สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการออกแบบตกแต่งภายในแบบสามมิติ

3D INTERIOR LAYOUT DESIGN

นายนฤวัจน์ เพชรแสงงาม
นายชินวัฒน์ ชูเนื่อง

เลขหมู่.....
เลขทะเบียน 61372
17 ก.ค. 2549
วัน,เดือน,ปี.....
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการออกแบบตกแต่งภายในแบบสามมิติ

3D INTERIOR LAYOUT DESIGN

โดย

นายนฤวัจน์ เพชรแสงงาม

นายชินวัฒน์ ชูเนื่อง

อาจารย์ที่ปรึกษา

ดร.อรัญญา วลัยรัชต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบออกแบบตกแต่งภายในแบบสามมิติ

3D INTERIOR LAYOUT DESIGN

คณะผู้จัดทำ นายนฤวัจน์ เพชรแสงงาม รหัส 45015372

นายชินวัฒน์ ชูเนื่อง รหัส 45015362

..... อาจารย์ที่ปรึกษา

(ดร.อรัญญา วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการออกแบบตกแต่งภายในแบบสามมิติ

นายนฤวัจน์ เพชรแสงงาม 45015372
นายชินวัฒน์ ชูเนื่อง 45015362
ดร. อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ระบบการออกแบบตกแต่งภายในแบบสามมิติ เป็นระบบที่จำลองการจัดวางเฟอร์นิเจอร์ภายในห้อง ซึ่งผู้ใช้สามารถจัดวางเฟอร์นิเจอร์ต่างๆ รวมทั้งสามารถเคลื่อนย้าย ปรับเปลี่ยนตำแหน่งหรือหมุนเฟอร์นิเจอร์ให้ได้ตามตำแหน่งตามที่ต้องการ โดยที่อินพุตของระบบเป็นภาพถ่ายจากสถานที่จริงของห้อง ตามมุมมองต่างๆ ที่ผู้ใช้ต้องการเห็นมาเป็นข้อมูลให้กับโปรแกรมเพื่อให้โปรแกรมสามารถสร้างห้องจำลองได้เหมือนจริงมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3D INTERIOR LAYOUT DESIGN

Mr. Naruwat Petchsangngam
Mr. Chinnawat Chunarng
Dr. Aranya Warairacht Advisor
Academic Year 2004

ABSTRACT

3D Interior Layout Design is a system which is not only able to imitate the furniture manipulation inside a room, but also can arbitrarily translate and rotate furniture. For the input of the program is a photo from a various view of rooms. After that transform it becomes an input for the program ,which user wants to see, so that the program is able to create as much close to the real room as possible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 ขอบเขตการวิจัย	1
1.3 วัตถุประสงค์	1
1.4 ผลที่คาดว่าจะได้รับ	1
1.5 วิธีการดำเนินงาน	1
1.5.1 Input	2
1.5.2 Create Room	2
1.5.3 Manipulate Furniture	4
1.5.4 Display Output	5
บทที่ 2 ทฤษฎี	6
2.1 ทฤษฎีคอมพิวเตอร์กราฟฟิก	6
2.1.1 Transformations	6
2.1.2 Viewing ทฤษฎีการมองเห็นภาพ	7
2.1.2.1 Classical Viewing	8
2.1.2.2 Perspective Viewing	8
2.1.2.3 Viewing กับ Computer	9
2.1.2.4 Viewport	9
2.2 ความรู้เกี่ยวกับ DirectX	9
2.2.1 API ของ DirectX	9
2.2.2 DirectInput	10
2.2.3 DirectDraw	10
2.2.4 Direct3D	10
2.2.5 การทำงานของ Direct3D	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

