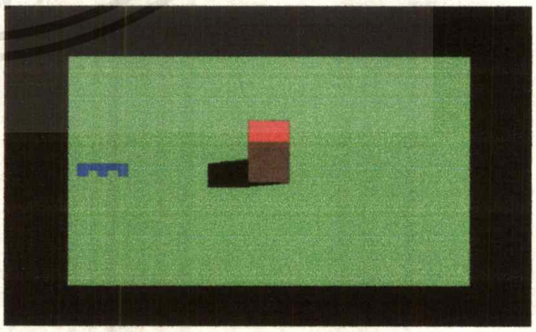
<u>ผลที่ได้จากทางโปรแกรม</u>



เวลา 14.30 น.



เวลา 14.45 น.



เวลา 15.00 น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

จากการที่ได้ศึกษาถึงสมการและความสัมพันธ์ต่างๆ ตั้งแต่บทที่ 1 เป็นต้นมา จะทำให้ได้ทราบถึงความสัมพันธ์ระหว่างโลกกับดวงอาทิตย์ ตำแหน่งของดวงอาทิตย์ในวันและเวลาต่างๆ และยังทำให้ได้ทราบถึงลักษณะและทิศทางของเงาที่เกิดขึ้น ซึ่งเป็นข้อมูลและหลักการเบื้องต้นที่ควรทราบ นอกจากนั้นยังได้ทำการทดลองวัดค่ามุมต่างๆ จากเงาที่เกิดจากแสงแดดจริงๆ โดยเมื่อนำมาทำการเปรียบเทียบกับค่าต่างๆ ที่ได้จากการคำนวณ

<u>ปัญหาในการทดลอง</u>

จะพบได้ว่ามีค่าผิดพลาดบ้างเนื่องจากการปัดค่า แต่ก็อยู่ในขั้นที่ยอมรับได้ ซึ่งข้อผิดพลาดที่เกิดขึ้นน่าจะเป็นผลมาจากอุปกรณ์ที่ใช้วัด ส่วนสำหรับการทดลองนั้นได้ทำการวัดค่าความยาวต่างๆ ที่ จำเป็นในการหาค่าแล้วนำไปทำการคำนวณหามุม Altitude α และ Azimuth ϕ เพื่อที่จะนำไปเปรียบเทียบกับผลที่ได้จากโปรแกรมโดยได้แสดงไว้ใน ตารางที่ 4.1

<u>ปัญหาในการเขียนโปรแกรม</u>

เมื่อทำการเขียนโปรแกรมจะพบปัญหาว่าจริงๆแล้วตำแหน่งของดวงอาทิตย์ไม่สามารถทำให้อยู่ที่ระยะอินฟินิตี้ได้ จึงทำให้เงาที่เกิดขึ้นอาจจะสั้นไปกว่าที่เป็นจริงเล็กน้อย แต่ทิศทางของเงาที่เกิดขึ้นจะมีทิศทางที่ถูกต้อง และปัญหาเรื่องการแสดงผลอาจจะดูไม่ราบเรียบเท่าที่ควรเนื่องจากโปรแกรมที่เขียนนั้นอ้างอิงไลเบรี่ของ OpenGL ซึ่งไม่ได้ใช้ไลเบรี่ MFC (Microsoft Foundation Class) ซึ่งเกี่ยวกับการเขียนกราฟฟิคส์ของไมโครซอฟท์ จึงทำให้การแสดงผลซึ่งต้องอาศัยการ์ดแสดงผลที่สนับสนุน OpenGL นั้นไม่ราบเรียบเท่าที่ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>สรุปและข้อเสนอแนะ</u>

ดังนั้นจึงพอจะสรุปได้ว่าโปรแกรมคอมพิวเตอร์ที่ได้พัฒนาขึ้นนี้ สามารถนำไปใช้ในการทำนายตำแหน่งของดวงอาทิตย์ ซึ่งจะทำให้เราได้ทราบถึงทิศทางของเงาที่จะเกิดขึ้นซึ่งจะมีประโยชน์ในการเลือกทิศทางการวางอาคาร เพื่อที่จะให้แดดเข้าน้อยที่สุด หรือว่าให้มีร่มเงามากที่สุด ไม่ว่าจะโดยการหากระจกกรองแสงมาทำการติดตั้งหรือวิธีอื่นๆ ซึ่งเป็นการประหยัดพลังงานจากเครื่องปรับอากาศ หรือว่าอาจจะให้แสงแดดส่องเข้าได้ในเวลากลางวันเพื่อให้มีแสงสว่างโดยที่ไม่ต้องเปิดไฟ โดยต่อไปอาจจะมีการพัฒนาให้มีการเปลี่ยนรูปร่างวัตถุที่ใช้หาลักษณะของเงา พร้อมทั้งสร้างให้มีความเหมือนจริงได้ยิ่งขึ้น ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solar.c

```
#include <windows.h>
#include <gl\gl.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";
static HINSTANCE hInstance;

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Get direction of shading device and build pyramid
BOOL North, South ,West ,East, Pyramid;

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

// Rotation
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 of shading device
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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Dialog Procedure for Time box
BOOL APIENTRY TimeDlgProc (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};
        int date, month, i, check,n ,total_min;
        float Latitude, n1, time;
        float declination, hourangle, altitude, azimuth, radian;

        //Set date month and calculate time total_minute
        date = Day;
        month = Month;
        time = (float)(Hour + (Min/60.0f));
        total_min = (Hour*60) + Min;

        //Set n = 0
        n=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Calculate Latitude and Uint is radian
Latitude = 14.0f*pi/180.0f;

//Calculate the number of day in year
for(i=1;i<month;i++)
        n+= month1[i];
n = n+date;

//Store n into check
check=n;

//Calculate the Radian
radian = (284.0f+n) * 2.0f * pi / 365.0f;



// Calculate the Declination and Unitt is degree
declination = 23.45f*(float)sin(radian);

//  Change unit from degree to radian
declination = declination * pi/180.0f;

//Calculate the Hour angle
n1 = 0.25f * (time-12.0f) * 60.0f;

// Change unit from degree to radian
hourangle = n1 * pi / 180.0f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Calculate Altitude and Azimuth angle and unit is radian
        altitude = (float)asin( ( sin(Latitude) * sin(declination) ) + ( cos(Latitude) * cos
(declination) * cos(hourangle) ) );
        azimuth = (float)asin( cos(declination) * ( sin(hourangle) / cos(altitude) ) );

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

        // Set Positive angle
        if ((total_min) > 720.0)
                azimuth =azimuth;

/* #######  (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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                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;
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

case 99:
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
    break;

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
      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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case (231):
if (total_min > 605)
        azimuth = 360.0f+azimuth;
    else
        azimuth = 180.0f-azimuth;
    break;

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

// Change unit from degree to radian
altitude = altitude*pi/180.0f;
azimuth = azimuth*pi/180.0f;

// Calculate the Light position
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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // 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;
        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];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// 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];
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];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //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);
        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()
        {
        //Build shadow
        // 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 }};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Any three point on north cube
        GLfloat pointsN[3][3] = {{ 30.01f, 10.0f, 10.0f },
                                 { 30.01f, 0.0f, 10.0f },
                                 { 30.01f, 0.0f, -10.0f }};

        // Any three point on south cube
        GLfloat pointsS[3][3] = {{ -30.01f, 10.0f, 10.0f },
                                 { -30.01f, 10.0f, 0.0f },
                                 { -30.01f, 0.0f, 0.0f }};

        // Any three point on east cube
        GLfloat pointsE[3][3] = {{ 10.0f, 10.0f, 30.01f },
                                 { -10.0f, 10.0f, 30.01f },
                                 { -10.0f, 0.0f, 30.01f }};

        // Any three point on west cube
        GLfloat pointsW[3][3] = {{ 10.0f, 10.0f, -30.01f },
                                 { 10.0f, -10.0f, -30.01f },
                                 { 0.0f, -10.0f, -30.01f }};

        //Calculate light position
        CalLightPos(lightPos);
        glEnable(GL_DEPTH_TEST);   // Hidden surface removal
        glEnable(GL_CULL_FACE);    // Do not calculate inside of cube
        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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

// Calculate projection matrix to draw shadow on the ground
MakeShadowMatrix(points, lightPos, shadowMat);

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

		}

		}

// This function just specifically draws the cube
void Draw(BOOL bShadow)
		{
		float normal[3]; // Storage for calculated surface normal

		// Set to black for the shadow once
		if(!bShadow)
			//Set colour is red
			glRGB(255, 0, 0);
		else
			//Set colour is black
			glRGB(0,0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Call only once for all remaining points
glBegin(GL_QUADS);

        {
        // Vertices for Front panel
        float v[4][3] =  {{ -30.0f, 30.0f, 30.0f},
                          { -30.0f, -30.0f, 30.0f},
                          { 30.0f, -30.0f, 30.0f},
                          {30.0f,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();
        }


        {
        // Vertices for Back panel
        float v[4][3] =  {{ -30.0f, 30.0f, -30.0f},
                          {`30.0f, 30.0f, -30.0f},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    { 30.0f, -30.0f, -30.0f},
                    {-30.0f,-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();
}


{
// Vertices for Top panel
float v[4][3] =   {{ 30.0f, 30.0f, 30.0f},
                   { 30.0f, 30.0f, -30.0f},
                   { -30.0f, 30.0f, -30.0f},
                   {-30.0f,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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glNormal3fv(normal);
        glVertex3fv(v[0]);
        glVertex3fv(v[1]);
        glVertex3fv(v[2]);
        glVertex3fv(v[3]);
    //glEnd();
    }

    {
    // Vertices for Bottom panel
    float v[4][3] =  {{ -30.0f, -30.0f, 30.0f},
                     { -30.0f, -30.0f, -30.0f},
                     { 30.0f, -30.0f, -30.0f},
                     {30.0f,-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

