บทที่ 3

การออกแบบและพัฒนาโปรแกรม

โปรแกรมการสร้างภาพแบบจำลอง 3 มิติ มีวัตถุประสงค์เพื่อใช้เป็นแนวทางในการศึกษาการพัฒนาโปรแกรมกราฟิก โดยใช้ชุดคำสั่งซันฟิกส์รุ่น 2.0 ร่วมกับภาษาซี ส่วนตัวประสานกับผู้ใช้บนระบบโอเพนวินโดว์รุ่น 3.0 ซึ่งเป็นระบบเอกซ์วินโดว์แบบโอเพนลุก จะใช้เอกซ์วิว ทูลคิต ในการพัฒนา ดังนั้นโปรแกรมนี้จะทำงานโดยใช้โนติฟายเออร์

เนื่องจากโปรแกรมนี้ถูกสร้างขึ้นเพื่อใช้ศึกษาและทดสอบซันฟิกส์รุ่น 2.0 และใช้เป็นแนวทางในการพัฒนาโปรแกรมด้วยซันฟิกส์รุ่น 2.0 ต่อไป ดังนั้นโปรแกรมจะประกอบด้วยการทำงานและการประยุกต์จากทฤษฎีด้านคอมพิวเตอร์กราฟิกขั้นพื้นฐานเป็นส่วนใหญ่ การใช้ทฤษฎีคอมพิวเตอร์กราฟิกขั้นสูงจะใช้เมื่อเกี่ยวข้องกับซันฟิกส์รุ่น 2.0 โดยตรงเท่านั้น เช่น การใช้เส้นโค้งบีสไปลน์ เป็นต้น ซึ่งในบทนี้จะกล่าวถึงลักษณะของโปรแกรมและการทำงานของโปรแกรม

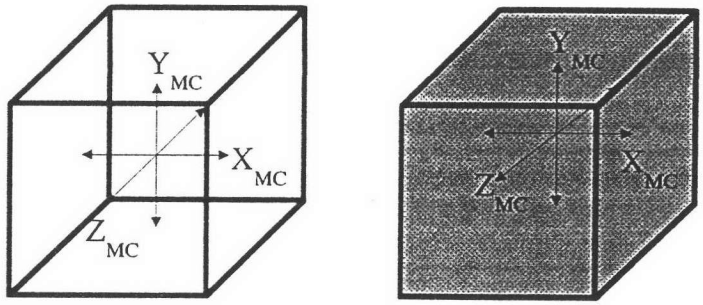
ลักษณะของโปรแกรม

1. หลักการทำงานของโปรแกรม

การสร้างภาพแบบจำลอง 3 มิติในโปรแกรมนี้ สามารถแบ่งขั้นตอนการสร้างภาพออกได้เป็น 3 ส่วน คือ ส่วนการสร้างวัตถุ ส่วนการสร้างแบบจำลอง และส่วนการสร้างฉาก (Scene)

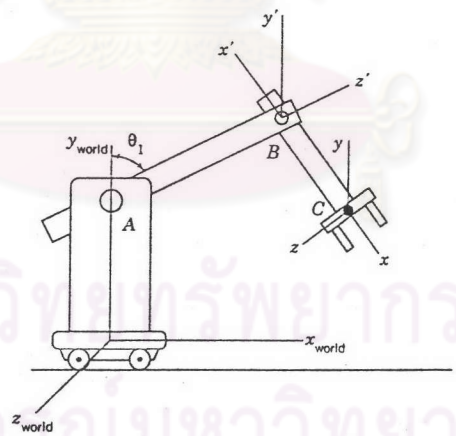
คำว่า วัตถุ จะหมายถึง โครงสร้างของฟิกส์ 1 โครงสร้าง ที่ประกอบด้วยลักษณะประจำของวัตถุ และรูปทรง 3 มิติที่กำหนดด้วยรูปทรงพื้นฐานของฟิกส์ 1 ชนิด ซึ่งลักษณะประจำของวัตถุจะประกอบด้วยลักษณะประจำของรูปทรงพื้นฐานชนิดนั้น ลักษณะการสะท้อนแสง การซ่อนพื้นผิว และลักษณะการเชดสี

ส่วนรูปทรง 3 มิติ คือ ข้อมูลที่ใช้กำหนดรูปทรงในแบบ 3 มิติ ให้กับรูปทรงพื้นฐานชนิดนั้น ซึ่งมี 2 ลักษณะ คือ การกำหนดจุดยอดให้กับรูปทรงพื้นฐานในกลุ่มของพื้นที่เติมเต็ม และการกำหนดจุดควบคุม (Control Point) ลำดับ (Order) และน็อต เวกเตอร์ (Knot Vector) ให้กับรูปทรงพื้นฐานในกลุ่มของบีสไปลน์ ข้อมูลของรูปทรง 3 มิติ นี้ จะถูกกำหนดอยู่ในพิกัดแบบจำลองของวัตถุนั้น โดยแต่ละวัตถุก็จะมีพิกัดแบบจำลองเป็นของตัวเอง



รูป 3.1 ลักษณะของวัตถุ

แบบจำลอง คือ การนำเอาวัตถุหรือแบบจำลองอื่นๆ มาประกอบกันเป็นแบบจำลองใหม่ โดยการแปลงพิกัดแบบจำลองของวัตถุหรือแบบจำลองนั้น ให้เป็นตำแหน่งที่ถูกต้องบนพิกัดแบบจำลองของแบบจำลองใหม่ แบบจำลองจะมีลักษณะเป็นโครงสร้างของฟิกส์ที่มีองค์ประกอบ 2 ชนิด คือ องค์ประกอบที่เรียกใช้โครงสร้างอื่น ซึ่งเป็นการอ้างถึงวัตถุหรือแบบจำลองที่เป็นส่วนประกอบของแบบจำลองนั้น และองค์ประกอบชนิดการแปลง เพื่อใช้กำหนดตำแหน่งให้กับส่วนประกอบต่างๆ ของแบบจำลอง



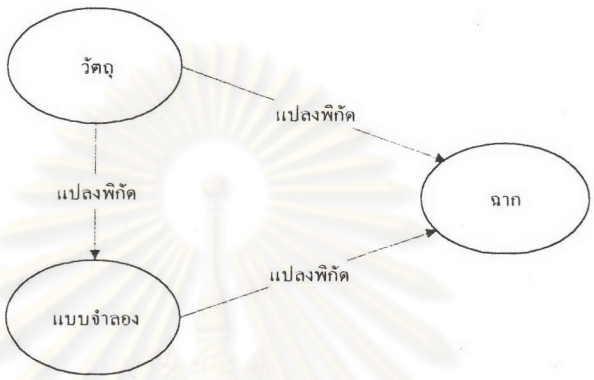
รูป 3.2 แบบจำลอง (Anand, 1993)

ฉาก คือ การนำเอาแบบจำลองหรือวัตถุมาวางรวมกันในฉาก โดยแปลงจากพิกัดแบบจำลองของแบบจำลองหรือวัตถุนั้นมาเป็นพิกัดโลก แล้วจึงทำการกำหนดการให้แสงและทำการเรนเดอร์ภาพ แสงที่ใช้ในโปรแกรมนี้ได้มี 4 ชนิด คือ แอมเบียนต์ (Ambient) ไดเรกชันนัล (Directional) โพซิชันนัล (Positional) และสปอต (Spot) ซึ่งข้อมูลที่ใช้ในการกำหนดการให้แสง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จะแตกต่างกันไปตามชนิดของแสง โครงสร้างของฉากจะมีองค์ประกอบ 3 ชนิด คือ องค์ประกอบที่เรียกใช้โครงสร้างอื่น องค์ประกอบชนิดการแปลง และองค์ประกอบที่ใช้กำหนดการให้แสง

จากที่กล่าวมาข้างต้นสามารถแสดงให้เห็นความสัมพันธ์ระหว่างส่วนประกอบทั้ง 3 ได้ดังรูปที่ 3.3



รูป 3.3 ความสัมพันธ์ของการสร้างวัตถุ การสร้างแบบจำลอง และการสร้างฉาก

ในขั้นตอนการสร้างภาพทั้ง 3 ขั้นตอนนี้ จะประกอบด้วยการทำงาน 3 ชนิด คือ การสร้าง (Create) วัตถุ แบบจำลองหรือฉากใหม่ การเก็บ (Save) และการค้นคืน (Retrieve) และการเลือก (Pick หรือ Select) และการแก้ไข (Edit) โดยการทำงานแต่ละอย่างจะมีรายละเอียดแตกต่างกันไป

1.1 การสร้างวัตถุ

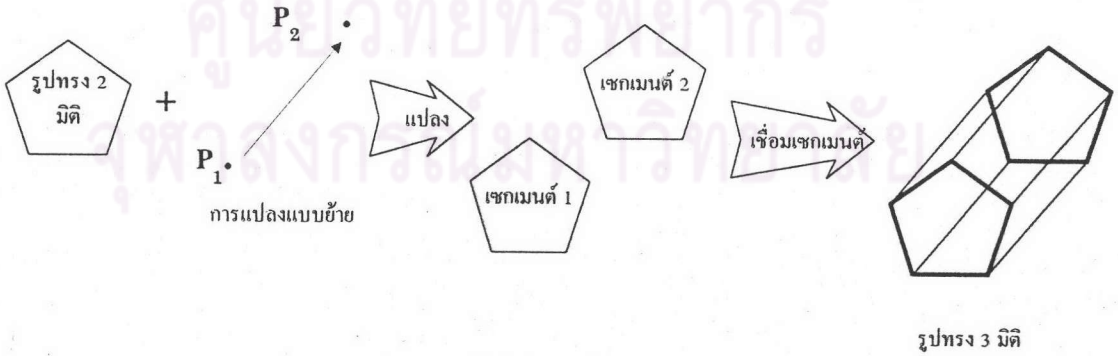
การสร้างวัตถุใหม่ทำได้โดยการนำรูปทรง 3 มิติมากำหนดลักษณะประจำของวัตถุ แล้วนำไปใส่ในโครงสร้างของวัตถุนั้น การสร้างรูปทรง 3 มิติสามารถทำได้ 2 วิธี คือ การนำรูปทรง 3 มิติที่มีอยู่แล้วมาใช้ และการสร้างรูปทรง 3 มิติจากรูปทรง 2 มิติ (Shape) รูปทรง 3 มิติที่ถูกสร้างเสร็จแล้วยังสามารถนำมาทำการแก้ไขได้ด้วย

การนำรูปทรง 3 มิติที่มีอยู่แล้วมาใช้ มี 2 ลักษณะ คือ การนำรูปทรง 3 มิติจากแฟ้มมาใช้ และการสร้างรูปทรง 3 มิติจากรูปทรง 3 มิติที่เตรียมไว้แล้ว เช่น ทรงลูกบาศก์ โดยผู้ใช้สามารถทำการกำหนดขนาดของรูปทรง 3 มิติได้

การสร้างรูปทรง 3 มิติจากรูปทรง 2 มิติ คือ การนำรูปทรง 2 มิติ มาทำการกวาด ซึ่งสามารถเลือกได้ 2 วิธี คือ การกวาดแบบย้าย (Translation Sweep) หรือการกวาดแบบหมุน (Rotation Sweep) สำหรับการกวาดแบบย้ายอาจเรียกได้อีกอย่างว่า การสร้างรูปยื่น ส่วนการกวาดแบบหมุนอาจเรียกว่า การหมุนพื้นผิว (Surface Revolution) หรือเรียกสั้นๆ ว่าการกวาด

การสร้างรูปทรง 2 มิติจะทำได้ 2 วิธี คือ การวาดรูปทรง 2 มิติขึ้นมาใหม่ และการนำรูปทรง 2 มิติที่มีอยู่แล้วมาใช้ รูปทรง 2 มิติจะถูกกำหนดด้วยรูปทรงพื้นฐานของฟีกส์ 2 ชนิด คือ โพลีไลน์และเส้นโค้งบีสไปลน์ โพลีไลน์จะใช้สำหรับรูปทรงประเภทเส้นและพื้นที่แบบโพลีกอน โดยพื้นที่แบบโพลีกอนก็คือโพลีไลน์ที่มีจุดเริ่มต้นและจุดสุดท้ายเป็นจุดเดียวกัน การที่ใช้โพลีไลน์แสดงรูปพื้นที่พื้นที่ แทนการใช้รูปทรงพื้นฐานชนิดพื้นที่เติมเต็ม ทั้งนี้เนื่องจากปัญหาเรื่องลำดับความสำคัญ (Priority) ของฟีกส์ คือ รูปที่ถูกวาดครั้งหลังสุดจะถูกวาดทับรูปที่วาดก่อน ทำให้ไม่สามารถใช้อุปกรณ์นำเข้าของฟีกส์แบบพิกในการเลือกรูปทรงที่วาดก่อนได้ ส่วนเส้นโค้งบีสไปลน์นั้น เมื่อผู้ใช้กำหนดจุดควบคุมและลำดับแล้ว โปรแกรมจะสร้างน็อต เวกเตอร์ชนิดยูนิฟอร์มแบบเปิด (Open Uniform) ให้ ดังนั้นเส้นโค้งบีสไปลน์ที่ถูกวาดครั้งแรกจะมีลักษณะเป็นเส้นโค้งบีสไปลน์ชนิดยูนิฟอร์มแบบเปิดไม่แบ่งส่วน (Open Uniform Non-Rational B-Spline Curve) หลังจากนั้นจะยอมให้ผู้ใช้กำหนดน็อต เวกเตอร์เองได้ เพื่อเปลี่ยนลักษณะเป็นแบบยูนิฟอร์มหรือนอนยูนิฟอร์ม (Non-Uniform) โดยโปรแกรมจะตรวจสอบข้อจำกัดบางประการให้เท่านั้น แต่เส้นโค้งบีสไปลน์ที่ใช้จะต้องเป็นชนิดไม่แบ่งส่วนเสมอ เพื่อรักษาความเป็นมาตรฐานเดียวกันของการกำหนดรูปทรงไว้ เนื่องจากเส้นโค้งบีสไปลน์แบบแบ่งส่วนจะต้องกำหนดเป็นระบบพิกัดแบบโฮโมจีเนียส (Homogeneous) ที่เปลี่ยนแปลงค่าน้ำหนัก (Weight) ได้ ในขณะที่การกำหนดค่าพิกัดที่ใช้ในโปรแกรมทั้งหมดเป็นระบบพิกัด 3 มิติธรรมดา ส่วนค่าพิกัด 2 มิติ คือ ค่าพิกัด 3 มิติที่มีค่าแกน z เป็น 0.0

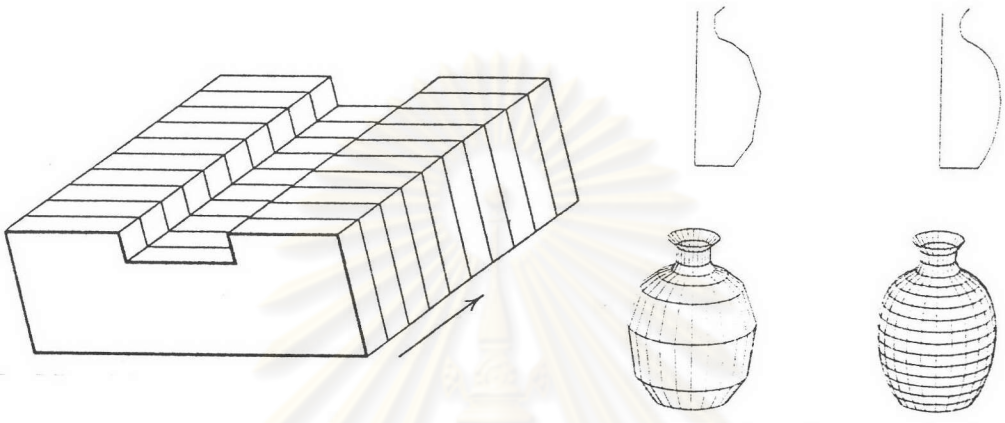
การสร้างรูปทรง 3 มิติจากรูปทรง 2 มิติ จะเป็นการนำรูปทรง 2 มิติมาผ่านการแปลง เพื่อใช้กำหนดรูปทรงของเซกเมนต์แต่ละเซกเมนต์ของรูปทรง 2 มิติ ดังรูปที่ 3.4



รูป 3.4 การสร้างรูปทรง 3 มิติจากรูปทรง 2 มิติ

เมื่อได้เซกเมนต์แต่ละส่วนแล้ว จะทำการเชื่อมเซกเมนต์แต่ละส่วนเข้าด้วยกัน ก็จะได้รูปทรง 3 มิติ 1 รูป

การกำหนดลักษณะของการแปลงให้เลือกทำได้วิธีใดวิธีหนึ่งใน 2 วิธี คือ การกวาดแบบย้าย ซึ่งจะกำหนดการแปลงแบบย้าย และการกวาดแบบหมุน จะเป็นการกำหนดการแปลงแบบหมุน



ก
แบบย้าย (Anand, 1993)

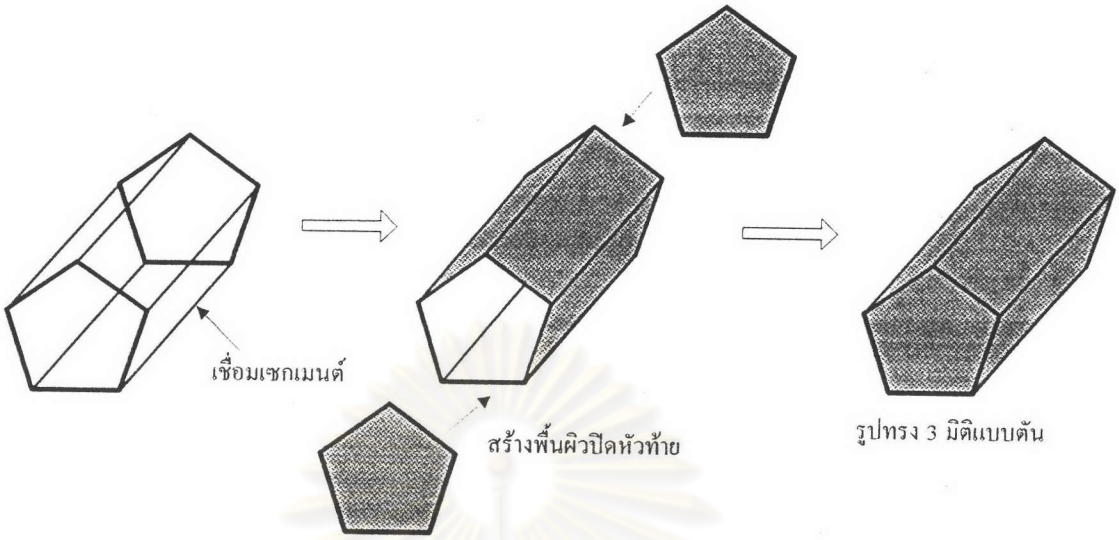
ข
แบบหมุน (Olfe, 1995)