2.2.6 ฟังก์ชั่นของ DirectX ที่นำมาใช้จัดการเกี่ยวกับกล้อง 11

2.3 เวอร์เท็กซ์ของ Direct3D กับกระบวนการให้แสง 11

2.4 ภาพรวมของ T&L Pipeline 12

2.4.1 การแปรไปสู่พิกัดเวิลด์ 13

2.4.2 การแปรไปสู่พิกัดวิว 14

2.4.3 การให้แสง 15

2.4.4 การแปรไปสู่พิกัดโปรเจ็กต์และ Viewing Frustum 15

2.4.5 การขริบ 17

2.4.6 การหารด้วยพิกัด W หรือกระบวนการทำ Nonhomogeneous 18

2.4.7 การปรับขนาดวิวพอร์ต 18

2.4.8 เวอร์เท็กซ์ 18

2.4.8.1 D3DVERTEX (ไม่แปรพิกัดและไม่ให้แสงเวอร์เท็กซ์) 19

2.4.8.2 D3DLVERTEX (ไม่แปรพิกัดแต่ให้แสงเวอร์เท็กซ์) 19

2.4.8.3 D3DTLVERTEX (แปรพิกัดและให้แสงเวอร์เท็กซ์) 19

บทที่ 3 การวิเคราะห์และออกแบบระบบ 20

3.1 Use-Case Diagram. 20

3.1.1 Use-Case::System. 20

3.1.2 Use-Case::3DI Program. 21

3.1.3 Use-Case::Create Room. 21

3.1.4 Use-Case::Furniture Manipulation. 22

3.1.5 Use-Case::Viewing. 23

3.1.6 Use-Case::Viewing Manipulation. 23

3.2 User Interface 25

3.2.1 View Control 26

3.2.1.1 View 26

3.2.1.2 Keyboard Control Key 26

3.2.2 Tool 27

3.2.3 Furniture Catalog 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
3.2.4.3 Menu View	30
3.2.4.4 Menu Insert	30
3.2.4.5 Menu Window	31
3.2.4.6 Menu Help	31
บทที่ 4 การทดลอง	32
4.1 การทดลองการสร้างห้อง	32
4.2 การทดลองการจัดวางเฟอร์นิเจอร์และการแสดงห้องในมุมมองแบบต่างๆ	33
4.3 การทดลองการนำภาพถ่ายเข้ามาแสดงภายในโปรแกรม	35
บทที่ 5 บทสรุป	40
5.1 สรุป	40
5.2 ปัญหาที่เกิดขึ้น	40
5.3 แนวทางการพัฒนาต่อ	40
ภาคผนวก	41
บรรณานุกรม	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 1-1 Block Diagram แสดงหลักการทำงานโดยรวมของโปรแกรม 2
รูปที่ 1-2 แสดงมุมมองตัวอย่างภายในห้องและแสดงจุดอ้างอิงของการถ่ายภาพ 3
รูปที่ 1-3 แสดงมุมมองตัวอย่างจากด้านบน 3
รูปที่ 1-4 แสดงมุมมองตัวอย่างจากด้านข้าง 4
รูปที่ 1-5 แสดงมุมมองแบบสามมิติของความสัมพันธ์ระหว่างมุมกล้องกับจุดอ้างอิง 4
รูปที่ 2-1 แสดงการ Translation 6
รูปที่ 2-2 แสดงการ Rotation 7
รูปที่ 2-3 แสดงการ Viewing 7
รูปที่ 2-4 แสดงการย้าย COP ไปไว้ที่ Infinity 8
รูปที่ 2-5 รูปแสดง Classical Perspective view 8
รูปที่ 2-6 ขั้นตอนทั้งสองในกระบวนการเรนเดอร์อ็อบเจ็กต์สามมิติ 11
รูปที่ 2-7 กระบวนการ T&L Pipeline 12
รูปที่ 2-8 การแปรไปสู่พิกัดเวิลด์ (ระบบพิกัดเวิลด์และ Local) 14
รูปที่ 2-9 การแปรไปสู่พิกัดวิว 14
รูปที่ 2-10 แสดงองค์ประกอบของ Viewing Frustum 16
รูปที่ 2-11 โปรเจ็กต์ชันแบบ Perspective 17
รูปที่ 3-1 แสดง Use-Case Diagram โดยรวมของระบบ 20
รูปที่ 3-2 แสดง Use-Case Diagram โดยรวมของโปรแกรม 21
รูปที่ 3-3 แสดง Use-Case Diagram ในส่วนการ Create Room 21
รูปที่ 3-4 แสดง Use-Case Diagram ในส่วนการ Furniture Manipulation 22
รูปที่ 3-5 แสดง Use-Case Diagram ในส่วนการ Viewing 23
รูปที่ 3-6 แสดง Use-Case Diagram ในส่วนการ Viewing Manipulation 23
รูปที่ 3-7 แสดงหน้าจอทั้งหมดของโปรแกรม 3DI 25
รูปที่ 3-8 แสดงส่วนของ View Control 26
รูปที่ 3-9 ส่วนของ Tool 27
รูปที่ 3-10 แสดงส่วนของ Furniture Catalog 28
รูปที่ 3-11 ตัวอย่างของ Furniture ที่สามารถเลือกใช้ได้ 28
รูปที่ 3-12 แสดงฟังก์ชั่นการทำงานใน Menu File 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 3-13 แสดงฟังก์ชั่นการทำงานใน Menu Edit 29
รูปที่ 3-14 แสดงฟังก์ชั่นการทำงานใน Menu View 30
รูปที่ 3-15 แสดงตัวอย่าง Dialog เมื่อเลือก Menu Image View 30
รูปที่ 3-16 แสดงฟังก์ชั่นการทำงานใน Menu Insert 30
รูปที่ 3-17 แสดงตัวอย่าง Dialog ของ Insert Image 30
รูปที่ 3-18 แสดงฟังก์ชั่นการทำงานใน Menu Window 31
รูปที่ 3-19 แสดงฟังก์ชั่นการทำงานใน Menu Help 31
รูปที่ 4-1 แสดงการกำหนดขนาดห้องและมุมมองเริ่มแรกของโปรแกรม 32
รูปที่ 4-2 แสดงห้องที่โปรแกรมสร้างขึ้น 33
รูปที่ 4-3 แสดงตัวอย่างการจัดวางเฟอร์นิเจอร์ใน Front View 33
รูปที่ 4-4 แสดงตัวอย่างการจัดวางเฟอร์นิเจอร์ใน Top View 34
รูปที่ 4-5 แสดงตัวอย่างที่ 1 สำหรับมุมมองแบบ Free View 34
รูปที่ 4-6 แสดงตัวอย่างที่ 2 สำหรับมุมมองแบบ Free View 35
รูปที่ 4-7 รูปถ่ายจากสถานที่จริงมุมมองที่ 1 35
รูปที่ 4-8 แสดงตัวอย่าง Image View มุมมองที่ 1 36
รูปที่ 4-9 แสดงตัวอย่างมุมมองที่ 1 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย 36
รูปที่ 4-10 รูปถ่ายจากสถานที่จริงมุมมองที่ 2 37
รูปที่ 4-11 แสดงตัวอย่าง Image View มุมมองที่ 2 37
รูปที่ 4-12 แสดงตัวอย่างมุมมองที่ 2 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย 38
รูปที่ 4-13 รูปถ่ายจากสถานที่จริงมุมมองที่ 3 39
รูปที่ 4-14 แสดงตัวอย่าง Image View มุมมองที่ 3 39
รูปที่ 4-15 แสดงตัวอย่างมุมมองที่ 3 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบัน โปรแกรมออกแบบตกแต่งภายในแบบสามมิติเป็นแสดงภาพจากห้องที่สร้างขึ้นภายในโปรแกรม ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่เหมือนจริงเท่าที่ควร เนื่องจากข้อจำกัดด้านการสร้างและแสดงภาพแบบสามมิติในคอมพิวเตอร์ ดังนั้น โครงงานนี้จึงเกิดขึ้นเพื่อดำเนินการวิจัยโดยให้ชื่อโครงงานนี้ว่า " ระบบการออกแบบตกแต่งภายในแบบสามมิติ (3D INTERIOR LAYOUT DESIGN) " โดยมีจุดมุ่งหมายเพื่อสร้างโปรแกรมที่นำภาพถ่ายจากสถานที่จริงเข้ามาใช้ในการแสดงภาพในการออกแบบตกแต่งภายใน ทำให้การแสดงภาพมีความสมจริงเนื่องจากภาพถ่ายจากสถานที่จริงมีรายละเอียดถูกต้องมากกว่าภาพที่สร้างขึ้นเองในคอมพิวเตอร์