```
{
// Vertices for Left panel
float v[4][3] =  {{ 30.0f, -30.0f, 30.0f},
                  { 30.0f, -30.0f, -30.0f},
                  { 30.0f, 30.0f, -30.0f},
                  {30.0f,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();
}

{

// Vertices for Right panel
float v[4][3] =  {{ -30.0f, 30.0f, 30.0f},
                  { -30.0f, 30.0f, -30.0f},
                  { -30.0f, -30.0f, -30.0f},
                  {-30.0f,-30.0f,30.0f}};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                // 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();
                }

        // Done drawing points
        glEnd();

        }

// Draw the pyramid
void DrawPyramid(BOOL bShadow,BOOL Pyramid)
{
        float normal[3]; // Storage for calculated surface normal

        // Set to black for the shadow once
        if(!bShadow)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            //Set colour is red
            glRGB(255, 0, 0);
    else
            //Set colour is black
            glRGB(0,0,0);

    // Draw the pyramid
    if (Pyramid)
    {
    glBegin(GL_TRIANGLES);
            {
            // Vertices for front panel
            float v[3][3] =  {{ -30.0f, 30.0f, 30.0f},
                              { 30.0f, 30.0f, 30.0f},
                              { 0.0f, 60.0f, 0.0f}};

            // Calculate the normal for the plane
            calcNormal(v,normal);

            // Draw the triangle using the plane normal
            // for all the vertices
                    glNormal3fv(normal);
                    glVertex3fv(v[0]);
                    glVertex3fv(v[1]);
                    glVertex3fv(v[2]);
            }
            {
            // Vertices for left panel
            float v[3][3] =  {{ 30.0f, 30.0f, 30.0f},
                              { 30.0f, 30.0f, -30.0f},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        { 0.0f, 60.0f, 0.0f}};

// Calculate the normal for the plane
calcNormal(v,normal);

// Draw the triangle using the plane normal
// for all the vertices

        glNormal3fv(normal);
        glVertex3fv(v[0]);
        glVertex3fv(v[1]);
        glVertex3fv(v[2]);
}

{
// Vertices for behind panel
float v[3][3] =  {{ 30.0f, 30.0f, -30.0f},
                  { -30.0f, 30.0f, -30.0f},
                  { 0.0f, 60.0f, 0.0f}};

// Calculate the normal for the plane
calcNormal(v,normal);

// Draw the triangle using the plane normal
// for all the vertices

        glNormal3fv(normal);
        glVertex3fv(v[0]);
        glVertex3fv(v[1]);
        glVertex3fv(v[2]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

{

// Vertices for right panel
float v[3][3] =  {{ -30.0f, 30.0f, -30.0f},
                  { -30.0f, 30.0f, 30.0f},
                  { 0.0f, 60.0f, 0.0f}};

// Calculate the normal for the plane
calcNormal(v,normal);

// Draw the triangle using the plane normal
// for all the vertices

        glNormal3fv(normal);
        glVertex3fv(v[0]);
        glVertex3fv(v[1]);
        glVertex3fv(v[2]);
}
glEnd();
}

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw the east
void DrawEast(void)
{
          int i;
          float z ,normal[3];

          // set colour is Blue
          glRGB(0,0,255);

          // Draw the East
          glBegin(GL_QUADS);
                    {
                    // Vertices for ground 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]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        glVertex3fv(v[3]);
                }

        for (i=0; i<3; i++)
        {
                // Vertices for ground 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

```
}
```

```
// Draw the Shading Device
void ShadingDevice(BOOL bShadow, char direct)
{
        float normal[3];

        if (!bShadow)
                // set colour is Red
                glRGB(255,0,0);
        else
                // set colour is Black
                glRGB(0,0,0);

        //Check direction of shading device
        switch(direct)
                {
                        // North
                        case 'N':
                                {

                                // Vertices for north panel
                                float v[4][3] =  {{30.0f ,30.0f , 30.0f},
                                                 { 60.0f,30.0f , 30.0f},
                                                 { 60.0f ,30.0f , -30.0f},
                                                 { 30.0f ,30.0f , -30.0f}};

                                // 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();
}
break;

// South
case 'S':
        {
        // Vertices for south panel
        float v[4][3] =  {{-30.0f ,30.0f , 30.0f},
                          { -30.0f,30.0f , -30.0f},
                          { -60.0f ,30.0f , -30.0f},
                          { -60.0f ,30.0f , 30.0f}};

                // 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();
            }
    break;

    //East
    case 'E':
            {
            // Vertices for east panel
            float v[4][3] =  {{30.0f ,30.0f , 30.0f},
                             { -30.0f,30.0f , 30.0f},
                             { -30.0f ,30.0f , 60.0f},
                             { 30.0f ,30.0f , 60.0f}};

            // 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();
}
break;


//West
case 'W':
        {
        // Vertices for west panel
        float v[4][3] =  {{30.0f ,30.0f , -30.0f},
                          { 30.0f,30.0f , -60.0f},
                          { -30.0f ,30.0f , -60.0f},
                          { -30.0f ,30.0f , -30.0f}};

        // 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();
                                        }
                                break;

                                default:
                                        return;
                        }

        }


        // Called to draw scene
        void RenderScene(void)
                {
                char direct;

                //Check direction of shading device
                // Store direct into direction
                if (North)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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(yRot, 0.0f, 1.0f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// draw the cube
Draw(FALSE);
//Draw shading device
ShadingDevice(FALSE,direct);


//Draw Pyramid
DrawPyramid(FALSE,Pyramid);

// Restore transformations
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);
glPushMatrix();

// Now rotate the cube around in the new flattend space
glRotatef(yRot, 0.0f, 1.0f, 0.0f);

// Multiply by shadow projection matrix on ground
glMultMatrixf((GLfloat *)shadowMat);

// draw shadow of cube
Draw(TRUE);

//Draw shadow of Shading Device
ShadingDevice(TRUE,direct);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Draw shadow of Pyramid
DrawPyramid(TRUE,Pyramid);

// Restore the projection to normal
glPopMatrix();
glPushMatrix();

// Now rotate the cube around in the new flattend space
glRotatef(yRot, 0.0f, 1.0f, 0.0f);

// Multiply by shadow projection matrix on direction of cube
glMultMatrixf((GLfloat *)shadowMatDirect);

//Draw shadow of Shading Device
ShadingDevice(TRUE,direct);

// Restore the projection to normal
glPopMatrix();

// Draw the light source
glPushMatrix();
glRotatef(yRot, 0.0f, 1.0f, 0.0f);

//Draw the sun and translation
glTranslatef(lightPos[0],lightPos[1], lightPos[2]);
glRGB(255,255,0);
auxSolidSphere(4.0f);
glPopMatrix();
glPushMatrix();
glRotatef(yRot, 0.0f, 1.0f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Draw the ground is green
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 the East
DrawEast();

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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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
        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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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)
        {
        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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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,
                                NULL,
                                NULL,
                                hInstance,
                                NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// 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
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
		Pyramid = FALSE;
		break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// 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;

// 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();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                // 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_LEFT)
                        yRot -= 5.0f;

                if(wParam == VK_RIGHT)
                        yRot += 5.0f;
                // Keep rotations within 360 degrees
                yRot = (float)((const int)yRot % 360);

                InvalidateRect(hWnd,NULL,FALSE);
                }
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            // 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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Put the day ,month ,time
        case ID_INPUT_TIME:
        DialogBox (hInstance,

MAKEINTRESOURCE(IDD_DIALOG_TIME),
                hWnd,
                TimeDlgProc);
        InvalidateRect(hWnd,NULL,FALSE);
        UpdateWindow(hWnd);
        break;

// Turn on Draw shading device on north
case ID_SHAPE_CUBE_NORTH:
North = !North;
if(North)

CheckMenuItem(hMenu,ID_SHAPE_CUBE_NORTH,MF_BYCOMMAND |
MF_CHECKED);
else

CheckMenuItem(hMenu,ID_SHAPE_CUBE_NORTH,MF_BYCOMMAND |
MF_UNCHECKED);

InvalidateRect(hWnd,NULL,FALSE);
break;

// Turn on Draw shading device on south
case ID_SHAPE_CUBE_SOUTH:
South = !South;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(South)
CheckMenuItem(hMenu,ID_SHAPE_CUBE_SOUTH,MF_BYCOMMAND |
MF_CHECKED);
else
CheckMenuItem(hMenu,ID_SHAPE_CUBE_SOUTH,MF_BYCOMMAND |
MF_UNCHECKED);

InvalidateRect(hWnd,NULL,FALSE);
break;

// Turn on Draw shading device on WEST
case ID_SHAPE_CUBE_WEST:
West = !West;

if(West)

CheckMenuItem(hMenu,ID_SHAPE_CUBE_WEST,MF_BYCOMMAND |
MF_CHECKED);
else

CheckMenuItem(hMenu,ID_SHAPE_CUBE_WEST,MF_BYCOMMAND |
MF_UNCHECKED);

InvalidateRect(hWnd,NULL,FALSE);
break;

// Turn on Draw shading device on EAST
case ID_SHAPE_CUBE_EAST:
East = !East;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(East)

CheckMenuItem(hMenu,ID_SHAPE_CUBE_EAST,MF_BYCOMMAND |
MF_CHECKED);
else

CheckMenuItem(hMenu,ID_SHAPE_CUBE_EAST,MF_BYCOMMAND |
MF_UNCHECKED);

InvalidateRect(hWnd,NULL,FALSE);
break;

// Turn on Draw pyramid
case ID_SHAPE_PYRAMID:
Pyramid = !Pyramid;

if(Pyramid)

CheckMenuItem(hMenu,ID_SHAPE_PYRAMID,MF_BYCOMMAND |
MF_CHECKED);
else

CheckMenuItem(hMenu,ID_SHAPE_PYRAMID,MF_BYCOMMAND |
MF_UNCHECKED);

InvalidateRect(hWnd,NULL,FALSE);
break;

// Display the about box
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        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));

        }

        return (0L);
        }


// Dialog Time procedure.
BOOL APIENTRY TimeDlgProc (HWND hDlg, UINT message, UINT wParam, LONG
lParam)
        {

        switch (message)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// 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
(hDllb,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;
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
                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;

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            // Closed from sysbox
            case WM_CLOSE:
                    EndDialog(hDlg,TRUE);
                    break;
            }

    return FALSE;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by Resource.rc
#define IDR_MENU                    101
#define IDD_DIALOG_ABOUT            102
#define IDD_DIALOG_TIME             106
#define IDI_ICON1                   107
#define IDI_ICON2                   108
#define IDC_EDIT_DAY                1013
#define IDC_EDIT_MONTH              1017
#define IDC_EDIT_TIME               1018
#define IDC_EDIT_HOUR               1018
#define IDC_EDIT_MIN                1020
#define ID_FILE_EXIT                40001
#define ID_HELP_ABOUT               40002
#define ID_INPUT_TIME               40006
#define ID_SHAPE_CUBE_NORTH         40010
#define ID_SHAPE_CUBE_SOUTH         40011
#define ID_SHAPE_CUBE_EAST          40012
#define ID_SHAPE_CUBE_WEST          40013
#define ID_SHAPE_PYRAMID            40014

// Next default values for new objects
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE    110
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define _APS_NEXT_COMMAND_VALUE        40017
#define _APS_NEXT_CONTROL_VALUE        1021
#define _APS_NEXT_SYMED_VALUE          101
#endif
#endif
```