รูป 3.5 การสร้างรูปทรง 3 มิติด้วยวิธีการกวาด

เมื่อกำหนดวิธีการแปลงและจำนวนเซกเมนต์ที่ต้องการแล้ว โปรแกรมจะนำข้อมูลเหล่านั้นมาสร้างเป็นแถวลำดับของการแปลง ซึ่งยอมให้ผู้ใช้แก้ไขเปลี่ยนแปลงได้ หลังจากนั้นโปรแกรมจะทำการเชื่อมเซกเมนต์แต่ละส่วนเข้าด้วยกัน

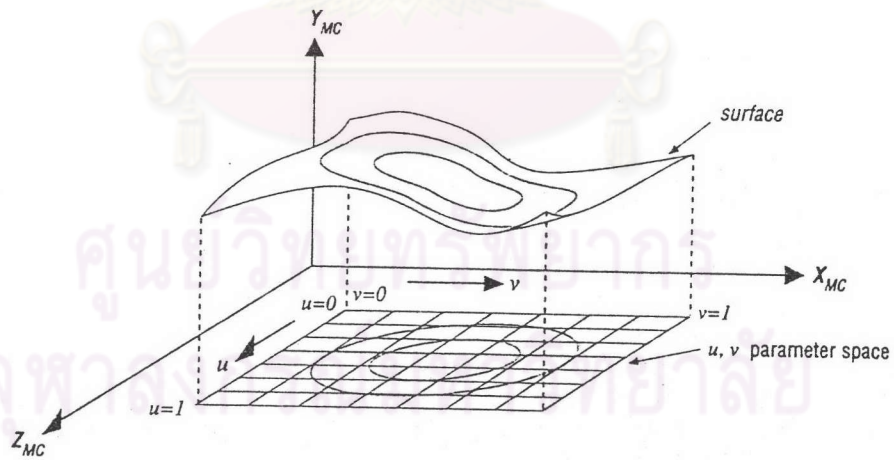
รูปทรงพื้นฐานที่ใช้ในการสร้างรูปทรง 3 มิติ มี 2 ชนิด คือ ชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูล และพื้นผิวแบบบีสไปลน์ ถ้ารูปทรง 2 มิติที่นำมาสร้างรูปทรง 3 มิติ เป็นรูปทรงพื้นฐานชนิดโพลีไลน์ รูปทรง 3 มิติที่จะสร้างได้ต้องเป็นชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูล แต่ถ้ารูปทรง 2 มิติเป็นเส้นโค้งบีสไปลน์ รูปทรง 3 มิติที่จะสร้างได้จะเป็นพื้นผิวแบบบีสไปลน์

การใช้รูปทรงพื้นฐานที่ต่างกันนี้ จะทำให้คุณสมบัติบางประการของวัตถุต่างกันด้วย คือ ชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูลจะสามารถสร้างรูปทรงลักษณะตันได้ โดยการเชื่อมเซกเมนต์แต่ละส่วนเข้าด้วยกัน แล้วทำการปิดหัวและท้ายของรูปทรงโดยการสร้างพื้นผิวที่เป็นรูปทรงเดียวกับเซกเมนต์หัวและท้าย



รูป 3.6 การสร้างรูปทรง 3 มิติแบบตันด้วยชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูล

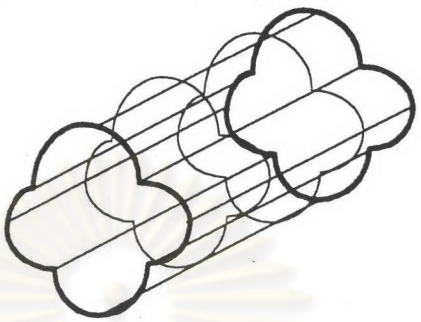
ส่วนรูปทรงแบบบีสไปลน์นั้น ในโปรแกรมนี้จะสามารถสร้างพื้นผิวบีสไปลน์ได้เฉพาะส่วนของการเชื่อมเซกเมนต์ แต่จะไม่สามารถสร้างรูปทรงตันด้วยวิธีดังกล่าวได้ เนื่องจากพื้นผิวแบบบีสไปลน์จะถูกกำหนดด้วยกริด ดังรูปที่ 3.7



รูป 3.7 ลักษณะของพื้นผิวแบบบีสไปลน์ (Gaskins, 1991)

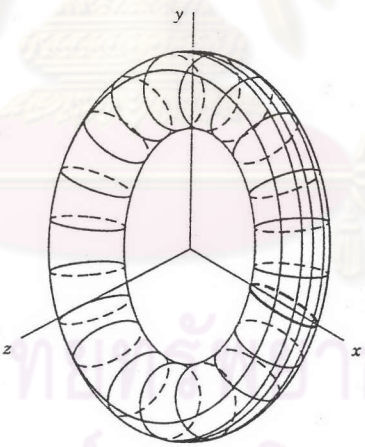
การสร้างรูปทรง 3 มิติแบบตันด้วยวิธีการสร้างพื้นผิวให้กับเซกเมนต์แรกและเซกเมนต์สุดท้าย คือ การกำหนดลักษณะของพื้นผิวให้มีรูปทรงเดียวกับรูปทรงของเซกเมนต์ ในกรณีของบีสไปลน์

โปรแกรมจะไม่สามารถกำหนดกริดควบคุมของพื้นผิวบีสไปลน์ให้มีรูปทรงถูกต้องตรงกับรูปทรงบีสไปลน์ของเซกเมนต์นั้นได้



รูป 3.8 เซกเมนต์ที่กำหนดด้วยบีสไปลน์

การสร้างรูปทรง 3 มิติ แบบตันด้วยพื้นผิวแบบบีสไปลน์ จะทำได้เพียงวิธีเดียวเท่านั้น คือ การใช้การกวาดแบบหมุน ที่หมุน 360 องศาเท่านั้น



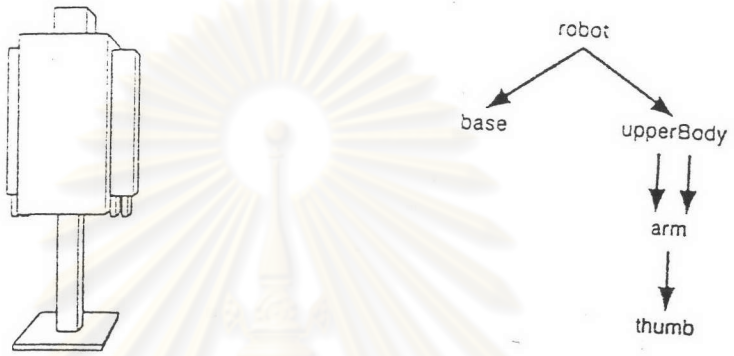
รูป 3.9 การสร้างรูปทรง 3 มิติแบบตันด้วยการกวาดแบบหมุน (Anand, 1993)

เมื่อได้รูปทรง 3 มิติแล้ว จะสร้างวัตถุได้โดยกำหนดลักษณะประจำให้วัตถุ ซึ่งประกอบด้วยลักษณะประจำของรูปทรงพื้นฐาน ลักษณะการสะท้อนแสงของวัตถุ การซ่อนพื้นผิว และการเชดสี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1.2 การสร้างแบบจำลอง

การสร้างแบบจำลองทำได้โดยการนำเอาวัตถุต่างๆ มาประกอบกันเป็นแบบจำลอง และแบบจำลองหนึ่งสามารถมีแบบจำลองอื่นๆ เป็นส่วนประกอบได้ โครงสร้างภายในของแบบจำลองจะมีลักษณะแบบเชิงลำดับชั้น ดังรูปที่ 3.10



รูป 3.10 ลักษณะของแบบจำลองเชิงลำดับชั้น (Foley, 1996)

เมื่อมีการแปลงแบบจำลอง ส่วนประกอบภายในของแบบจำลองจะมีการเคลื่อนไหวที่สัมพันธ์กันหมด แบบจำลองในโปรแกรมนี้จะไม่สามารถทำการเคลื่อนไหวเฉพาะส่วนได้ เช่น การเคลื่อนไหวในลักษณะที่มีจุดหมุน (Pivot Point) แบบข้อต่อ แขน ขา เป็นต้น การแก้ไขตำแหน่งของส่วนประกอบภายในสามารถทำได้เฉพาะขณะที่กำลังสร้างหรือแก้ไขแบบจำลองนั้น แต่เมื่อแบบจำลองนั้นไปเป็นส่วนประกอบของแบบจำลองอื่น การแก้ไขตำแหน่งเฉพาะส่วนจะทำไม่ได้

1.3 การสร้างฉาก

การสร้างฉากทำได้โดยนำเอาแบบจำลองต่างๆ มาวางรวมกันในฉาก แล้วกำหนดการให้แสง การกำหนดการให้แสงจะมี 2 ส่วน คือ การกำหนดตำแหน่ง และการกำหนดลักษณะของแสง เนื่องจากข้อจำกัดบางประการของซันฟิกส์รุ่น 2.0 ภาพของฉากที่ได้จากการเรนเดอร์จึงเป็นเพียงการทำเชดสีให้กับแบบจำลองแต่ละตัวภายในฉากให้สัมพันธ์กับแหล่งกำเนิดแสงในฉากเท่านั้น ซันฟิกส์รุ่น 2.0 ไม่สามารถคำนวณผลกระทบของแบบจำลองหนึ่งต่ออีกแบบจำลองหนึ่งได้ จึงไม่สามารถสร้างภาพเงาของแบบจำลองหนึ่งลงบนอีกแบบจำลองหนึ่งได้

1.4 การเก็บและการค้นคืน

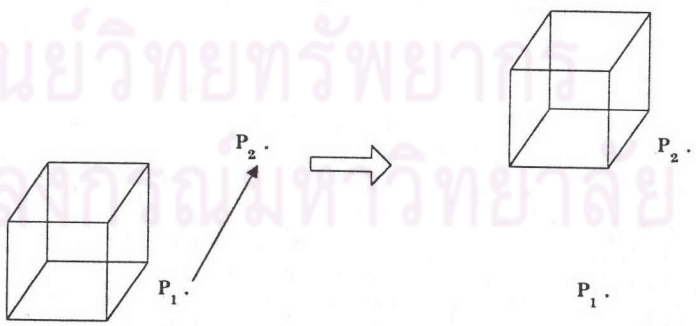
การเก็บและการค้นคืนจะเป็นการเก็บข้อมูลของการทำงานแต่ละส่วนลงในแฟ้ม ส่วนการค้นคืนจะเป็นการนำเอาข้อมูลในแฟ้มกลับมาใช้ในโปรแกรม แฟ้มที่ใช้ในการเก็บและค้นคืน คือ แฟ้มถาวรของฟิกส์

1.5 การเลือกและการแก้ไข

การเลือกวัตถุหรือแบบจำลองเพื่อทำการแก้ไข จะใช้อุปกรณ์นำเข้าของฟิกส์ชนิดพิกในการเลือกวัตถุหรือแบบจำลองซึ่งเก็บอยู่ในรูปของโครงสร้างมาทำการแก้ไข การเลือกสามารถเลือกได้ทีละ 1 วัตถุ หรือ 1 แบบจำลองเท่านั้น ในกรณีที่เป็นแบบจำลอง ไม่ว่าเลือกได้ส่วนประกอบใดของแบบจำลอง โปรแกรมจะทำการเลือกแบบจำลองนั้นทั้งหมดเพื่อทำการแก้ไข

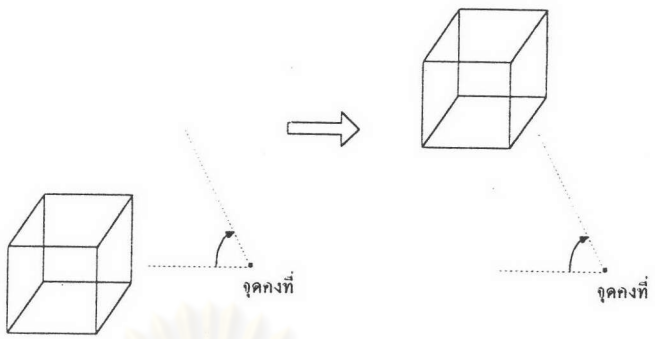
การแก้ไขวัตถุหรือแบบจำลองในโปรแกรมนี้ หมายถึง การเปลี่ยนแปลงตำแหน่งของจุดยอด วัตถุหรือแบบจำลอง การเปลี่ยนตำแหน่งสามารถทำได้ 3 วิธี คือ การย้าย การหมุน และการสเกล (Scale) ในกรณีของการเปลี่ยนตำแหน่งวัตถุหรือแบบจำลอง จะสามารถกำหนดให้มีการเปลี่ยนแปลงตำแหน่งให้สัมพันธ์กับจุดคงที่จุดหนึ่งในพิกัดโลกได้

ก) การเปลี่ยนตำแหน่งจุดยอด (การแปลงแบบย้าย)

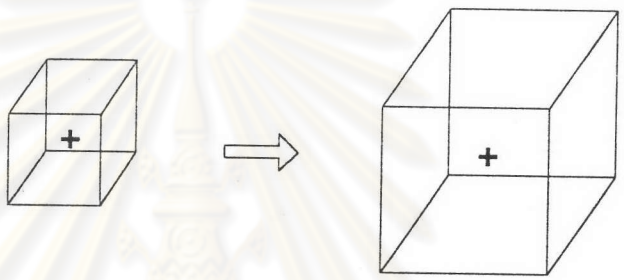


ข) การย้ายตำแหน่งวัตถุ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ก) การหมุนวัตถุรอบจุดคงที่



ง) การสเกลวัตถุจากจุดคงที่

รูป 3.11 การแก้ไขวัตถุหรือแบบจำลอง

2. ลักษณะของโปรแกรมและตัวประสานกับผู้ใช้

โปรแกรมการสร้างภาพแบบจำลอง 3 มิติ จะแบ่งการทำงานออกเป็น 2 ส่วน คือ โปรแกรมหลัก และโปรแกรมย่อย โดยโปรแกรมหลักจะมีหน้าที่ในการควบคุมดูแลการทำงานทั้งหมดของโปรแกรม ตั้งแต่ต้นจนจบการทำงานของโปรแกรม (เช่น การกำหนดค่าเริ่มต้น การเตรียมการจบโปรแกรม และการดูแลการติดต่อกับเอกกซ์วินโดว์ เป็นต้น) ส่วนโปรแกรมย่อยจะเป็นโปรแกรมที่มีหน้าที่เฉพาะอย่าง โปรแกรมย่อยจะมีทั้งหมด 6 โปรแกรม คือ

2.1 โปรแกรมสำหรับการจัดการข้อมูลและแฟ้ม (Data List & File Tool) มีหน้าที่ในการจัดการกับแฟ้มถาวรสำหรับการเรียกเก็บและการค้นคืน รวมไปถึงดูแลการจัดเก็บข้อมูลของโปรแกรมย่อยอื่นๆ ด้วย

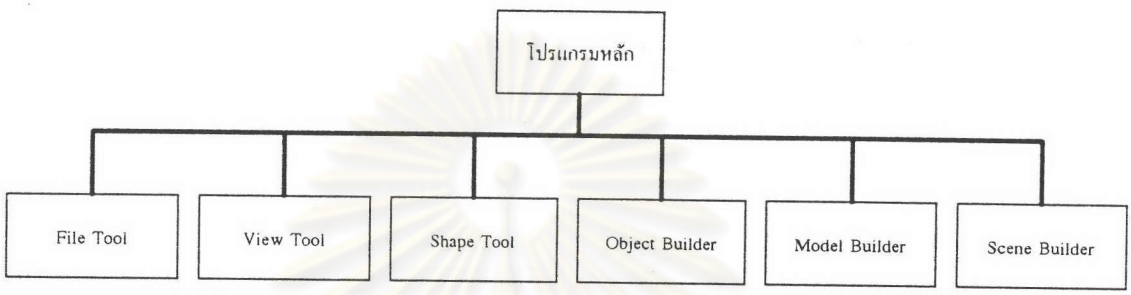
2.2 โปรแกรมสำหรับการจัดการวิว (View Tool) มีหน้าที่ในการเปลี่ยนแปลงวิวของเวิร์กสเตชันแต่ละตัว

2.3 โปรแกรมสำหรับการสร้างรูปทรง 2 มิติ (Shape Tool) ใช้สำหรับการวาดและแก้ไขรูปทรง 2 มิติ

2.4 โปรแกรมสำหรับการสร้างวัตถุ (Object Builder) ใช้ในการสร้างวัตถุ

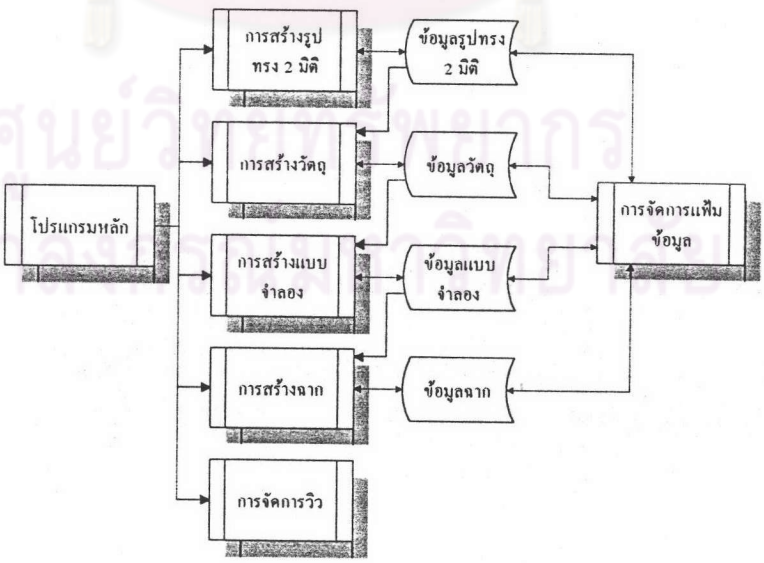
2.5 โปรแกรมสำหรับการสร้างแบบจำลอง (Model Builder) ใช้ในการสร้างแบบจำลอง

2.6 โปรแกรมสำหรับการสร้างฉาก (Scene Builder) ใช้ในการสร้างฉาก



รูป 3.12 ความสัมพันธ์ระหว่างโปรแกรมหลักและโปรแกรมย่อย

การทำงานแต่ละส่วนจะมีที่เก็บข้อมูลเฉพาะส่วน ซึ่งที่เก็บข้อมูลนี้จะมีลักษณะเป็นแถวลำดับสำหรับเก็บข้อมูลต่างๆ เช่น วัตถุ ตัวระบุโครงสร้าง เป็นต้น การทำงานแต่ละส่วนจะไม่สามารถอ้างถึงกันโดยตรงได้ แต่จะอ้างถึงข้อมูลของแต่ละส่วนได้ โดยอ้างถึงที่เก็บข้อมูล การเก็บและค้นคืนจำเป็นต้องอาศัยที่เก็บข้อมูลเหล่านี้ด้วย ความสัมพันธ์ของที่เก็บข้อมูลกับการทำงานแต่ละส่วนสามารถแสดงได้ดังรูปที่ 3.13