1.2 ขอบเขตการวิจัย

โปรแกรมสามารถจำลองการตกแต่งภายในห้องแบบสามมิติ โดยใช้ตัวอย่างห้องที่มีลักษณะเป็นห้องโล่งสี่เหลี่ยมเป็นกรณีศึกษา ส่วนเฟอร์นิเจอร์ใช้เป็นโมเดล 3 มิติ สร้างจากโปรแกรม 3D Studio Max ขนาดเท่าของจริง โดยผู้ใช้สามารถนำเฟอร์นิเจอร์ที่มีในโปรแกรมมาจัดวางภายในห้องได้ตามต้องการนอกจากนั้นผู้ใช้ยังสามารถเลือกมุมมองภาพถ่ายหรือมุมมองอิสระได้

1.3 วัตถุประสงค์

1. เพื่อศึกษาวิธีการวิเคราะห์ ออกแบบและสร้างระบบที่สามารถนำมาประยุกต์ใช้งานได้
2. เพื่อศึกษาหลักการทฤษฎีของ Computer Graphics ศึกษาโปรแกรมสร้างภาพ 3 มิติ

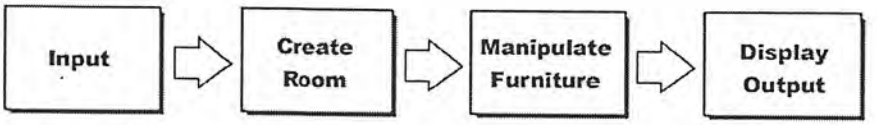
1.4 ผลที่คาดว่าจะได้รับ

1. ได้รับความรู้ความสามารถในการเขียน โปรแกรม Visual C++ และ DirectX.
2. ได้รับความรู้ความสามารถในการสร้างวัตถุสามมิติจากโปรแกรม 3D Studio Max.