Resource.rc

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
/////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

/////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

/////////////////////////////////////////////////////////////////////////////
// Unknown language: 0x1E, 0x1 resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_THA)
#ifdef _WIN32
LANGUAGE 0x1E, 0x1
#pragma code_page(874)
#endif //_WIN32
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/////////////////////////////////////////////////////////////////////////////
//
// Dialog
//

IDD_DIALOG_TIME DIALOG DISCARDABLE  0, 0, 212, 106
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Input Data"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON   "OK",IDOK,41,78,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,121,78,50,14
    EDITTEXT        IDC_EDIT_DAY,20,38,31,14,ES_AUTOHSCROLL | ES_NUMBER
    EDITTEXT        IDC_EDIT_MONTH,67,38,31,14,ES_AUTOHSCROLL | ES_NUMBER
    EDITTEXT        IDC_EDIT_HOUR,114,38,31,14,ES_AUTOHSCROLL | ES_NUMBER
    LTEXT           "Date",IDC_STATIC,25,18,16,8
    LTEXT           "Month",IDC_STATIC,73,18,23,8
    LTEXT           "Hour",IDC_STATIC,121,18,16,8
    EDITTEXT        IDC_EDIT_MIN,161,38,31,14,ES_AUTOHSCROLL | ES_NUMBER
    LTEXT           "Min",IDC_STATIC,169,18,16,8
END


/////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN
   IDD_DIALOG_TIME, DIALOG
        BEGIN
                LEFTMARGIN, 7
                RIGHTMARGIN, 205
                TOPMARGIN, 7
                BOTTOMMARGIN, 92
        END
END
#endif   // APSTUDIO_INVOKED



/////////////////////////////////////////////////////////////////////////////
//
// Icon
//

// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.
IDI_ICON2           ICON    DISCARDABLE     "icon_1.ico"

#ifndef _MAC
/////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x40004L
 FILETYPE 0x1L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "041e04b0"
        BEGIN
                VALUE "CompanyName", "kmitl\0"
                VALUE "FileDescription", "solar\0"
                VALUE "FileVersion", "1, 0, 0, 1\0"
                VALUE "InternalName", "solar\0"
                VALUE "LegalCopyright", "Copyright _ 1998\0"
                VALUE "OriginalFilename", "solar.exe\0"
                VALUE "ProductName", "kmitl solar\0"
                VALUE "ProductVersion", "1, 0, 0, 1\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x41e, 1200
    END
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#endif   // !_MAC

#endif   // Unknown language: 0x1E, 0x1 resources
/////////////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

/////////////////////////////////////////////////////////////////////////////
//
// Dialog
//
IDD_DIALOG_ABOUT DIALOG DISCARDABLE  0, 0, 210, 90
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Solar Project Team"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON   "OK",IDOK,80,65,50,16
    LTEXT    "Mr. Thanupong Ngaojaruwong 37014159",IDC_STATIC,43,15,143,8
    LTEXT    "Mr. Morakot Srianujata 37014337",IDC_STATIC,56,28,123,8
    LTEXT    "Mechanical Engineering Department, KMIT'L, 1997",
   IDC_STATIC,37,42,157,8
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG_ABOUT, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 203
        TOPMARGIN, 7
        BOTTOMMARGIN, 81
    END
END
#endif    // APSTUDIO_INVOKED


/////////////////////////////////////////////////////////////////////////////
//
// Menu
//

IDR_MENU MENU DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "E&xit",                       ID_FILE_EXIT
    END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    POPUP "&Input"
    BEGIN
        MENUITEM "&Time",                ID_INPUT_TIME
    END
    POPUP "&Shape"
    BEGIN
        POPUP "Shading Device"
        BEGIN
            MENUITEM "&North",           ID_SHAPE_CUBE_NORTH
            MENUITEM "&South",           ID_SHAPE_CUBE_SOUTH
            MENUITEM "&East",            ID_SHAPE_CUBE_EAST
            MENUITEM "West",             ID_SHAPE_CUBE_WEST
        END
        MENUITEM "Pyramid",              ID_SHAPE_PYRAMID
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About",               ID_HELP_ABOUT
    END
END


#ifdef APSTUDIO_INVOKED
/////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
   "#include ""afxres.h""\r\n"
   "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
   "\r\n"
   "\0"
END

#endif    // APSTUDIO_INVOKED

#endif    // English (U.S.) resources
/////////////////////////////////////////////////////////////////////////////


#ifndef APSTUDIO_INVOKED
/////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//


/////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายการทำงานของฟังก์ชันต่าง ๆ