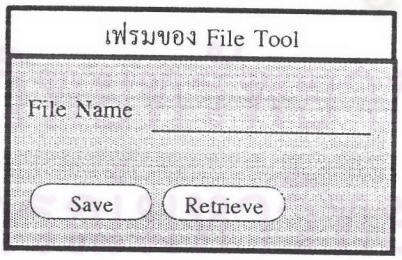


รูป 3.13 ความสัมพันธ์ของที่เก็บข้อมูลกับการทำงานแต่ละส่วน

ตัวประสานกับผู้ใช้ของโปรแกรมหลักจะเป็นเฟรม 1 เฟรม ที่ประกอบด้วยพาเนล 1 พาเนล เรียกว่า เมนู พาเนล พาเนลนี้จะประกอบด้วยไอเท็มของพาเนล ชนิดปุ่มเมนู ที่ใช้ในการเรียกโปรแกรมมาทำงาน สำหรับเฟรมของโปรแกรมหลักนี้จะถูกเปิดและแสดงตลอดเวลา การปิดเฟรมนี้จะหมายถึงการจบโปรแกรม

รูป 3.14 ตัวประสานกับผู้ใช้ของโปรแกรมหลัก

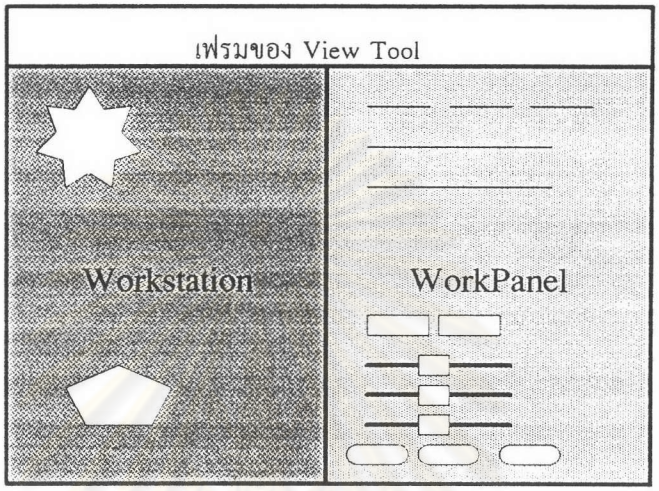
ตัวประสานกับผู้ใช้ของโปรแกรมการจัดการข้อมูลและแฟ้ม จะเป็นเฟรม 1 เฟรม มีพาเนล 1 พาเนล เรียกว่า เวิร์ก พาเนล ซึ่งใช้สำหรับกำหนดการจัดการแฟ้ม ไอเท็มบนพาเนลนี้ประกอบด้วย ไอเท็มชนิดข้อความ สำหรับการกำหนดชื่อแฟ้มหรือข้อมูลที่ต้องการจัดการ และไอเท็มชนิดปุ่ม (Button) สำหรับกำหนดการทำงานของโปรแกรมการจัดการข้อมูลและแฟ้ม



รูป 3.15 ตัวประสานกับผู้ใช้ของโปรแกรมการจัดการข้อมูลและแฟ้ม

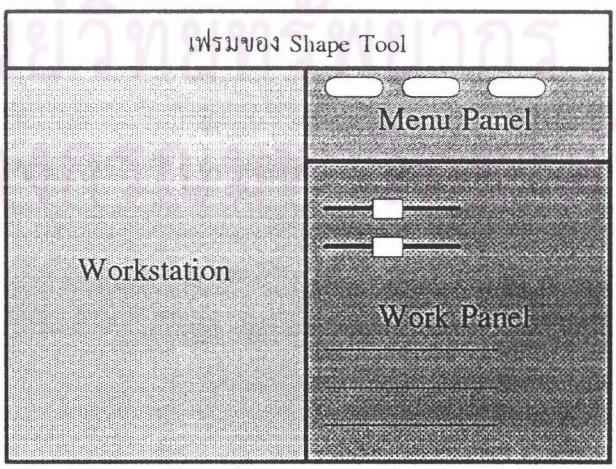
ตัวประสานกับผู้ใช้ของโปรแกรมการจัดการวิว เป็นเฟรม 1 เฟรม ที่ประกอบด้วยแคนวาส 1 แคนวาส และพาเนล 2 พาเนล ซึ่งใช้เป็นเวิร์ก พาเนลทั้ง 2 พาเนล เวิร์ก พาเนลทั้ง 2 พาเนลนี้จะใช้ในการรับข้อมูลเพื่อใช้กำหนดข้อมูลวิว พาเนลหนึ่งเรียกว่า ฟิกส์ พาเนล (phigs_panel) ใช้สำหรับรับข้อมูลวิวแบบฟิกส์ ส่วนอีกพาเนลหนึ่งเรียกว่า คอมมอน พาเนล

(common_panel) ใช้สำหรับรับข้อมูลวิวในแบบทั่วไป เช่น การซูม (Zoom) พาเนลทั้ง 2 พาเนลนี้จะอยู่ในตำแหน่งเดียวกัน เมื่อพาเนลหนึ่งถูกแสดงอีกพาเนลหนึ่งก็จะถูกซ่อน สำหรับแคนวาสจะใช้เป็นเวิร์กสเตชันสำหรับแสดงภาพ



รูป 3.16 ตัวประสานกับผู้ใช้ของโปรแกรมการจัดการวิว

ตัวประสานกับผู้ใช้ของโปรแกรมการสร้างรูปทรง 2 มิติ จะเป็นเฟรม 1 เฟรม ที่ประกอบด้วย แคนวาส 1 แคนวาส ซึ่งใช้เป็นที่แสดงภาพรูปทรง 2 มิติ และพาเนล 2 พาเนล โดยพาเนลหนึ่งจะใช้เป็นเมนู พาเนล เพื่อกำหนดสถานะการทำงาน ส่วนอีกพาเนลหนึ่งจะใช้เป็นเวิร์กพาเนล สำหรับการกำหนดข้อมูลในการทำงานแต่ละอย่าง เช่น ใช้กำหนดอันดับให้กับเส้นโค้งบีสไปลน์



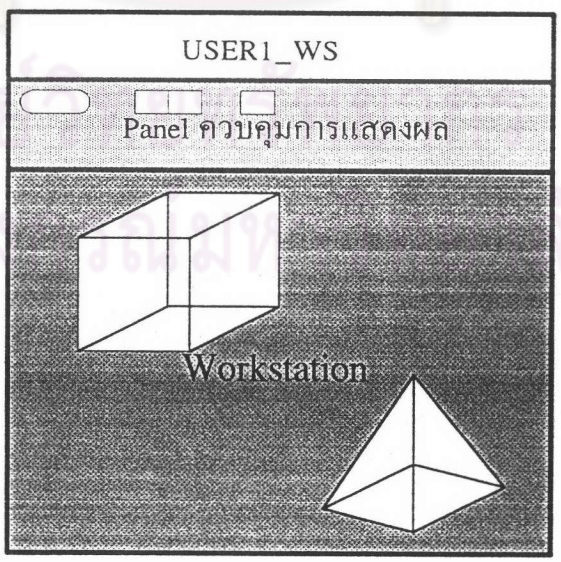
รูป 3.17 ตัวประสานกับผู้ใช้ของโปรแกรมการสร้างรูปทรง 2 มิติ

สำหรับโปรแกรมการสร้างวัตถุ แบบจำลอง และฉาก จะประกอบด้วยเฟรม 2 ชนิด คือ เฟรมที่ใช้สำหรับการแสดงภาพ และเฟรมควบคุมการทำงาน

เฟรมสำหรับการแสดงภาพประกอบด้วยแคนวาส 1 แคนวาส ที่ใช้เป็นเวิร์กสเตชันในการแสดงภาพ และพาเนล 1 พาเนล ที่ใช้สำหรับกำหนดลักษณะเฟรมและการแสดงผล เฟรมสำหรับการแสดงภาพนี้มี 9 เฟรม ซึ่งใช้แสดงภาพของวัตถุ แบบจำลอง หรือฉาก ในวิวต่างๆ กัน ดังนี้

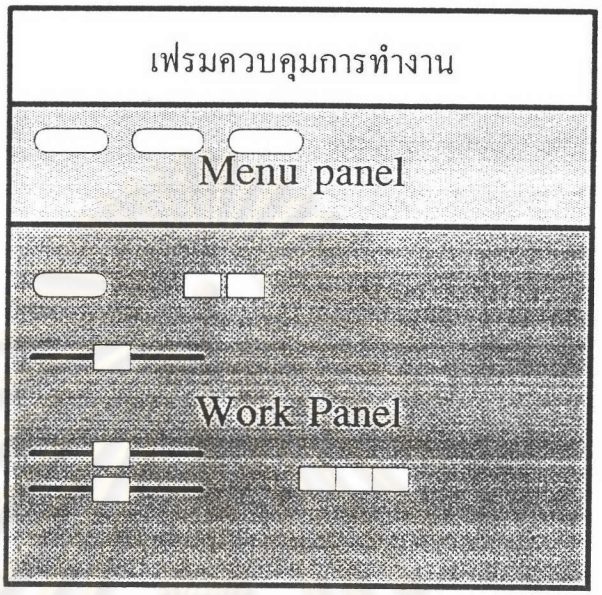
- เฟรม TOP_WS ใช้สำหรับแสดงภาพด้านบน
- เฟรม BOTTOM_WS ใช้สำหรับแสดงภาพด้านล่าง
- เฟรม FRONT_WS ใช้สำหรับแสดงภาพด้านหน้า
- เฟรม BACK_WS ใช้สำหรับแสดงภาพด้านขวา
- เฟรม RIGHT_WS ใช้สำหรับแสดงภาพด้านขวา
- เฟรม LEFT_WS ใช้สำหรับแสดงภาพด้านซ้าย
- เฟรม USER1_WS ใช้สำหรับแสดงภาพวิวอื่นๆ
- เฟรม USER2_WS ใช้สำหรับแสดงภาพวิวอื่นๆ
- เฟรม USER3_WS ใช้สำหรับแสดงภาพวิวอื่นๆ

เฟรม 6 เฟรมแรกจะกำหนดค่าวิว เพลน นอร์มัล และวิว อัป เวกเตอร์ ที่แน่นอนไว้ผู้ใช้ไม่สามารถเปลี่ยนแปลงได้โดยโปรแกรมการจัดการวิว แต่ข้อมูลอื่นของวิวสามารถเปลี่ยนแปลงได้ ส่วน 3 เฟรมสุดท้าย ผู้ใช้สามารถเปลี่ยนแปลงข้อมูลต่างๆ ของวิวได้อย่างอิสระ



รูป 3.18 เฟรมที่ใช้เป็นเวิร์กสเตชัน

ส่วนเฟรมควบคุมการทำงานแต่ละโปรแกรมจะมีโปรแกรมละ 1 เฟรม เฟรมชนิดนี้จะมี 2 พาเนล โดยมีพาเนลหนึ่งเป็นเมนู พาเนล และอีกพาเนลหนึ่งเป็นเวิร์ก พาเนล



รูป 3.19 เฟรมควบคุมการทำงาน

การทำงานของโปรแกรม

1. ข้อมูลและตัวแปรสำคัญที่ใช้ในโปรแกรม

ข้อมูลและตัวแปรสำคัญของโปรแกรมนี้ สามารถแบ่งได้เป็น 2 กลุ่ม คือ ข้อมูลของระบบ และข้อมูลของโปรแกรมย่อย ข้อมูลของระบบ คือ ข้อมูลที่จะถูกอ้างถึงเสมอในโปรแกรมย่อยทุกโปรแกรม ส่วนข้อมูลของโปรแกรมย่อยจะเป็นข้อมูลที่ถูกใช้เฉพาะในโปรแกรมย่อยบางโปรแกรม

ในหัวข้อนี้จะกล่าวถึงข้อมูลทั้ง 2 กลุ่ม เฉพาะส่วนที่สำคัญเท่านั้น สำหรับชื่อของชนิดข้อมูลของภาษาซีที่เป็นของซันฟิกส์รุ่น 2.0 จะขึ้นต้นด้วยตัว "P" เสมอ ส่วนชื่อของชนิดข้อมูลที่ถูกสร้างขึ้นใหม่เพื่อใช้กับโปรแกรมนี้จะขึ้นต้นด้วยตัว "T" เสมอ

1.1 ข้อมูลของระบบ

ข้อมูลของระบบที่สำคัญมี 3 กลุ่ม คือ กลุ่มที่ใช้เก็บข้อมูลความสามารถและข้อจำกัดของโปรแกรม กลุ่มที่ใช้เก็บข้อมูลที่เกี่ยวกับเวิร์กสเตชัน และกลุ่มที่ใช้เก็บข้อมูลสถานะและการทำงานของโปรแกรม

1.1.1 ข้อมูลที่ใช้เก็บข้อมูลความสามารถและข้อจำกัดของโปรแกรม

ข้อมูลในกลุ่มนี้มักเป็นข้อมูลชนิดจำนวนเต็ม อิงดรรชนี (Floating Point) หรือ Plimit3 ของซันฟิกส์รุ่น 2.0 ซึ่งใช้เก็บข้อมูลขนาดหรือจำนวนของความสามารถหรือข้อจำกัดบางอย่างของโปรแกรม เช่น จำนวนเวิร์กสเตชันที่เปิดได้พร้อมกันสูงสุด จำนวนดัชนีของสีที่มีได้มากที่สุดในตารางสี เป็นต้น

ข้อมูลเหล่านี้สามารถหาได้จากฟังก์ชันสอบถามของซันฟิกส์รุ่น 2.0 เช่น pinq_phigs_facs() เป็นต้น จะมีข้อมูลบางตัวที่ผู้วิจัยกำหนดขึ้นมาเอง เพื่อให้สมรรถนะของโปรแกรมดีขึ้น เช่น จำนวนของโครงสร้างของฟิกส์ที่แสดงได้มากที่สุดใน 1 เวิร์กสเตชัน (เนื่องจากเครื่องคอมพิวเตอร์ที่ใช้มีความเร็วในการแสดงผลกราฟิกค่อนข้างต่ำ ดังนั้นเมื่อจำนวนโครงสร้างเพิ่มขึ้นมากเท่าใด การปรับภาพจะยิ่งช้าลงเท่านั้น) เป็นต้น

ตัวแปรของข้อมูลเหล่านี้มักขึ้นต้นด้วยคำว่า Max_ เช่น

```
Pint   Max_Ws;          /* จำนวนเวิร์กสเตชันสูงสุดที่สามารถแสดงได้พร้อมกัน */
Pint   Max_Colr_rep;    /* จำนวนสีที่มีได้มากที่สุดในตารางสี */
Plimit3  Max_Space      /* ขอบเขตของพิกัดโลก */
```

1.1.2 ข้อมูลที่เกี่ยวข้องกับเวิร์กสเตชัน

ข้อมูลที่เกี่ยวข้องกับเวิร์กสเตชันมี 3 ชนิด คือ ข้อมูลที่ใช้เก็บค่าควบคุมต่างๆ ที่เกี่ยวข้องกับเอกซ์วินโดว์ ข้อมูลวิว และข้อมูลที่เกี่ยวกับการแสดงผลบนเวิร์กสเตชัน

ข้อมูลที่เกี่ยวกับค่าควบคุมของเอกซ์วินโดว์ประกอบด้วย ค่าควบคุมของเฟรม พาเนล และแคนวาส รวมทั้งค่าตัวระบุการเชื่อมต่อ (Connection Identifier) ของฟิกส์ด้วย ตัวแปรของข้อมูลเหล่านี้จะมีลักษณะเป็นแถวลำดับของค่าควบคุม ที่มีขนาดเท่ากับ Max_Ws และชื่อของตัวแปรเหล่านี้จะขึ้นต้นด้วยตัว "D" เสมอ

```
Frame   Dframe[Max_Ws];          /* ค่าควบคุมของเฟรม */
Panel   Dpanel[Max_Ws];          /* ค่าควบคุมของพาเนล */
Canvas  Dcanvas[Max_Ws];         /* ค่าควบคุมของแคนวาส */
Pconn_id_drawable  Dconn[Max_Ws];  /* ค่าของตัวระบุการเชื่อมต่อ */
```

โดยค่าตำแหน่งในแถวลำดับของข้อมูลเหล่านี้จะเป็นค่าเดียวกับตัวระบุเวิร์กสเตชัน เช่น Dframe [4] จะหมายถึง ค่าควบคุมของเฟรมของเวิร์กสเตชันที่มีตัวระบุเป็น 4 เป็นต้น

ข้อมูลวิวที่ใช้ฟังก์ชันสอบถามของซันฟิกส์รุ่น 2.0 เรียกมาใช้ จะมีลักษณะเป็นเมทริกซ์ (Matrix) ขนาด 4X4 ของวิว โอเรียนเตชัน และวิว แมปปิง ซึ่งข้อมูลเหล่านี้ได้ถูกผสมกันแล้วจึงไม่สามารถนำกลับมาใช้ใหม่ได้ ดังนั้นจึงต้องสร้างข้อมูลใหม่เพื่อเก็บข้อมูลวิว เช่น

```
    Tview_info  ws_view[Max_Ws];
typedef  struct  {
    Ppoint3  vrp;
    Pvec3  vpn;
    Pvec3  vuv;
} Tview_orientation;
typedef  struct  {
    Tview_orientation  ori;      /* ข้อมูลของวิวโอเรียนเตชัน */
    Pview_map3  map;             /* ข้อมูลของวิว แมปปิง */
} Tview_info;
```

ตัวแปร ws_view เป็นแถวลำดับที่ใช้เก็บข้อมูลวิวของเวิร์กสเตชันแต่ละตัว ค่าตำแหน่งในแถวลำดับของ ws_view จะมีความสัมพันธ์กับค่าตัวระบุของเวิร์กสเตชัน เช่น ws_view[4] หมายถึง ข้อมูลวิวของเวิร์กสเตชันที่มีตัวระบุเป็น 4

ข้อมูลเกี่ยวกับการแสดงผลของเวิร์กสเตชัน จะเก็บข้อมูลหลายอย่างที่ใช้บอกให้เวิร์กสเตชันรู้ว่าจะต้องแสดงภาพอะไร ในลักษณะใด เช่น รายชื่อของโครงสร้างที่ต้องแสดงภาพ วิธีการซ่อนเส้นและพื้นผิว เป็นต้น โดยความเป็นจริงแล้วข้อมูลเหล่านี้จะมีอยู่ในรายการสถานะของเวิร์กสเตชันและตารางอธิบายข้อมูลเวิร์กสเตชันซึ่งสามารถใช้ฟังก์ชันสอบถามของซันฟิกส์รุ่น 2.0 ได้ แต่จะทำให้โปรแกรมทำงานช้าลง จึงต้องเลือกข้อมูลที่โปรแกรมต้องอ้างถึงบ่อยๆ มาเก็บไว้ในตัวโปรแกรมเอง

1.1.3 ข้อมูลของสถานะและการทำงานของโปรแกรม

ข้อมูลในกลุ่มนี้จะใช้บอกถึงสถานะของโปรแกรม และสถานะการทำงานปัจจุบันของโปรแกรม เช่น current_prog ใช้บอกว่าขณะนี้กำลังทำงานในโปรแกรมย่อยใด เป็นต้น