1.5 วิธีการดำเนินงาน

หลักการทำงานเบื้องต้นของระบบ สามารถอธิบายได้ตาม Block Diagram รูปที่ 1-1 ซึ่งประกอบด้วยส่วนการทำงานหลักๆ 4 ส่วนด้วยกันคือ Input, Create Room, Manipulate Furniture และ Display Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-1 Block Diagram แสดงหลักการทำงานโดยรวมของโปรแกรม

1.5.1 Input

คือ Input ที่จะป้อนให้กับระบบ ได้แก่

1 ไฟล์ภาพ (Image)

เป็นไฟล์ภาพสีที่ได้มาจากการใช้กล้องดิจิตอล ถ่ายภาพจากห้องที่ต้องการทำการจัดวางเฟอร์นิเจอร์ในมุมมองที่ผู้ใช้ต้องการมองเห็น โดยจะอยู่ในรูปแบบของ BMP format ในโหมด RGB Color 24 บิต ขนาด 800*600 pixel (ซึ่งเป็นขนาดของ Window จะที่แสดงให้ผู้ใช้เห็นในโปรแกรม)

2 ขนาดห้อง (Room size)

ทำการวัดขนาดความกว้าง ความยาว และความสูงของห้องที่เราจะถ่ายภาพ ซึ่งจะใช้หน่วยวัดเป็นเมตร

3 ตำแหน่งของกล้อง (Camera Position)

ทำการวัดตำแหน่งของกล้องที่ทำการถ่ายภาพ เทียบกับขนาดของห้อง โดยค่าที่วัดได้แก่ ระยะห่างทางด้านกว้าง ด้านยาว และความสูงจากพื้น มีหน่วยวัดเป็นเมตร

4 ตำแหน่งอ้างอิงของภาพ (Look at Position)

คือตำแหน่งจุดศูนย์กลางของภาพ หรือ Look At เป็นจุดที่ใช้อ้างอิงตำแหน่งที่กล้องจะทำการถ่ายภาพไปที่จุดนั้นเป็นจุดศูนย์กลางของภาพ ในมุมมองนั้น โดยค่าที่วัดเป็นลักษณะเดียวกับตำแหน่งของกล้อง คือวัดระยะเทียบกับขนาดห้อง ได้แก่ ระยะห่างทางด้านกว้าง ด้านยาว และความสูงจากพื้น มีหน่วยวัดเป็นเมตร

1.5.2 Create Room

จากข้อมูล Input ที่ได้มาวิเคราะห์เพื่อคำนวณสร้างห้องจำลองให้ได้ขนาดอัตราส่วนและมุมมองที่ตรงกับภาพถ่าย

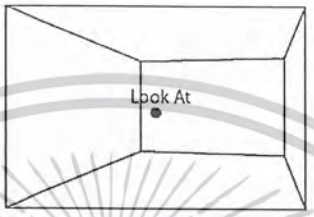
ข้อมูลที่เป็นขนาดห้อง จะนำมาสร้างพื้นที่ห้องขึ้นในโปรแกรม พื้นที่ในส่วนนี้จะเป็นพื้นที่ที่นำมาจัดวางเฟอร์นิเจอร์ต่างๆต่อไป

ข้อมูลตำแหน่งของกล้องและตำแหน่งอ้างอิงของภาพ ก็จะนำมากำหนดให้กับตำแหน่งกล้องและตำแหน่งการมอง(Look At) ภายในโปรแกรม ตามลำดับ ซึ่งเป็นตัวกำหนดมุมมองที่ผู้ใช้ต้องการจะเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำการสร้างห้องจำลองขึ้นโดยมีขนาด และมุมมองตาม Input ที่ได้รับเข้ามา ซึ่งจำทำให้มุมมองที่ได้จากโปรแกรมตรงกับในภาพถ่าย โดยโปรแกรมจะไม่แสดงห้องที่สร้างขึ้นให้ผู้ใช้เห็นแต่จะใช้ในการอ้างอิงในการจัดวางเฟอร์นิเจอร์ เพื่อให้เฟอร์นิเจอร์วางในลักษณะที่ถูกต้อง