Procedure

TimeDlgProc	: ใช้สำหรับรับค่าวัน และเวลาทางคีย์บอร์ด แล้วไปทำการคำนวณ

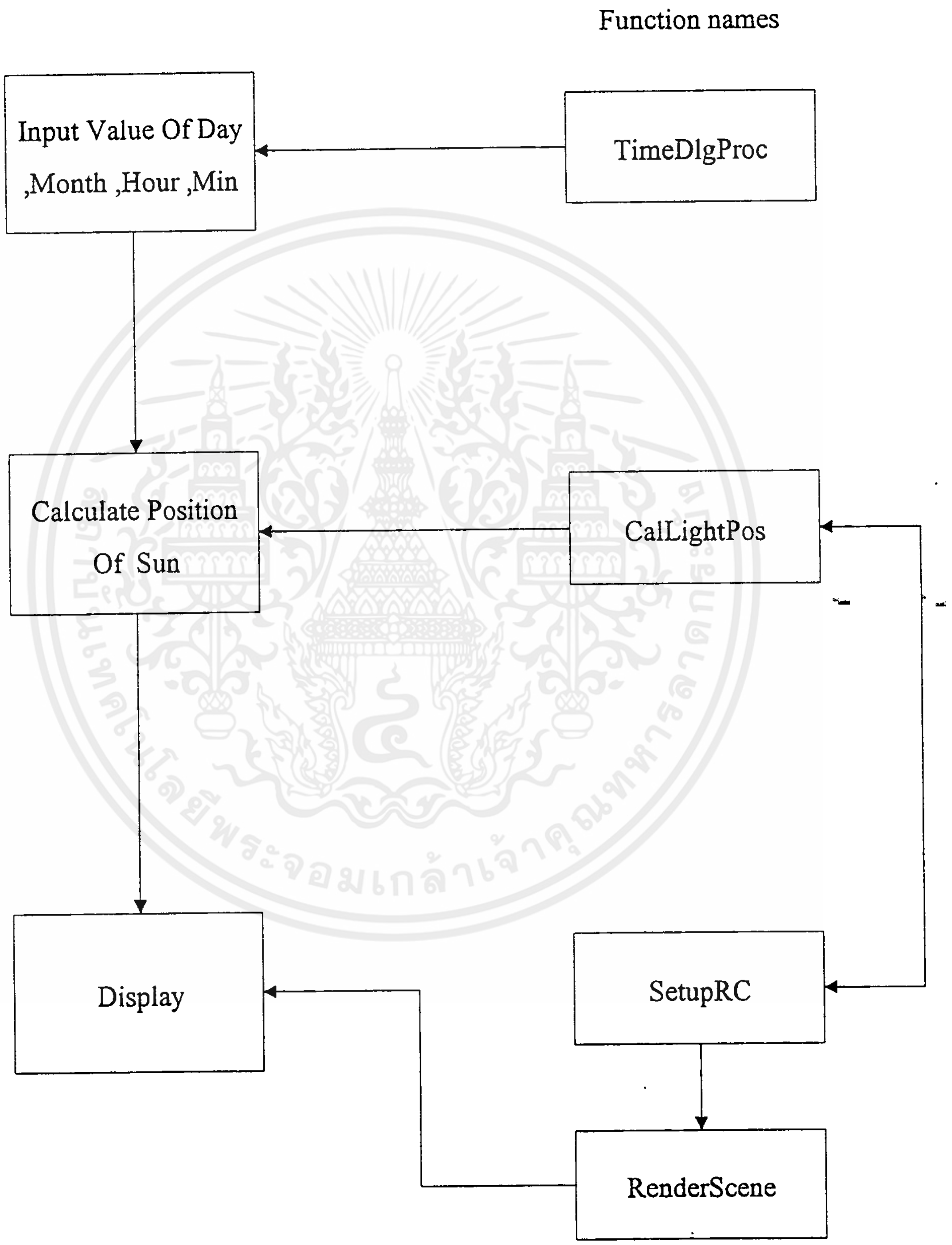
Function

RenderScene	: ใช้เพื่อสร้าง Polygons รูปต่างๆ และสร้างเงา
SetupRC	: ไว้สำหรับการเตรียมข้อมูลต่างๆ ที่จำเป็นต้องใช้ในการแสดงผล
CalLightPos	: สำหรับคำนวณหาตำแหน่งของดวงอาทิตย์
MakeShadowMatrix	: คำนวณหาตำแหน่งที่จะแสดงเงา
CalcNormal	: คำนวณหา Normal Vector
ReduceToUnit	: คำนวณหา Vector 1 หน่วย
DrawEast	: สร้างรูปตัว E
Draw	: สร้างรูปกล่อง
ChangeSize	: ป้องกันสัดส่วน ไม่ให้เปลี่ยนแปลง เมื่อมีการเปลี่ยนขนาดของวินโดว์ และยึดตำแหน่งมุมมองให้คงที่
ShadingDevice	: แสดงลักษณะอุปกรณ์กันแดดที่จำลองขึ้น ในทิศต่างๆ ที่เลือกไว้
DrawPyramid	: แสดงรูปปิรามิดบนรูปลูกบาศก์

แสดงขั้นตอนการทำงานของโปรแกรม

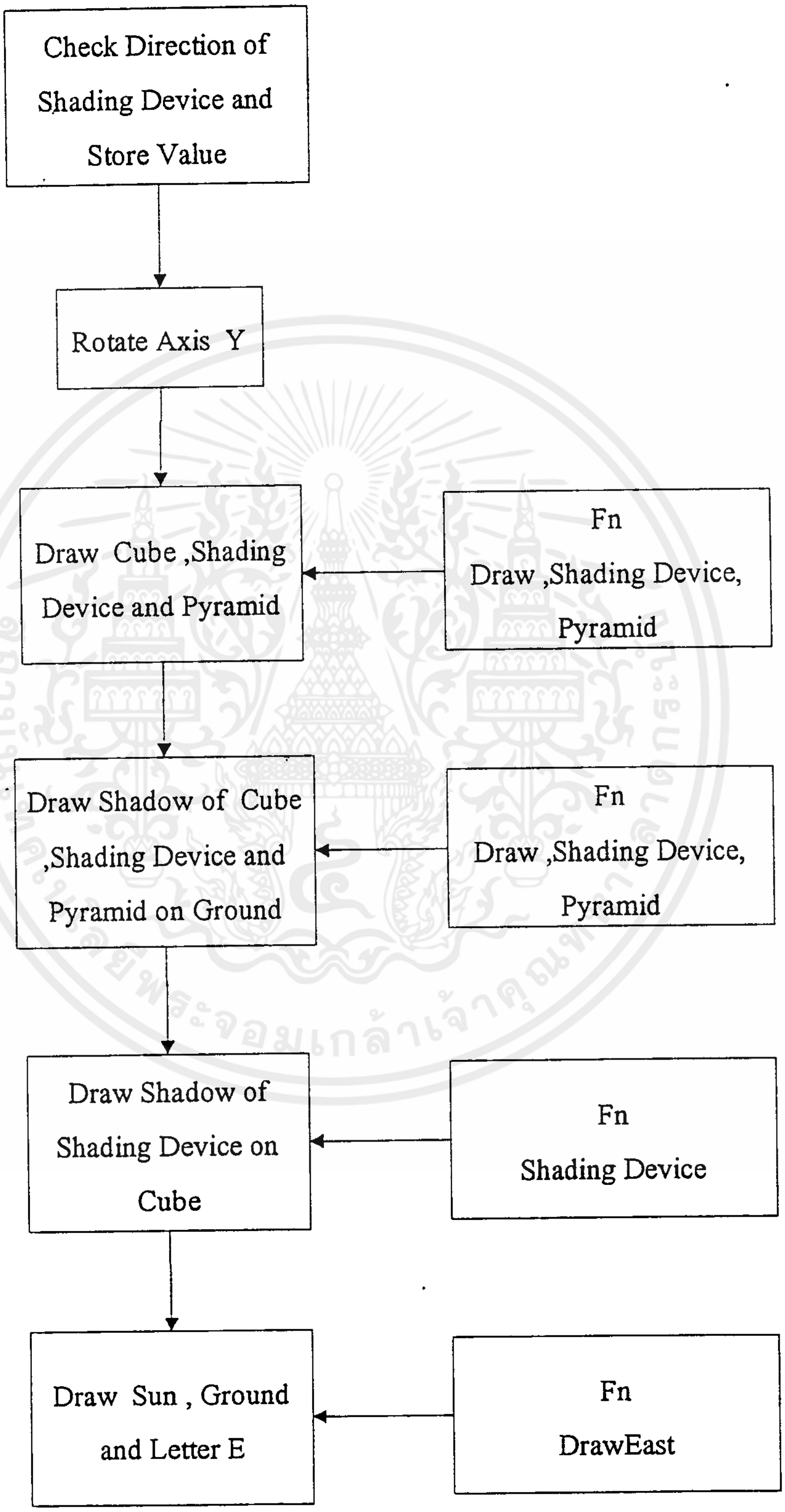
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงขั้นตอนการทำงานของโปรแกรม



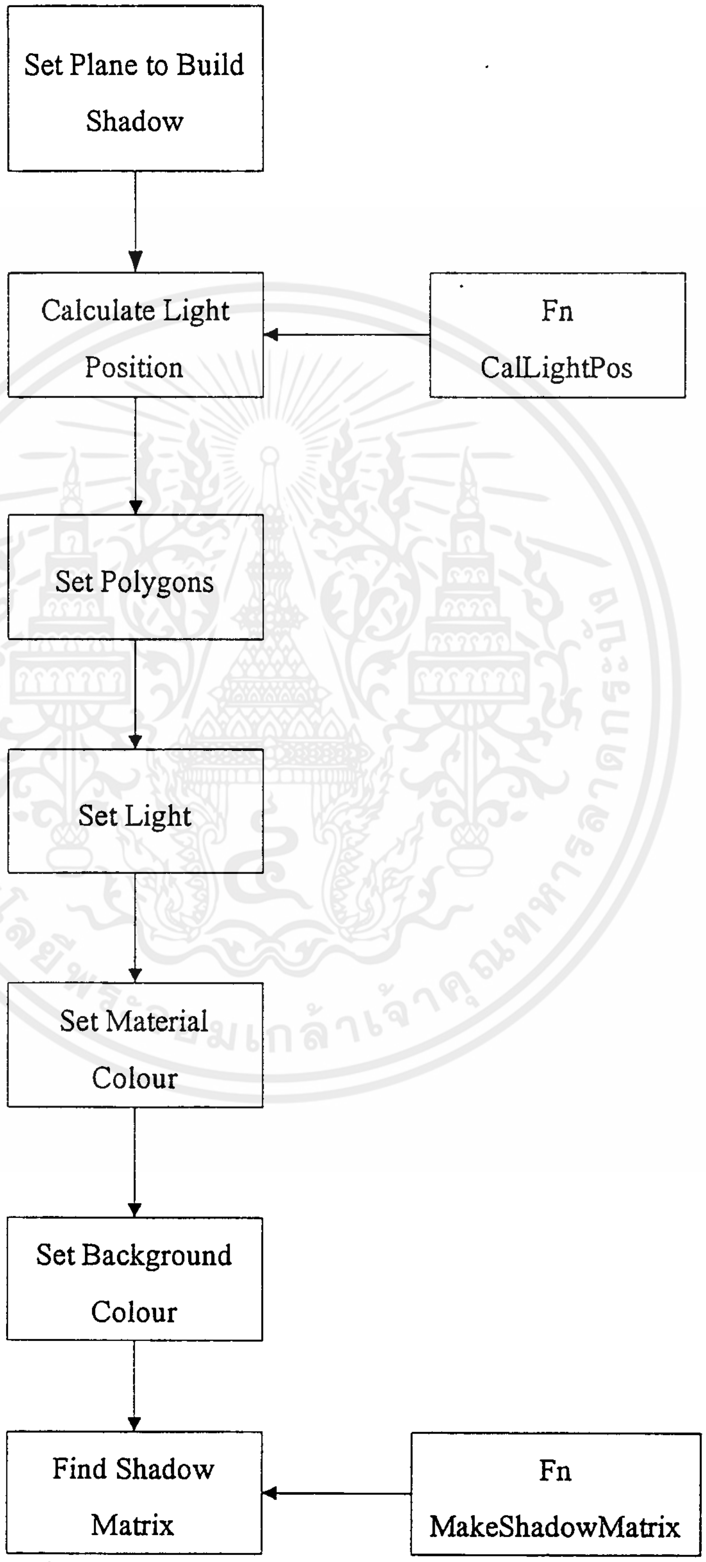
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function RenderScene