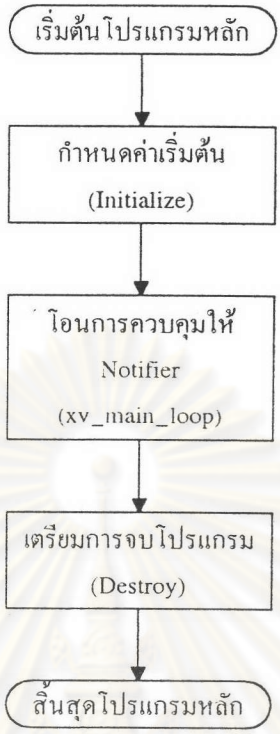
1.2 ข้อมูลของโปรแกรมย่อย

ข้อมูลของโปรแกรมย่อยจะมีความแตกต่างกันไปตามชนิดของโปรแกรม แต่จะมีข้อมูลกลุ่มหนึ่งที่เหมือนกัน คือ Prog_job ซึ่งใช้บอกถึงสถานะการทำงานปัจจุบัน เช่น การวาดเส้น การย้ายจุดยอด และการเลือกวัตถุ เป็นต้น และ Prog_job_stat จะใช้บอกสถานะปัจจุบันของการทำงานนั้น เช่น ถ้า Prog_job เป็นการเลือกวัตถุ ค่า Prog_job_stat จะมีได้ 2 ค่า คือ ยังไม่มีการเลือก หรือเลือกแล้ว

2. <u>การทำงานของโปรแกรม</u>

โปรแกรมจะเริ่มทำงานจากการสร้างเฟรมต่างๆ และกำหนดค่าเริ่มต้นของโปรแกรม หลังจากนั้นโปรแกรมจะทำงานโดยใช้โนติฟายเออร์และอีเวนต์ ซึ่งถูกควบคุมโดยเอกซ์วินโดว์ การทำงานโดยใช้โนติฟายเออร์จะเกิดขึ้นเมื่อมีอีเวนต์เกิดขึ้นในวัตถุของเอกซ์วิว เช่น เฟรม พาเนล ส่วนการทำงานในลักษณะอีเวนต์จะเกิดขึ้นบนเพ้นต์วินโดว์ของแคนวาส เมื่อมีโนติฟายเออร์ของเฟรมหลักบอกให้จบโปรแกรม โปรแกรมก็จะเตรียมการจบโปรแกรม เช่น การคืนเนื้อที่ในหน่วยความจำ ปิดเวิร์กสเตชัน ปิดวินโดว์ต่างๆ รวมทั้งปิดการใช้งานซันฟิกส์รุ่น 2.0 และเอกซ์วิว เป็นต้น ซึ่งสามารถแสดงได้ดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

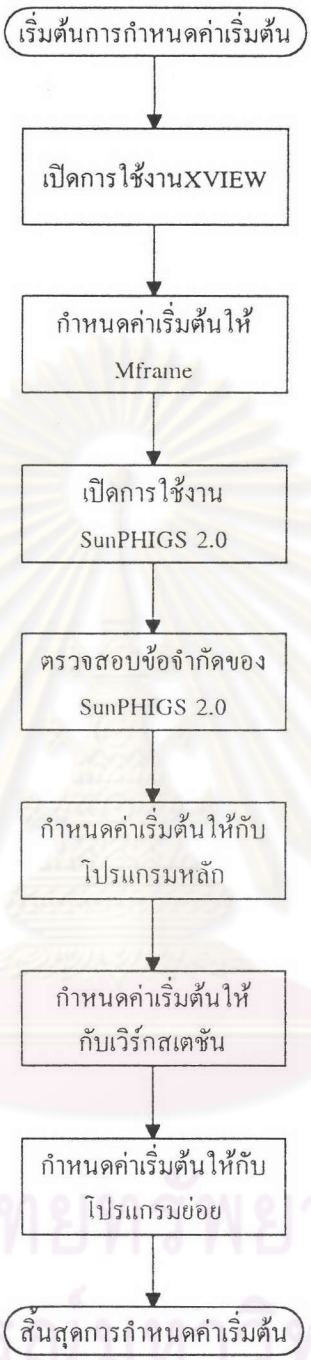


รูป 3.20 ผังงานหลักของโปรแกรม

2.1 กระบวนการกำหนดค่าเริ่มต้น (Initialize)

ในส่วนของการกำหนดค่าเริ่มต้นจะเริ่มโดยการเปิดใช้งานเอกซ์วิว การสร้างเฟรมหลัก คือ Mframe (ซึ่งเฟรมทุกเฟรมในโปรแกรมจะมีเจ้าของเป็น Mframe เสมอ) แล้วจึงเปิดการใช้งานฟิกส์ หลังจากนั้นจะเป็นการตรวจสอบข้อจำกัดของโปรแกรมบนเครื่องที่ใช้ แล้วจึงกำหนดค่าเริ่มต้นต่างๆ ของโปรแกรม เช่น ค่าวิว ต่อมาจึงทำการกำหนดค่าเริ่มต้นของเวิร์กสเตชัน จากนั้นจึงทำการกำหนดค่าเริ่มต้นของโปรแกรมย่อยต่างๆ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 3.21 ผังงานกระบวนการกำหนดค่าเริ่มต้น

2.2 กระบวนการเตรียมการจบโปรแกรม (Destroy)

กระบวนการนี้ใช้เพื่อเตรียมการจบโปรแกรม โดยการคืนที่ในหน่วยความจำให้ระบบ และปิดการทำงานต่างๆ ดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูป 3.22 ผังงานกระบวนการเตรียมการจบโปรแกรม

2.3 กระบวนการของ XV_main_loop

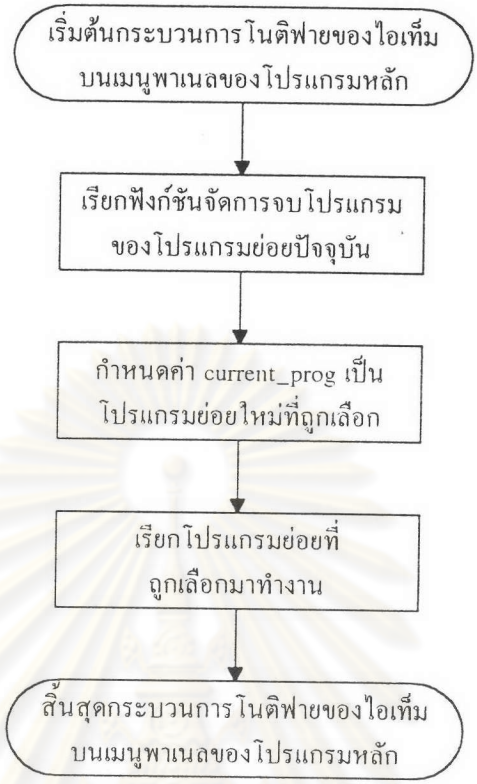
การทำงานในส่วนนี้ จะถูกควบคุมโดยกระบวนการโนติฟายเออร์ของเอกซ์วิว โนติฟายเออร์มีการทำงานใน 2 ลักษณะ คือ เป็นกระบวนการอีเวนต์เมื่อมีอีเวนต์เกิดขึ้นบนเพ้นต์วินโดว์ของแคนวาส และเป็นกระบวนการโนติฟายเมื่อมีอีเวนต์เกิดกับวัตถุอื่นๆ ของเอกซ์วิว เช่น เฟรม พาเนล และปุ่มบนพาเนล เป็นต้น เมื่อเกิดกระบวนการอีเวนต์ขึ้นโนติฟายเออร์จะเรียกฟังก์ชันควบคุมอีเวนต์ของเพ้นต์วินโดว์นั้นขึ้นมาใช้ โดยส่งค่าพารามิเตอร์เป็นค่าควบคุมวินโดว์และอีเวนต์ให้ ส่วนกระบวนการโนติฟายจะเป็นการทำงานในลักษณะแจ้งให้รู้ว่าไอเท็ม (Item) ตัวนั้นถูกเรียกใช้ โนติฟายเออร์ก็จะเรียกใช้ฟังก์ชันโนติฟายของวัตถุนั้น

ในโปรแกรมการสร้างภาพแบบจำลอง 3 มิตินี้ ฟังก์ชันควบคุมอีเวนต์ทั้งหมดของเวิร์กสเตชัน (ซึ่งใช้เพ้นต์วินโดว์ของแคนวาสในการแสดงภาพ) จะใช้ฟังก์ชัน

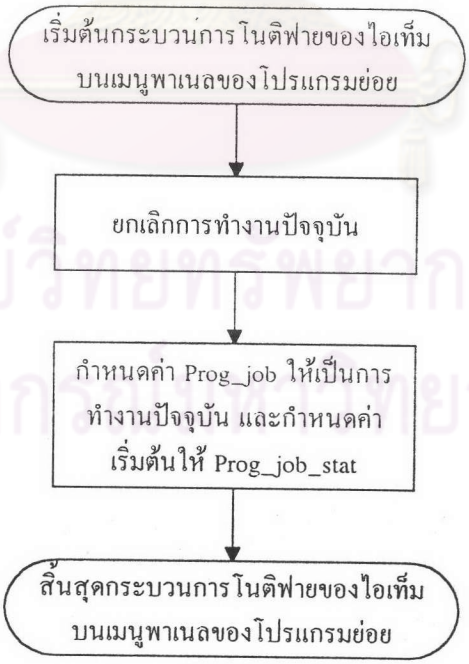
tevent_draw_ws() ซึ่งจะมีการทำงาน 2 ขั้นตอน คือ การจัดการกับอีเวนต์หลัก ซึ่งการตอบสนองต่ออีเวนต์นี้จะเหมือนกันในทุกๆ โปรแกรมย่อย เช่น การเปลี่ยนขนาดวินโดว์ (WIN_RESIZE) ในโปรแกรมนี้จะเรียกอีเวนต์เหล่านี้ว่าเป็นอีเวนต์ครอบคลุม (Global Event) ส่วนขั้นตอนที่ 2 จะเป็นการทำงานกับอีเวนต์บางอย่างที่จะมีการตอบสนองต่ออีเวนต์แตกต่างกันไปในโปรแกรมย่อยแต่ละโปรแกรม เช่น ในโปรแกรมเดียวกันนั้นการกดปุ่มซ้ายของเมาส์ในบางกรณีอาจหมายถึงการสั่งให้วาดภาพ แต่บางกรณีอาจหมายถึงการเลือกโครงสร้างที่กำลังแสดงอยู่ เป็นต้น การทำงานในส่วนนี้จะเป็นการตรวจสอบเงื่อนไขต่างๆ และเรียกใช้ฟังก์ชันการจัดการอีเวนต์ของโปรแกรมย่อย

กระบวนการโนติฟายของโปรแกรมการสร้างภาพแบบจำลอง 3 มิตินี้ จะมีการทำงานใน 2 ลักษณะ คือ กระบวนการโนติฟายที่มีการกำหนดค่าสถานะ การเรียกใช้ฟังก์ชันหรือโปรแกรมย่อยอื่นๆ และกระบวนการโนติฟายที่มีการกำหนดค่าข้อมูลและมีการทำงานเฉพาะอย่าง กระบวนการโนติฟายลักษณะแรกจะเกิดขึ้นกับไอเท็มต่างๆ บนเมนู พาเนลของทั้งโปรแกรมหลักและโปรแกรมย่อย ส่วนลักษณะที่ 2 จะเกิดขึ้นกับไอเท็มต่างๆ บนเวิร์ก พาเนลของโปรแกรมย่อย ปุ่มต่างๆ บนเมนู พาเนลของ Mframe (เฟรมของโปรแกรมหลัก) จะมีฟังก์ชันโนติฟายที่ใช้ในการเรียกโปรแกรมย่อยอื่นๆ ส่วนปุ่มรายการเลือกต่างๆ บนเมนู พาเนลของโปรแกรมย่อย จะเป็นการกำหนดค่าสถานะ เพื่อให้ฟังก์ชันควบคุมอีเวนต์ของโปรแกรมย่อยนำไปใช้ต่อไป เช่น ถ้ามีการกดปุ่ม Draw Polyline บน เมนู พาเนลของโปรแกรมการสร้างรูปทรง 2 มิติ ฟังก์ชันโนติฟายของปุ่มนี้จะไปกำหนดค่า Prog_job ของโปรแกรมการสร้างรูปทรง 2 มิติให้เป็น DRAW_POLYLINE เมื่อมีการกดปุ่มซ้ายของเมาส์และลากเมาส์ไปมาบนเวิร์กสเตชัน ฟังก์ชันควบคุมอีเวนต์ของโปรแกรมการสร้างรูปทรง 2 มิติ จะทำการวาดภาพโพลีไลน์ขึ้นมา ส่วนปุ่มบนเวิร์ก พาเนล จะใช้กำหนดข้อมูลและทำงานตามลักษณะของโปรแกรมย่อย เช่น การกดปุ่ม Paralell บนเวิร์ก พาเนลของโปรแกรมการจัดการวิว จะเป็นการกำหนดชนิดของการโปรเจกชันของวิวให้เป็นชนิดขนานและทำการปรับภาพของเวิร์กสเตชันนั้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

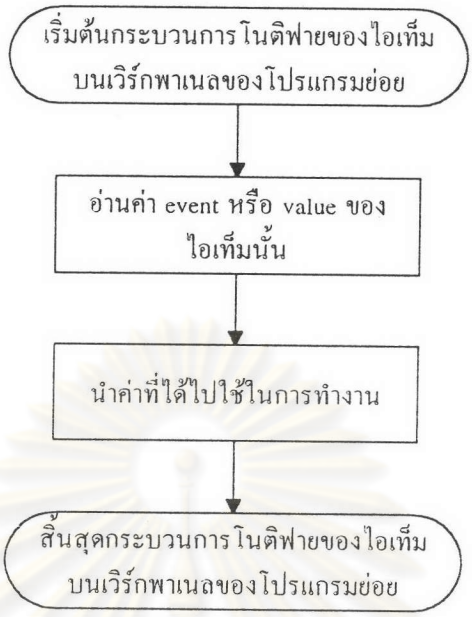


รูป 3.23 ผังงานของกระบวนการโนติฟายของไอเท็มบนเมนู พาเนลของโปรแกรมหลัก



รูป 3.24 ผังงานของกระบวนการโนติฟายของไอเท็มบนเมนู พาเนลของโปรแกรมย่อย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

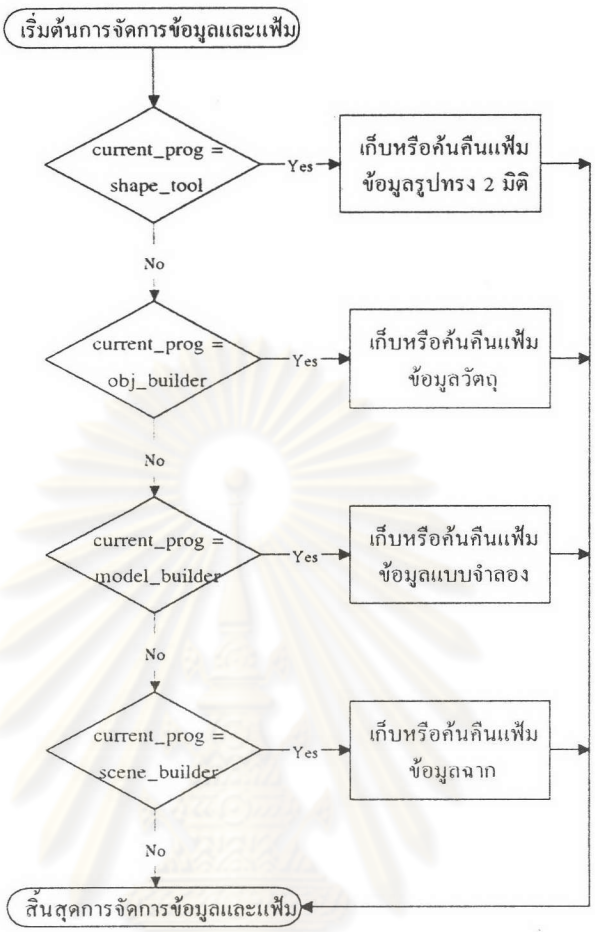


รูป 3.25 ผังงานของกระบวนการโนติฟายของไอเท็มบนเวิร์ก พาเนลของโปรแกรมย่อย

2.4 การทำงานของโปรแกรมการจัดการข้อมูลและแฟ้ม

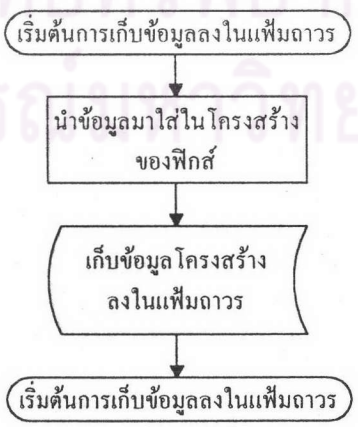
เมื่อมีการกดปุ่ม File บนเมนู พาเนล ของ Mframe โปรแกรมหลักจะเรียกโปรแกรมการจัดการข้อมูลและแฟ้มขึ้นมาทำงาน โปรแกรมการจัดการข้อมูลและแฟ้มจะเลือกการทำงานให้เหมาะสมกับโปรแกรมย่อยอื่นที่ทำงานอยู่ในขณะนั้น โดยดูจาก current_prog ซึ่งสามารถแสดงผังงานได้ดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 3.26 ผังงานการทำงานของโปรแกรมการจัดการข้อมูลและแฟ้ม

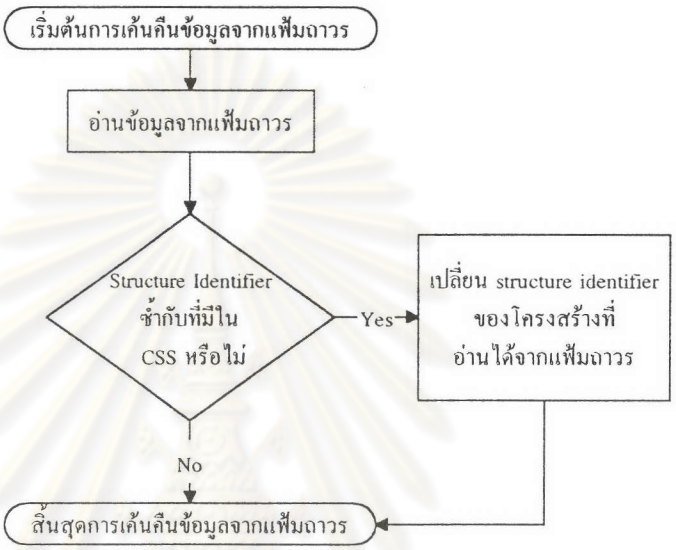
เมื่อมีการเก็บข้อมูลลงในแฟ้ม โปรแกรมการจัดการข้อมูลและแฟ้มจะทำการนำข้อมูลต่างๆ มาใส่ไว้ในโครงสร้างของฟิกส์ และใช้แฟ้มถาวรของฟิกส์ในการเก็บข้อมูลนั้น