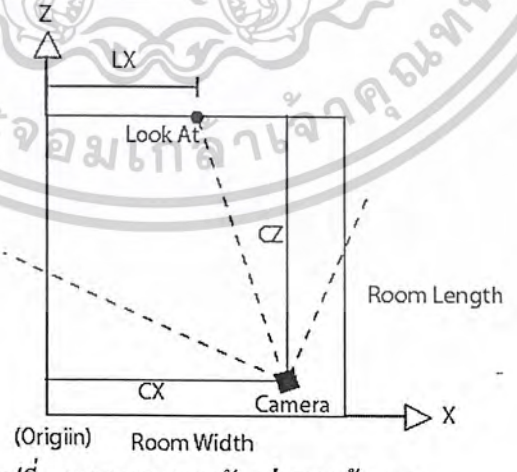
ไฟล์ภาพถ่ายห้องที่อยู่ในรูปแบบตามที่กำหนดไว้จะถูกนำมาใช้เป็นพื้นหลัง (Back Ground) เพื่อแสดงภาพให้ผู้ใช้ได้เห็น



รูปที่ 1-2 แสดงมุมมองตัวอย่างภายในห้องและแสดงจุดอ้างอิงของการถ่ายภาพ

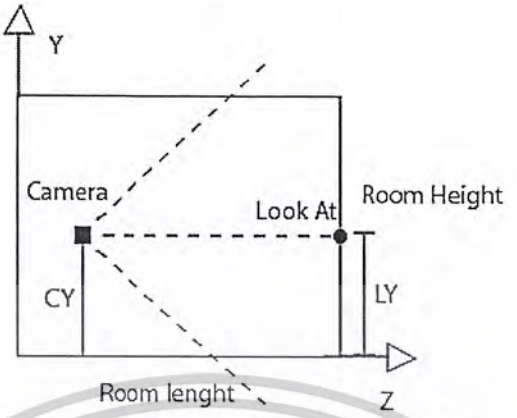
จากมุมมองตัวอย่าง เมื่อเรามองจากด้านบนและด้านข้างก็จะเห็นลักษณะโดยรวมดังภาพด้านล่าง ซึ่งก็คือ Input ที่เราป้อนให้กับระบบ

- Room Width คือ ความกว้างของห้อง
- Room Length คือ ความยาวของห้อง
- Room Height คือ ความสูงของห้อง
- CX, CY, CY คือ ตำแหน่งของกล้อง (Camera)
- LX, LY, LZ คือ ตำแหน่งอ้างอิงของภาพ (Look At)

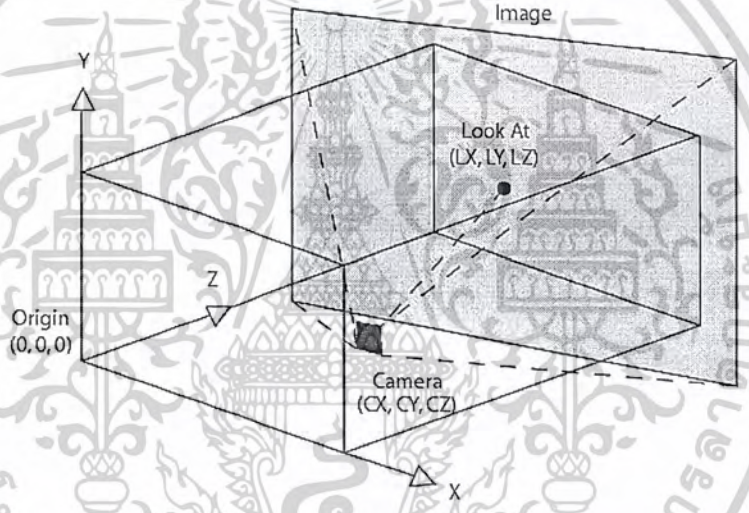


รูปที่ 1-3 แสดงมุมมองตัวอย่างจากด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-4 แสดงมุมมองตัวอย่างจากด้านข้าง



รูปที่ 1-5 แสดงมุมมองแบบสามมิติของความสัมพันธ์ระหว่างมุมกล้องกับจุดอ้างอิง

1.5.3 Manipulate Furniture

เป็นส่วนที่เกี่ยวกับการจัดการเฟอร์นิเจอร์รวมถึงการตรวจสอบความถูกต้องในการจัดวาง โดยผู้ใช้จะเลือกเฟอร์นิเจอร์ต่างๆที่มีอยู่แล้วในตัวโปรแกรมและทำการจัดวางเฟอร์นิเจอร์เหล่านั้นเข้าไปในห้อง เฟอร์นิเจอร์ที่ถูกใส่ให้กับโปรแกรมจะสอดคล้องกับสภาพแวดล้อมของห้องในภาพถ่าย ทั้งขนาดและมุมมอง โดยลักษณะการจัดวางเฟอร์นิเจอร์ได้แก่

- การเคลื่อนย้ายตำแหน่ง (Translation)
- การหมุน (Rotation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะทำการการตรวจสอบความถูกต้องในการจัดวาง เช่น การตรวจสอบพื้นที่ในการวางเฟอร์นิเจอร์ ไม่ให้ทับซ้อนกัน ตำแหน่งในการวางเฟอร์นิเจอร์อยู่ภายในพื้นที่ห้อง เป็นต้น