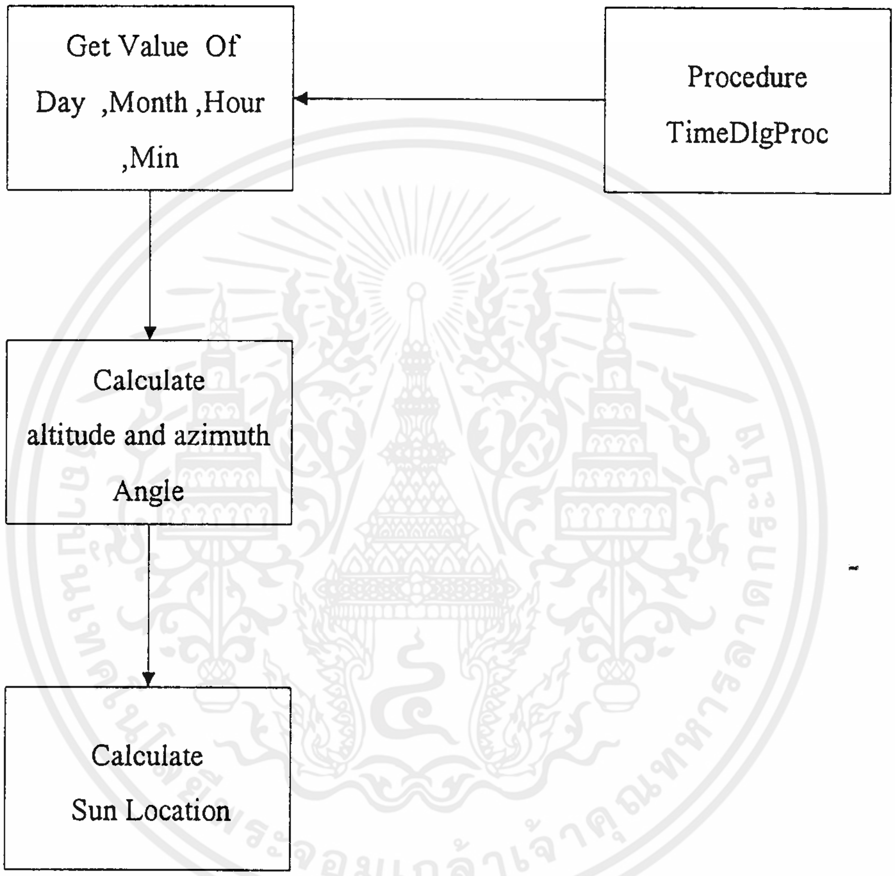
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function SetupRC



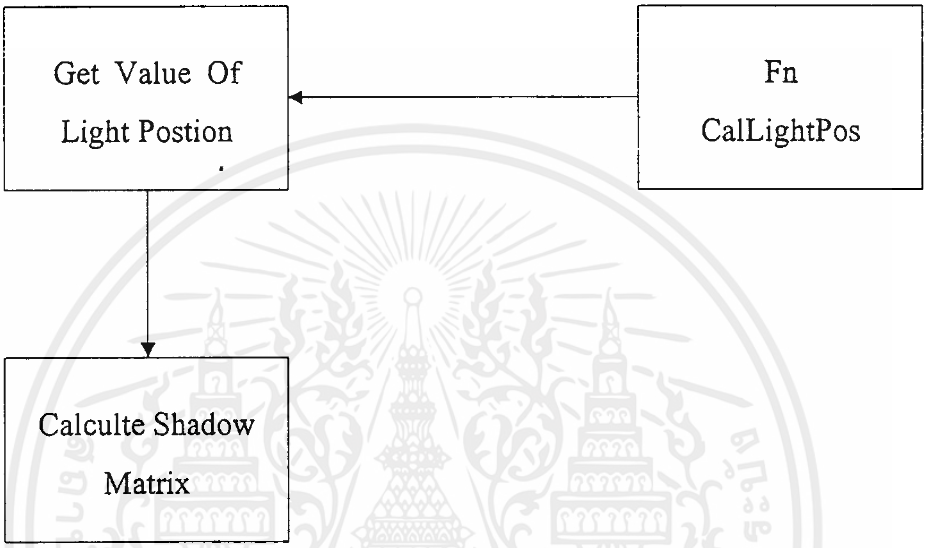
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function CalLightPos



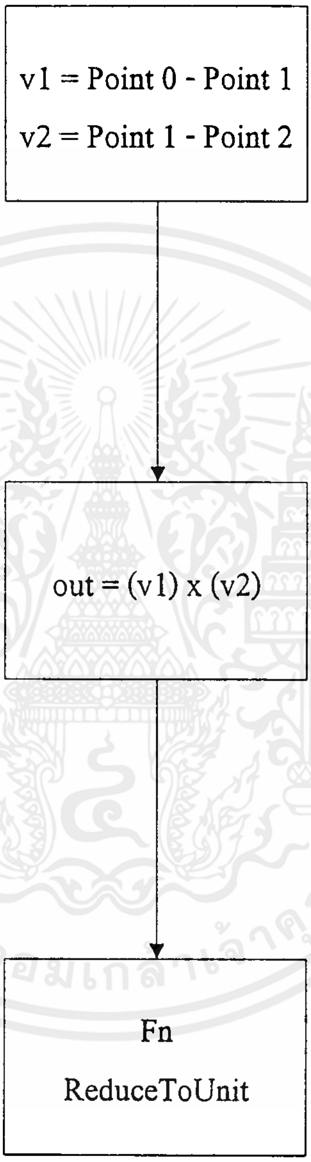
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function MakeShadowMatrix



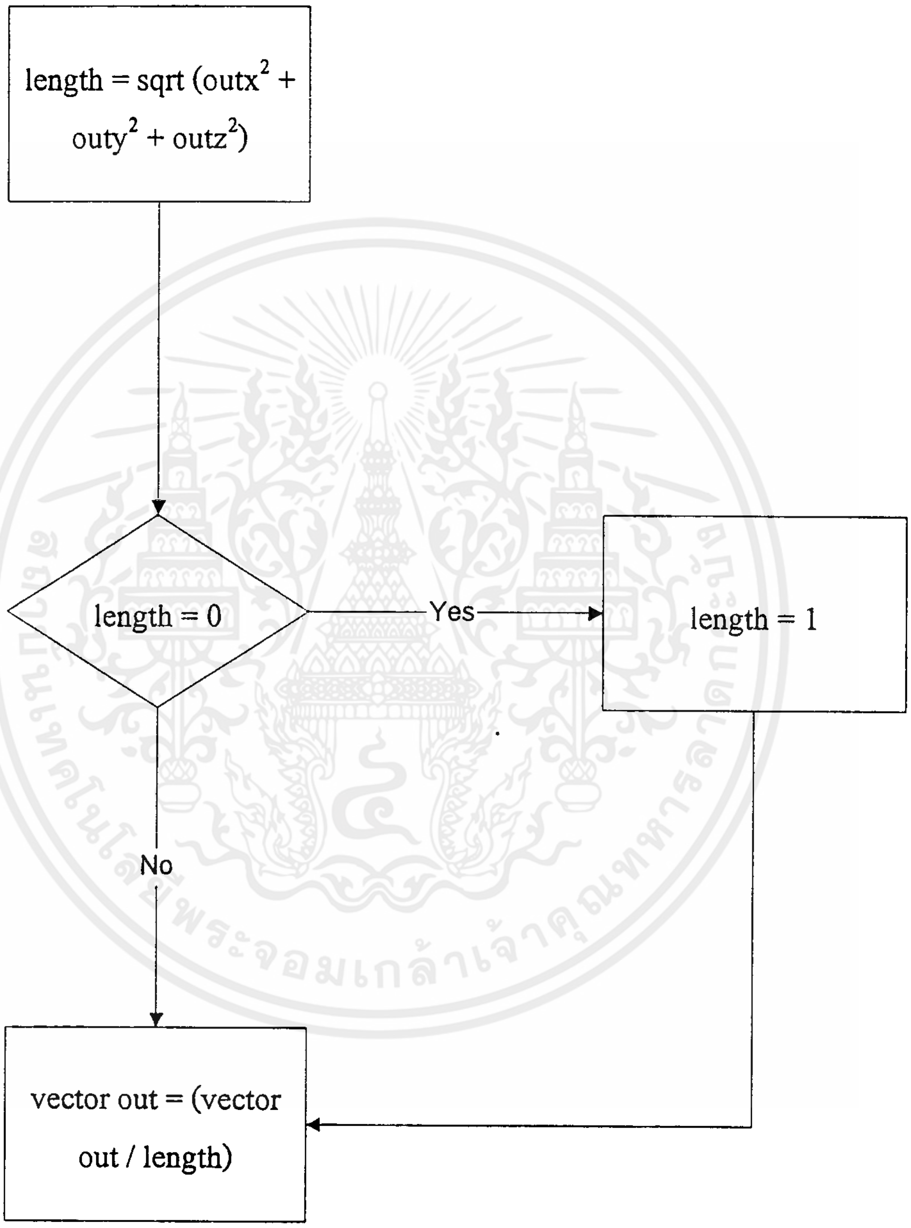
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function CalcNormal