รูป 3.27 ผังงานการเก็บข้อมูลลงในแฟ้ม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เมื่อมีการค้นคืนข้อมูลจากแฟ้มถาวร โปรแกรมการจัดการข้อมูลและแฟ้มจะทำการอ่านข้อมูลจากแฟ้มถาวรของฟิกส์มา แล้วตรวจว่าตัวระบุโครงสร้างของข้อมูลในแฟ้มซ้ำกับตัวระบุโครงสร้างในหน่วยเก็บโครงสร้างส่วนกลางหรือไม่ ถ้าซ้ำกันโปรแกรมการจัดการข้อมูลและแฟ้มจะทำการเปลี่ยนตัวระบุโครงสร้างของข้อมูลในแฟ้มถาวรเป็นค่าอื่น

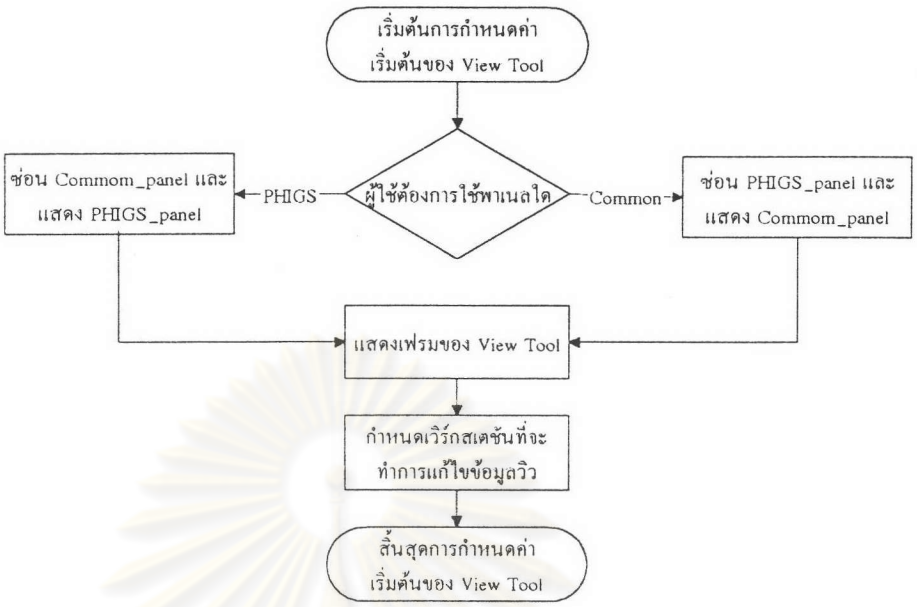


รูป 3.28 ผังงานการค้นคืนข้อมูลจากแฟ้ม

2.5 โปรแกรมการจัดการวิว

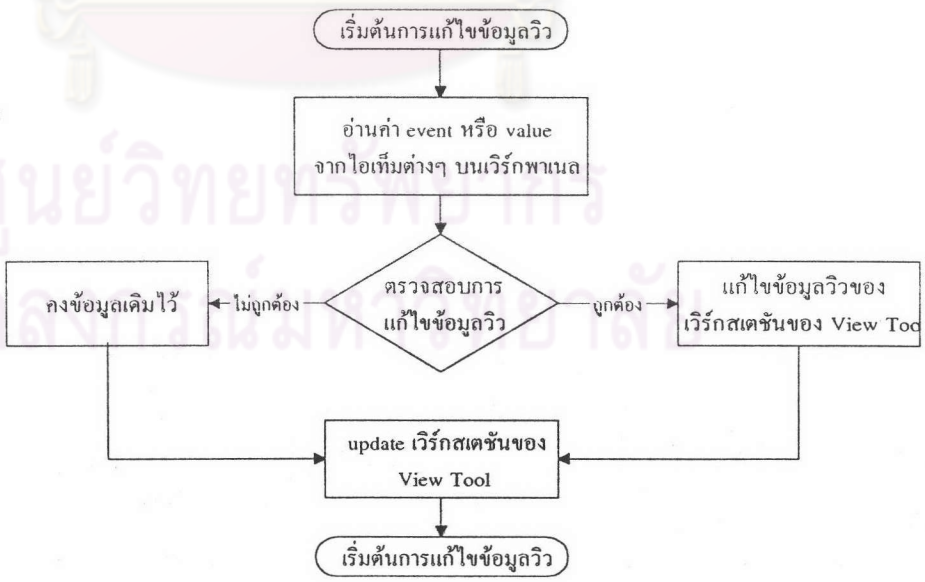
เมื่อมีการกดปุ่ม View บนเมนู พาเนลของ Mframe จะมีการแสดงรายการเลือกออกมา ในรายการเลือกนี้จะมี 2 ไอเท็ม คือ "Common" หรือ "PHIGS" ถ้ามีการเลือก Common จะเป็นการเรียกให้โปรแกรมการจัดการวิวทำงาน ซึ่งจะทำการซ่อน ฟิกส์ พาเนล (PHIGS Panel) และแสดงเฉพาะคอมมอน พาเนล (Common Panel) แต่ถ้าเลือก PHIGS ก็จะทำงานตรงกันข้ามกัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 3.29 ผังงานการกำหนดค่าเริ่มต้นของโปรแกรมการจัดการวิว เมื่อถูกโปรแกรมหลักเรียกใช้

เมื่อมีการเปลี่ยนแปลงข้อมูลบนเวิร์ก พาเนลของโปรแกรมการจัดการวิว โปรแกรมจะทำการตรวจสอบข้อมูลเหล่านั้นว่าใช้ได้หรือไม่ เช่น ตรวจว่าวิว เพลน นอร์มัลไม่เป็น (0, 0, 0) หรือไม่ขนานกับวิว อัป เวคเตอร์ เป็นต้น จากนั้นจึงทำการแก้ไขข้อมูลวิวของโปรแกรมการจัดการวิว แล้วปรับภาพ



รูป 3.30 ผังงานการทำงานของโปรแกรมการจัดการวิว เมื่อมีการแก้ไขข้อมูลบนเวิร์ก พาเนล

เมื่อมีการกดปุ่ม Ok บนเวิร์ก พาเนล โปรแกรมการจัดการวิวจะทำการเปลี่ยนแปลงข้อมูลวิวของเวิร์กสเตชันนั้น และทำการปรับภาพ แต่ถ้ากดปุ่ม Cancel จะทำการปิดเฟรมของโปรแกรมการจัดการวิว โดยไม่แก้ไขข้อมูลวิวในตัวระบุเวิร์กสเตชันที่กำหนด

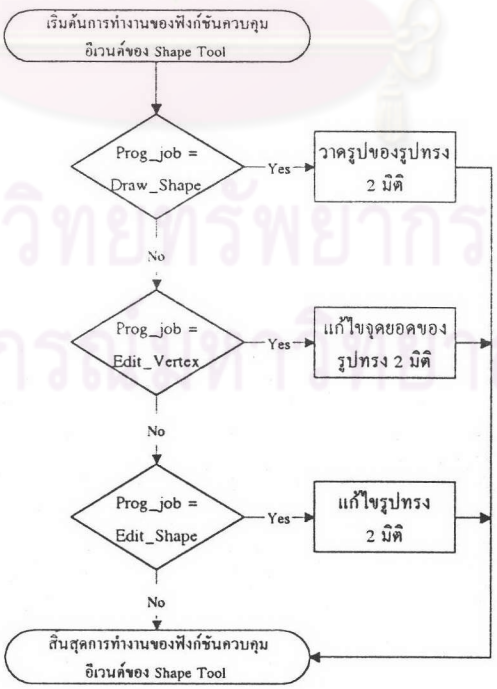
2.6 โปรแกรมการสร้างรูปทรง 2 มิติ

เมื่อมีการกดปุ่ม Shape บนเมนู พาเนล โปรแกรมหลักจะทำการเรียกโปรแกรมการสร้างรูปทรง 2 มิติให้ทำงาน

การทำงานโปรแกรมการสร้างรูปทรง 2 มิติ จะประกอบด้วยการสร้างรูปทรง 2 มิติขึ้นมาใหม่ หรือนำรูปทรง 2 มิติที่เตรียมไว้แล้วมาใช้ การแก้ไขจุดยอดของรูปทรง 2 มิติ เช่น การย้ายตำแหน่งของจุดยอด และการแก้ไขรูปทรง 2 มิติ เช่น การหมุนรูปทรง 2 มิติ การสเกล เป็นต้น

ไอเท็มของเมนู พาเนล จะใช้ในการกำหนดสถานะการทำงาน และแสดงเวิร์กพาเนลที่เกี่ยวข้องกับการทำงานนั้นๆ เช่น เมื่อกดปุ่ม Draw B-Spline จะเป็นการกำหนด Prog_job ให้เป็น DRAW_B_SPLINE และแสดงเวิร์ก พาเนลของการกำหนดข้อมูลบีสไปลน์ เช่น การกำหนดอันดับ การแก้ไขน็อต เวกเตอร์ เป็นต้น

การทำงานในฟังก์ชันควบคุมอีเวนต์ของโปรแกรมการสร้างรูปทรง 2 มิติ จะเลือกการทำงานโดยดูจาก Prog_job ดังนี้



รูป 3.31 ผังงานการทำงานของฟังก์ชันควบคุมอีเวนต์ของโปรแกรมการสร้างรูป 2 มิติ

2.7 โปรแกรมการสร้างวัตถุ แบบจำลอง และฉาก

การทำงานของโปรแกรมทั้ง 3 โปรแกรมนี้ จะมีลักษณะการทำงานเช่นเดียวกับโปรแกรมสร้างรูปทรง 2 มิติ คือ เมื่อกดปุ่ม Object, Model หรือ Scene บนเมนู พาเนล จะเป็นการเรียกโปรแกรมย่อยเหล่านี้มาทำงาน

การทำงานของโปรแกรมการสร้างวัตถุ ประกอบด้วยการสร้างวัตถุใหม่หรือการเรียกเอาวัตถุที่เตรียมไว้แล้วมาใช้ การแก้ไขรูปทรงวัตถุ และการกำหนดลักษณะประจำของวัตถุ (ส่วนการเก็บและค้นคืนจะถูกจัดการโดยโปรแกรมการจัดการข้อมูลและแฟ้ม)

การทำงานของโปรแกรมสร้างแบบจำลองประกอบด้วย การสร้างแบบจำลอง การแก้ไขตำแหน่งองค์ประกอบภายในแบบจำลอง

การทำงานของโปรแกรมสร้างฉากประกอบด้วย การเรียกใช้แบบจำลอง และแก้ไขตำแหน่งของแบบจำลอง การกำหนดแสงและแก้ไขแสง

เช่นเดียวกันกับโปรแกรมการสร้างรูปทรง 2 มิติ เมนู พาเนลของแต่ละโปรแกรม จะใช้ในการกำหนดสถานะการทำงาน เวิร์ก พาเนลใช้กำหนดข้อมูลที่เกี่ยวข้องกับการทำงานในขณะนั้น และฟังก์ชันควบคุมอีเวนต์ของแต่ละโปรแกรมย่อยจะทำงานตามสถานะของการทำงานที่ถูกกำหนดจากเมนู พาเนล

3. <u>การทำงานของโปรแกรม และการนำซันฟิกส์รุ่น 2.0 มาประยุกต์ใช้</u>

3.1 การกำหนดค่าเริ่มต้นของซันฟิกส์รุ่น 2.0

การเปิดการใช้งานซันฟิกส์รุ่น 2.0 สามารถทำได้ 2 วิธี คือ การใช้ฟังก์ชัน popen_phigs() สำหรับการเปิดใช้งานอย่างธรรมดาและมีการกำหนดสภาวะการทำงานร่วมกับเอกซ์ วินโดว์ เป็นค่าโดยปริยาย (Default) ส่วนอีกฟังก์ชัน คือ popen_xphigs() ซึ่งใช้เมื่อต้องการกำหนดสภาวะการทำงานบางประการเอง ในโปรแกรมนี้จะเปิดใช้งานซันฟิกส์รุ่น 2.0 ในแบบธรรมดา คือ ใช้ฟังก์ชัน popen_phigs()

ชนิดของเวิร์กสเตชันที่ใช้ในโปรแกรมจะใช้เวิร์กสเตชันฐานเป็นแบบเอกซ์ดรอเอเบิล และได้มีการกำหนดลักษณะบางประการของเวิร์กสเตชันฐาน ดังนี้

```
Ws_Type = phigs_ws_type_create (phigs_ws_type_x_drawable,
        PHIGS_X_BUF_MODE, PHIGS_BUF_DOUBLE,
        /* กำหนดให้ใช้ Double Buffering */
        PHIGS_HANDLE_EXPOSE, 1,
        /* ให้ PHIGS Redraw เวิร์กสเตชันเมื่อมีอีเวนต์ชนิด EXPOSE */
```

```
PHIGS_HANDLE_DESTROY, 1,
/* ให้ PHIGS ปิดเวิร์กสตเชันเมื่อมีอีเวนต์ชนิด DestroyNotify */
PHIGS_DC_MODEL, PHIGS_DC_LIMITS_ADJUST_TO_WINDOW,
/* ให้เวิร์กสเตชันสามารถปรับขนาดตามวินโดว์ได้ */
PHIGS_COLOUR_MODEL,TRUE,      /* กำหนดให้ใช้สีแบบ True Color */
NULL);
```

3.2 การสร้างเวิร์กสเตชันสำหรับการแสดงภาพ

การสร้างเวิร์กสเตชันทำได้โดยการสร้างเฟรมของเวิร์กสเตชันก่อน โดยมี Mframe เป็นเจ้าของ (Mframe เป็นเฟรมหลักของโปรแกรม) แล้วเก็บค่าควบคุมไว้ที่ Dframe ของเวิร์กสเตชันนั้น หลังจากนั้นจึงสร้างแคนวาสของเฟรม แล้วเก็บค่าควบคุมไว้ที่ Dcanvas ของเวิร์กสเตชัน จากนั้นจึงนำค่า XV_DISPLAY และค่าควบคุมของเพ้นต์ วินโดว์ของแคนวาสนั้นไปเก็บไว้ใน Dconn แล้วจึงสร้างเวิร์กสเตชันของฟิกส์ด้วยค่าเหล่านี้ ดังตัวอย่างดังนี้

```
/* wsid หมายถึง WORKSTATION IDENTIFIER */
  Dframe[wsid] = (Frame) xv_create(Mframe, FRAME, NULL);
  Dcanvas[wsid] = (Canvas) xv_create(Dframe[wsid], CANVAS, NULL);
  Dconn[wsid].display = (Display *) xv_get(Dcanvas[wsid], XV_DISPLAY);
  Dconn[wsid].drawagle_id = (Window) xv_get((Xv_Window) canvas_paint_window
                              (Dcanvas[wsid], SV_SID));
  popen_ws(wsid, (void *) Dconn[wsid], Ws_Type);
```

หลังจากนั้นโปรแกรมจะทำการกำหนดลักษณะของแคนวาสให้หด (Shrink) และขยาย (Expand) ตามขนาดของเฟรม กำหนดให้มีการทำแบกกิง สโตร์ (Backing Store) กำหนดตำแหน่งและขนาดเริ่มต้นของแคนวาส และกำหนดให้ใช้ฟังก์ชันรีดรอ (Redraw) ของโปรแกรมนี้ คือ tredraw_ws() ซึ่งแสดงได้ดังนี้

```
xv_set(Dcanvas[wsid],
      CANVAS_RETAINED, TRUE,
      CANVAS_AUTO_EXPAND, TRUE, CANVAS_AUTO_SHRINK, TRUE,
```

```
XV_X, (wsid *50), XV_Y, (wsid *50),
CANVAS_WIDTH, 500, CANVAS_HEIGHT, 500,
CANVAS_X_PAINT_WINDOW, TRUE, CANVAS_REPAINT_PROC, tredraw_ws,
NULL);
```

จากนั้นโปรแกรมจะทำการกำหนดให้เพ้นต์วินโดว์ของแคนวาสนั้น ใช้ฟังก์ชันควบคุมอีเวนต์ของโปรแกรม คือ tevent_draw_ws() และกำหนดให้จัดการกับอีเวนต์เฉพาะที่เกี่ยวข้องกับโปรแกรมดังนี้

```
xv_set (canvas_paint_window(Dcanvas[wsid]),
        WIN_EVENT_PROC, tevent_draw_ws,
        WIN_CONSUME_EVENTS,
            WIN_ASCII_EVENTS, WIN_MOUSE_BUTTONS,
            LOC_MOVE, LOC_WINEXIT, LOC_WINENTER, LOC_DRAG,
            KBD_DONE, KBD_USE,
            WIN_REPAINT, WIN_RESIZE,
            NULL,
        WIN_CONSUME_X_EVENT_MASK, StructureNotifyMask, NULL,
        NULL);
```

ต่อมาโปรแกรมจะทำการกำหนดค่าวิวเริ่มต้น และชนิดของการซ่อนเส้นและพื้นผิวให้กับเวิร์กสเตชันนั้นๆ ดังนี้

- เวิร์กสเตชันวิวด้านบน (Top View Workstation) จะมีวิว เพลน นอร์มัล (View Planen Normal ย่อว่า VPN) เป็น (0,1,0) และวิว อัป เวคเตอร์ (View Up Vector ย่อว่า VUV) เป็น (0,0,-1)
- เวิร์กสเตชันวิวด้านล่าง (Bottom View Workstation) จะมี VPN เป็น (0,-1,0) และ VUV (0,0,-1)
- เวิร์กสเตชันวิวด้านหน้า (Front View Workstation) จะมี VPN เป็น (0,0,1) และ VUV เป็น (0,1,0)

- เวิร์กสเตชันวิวด้านหลัง (Back View Workstation) จะมี VPN เป็น (0,0,-1) และ VUV เป็น (0,1,0)

- เวิร์กสเตชันวิวด้านขวา (Right View Workstation) จะมี VPN เป็น (1,0,0) และ VUV เป็น (0,1,0)

- เวิร์กสเตชันวิวด้านซ้าย (Left View Workstation) จะมี VPN เป็น (-1,0,0) และ VUV เป็น (0,1,0)

- เวิร์กสเตชันของผู้ใช้1 (User1 View Workstation) จะมี VPN เป็น (1,1,1) และ VUV เป็น (0,1,0)

- เวิร์กสเตชันของผู้ใช้2 (User2 View Workstation) จะมี VPN และ VUV เช่นเดียวกับเวิร์กสเตชันของผู้ใช้ 1 แต่มีชนิดของการโปรเจกชัน (Projection) เป็นชนิดเพอร์สเปกทีฟ

- เวิร์กสเตชันของผู้ใช้3 (User3 View Workstation) จะมี VPN เป็น (-1,-1,-1) และ VUV เป็น (0,1,0) และมีชนิดของการโปรเจกชันเป็นชนิดเพอร์สเปกทีฟ