เฟอร์นิเจอร์ที่นำมาใช้ได้มาจากโปรแกรมสร้าง Model 3 มิติ 3D studio Max โดยจะถูก save ให้อยู่ในรูปแบบที่โปรแกรมสามารถเรียกใช้ได้ คือ ไฟล์.x

1.5.4 Display Output

หลังจากจัดวางเฟอร์นิเจอร์และตรวจสอบความถูกต้องต่างๆเรียบร้อยแล้ว ก็จะทำการแสดงผลภาพห้องและเฟอร์นิเจอร์ต่างๆ ให้ผู้ใช้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 ทฤษฎีคอมพิวเตอร์กราฟิก

ทฤษฎีทางคอมพิวเตอร์กราฟิกที่เกี่ยวข้องกับโครงงานนี้ได้แก่

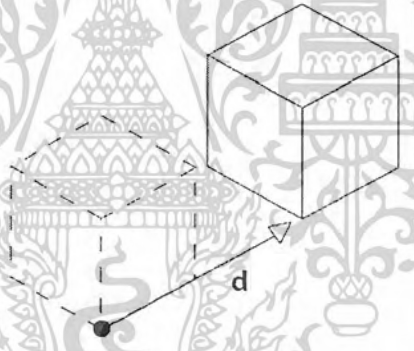
2.1.1 Transformations

คือการย้ายจุดหรือเวกเตอร์ไปไว้ในอีกจุดหรือเวกเตอร์อื่น ทฤษฎีการทรานฟอร์มเมชันที่นำมาใช้ในการกำหนดตำแหน่งกล้องและการจัดวางเฟอร์นิเจอร์คือ

- Translation การย้ายตำแหน่ง

Translation คือการปฏิบัติการเคลื่อนที่จุด โดยมีการกำหนดระยะทางหรือ ทิศทาง ดังแสดงในรูปที่ 2-1 ในการกำหนด Translation นั้น เราต้องการกำหนดเฉพาะ displacement ของเวกเตอร์ d เท่านั้น เพราะจุดที่จะถูก transform สามารถหาได้จากจากสมการดังนี้

$$P' = P + d$$

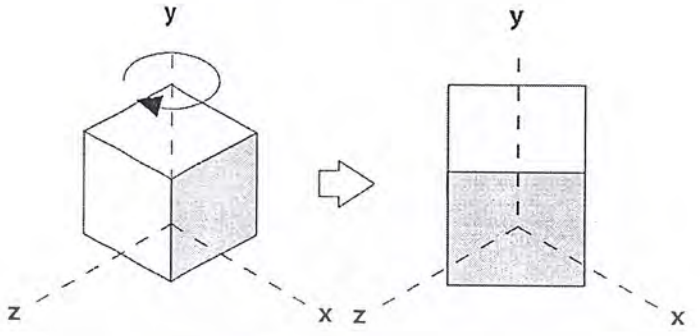


รูปที่ 2-1 แสดงการ Translation

- Rotation การหมุน

การทำ Rotation จะมีจำนวนของพารามิเตอร์ที่มาเกี่ยวข้องมากกว่าการทำ Translation นั่นคือ จุดอ้างอิงในการหมุน (Fix point) มุมในการหมุน และ เวกเตอร์ที่เราจะหมุนรอบแกน

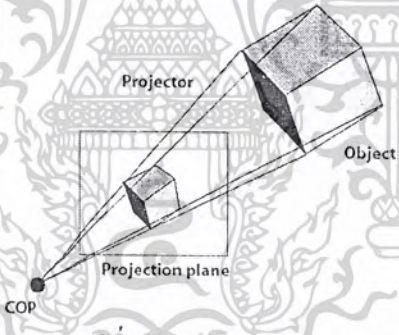
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 แสดงการ Rotation รอบแกน y

2.1.2 Viewing ทฤษฎีการมองเห็นภาพ

การมองเห็นในความเป็นจริง Classical viewing และ การมองเห็นแบบคอมพิวเตอร์ Computer viewing มีส่วนประกอบพื้นฐานที่เหมือนกันคือ โดยแสดงในรูปที่ 2-3 เรามี object, viewer , projectors และ projection plane projector จะมาบรรจบที่ center of projection (COP) ซึ่ง COP จะสัมพันธ์กับจุดศูนย์กลางของเลนส์กล้องหรือ ดวงตามนุษย์ และในระบบ Computer Graphic ก็คือจุดศูนย์กลางของ Camera frame