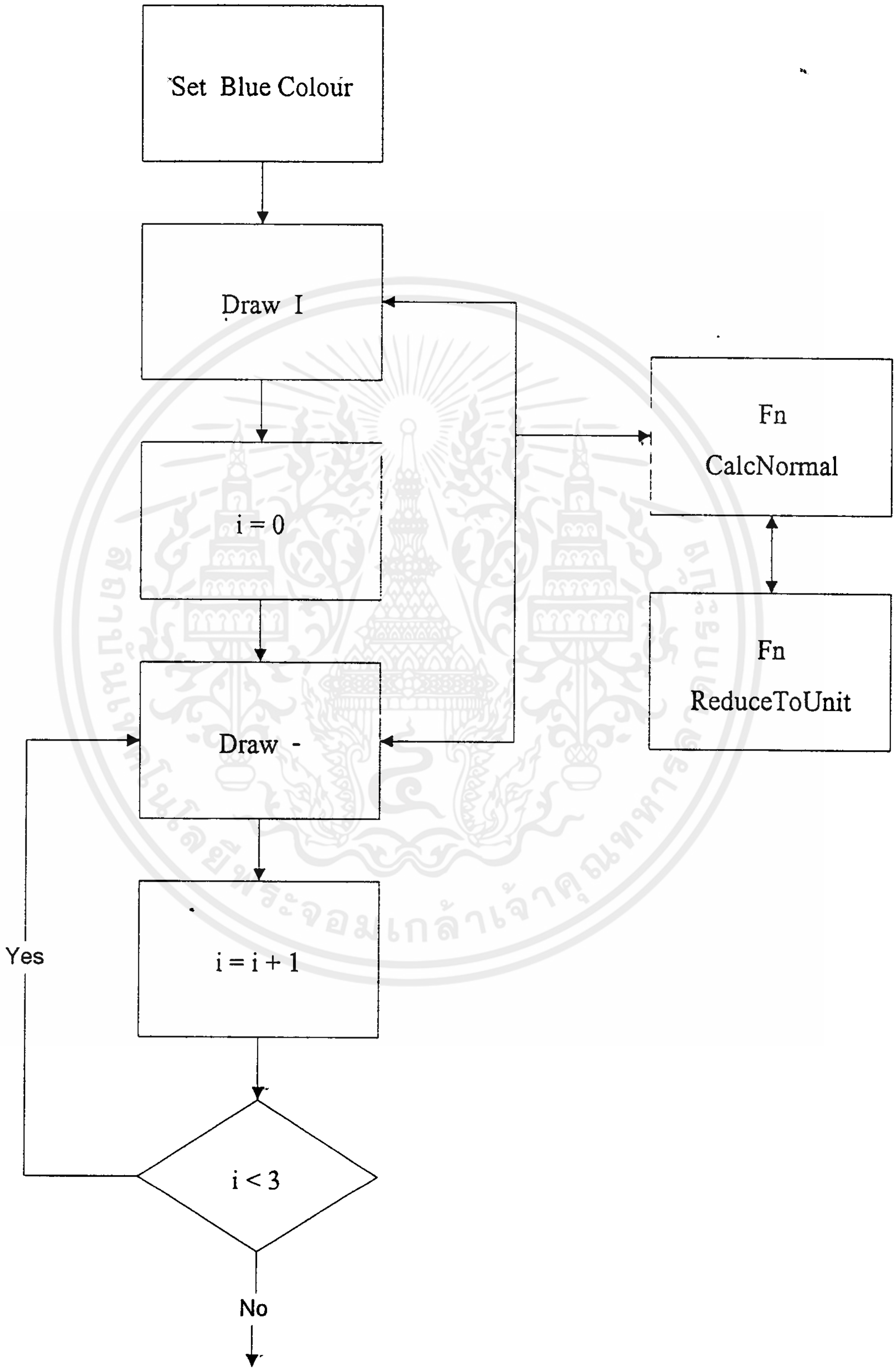
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ReduceToUnit



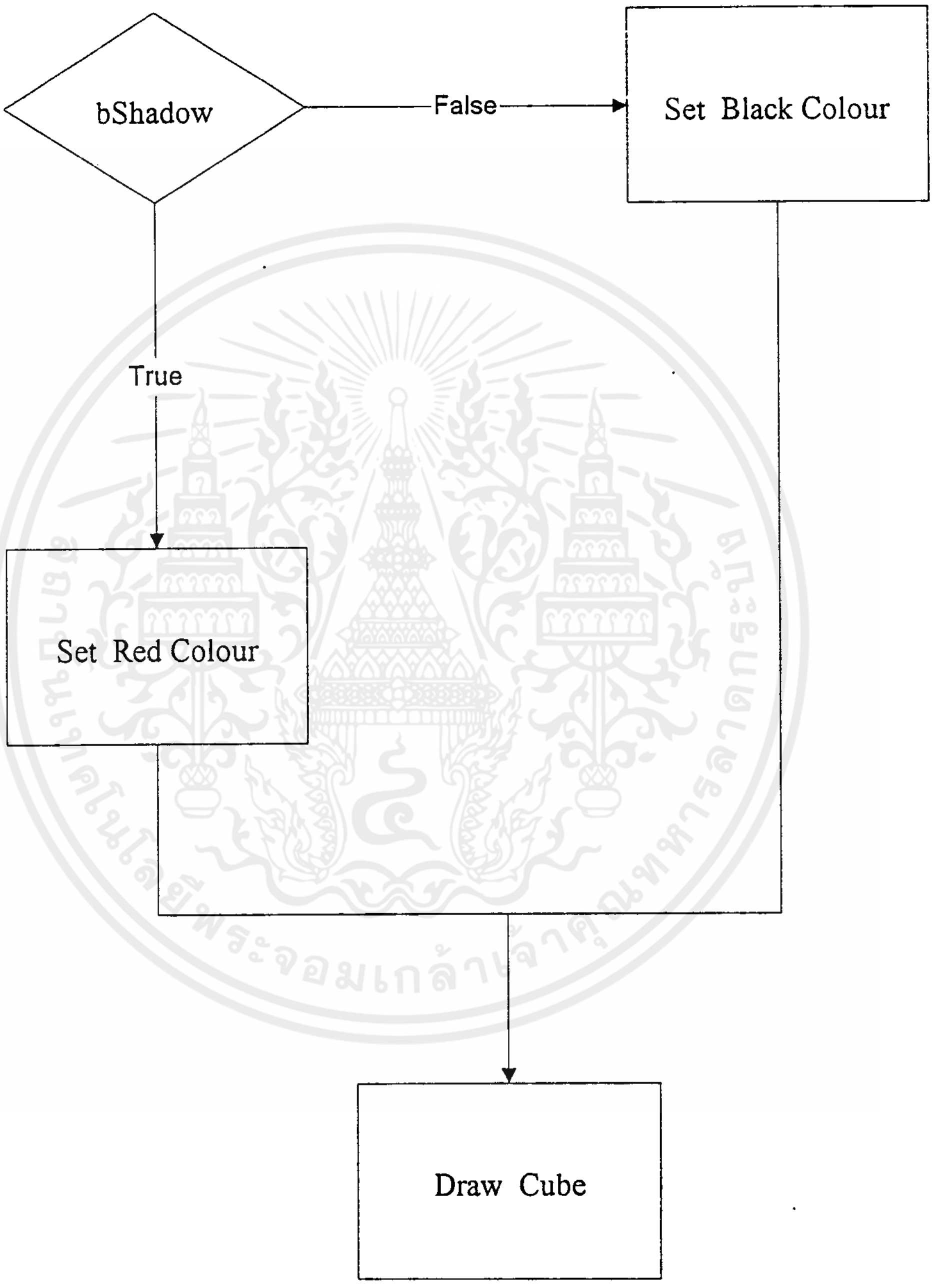
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawEast



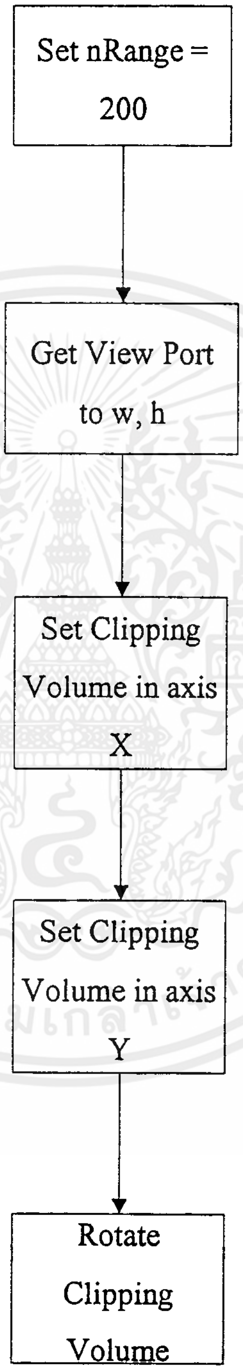
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function Draw



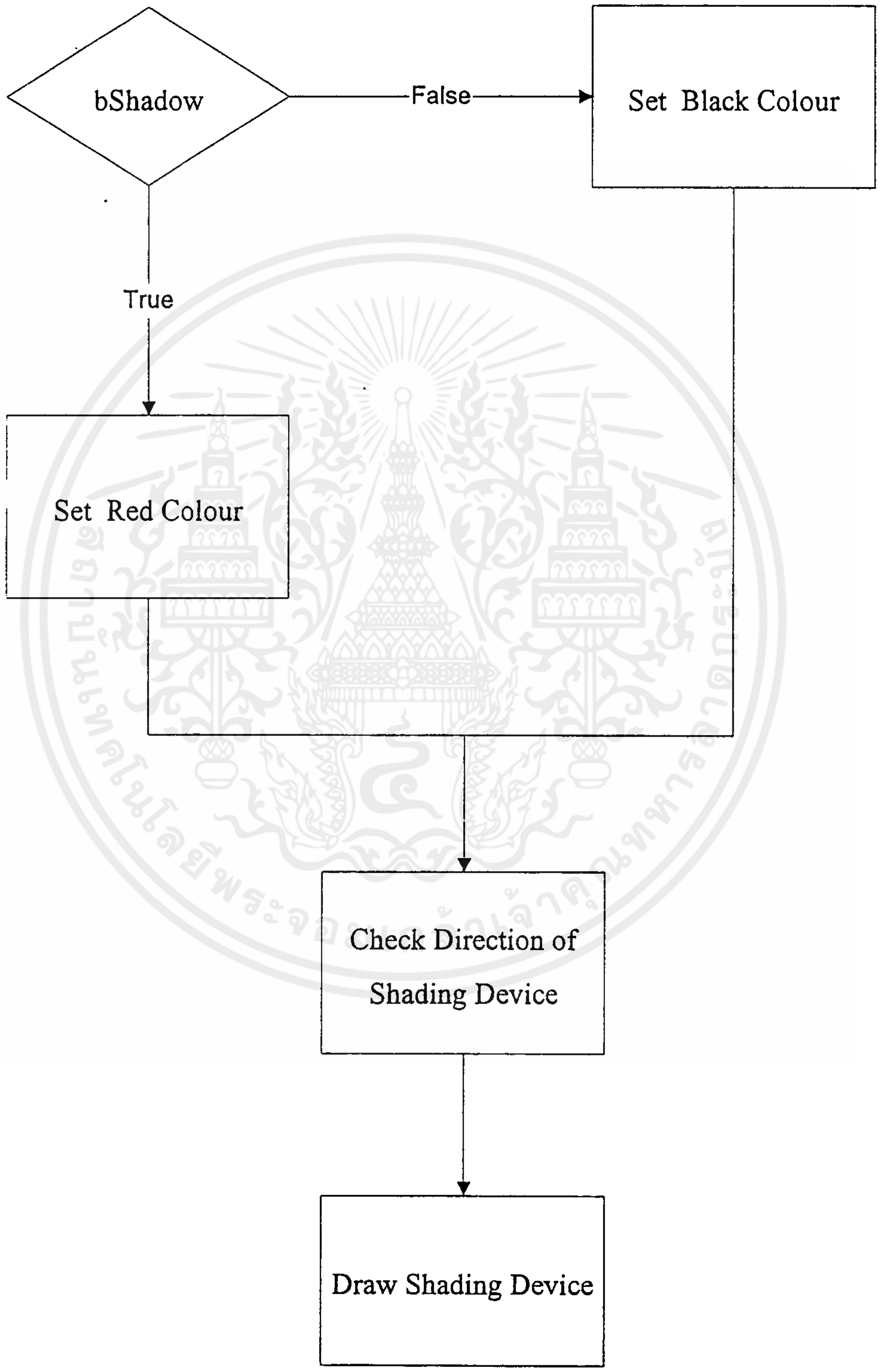
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ChangeSize



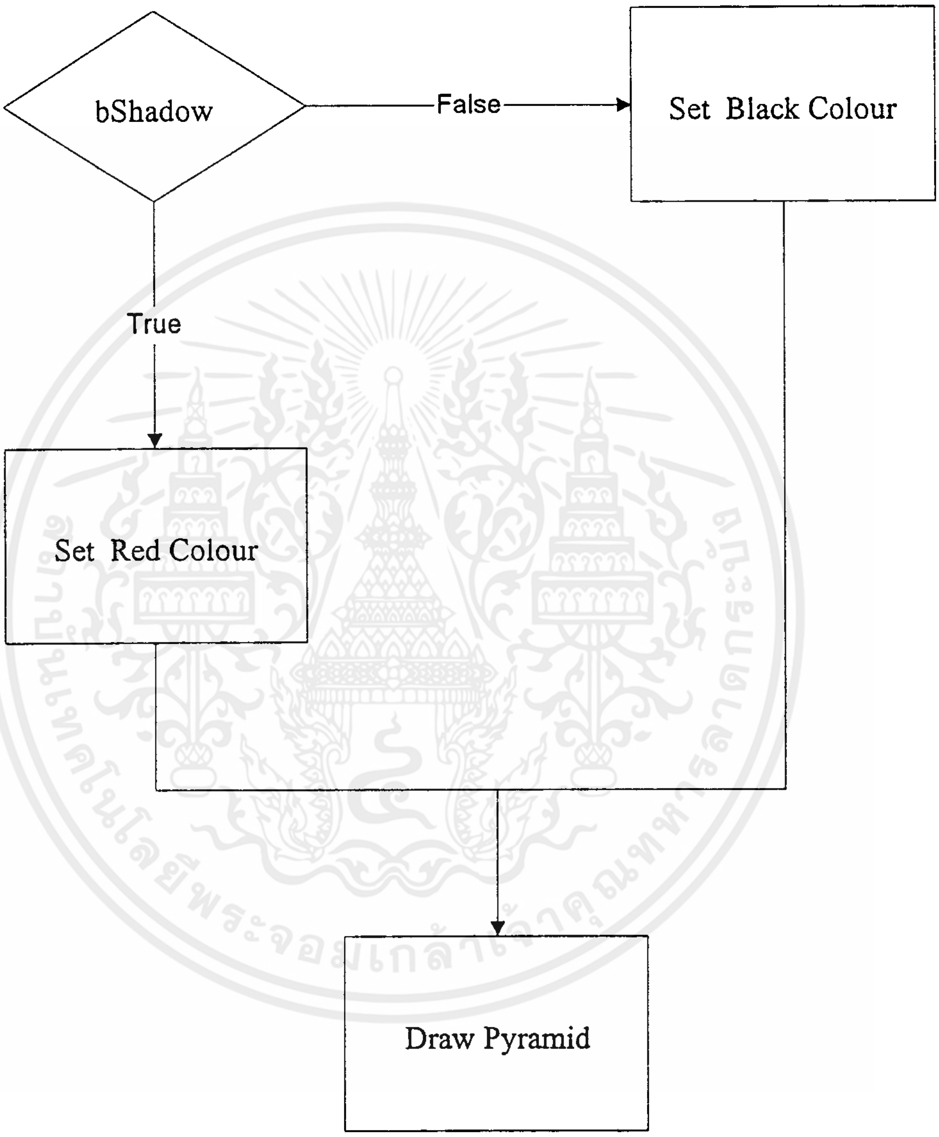
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function ShadingDevice