- ส่วนเวิร์กสเตชันอื่นๆ จะมีค่าเหมือนกับวิวด้านหน้า

เวิร์กสเตชันทั้งหมดจะมีค่าอื่นๆ ของข้อมูลการกำหนดวิวดังนี้

- จุดอ้างอิงของวิว (View Reference Point) เป็น (0,0,0)
- ขอบเขตของวินโดว์ (Window Limit) มีค่า

  X Min. เท่ากับ -20.0 และ X Max. เท่ากับ 20.0

  Y Min. เท่ากับ -20.0 และ Y Max. เท่ากับ 20.0
- ขอบเขตของโปรเจกชัน วิวพอร์ต (Projection Viewport Limit) มีค่า

  X Min. เท่ากับ 0.0 และ X Max. เท่ากับ 1.0

  Y Min. เท่ากับ 0.0 และ Y Max. เท่ากับ 1.0

  Z Min. เท่ากับ 0.0 และ Z Max. เท่ากับ 1.0
- ชนิดของการโปรเจกชัน (Projection Type) มีค่าเป็น PTYPE_PARAL คือ กำหนดให้โปรเจกชันเป็นชนิดขนาน (Parallel)
- จุดอ้างอิงของการโปรเจกชัน (Projection Reference Point) มีค่าเป็น(0,0,40)

  ค่า x, y จะถูกกำหนดให้อยู่ตรงกลางวินโดว์ เพื่อไม่ให้เกิดการเอียงของการโปรเจคชัน (Oblique Projection) ขึ้น ส่วนค่า z คือ ค่าระนาบด้านหน้า (Front Plane)
- ค่าระนาบของวิว มีค่าเท่ากับ 0.0
- ค่าระนาบด้านหลัง มีค่าเท่ากับ -40.0

- ค่าระนาบด้านหน้า มีค่าเท่ากับ 40.0

ค่าระนาบด้านหน้าและด้านหลัง ถูกกำหนดให้มีค่าเท่ากับขนาดของขอบเขตของวินโดว์ ถ้ากำหนดให้มีค่าน้อยกว่านี้ ภาพที่ปรากฏจะถูกตัด (Clip) ก่อนที่จะพ้นขอบเขตของวินโดว์

3.3 การควบคุมอีเวนต์

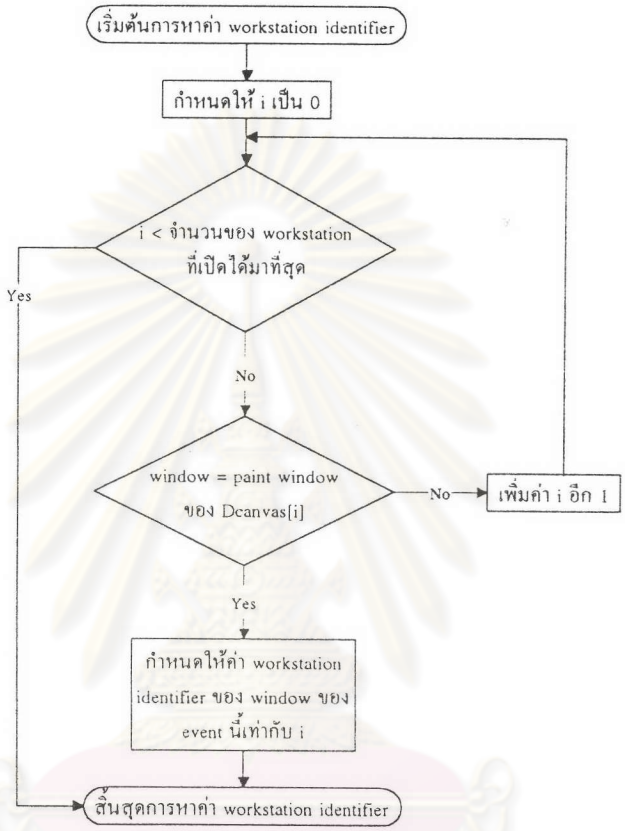
เมื่อมีอีเวนต์เกิดขึ้นบนเพ้นต์วินโดว์ของเวิร์กสเตชัน เอกซ์วินโดว์จะเรียกใช้ฟังก์ชันควบคุมอีเวนต์ของโปรแกรม คือ tevent_draw_ws() ดังที่กำหนดไว้ในตอนสร้างเวิร์กสเตชัน ฟังก์ชัน tevent_draw_ws() จะมีการทำงานดังนี้

รูป 3.32 ผังงานการทำงานของฟังก์ชัน tevent_draw_ws()

ค่าที่เอกซ์วินโดว์ส่งมาให้ฟังก์ชันควบคุมอีเวนต์นี้ คือ ค่าควบคุมของวินโดว์ที่เกิดอีเวนต์และค่าอีเวนต์ ซึ่งมีรูปแบบดังนี้

```
void  tevent_draw_ws(window, event)
   Xv_Window  window;
   Event  *event;
```

แต่ฟังก์ชันต่างๆ ของซันฟิกส์รุ่น 2.0จะอ้างถึงเวิร์กสเตชันด้วยตัวระบุเวิร์กสเตชัน ดังนั้นจึงต้องนำค่าควบคุมวินโดว์ที่เอกซ์วินโดว์ส่งมา ไปหาค่าตัวระบุเวิร์กสเตชันอีกที โดยการนำค่าวินโดว์ไปเทียบกับค่าของเพ้นต์วินโดว์ดังนี้



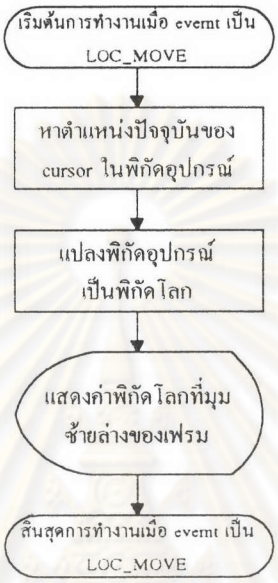
รูป 3.33 ผังงานกระบวนการของการหาค่าตัวระบุเวิร์กสเตชัน

เมื่อได้ค่าตัวระบุเวิร์กสเตชันแล้วขั้นตอนต่อมาในฟังก์ชันควบคุมอีเวนต์นี้ คือ การจัดการกับอีเวนต์ครอบคลุม (Global Event) ซึ่งจะเป็นการทำงานที่ตอบสนองต่ออีเวนต์ที่เข้ามาเหมือนกันทุกเวิร์กสเตชันและไม่เกี่ยวข้องกับโปรแกรมย่อยด้วย อีเวนต์เหล่านี้ได้แก่ การเลื่อนตัวชี้ตำแหน่ง การรีเพ้นต์ (Repaint) วินโดว์ และการเปลี่ยนขนาดของวินโดว์

3.3.1 การเลื่อนตัวชี้ตำแหน่ง (LOC_MOVE)

เมื่อมีการเปลี่ยนตำแหน่งของตัวชี้ตำแหน่งบนเวิร์กสเตชัน โปรแกรมจะทำการหาค่าตำแหน่งใหม่ของตัวชี้ตำแหน่ง และแสดงค่าตำแหน่งใหม่ของตัวชี้ตำแหน่งนั้น ที่ตำแหน่งล่างซ้ายของเฟรม

ค่าตำแหน่งของตัวชี้ตำแหน่งที่ได้มาจากเอกซ์วินโดว์ จะอยู่ในระบบพิกัดของอุปกรณ์ ซึ่งต้องแปลงให้เป็นค่าในพิกัดโลกด้วยฟังก์ชันเอสเคปหมายเลข 5 ของซันฟิกส์รุ่น 2.0



รูป 3.34 ผังงานกระบวนการจัดการอีเวนต์ครอบคลุม เมื่ออีเวนต์เป็น LOC_MOVE

3.3.2 การรีเพ้นต์วินโดว์ (WIN_REPAINT)

นอกจากการกำหนดให้แคนวาสใช้ฟังก์ชันรีเพ้นต์ของโปรแกรมแล้ว ยังสามารถควบคุมการรีเพ้นต์ของแคนวาสได้อีกทางหนึ่ง โดยการสั่งให้มีการรีเพ้นต์เมื่ออีเวนต์ที่เข้ามาเป็น WIN_REPAINT

การสั่งให้ซันฟิกส์รุ่น 2.0 ทำการรีเพ้นต์เวิร์กสเตชัน สามารถสั่งได้โดยใช้ฟังก์ชัน predraw_all_structs

3.3.3 การเปลี่ยนขนาดเฟรม (WIN_RESIZE)

เมื่อมีการเปลี่ยนขนาดเฟรม ซันฟิกส์รุ่น 2.0จะพยายามรักษาสภาพความสมมาตรของวิวพอร์ตเสมอ เมื่อขนาดของแคนวาสไม่สมมาตร คือ มีด้านกว้างยาวกว่าด้านสูงหรือด้านสูงยาวกว่าด้านกว้าง จะทำให้เกิดช่องว่างที่ไม่ใช้ขึ้น เพราะซันฟิกส์รุ่น 2.0จะปรับวิวพอร์ตให้มีขนาดกว้างและสูงเท่ากับด้านที่สั้นที่สุดของแคนวาส

การกำหนดวิวพอร์ตของซันฟิกส์รุ่น 2.0บนแคนวาสจะกำหนดได้ด้วยข้อมูล 2 ตัว คือ เวิร์กสเตชัน วิวพอร์ต (Workstation Viewport) และเวิร์กสเตชัน วินโดว์ (Workstation Window) การกำหนดเวิร์กสเตชัน วิวพอร์ต จะกำหนดโดยใช้พิกัดของอุปกรณ์ ซึ่ง

จะเป็นการกำหนดขนาดที่ใช้จริงบนแคนวาส ส่วนเวิร์กสเตชัน วินโดว์ จะเป็นการกำหนดโดยใช้การใช้นอร์มัลไลซ์ โปรเจกชัน โคออดิเนต ซึ่งจะเป็นการกำหนดขนาดของวินโดว์ที่ได้จากขั้นตอนวิว แมปปิงในทรานส์ฟอร์เมชัน ไพป์ไลน์

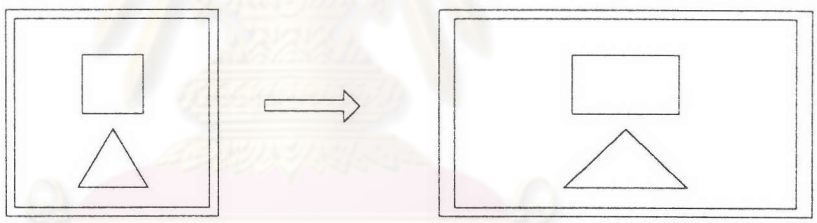
จากการทดลองพบว่ามี 3 วิธี ที่จะทำให้ซันฟิกส์รุ่น 2.0ใช้พื้นที่ทั้งหมดของแคนวาสสำหรับแสดงภาพได้ คือ

3.3.3.1 กำหนดขนาดเวิร์กสเตชัน วิวพอร์ต ให้เท่ากับขนาดของแคนวาส และกำหนดเวิร์กสเตชัน วินโดว์ดังนี้

window.X_MIN = window.Y_MIN = 0.0;

window.X_MAX = window.Y_MAX = 1.0;

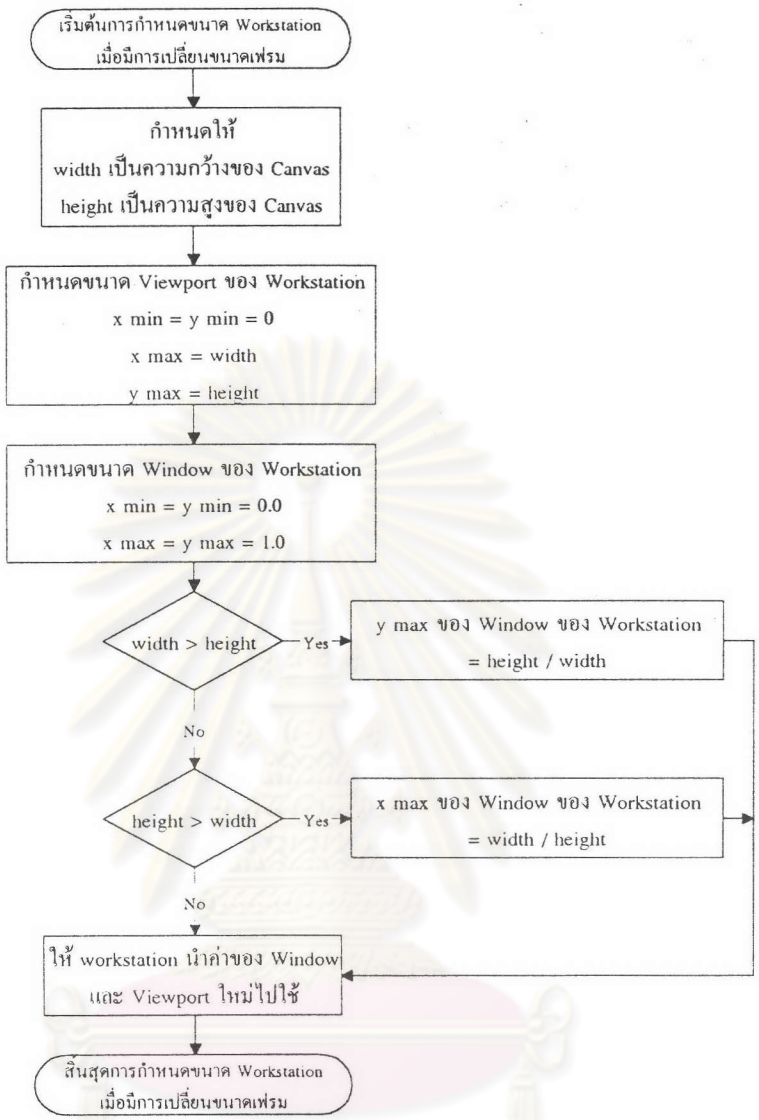
วิธีนี้จะทำให้ซันฟิกส์รุ่น 2.0สามารถใช้พื้นที่ทั้งหมดของแคนวาส ในการแสดงภาพได้ แต่ภาพที่ได้จะมีการเปลี่ยนแปลงขนาดไปและมีลักษณะไม่สมมาตร



รูป 3.35 ภาพวัตถุที่ได้จากการกำหนดเวิร์กสเตชัน วินโดว์เป็น window.X_MIN = window.Y_MIN = 0.0 และ window.X_MAX = window.Y_MAX = 1.0;

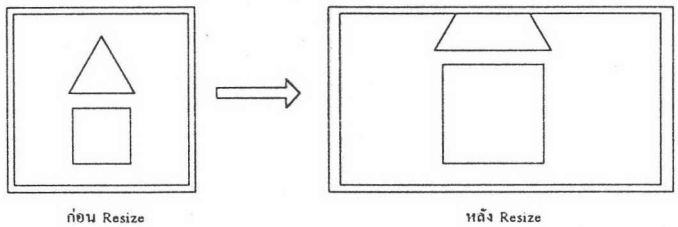
3.3.3.2 กำหนดขนาดเวิร์กสเตชัน วิวพอร์ต ให้เท่ากับขนาดของแคนวาส และกำหนดขนาดของเวิร์กสเตชัน วินโดว์ให้สัมพันธ์กับลักษณะของแคนวาส ดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 3.36 ผังงานกระบวนการจัดการอีเวนต์ครอบคลุม เมื่ออีเวนต์เป็น WIN_RESIZE

วิธีนี้จะทำให้มีการใช้พื้นที่ทั้งหมดของแคนวาส ในการแสดงภาพเช่นกัน และภาพที่ปรากฏจะมีขนาดเปลี่ยนแปลงไป และมีลักษณะสมมาตรแต่จะถูกตัดบางส่วนไป ดังรูปที่ 3.37



รูป 3.37 ภาพวัตถุที่ได้จากการกำหนดเวิร์กสเตชัน วินโดว์ให้มีความสัมพันธ์กับลักษณะของแคนวาส

3.3.3.3 กำหนดให้เฟรมเป็นสี่เหลี่ยมจตุรัสเสมอ

การกำหนดให้เฟรมมีลักษณะเป็นสี่เหลี่ยมจตุรัสเสมอ ทำได้โดยกำหนดค่าด้านสั้นให้มีค่าเท่ากับด้านยาว ในกรณีที่ต้องการให้เฟรมเป็นสี่เหลี่ยมจตุรัสที่ใหญ่ขึ้นจากขนาดเดิม หรืออาจกำหนดให้ด้านยาวมีขนาดเท่ากับด้านสั้น ในกรณีที่ต้องการให้เฟรมเป็นสี่เหลี่ยมจตุรัสที่มีขนาดเล็กลง

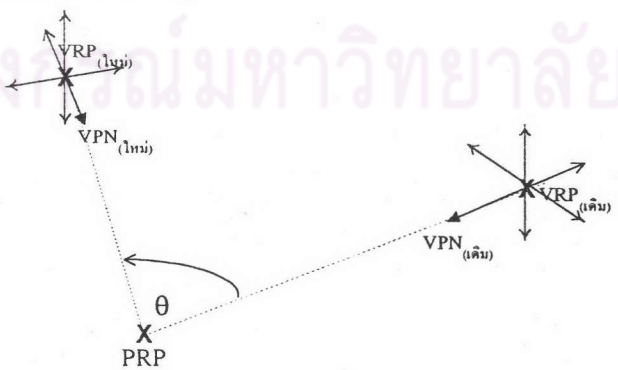
ในโปรแกรมนี้จะใช้วิธีที่ 2 และ 3 ร่วมกัน โดยใช้วิธีที่ 2 เมื่อมีอีเวนต์ของการเปลี่ยนขนาดวินโดว์ และให้ผู้ใช้สามารถเลือกได้ว่าจะทำให้เฟรมนั้นมีลักษณะเป็นสี่เหลี่ยมจตุรัสหรือไม่

3.4 การเปลี่ยนวิวในลักษณะของการซูม การแพน (Pan) การเดิน และการเดินรอบวัตถุในโปรแกรมการจัดการวิว

การเปลี่ยนวิวบนเวิร์กสเตชัน สามารถทำได้โดยการกำหนดข้อมูลวิวของเวิร์กสเตชันของซันฟิกส์ รุ่น 2.0 ประกอบด้วยวิว โอเรียนเตชัน วิว แมปปิง และการตัดภาพ ส่วนการกำหนดการเปลี่ยนแปลงวิวในลักษณอื่น เช่น การซูม การแพนนั้น ซันฟิกส์รุ่น 2.0 ไม่ได้จัดเตรียมไว้ให้ แต่สามารถทำได้โดยกำหนดข้อมูลวิวให้สอดคล้องกับการทำงานเหล่านั้น

การซูมภาพในโปรแกรมการจัดการวิวมี 2 ลักษณะ คือ การซูมเข้า (Zoom In) และการซูมออก (Zoom Out) ซึ่งสามารถกำหนดขนาดการซูมได้ด้วย Zoom Step การซูมสามารถทำได้โดยการเปลี่ยนขนาดขอบเขตของวินโดว์ในวิว แมปปิง (Gaskins, 1991) การซูมเข้า คือ การลดขนาดขอบเขตของวินโดว์ลงให้เท่ากับขนาดของ Zoom Step และการซูมออก คือ การเพิ่มขนาดขอบเขตของวินโดว์ขึ้น โดยให้ขนาดของวินโดว์บนเวิร์กสเชันเท่าเดิม