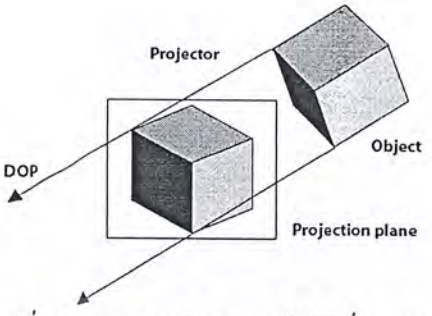


รูปที่ 2-3 แสดง Viewing

ทั้งในความเป็นจริงและใน Computer Graphic นั้นจะอนุญาตให้ viewer อยู่ในระยะทางที่ไม่จำกัดระยะทาง (infinity) จาก object ซึ่งจะเห็นว่าในขณะที่เราเลื่อน COP แบบ infinity ซึ่งทำให้ projector กลายเป็นเส้นขนาน COP จะกลายเป็น direction of projection (DOP) ดังที่แสดงอยู่ในรูปที่ 2-4 โดย View มี 2 แบบ ได้แก่

- **Perspective views** คือ view ที่มีการจำกัดระยะทางของ center of projection
- **Parallel views** คือ view ที่ไม่จำกัดระยะของ center of projection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 แสดง การย้าย COP ไปไว้ที่ Infinity

2.1.2.1 Classical viewing

เมื่อสถาปนิกวาดภาพของตึก เขารู้ว่าด้านไหนที่เขาต้องการที่จะแสดง ดังนั้นที่ที่เขาควรวาง viewer ควรสัมพันธ์กับ ตึก แต่ละ classical view ถูกกำหนดโดยความสัมพันธ์ระหว่าง object และ viewer

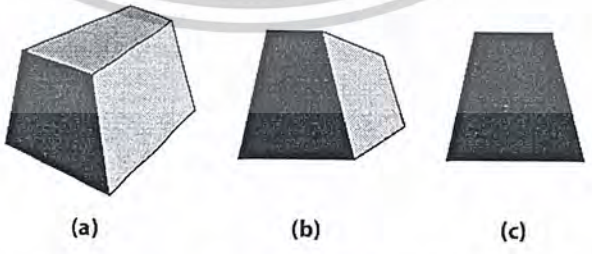
ใน classical view มีความคิดพื้นฐานของ principal face โดยชนิดของวัตถุที่ถูกมองอยู่ใน real-world applications สำหรับ object แบบ สี่เหลี่ยม เช่น ตึกจะมีด้าน front , back , top ,right, left

2.1.2.2 Perspective viewing

Perspective view ทั้งหมดถูกกำหนดคุณสมบัติการลดลงของขนาด เมื่อ object ถูกเลื่อนไกลจาก viewer ซึ่งภาพที่อยู่ไกลจะเล็กลงตามความเป็นจริง อย่างไรก็ตามเราไม่สามารถวัดค่าใดๆจาก Perspective view ได้เนื่องจากจำนวนการสั้นลงของเส้นขึ้นอยู่กับระยะทางที่เส้นนั้นไกลจากกล้อง

ใน classical perspective view ตัว viewer จะถูกวางอย่างสมมาตรซึ่งสัมพันธ์กับ projection plane ดังนั้น การสมมาตรนี้จะทำให้เกิดความสัมพันธ์ระหว่างด้านหลัง และเลนส์ของดวงตาสำหรับการมองของมนุษย์ หรือความสัมพันธ์ระหว่างด้านหลังกับเลนส์ของกล้อง

โดยทั่วไป Classical perspective views จะรู้จักกันในชื่อ one-point perspective, two-point perspective, three-point perspective โดยความแตกต่างระหว่างระหว่าง 3 กรณี อยู่ขึ้นอยู่กับจำนวนของ 3 ทิศทางหลักๆ ใน object ซึ่งขนานไปกับ projection plane ว่ามีจำนวนเท่าใด เส้นขนานทั้ง 3 ทิศทางหลักจะไปบรรจบกันที่จุดหนึ่ง เรียกว่า vanishing points



รูปที่ 2-5รูปแสดง Classical perspective views (a) 3, (b) 2, และ (c) 1-point perspectives

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.3 Viewing กับ Computer

Viewing ในคอมพิวเตอร์กราฟิก อยู่บนพื้นฐานของโมเดล synthetic camera เราสามารถสร้างทุก classical view ได้ อย่างไรก็ตามมีหลักการที่แตกต่างอยู่ คือ ทุก classical view อยู่บนพื้นฐานความสัมพันธ์ระหว่าง object , viewer และ projectors เราเน้นที่ความเป็นอิสระของการกำหนด object และ parameter ของกล้อง

2.1.2.4 Viewport

เป็นการกำหนดว่าจะนำ 3D-scene มาแสดงใน 2-D window อย่างไร view port จะกำหนดพื้นที่สี่เหลี่ยมลงใน Device ที่ Object นั้นจะถูกแสดง(Render)