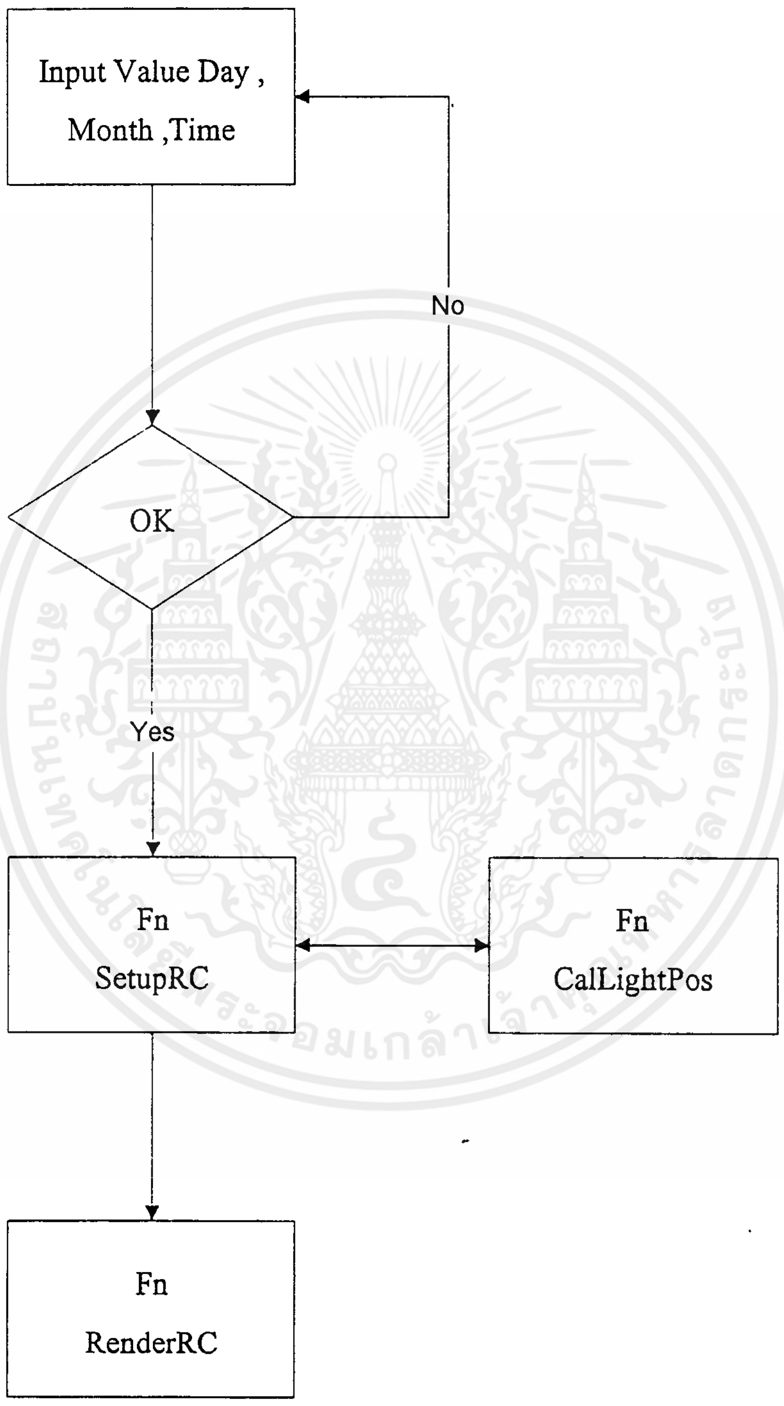
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Function DrawPyramid



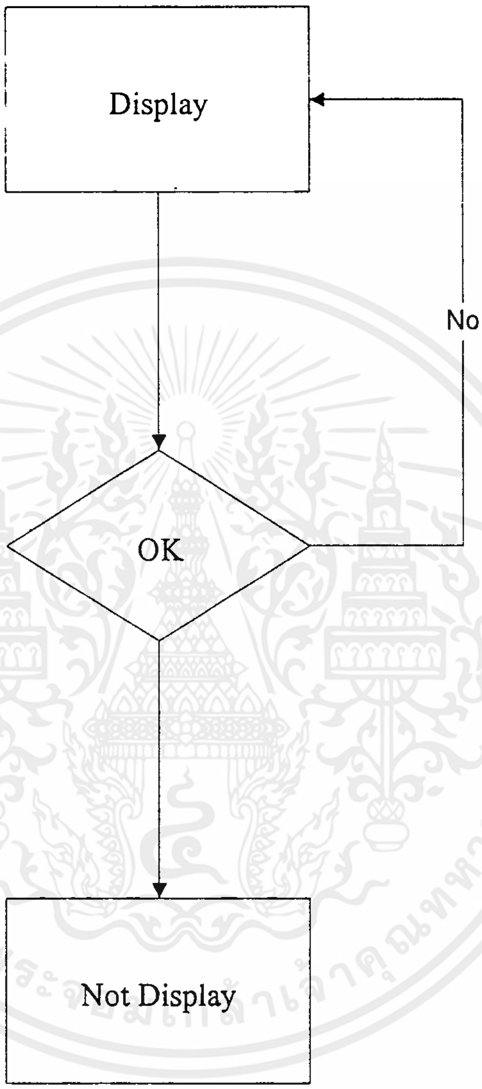
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Procedure TimeDlgProc

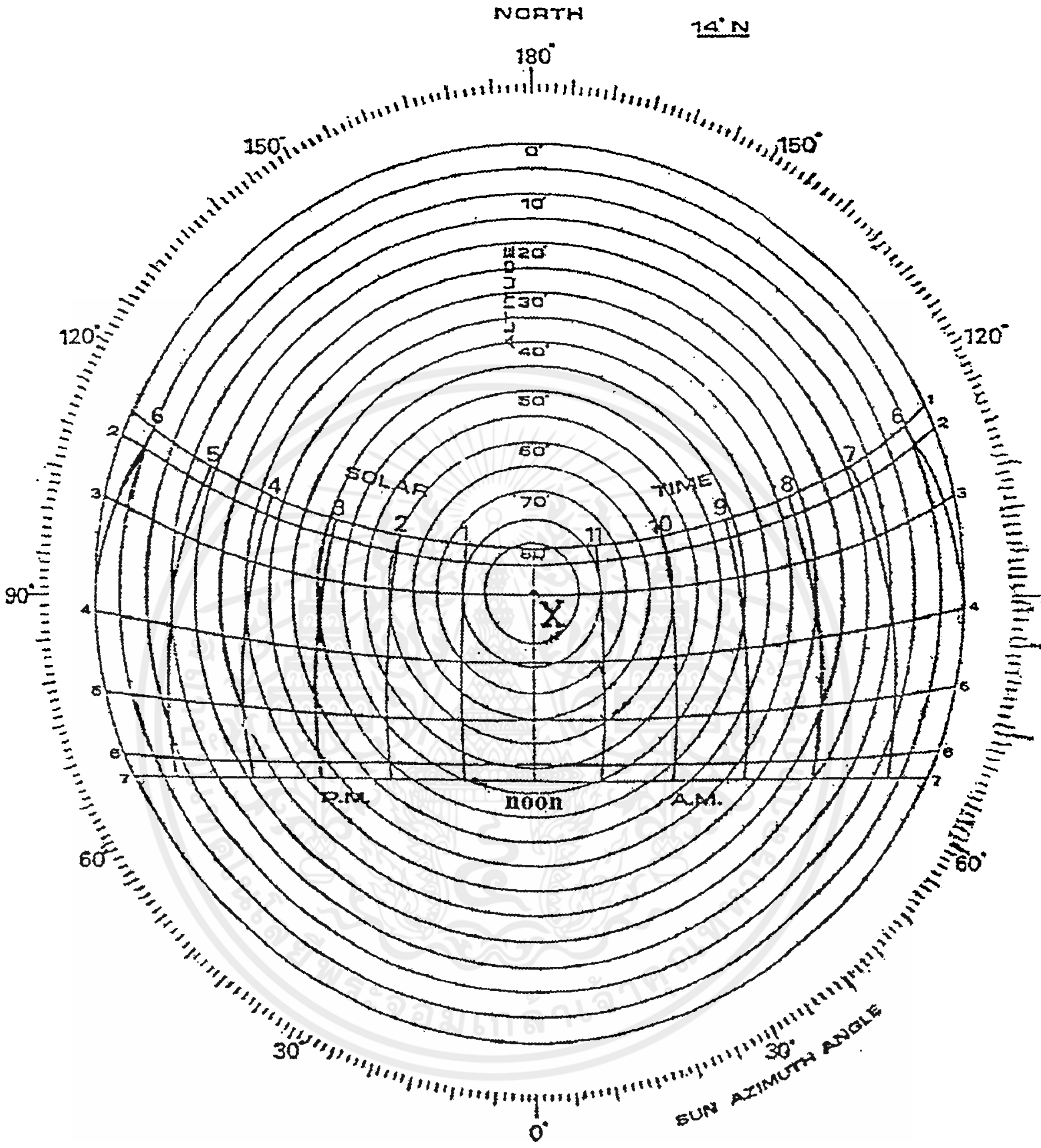


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Procedure AboutDlgProc



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง Sun Path Diagram ของประเทศไทย ที่ละติจูด 14° เหนือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Ted J. Jansen, "Solar Engineering Technology", New Jersy: Prentice - Hall, 1985.
2. Nathan and Ori Gurewich, "Visual C++ 5 in 21 Days", 4th.Ed., Sams Publishing,1997.
3. Chris H. Pappas, William H. Murray,III, "The Visual C++ Handbooks", 2nd .Ed., Osborne: McGraw – Hill
4. Richard S. Wright JR., Michael Sweet, "OpenGL Superbible", Waite Group Press™
5. Jan F. Kreider and Ari Rabl, "Heating and Cooling of Builders", New York: McGraw – Hill, 1994.
6. มัณฑนา ปราการสมุทร, "การเขียนชุดคำสั่งภาษาซี", พิมพ์ครั้งที่ 3, บริษัท ดวงกมลสมัย จำกัด, 2534.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้