การแพน คือ การที่ผู้มองอยู่กับที่ แล้วเปลี่ยนทิศทางของการมอง ทำได้โดยการกำหนดจุดอ้างอิงของวิว และวิว เพลน นอร์มัล ให้สัมพันธ์กับมุมมองของการแพน และตำแหน่งของผู้มอง ในโปรแกรมนนี้ได้กำหนดให้จุดอ้างอิงของโปรเจกชันเป็นตำแหน่งของผู้มอง



รูป 3.38 ลักษณะของการแพน

การหาจุดอ้างอิงของวิว และวิว เพลน นอร์มัลใหม่ที่เกิดจากการแพน ทำได้โดยใช้การแปลงแบบหมุนมาแปลงตำแหน่งจุดอ้างอิงของวิว โดยมีจุดอ้างอิงของโปรเจกชันเป็นจุดศูนย์กลาง แล้วจึงกำหนดให้วิว เพลน นอร์มัลใหม่มีทิศทางชี้จากจุดอ้างอิงของวิวไปยังวิว เพลน นอร์มัล

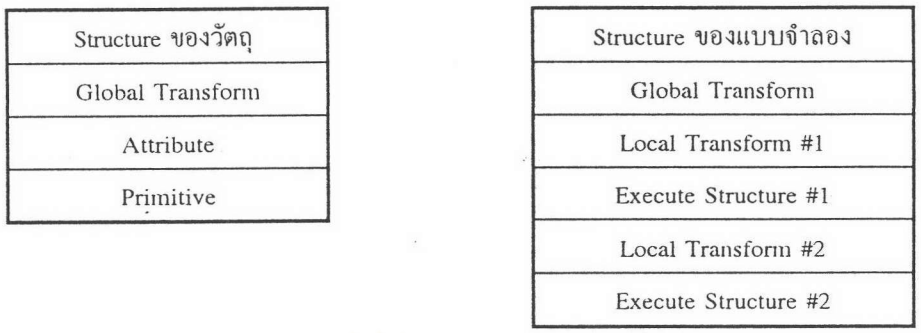
การเดินในโปรแกรมการจัดการวิว จะเปรียบเสมือนการเดินไปมาภายในฉากซึ่งสามารถเดินได้ใน 6 ทิศทาง คือ เดินหน้า ถอยหลัง ซ้าย ขวา ขึ้น และลง การเดิน คือ การเปลี่ยนตำแหน่งของจุดอ้างอิงของวิว โดยให้วิว เพลน นอร์มัล และวิว อัป เวคเตอร์ คงเดิม การเดินหน้า คือ การเลื่อนตำแหน่งจุดอ้างอิงของวิวไปตามแกน z (ในพิกัดอ้างอิงของวิว) ทางด้านค่าลบ การเดินถอยหลัง คือ การเลื่อนตำแหน่งจุดอ้างอิงของวิวไปตามแกน z ทางด้านค่าบวก การเดินไปทางซ้ายและขวา คือ การเลื่อนตำแหน่งจุดอ้างอิงของวิวไปตามแกน x ทางด้านค่าลบและค่าบวก ส่วนการเดินขึ้นและลง คือ การเลื่อนตำแหน่งจุดอ้างอิงของวิวไปตามแกน y ทางด้านค่าบวกและค่าลบ เนื่องจากค่าจุดอ้างอิงของวิวต้องกำหนดในพิกัดโลก ดังนั้นการเปลี่ยนแปลงค่าจุดอ้างอิงของวิวจึงต้องนำวิว เพลน นอร์มัล และวิว อัป เวคเตอร์ มาพิจารณาด้วย การเปลี่ยนตำแหน่งจุดอ้างอิงของวิวจะใช้การแปลงแบบย้ายมาช่วย เช่น การเดินถอยหลังเป็นการกำหนดเวกเตอร์การย้ายให้มีทิศทางเดียวกับวิว เพลน นอร์มัล และมีขนาดเท่ากับ Walk Step เป็นต้น

การเดินรอบวัตถุเปรียบเสมือนการกำหนดให้จุดสนใจของการมองคงที่ และผู้มองเดินไปรอบๆ จุดสนใจนั้น โดยมีระยะห่างเท่าเดิม ซึ่งก็คือการกำหนดจุดอ้างอิงของวิวให้คงที่แล้วเปลี่ยนแปลงเฉพาะค่าวิว เพลน นอร์มัล

เมื่อได้กำหนดการเปลี่ยนแปลงมุมของการมองแล้ว โปรแกรมจะทำการตรวจสอบข้อมูลวิวนั้นว่าถูกต้องหรือไม่ เช่น ค่าวิว เพลน นอร์มัลต้องไม่เป็น (0, 0, 0) และไม่ขนานกับวิว อัป เวคเตอร์ เป็นต้น ในกรณีที่ข้อมูลวิวมีข้อผิดพลาด โปรแกรมจะคงค่าเดิมของข้อมูลวิวไว้

3.5 การแปลงตำแหน่งของวัตถุหรือแบบจำลอง

โครงสร้างของวัตถุ หรือแบบจำลองจะมีองค์ประกอบที่ใช้กำหนดตำแหน่งของรูปทรงพื้นฐานรวมอยู่ด้วย ดังรูปที่ 3.39



รูป 3.39 องค์ประกอบของโครงสร้างของวัตถุและแบบจำลอง

การเปลี่ยนตำแหน่งวัตถุเพื่อให้สัมพันธ์กับตำแหน่งเดิม ทำได้โดยการสร้างเมทริกซ์ของการแปลงแล้วนำไปประกอบ (Compose) กับเมทริกซ์เดิมที่อยู่ในองค์ประกอบของการแปลงครอบคลุม (Global Transform) ซึ่งจะทำให้ตำแหน่งของวัตถุ หรือแบบจำลองเปลี่ยนตำแหน่งไปจากเดิม ส่วนการเปลี่ยนตำแหน่งของชิ้นส่วนในแบบจำลองต้องนำเมทริกซ์การแปลงนั้นมาประกอบกับเมทริกซ์การแปลงเฉพาะที่ของชิ้นส่วนนั้น

การแปลงในโปรแกรมนี้ สามารถกำหนดให้การแปลงแบบหมุนและสเกลมีความสัมพันธ์กับจุดคงที่ในพิกัดโลกได้ การกำหนดมุมหรือขนาดของการสเกลสามารถกำหนดได้โดยใช้ไอเท็มบนเวิร์ก พาเนลเท่านั้น

การแปลงแบบย้ายมี 2 ขั้นตอน คือ การเลือกวัตถุหรือแบบจำลองที่ต้องการย้ายและการกำหนดทิศทางการย้าย เมื่อผู้ใช้เลื่อนตัวชี้ตำแหน่งมาตรงกับวัตถุและกดปุ่มซ้ายบนเมาส์ซึ่งเป็นการเลือกวัตถุ โปรแกรมจะกำหนดให้ตำแหน่งปัจจุบันเป็นจุดเริ่มต้นของเวกเตอร์การย้าย และเมื่อเลื่อนตัวชี้ตำแหน่งไปและกดปุ่มซ้ายบนเมาส์อีกครั้ง จะเป็นการกำหนดจุดปลายของเวกเตอร์ ซึ่งโปรแกรมจะทำการหาขนาดของเวกเตอร์แล้วนำไปสร้างเมทริกซ์การย้าย ในกรณีที่มีการเลือกวัตถุแล้ว และตัวชี้ตำแหน่งถูกเลื่อนออกจากเวิร์กสเตชัน (LOC_WINEXIT) โปรแกรมจะถือว่าเป็นการยกเลิกการย้าย เนื่องจากจุดต้นและจุดปลายของเวกเตอร์จะต้องสัมพันธ์กัน แต่ค่าพิกัดโลกที่หาได้จากแต่ละวิวจะแตกต่างกันไป ดังนั้นการกำหนดจุดต้นและปลายของเวกเตอร์ในเวิร์กสเตชันที่ต่างกัน จะมีวิวต่างกันด้วย จึงทำให้เวกเตอร์การย้ายไม่เป็นไปตามต้องการ

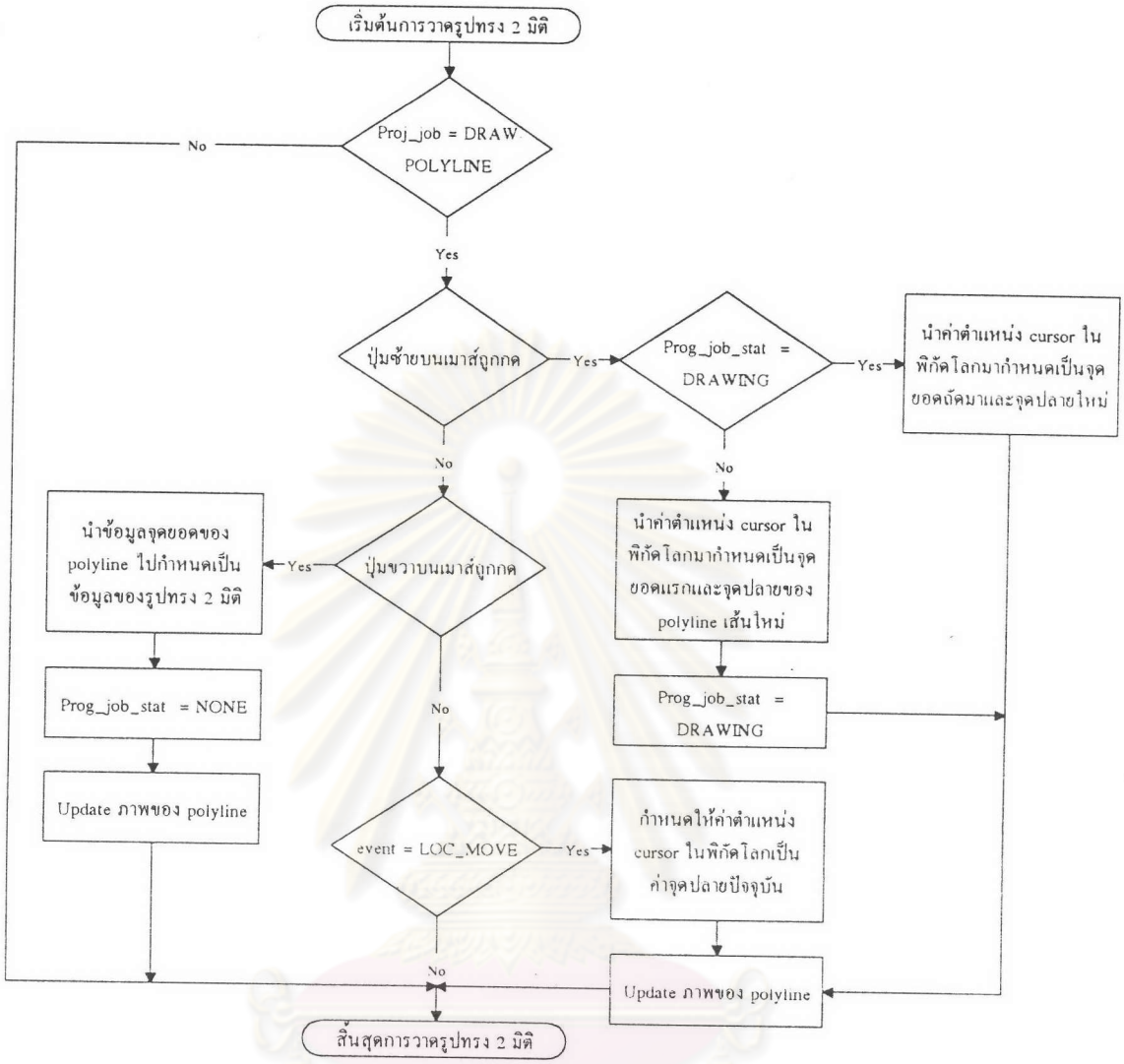
3.6 การวาดรูปทรง 2 มิติ

การวาดรูปทรง 2 มิติ เป็นการทำงานเดียวในโปรแกรมที่มีการปรับภาพและข้อมูลตามการเปลี่ยนแปลงตำแหน่งของตัวชี้ตำแหน่ง เนื่องจากเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรมนี้เป็นเครื่องที่มีความเร็วต่ำ การทำงานกับข้อมูลจำนวนมากจะทำงานได้ไม่ทันกับอีเวนต์ที่เกิดขึ้น แต่รูปทรงพื้นฐานของรูปทรง 2 มิติ จัดเป็นรูปทรงพื้นฐานที่มีข้อมูลที่เกี่ยวข้อง

น้อยที่สุดในโปรแกรมนี้ จึงสามารถนำมาใช้แสดงให้เห็นเป็นแนวทางของการปรับภาพตามการเคลื่อนไหวของตัวชี้ตำแหน่งได้

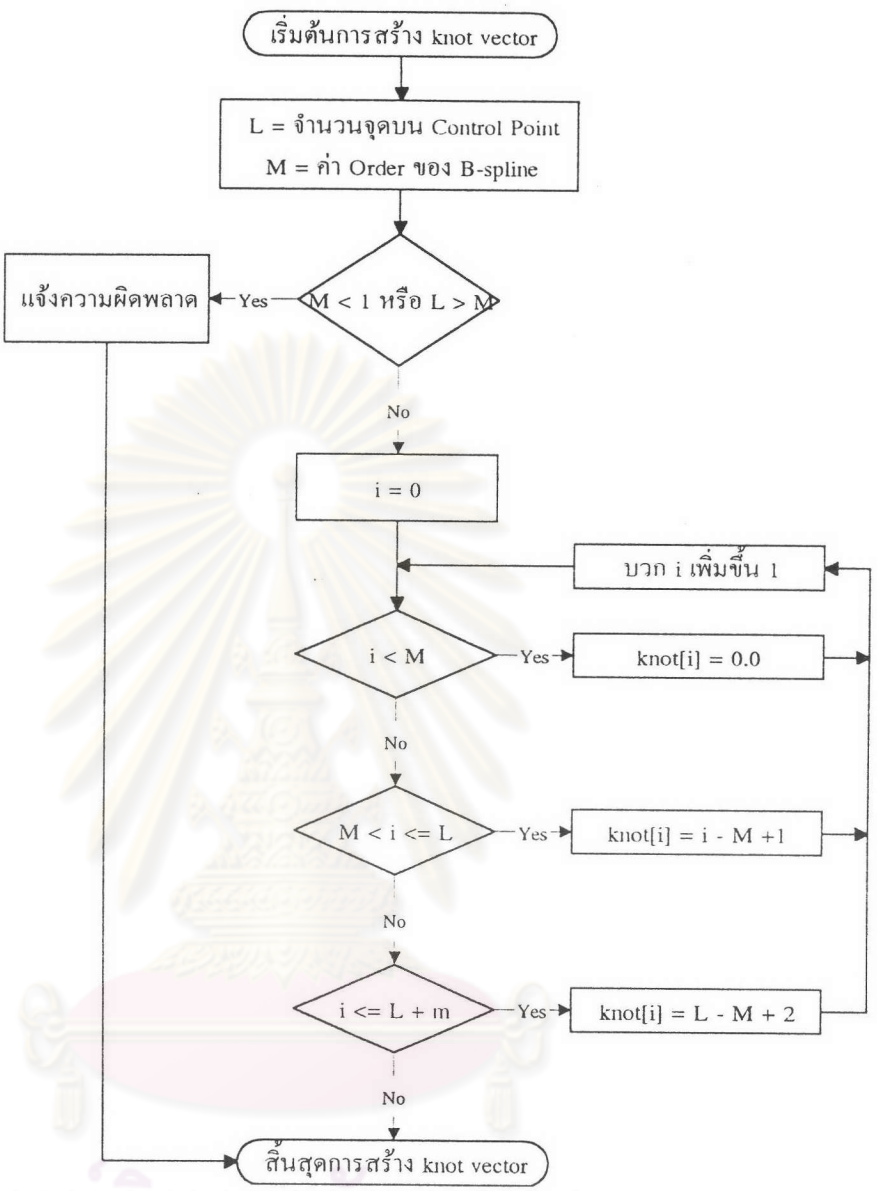
เมื่อมีการกดปุ่ม Draw Polyline บนเวิร์ก พาเนลของโปรแกรมการสร้างรูปทรง 2 มิติ Prog_job จะถูกกำหนดให้เป็น DRAW POLYLINE และ Prog_job_stat จะถูกกำหนดให้เป็น NONE เมื่อมีการกดปุ่มซ้ายบนเมาส์ในขณะที่ตัวชี้ตำแหน่งอยู่บนเวิร์กสเตชันของโปรแกรมการสร้างรูปทรง 2 มิติ จะหมายถึงการเริ่มวาดโพลีไลน์ โดยค่า Prog_job_stat จะถูกกำหนดเป็น DRAWING และโปรแกรมจะกำหนดค่าตำแหน่งปัจจุบันของตัวชี้ตำแหน่งเป็นจุดเริ่มต้นและจุดปลายของเส้นโพลีไลน์นั้น เมื่อมีการเลื่อนตัวชี้ตำแหน่งไปมาค่าพิกัดของจุดปลายก็จะเปลี่ยนไป เมื่อมีการกดปุ่มซ้ายบนเมาส์อีกครั้งจะเป็นการกำหนดจุดถัดมาของโพลีไลน์ และกำหนดจุดนั้นเป็นจุดปลายใหม่ด้วย และเมื่อเลื่อนตัวชี้ตำแหน่งไปมาอีก ค่าพิกัดของจุดปลายนั้นก็จะเปลี่ยนไปอีก ซึ่งจะทำให้เห็นเป็นภาพส่วนของเส้นตรงของเส้นโพลีไลน์เลื่อนตามตัวชี้ตำแหน่งไป เมื่อมีการกดปุ่มขวาบนเมาส์จะเป็นการยกเลิกการวาดภาพ และค่า Prog_job_stat จะกลับเป็น NONE ขั้นตอนการวาดภาพเหล่านี้จะอยู่ในฟังก์ชันควบคุมอีเวนต์ของเวิร์กสเตชันของโปรแกรมการสร้างรูปทรง 2 มิติ ดังนั้นการทำงานของผังงานในรูปที่ 3.40 หมายถึง การทำงานที่เกิดขึ้นเมื่อมีอีเวนต์เกิดขึ้น 1 ครั้ง บนเวิร์กสเตชันของโปรแกรมการสร้างรูปทรง 2 มิติ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



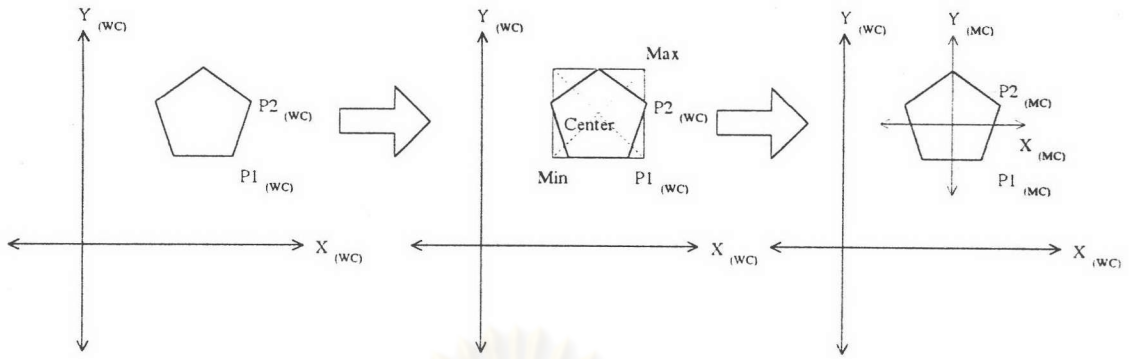
รูป 3.40 ผังงานของการวาดรูปทรง 2 มิติในฟังก์ชันควบคุมอีเวนต์ของโปรแกรมการสร้างรูปทรง 2 มิติ

สำหรับการใช้โพลีไลน์ในการวาดรูปพื้นที่ การทำงานจะแตกต่างออกไปเล็กน้อย คือ จุดสุดท้ายของโพลีไลน์จะมีพิกัดเดียวกับจุดแรกเสมอ ส่วนจุดที่ใช้ในการวาดส่วนของเส้นตรงของโพลีไลน์ที่เลื่อนตามตัวชี้ตำแหน่งจะเป็นจุดรองสุดท้ายแทน ส่วนการวาดเส้นโค้งบีสไปลน์จะใช้วิธีให้ผู้ใช้เลือกโพลีไลน์ที่วาดไว้แล้วมาใช้เป็นจุดควบคุม แล้วโปรแกรมจะทำการคำนวณน็อต เวกเตอร์ให้ การเปลี่ยนรูปทรงบีสไปลน์ใหม่ ทำได้โดยการเลือกโพลีไลน์เส้นใหม่หรือแก้ไขโพลีไลน์เส้นเดิม แล้วเลือกโพลีไลน์ที่แก้ไขแล้วนั้นซ้ำอีกที



รูป 3.41 ผังงานการสร้างน็อต เวกเตอร์ชนิดยูนิฟอร์มแบบเปิด

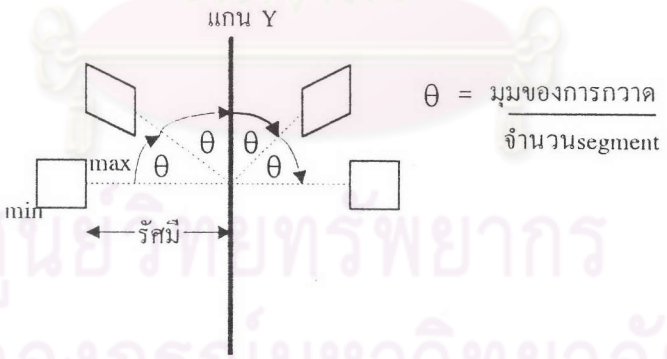
เมื่อได้รูปทรง 2 มิติแล้ว โปรแกรมจะทำการหาขอบเขตของรูปทรงเพื่อหาจุดศูนย์กลาง เมื่อได้หาจุดศูนย์กลางแล้วโปรแกรมจะทำการแปลงค่าพิกัดของจุดยอดต่างๆ บนโพลีไลน์ให้สัมพันธ์กับจุดศูนย์กลาง เพื่อกำหนดเป็นพิกัดแบบจำลองของรูปทรง 2 มิตินี้



รูป 3.42 การกำหนดพิกัดแบบจำลองของรูปทรง 2 มิติ

3.7 การสร้างรูปทรง 3 มิติจากรูปทรง 2 มิติ

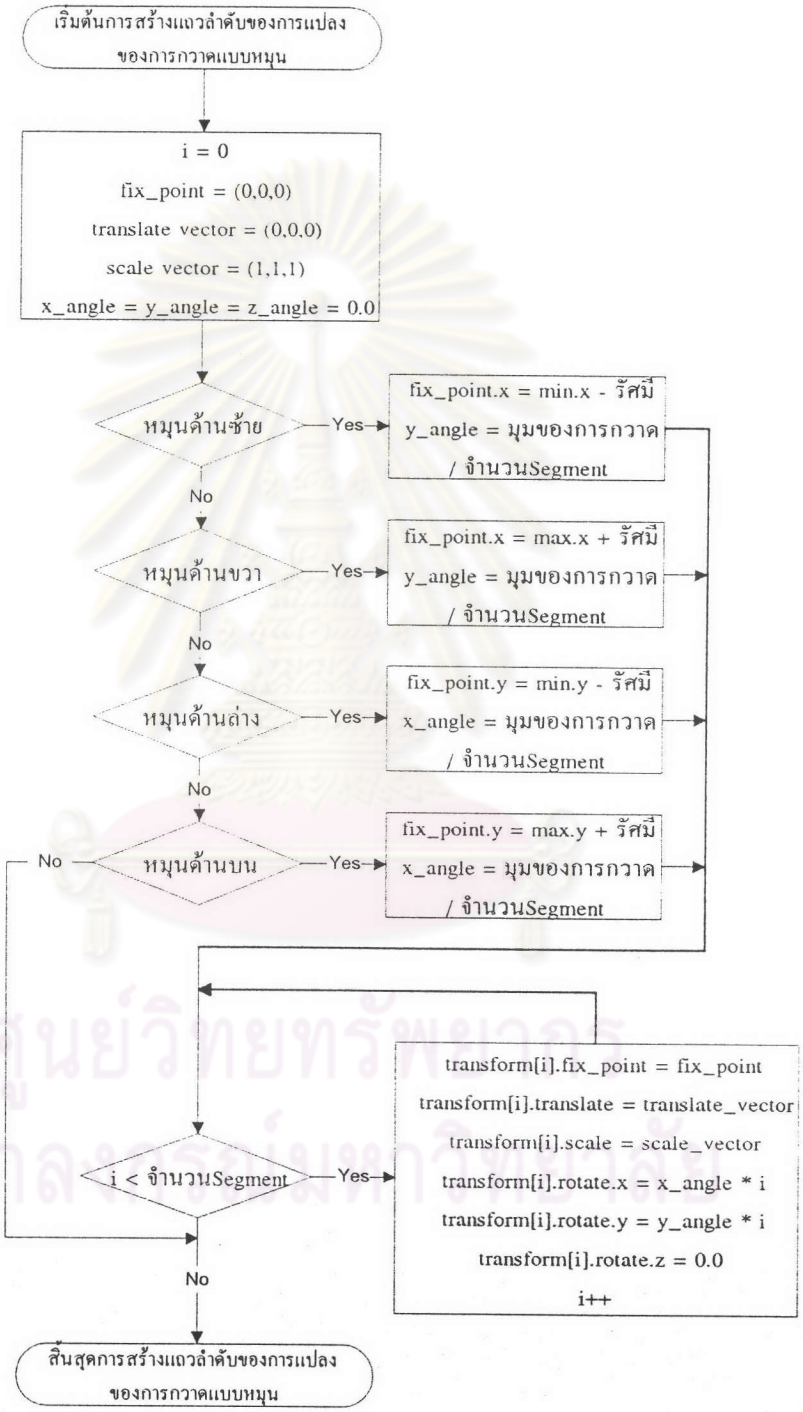
การสร้างรูปทรง 3 มิติด้วยวิธีการกวาดแบบหมุนทำได้โดยการเลือกรูปทรง 2 มิติ เพื่อใช้กำหนดรูปทรงของเซกเมนต์ และกำหนดข้อมูลของการกวาดแบบหมุน คือ มุมของการกวาด รัศมี จำนวนเซกเมนต์ และทิศทางของการกวาด ทิศทางที่กำหนดได้ในโปรแกรมนี้มี 4 ทิศทาง คือ ทางซ้าย ขวา บน และล่างของรูปทรง 2 มิติ เมื่อกำหนดทิศทางแล้ว โปรแกรมจะกำหนดจุดศูนย์กลางของการหมุน โดยมีระยะห่างจากขอบเขตของรูปทรง 2 มิติ เท่ากับรัศมี ส่วนมุมระหว่างเซกเมนต์จะหาได้โดยการนำค่ามุมของการกวาดมาหารด้วยจำนวนเซกเมนต์



รูป 3.43 การสร้างรูปทรง 3 มิติด้วยวิธีการกวาดแบบหมุน

เมื่อกำหนดข้อมูลของการกวาดแบบหมุนแล้ว โปรแกรมจะสามารถสร้างแถวลำดับของการแปลงได้โดยใช้ฟังก์ชัน pbuild_tran_matrix3 โดยกำหนดให้ fix point คือ จุดศูนย์กลาง และมุมของการหมุนเพิ่มขึ้นทีละ θ ตามลำดับเซกเมนต์ ถ้าทิศทางการหมุนเป็นบนหรือล่าง

จะกำหนดให้เป็นการหมุนรอบแกน x ถ้าเป็นซ้ายหรือขวาจะกำหนดให้หมุนรอบแกน y ส่วนเวกเตอร์ของการย้าย การสเกล และค่ามุมอื่นๆ เป็น (0, 0, 0), (1, 1, 1) และ 0.0 ตามลำดับ



รูป 3.44 ผังงานการสร้างแถวลำดับของการแปลงของการกวาดแบบหมุน

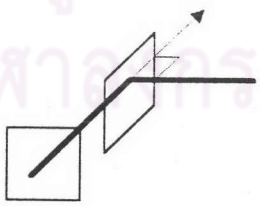
การกวาดแบบย้ายทำได้โดยการเลือกรูปทรง 2 มิติ 2 รูป เพื่อใช้กำหนดรูปทรงของเซกเมนต์ และเส้นทางของการย้าย (Path) รูปทรงที่นำมาใช้เป็นเส้นทางของการย้ายจะนำมาใช้ในการกำหนดเวกเตอร์ของการย้าย โดยการนำจุดยอดปัจจุบันมาลบกับจุดยอดแรก เนื่องจากรูปทรง 2 มิติจะถูกกวาดในระนาบของ xy แต่เส้นทางของการกวาดแบบย้ายจะใช้ระนาบของ xz ดังนั้นค่าพิกัด y จะถูกปรับมาใช้เป็นพิกัด z แทน ดังนี้

```
transform[n].translate.delta_x = vertex[n].x - vertex[0].x;
transform[n].translate.delta_y = 0.0;
transform[n].translate.delta_z = vertex[n].y - vertex[0].y;
```

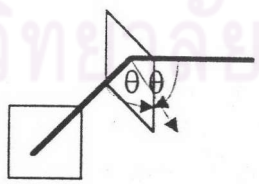
ในกรณีที่เส้นทางมีการหักมุมหรือเปลี่ยนแปลงทิศทาง โปรแกรมจะให้เลือกลักษณะการเปลี่ยนแปลงมุมของเซกเมนต์ได้ 4 วิธี คือ ไม่เปลี่ยนแปลง ทำมุมตั้งฉากกับส่วนของเส้นตรงของเส้นทางก่อนหน้า ทำมุมตั้งฉากกับส่วนของเส้นตรงของเส้นทางถัดไป และทำมุมเป็นครึ่งหนึ่งระหว่างมุมของส่วนของเส้นตรงทั้ง 2 ค่ามุมที่ใช้กำหนดการหมุนให้กับเซกเมนต์จะเป็นมุมของการหมุนรอบแกน y

ก. ไม่เปลี่ยนแปลง

ข. ตั้งฉากกับเส้นก่อนหน้า



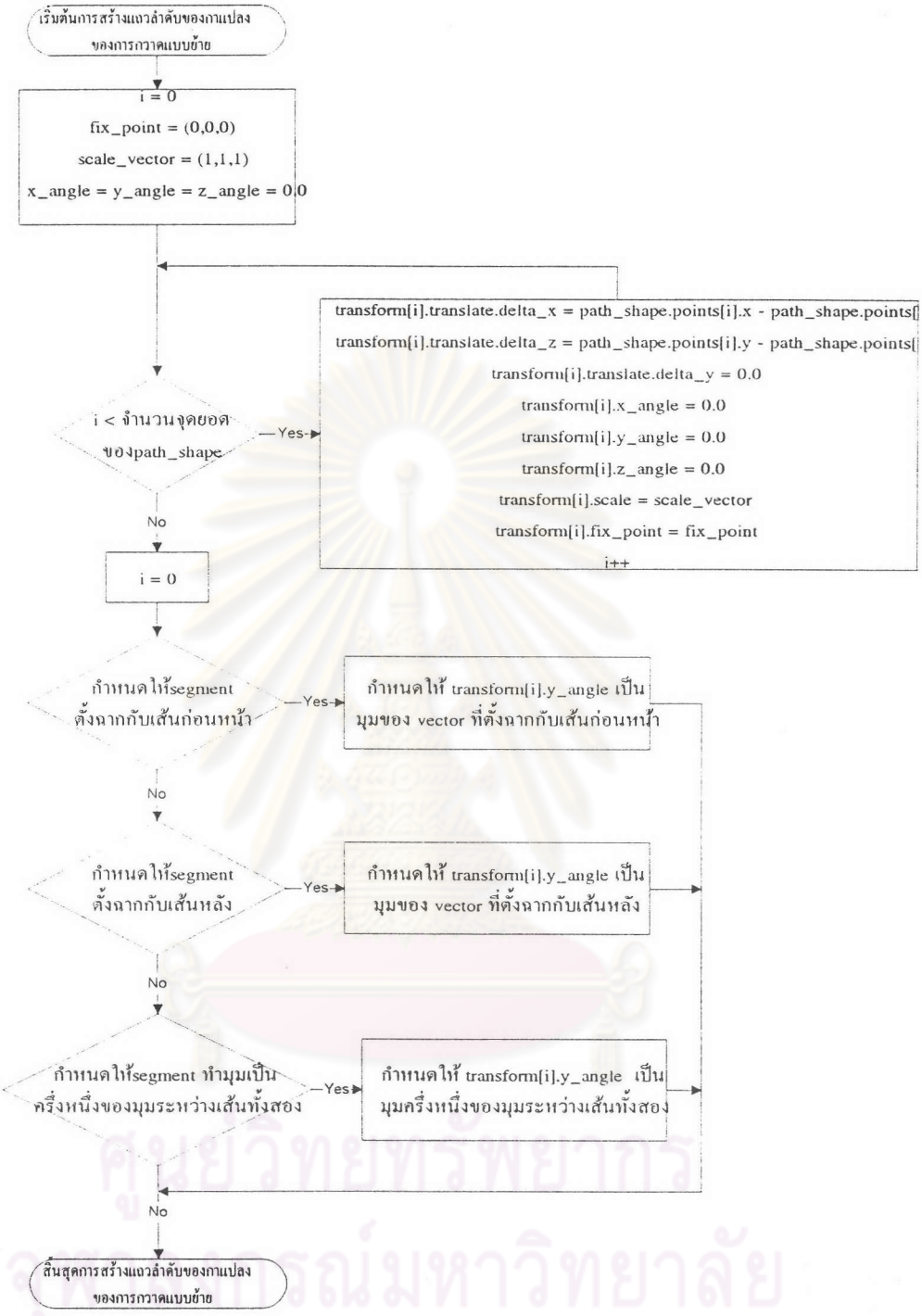
ค. ตั้งฉากกับเส้นหลัง



ง. ทำมุมเป็นครึ่งหนึ่ง

รูป 3.45 ลักษณะการเปลี่ยนแปลงมุมของเซกเมนต์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



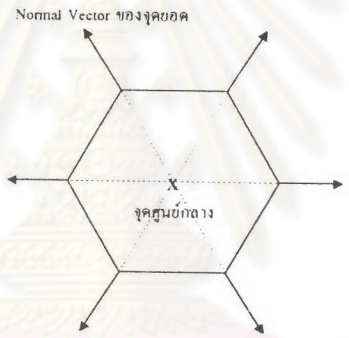
รูป 3.46 ผังงานการสร้างแถวลำดับของการแปลงของการกวาดแบบย้าย

3.8 การกำหนดรูปทรงพื้นฐานให้กับรูปทรง 3 มิติ

เมื่อกำหนดเซกเมนต์ของวัตถุได้แล้ว ถ้าเป็นการสร้างรูปโดยใช้รูปทรงพื้นฐานชนิดพื้นผิวแบบบีสไปลน์ โปรแกรมจะนำเอาเซกเมนต์ต่างๆ ไปใส่เรียงกันในแถวลำดับของกริด

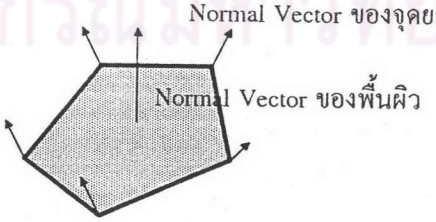
ควบคุมของบีสไปลน์ได้ทันที ส่วนกรณีของชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูลนั้น โปรแกรมต้องแปลงข้อมูลของเซกเมนต์ (ซึ่งเก็บข้อมูลจุดยอดในลักษณะแถวลำดับของ Ppoint3 คือ เก็บเฉพาะข้อมูลพิกัด) ไปเป็นแบบ Pptnorm3 ซึ่งจะเก็บทั้งข้อมูลพิกัดและนอร์มัล เวกเตอร์ของจุดยอด สำหรับนอร์มัล เวกเตอร์ของจุดยอดจะถูกนำไปใช้ในการหานอร์มัล เวกเตอร์ของพื้นผิว เพื่อใช้กำหนดด้านหน้าและด้านหลังของพื้นผิวต่อไป ด้านหน้าและด้านหลังของพื้นผิวจะมีความสำคัญต่อการกำหนดการซ่อนเส้นและพื้นผิว จึงต้องกำหนดนอร์มัล เวกเตอร์ของพื้นผิวให้สัมพันธ์กับลักษณะของวัตถุด้วย

การกำหนดนอร์มัล เวกเตอร์ให้กับจุดยอด โปรแกรมจะพยายามกำหนดให้เวกเตอร์ชี้ออกจากตัวเซกเมนต์ โดยการหาขอบเขตของเซกเมนต์และนำมากำหนดหาจุดศูนย์กลาง แล้วกำหนดทิศทางของเวกเตอร์ให้ชี้ออกจากจุดศูนย์กลางไปยังจุดยอดนั้น ดังรูปที่ 3.43



รูป 3.47 นอร์มัล เวกเตอร์ของจุดยอดที่ชี้ออกจากเซกเมนต์

ดังนั้นเมื่อนำเอาจุดยอดเหล่านี้ไปคำนวณหานอร์มัล เวกเตอร์ของพื้นผิว จะได้เวกเตอร์ที่ชี้ออกจากตัววัตถุ ซึ่งทำให้ด้านนั้นกลายเป็นด้านหน้าของพื้นผิว



รูป 3.48 นอร์มัล เวกเตอร์ของพื้นผิวที่ชี้ออกจากตัววัตถุ