2.2 ความรู้เกี่ยวกับ Direct X

DirectX เป็น API ของไมโครซอฟท์ที่พัฒนามาเพื่อใช้จัดเตรียมอินเตอร์เฟสสำหรับควบคุมฮาร์ดแวร์มัลติมีเดียบนระบบ Microsoft Windows และเป็นเครื่องมือให้โปรแกรมเมอร์ทำงานกับคำสั่งและโครงสร้างข้อมูลในระดับใกล้ชิดฮาร์ดแวร์ โดยไม่ต้องสร้างโค๊ดติดต่อในระดับล่างซึ่งวิธีติดต่อจะแตกต่างกันไปตามประเภทของอุปกรณ์ การเขียนโค้ดที่เป็นอิสระจากอุปกรณ์ในลักษณะนี้ ช่วยให้โปรแกรมเมอร์สามารถสร้างซอฟต์แวร์เพื่อทำงานดังกล่าวได้เป็นอย่างดี แม้ผู้ใช้จะปรับเปลี่ยนอุปกรณ์ตัวใหม่และเพิ่มการ์ดเร่งความเร็วแบบสามมิติ,เสียง,อุปกรณ์อินพุต และอื่นๆก็ตาม

API ของ DirectX ที่สร้างขึ้นมาบนรากฐานของ HAL(Hardware Abstraction Layer) สามารถซ่อนลักษณะเฉพาะของดีไวท์(Device) ที่เกี่ยวพันอยู่กับฮาร์ดแวร์ และเนื่องจาก DirectX ได้รับการออกแบบให้มีความสามารถในการเสริมขยายได้ในอนาคต ดังนั้นมันจึงสามารถรองรับความสามารถของฮาร์ดแวร์เร่งความเร็วต่างๆมากมายที่ปัจจุบันยังไม่มี ด้วยความสามารถจำลองการทำงานผ่าน HEL (Hardware Emulation Layer) เมื่อมีดีไวท์ที่สามารถนำมาแทนที่ดีไวท์ตัวเก่า

เมื่อใดก็ตามที่สร้างออบเจ็กต์ DirectX ให้กับดีไวท์นั้น DirectX จะเข้าไปถามฮาร์ดแวร์ผ่าน HAL เพื่อดึงเอาข้อมูลเกี่ยวกับดีไวท์ออกมา ข้อมูลที่ดึงออกมานี้มีอยู่ในตาราง Cap Bits (ย่อมาจาก Capability Bits) เป็นข้อมูลที่ใช้บอกความสามารถที่ฮาร์ดแวร์สามารถทำได้

2.2.1 API ของ DirectX

DirectX ประกอบไปด้วย API มากมายที่ได้รับการออกแบบเพื่อใช้พัฒนา และการจำลองแบบ (Simulation) แบบ 3 มิติ DirectX มีไลบรารีที่เก็บฟังก์ชันที่ใช้ในการเรนเดอร์แบบ 2 มิติและ 3 มิติ,สร้างเสียงแบบปกติและแบบ 3 มิติ, คนตรี,ติดต่อแป้นพิมพ์,จอยสติก และอุปกรณ์อินพุตชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุด API ที่มีอยู่ใน DirectX คือ DirectDraw, Direct3D, DirectMusic, DirectSound, DirectPlay, DirectInput และ DirectSetup

ส่วนที่เกี่ยวข้องกับโครงงานคือ ส่วนของ Direct Input ,Direct Draw, และ Direct 3D ซึ่งเราจะศึกษารูปแบบการเขียนโปรแกรมเพื่อใช้สร้างโปรแกรม

2.2.2 DirectInput

เป็น API ของ DirectX ที่สนับสนุนอินพุตความเร็วต่ำ(Low-latency Input)

2.2.3 DirectDraw

เป็นชุด API สำหรับใช้จัดการอุปกรณ์แสดงผล,ควบคุมข้อมูลบิตแมป หน่วยความจำที่ออกนอกพื้นที่สกรีน และสร้างการติดต่อที่รวดเร็วให้กับฟีเจอร์ของฮาร์ดแวร์ เช่น Blitting และ Page Flipping ซึ่งเป็นฟีเจอร์พื้นฐานที่ Direct3D สามารถทำได้

2.2.4 Direct3D

คือ API ที่มีความสามารถเขียนโปรแกรมกราฟิกสามมิติและติดต่อใช้งานฮาร์ดแวร์เร่งความเร็วแบบสามมิติ Direct3D ในยุคแรกมี API อยู่สองโหมดคือ โหมด Immediate (IM) และโหมด Retained (RM) ในโหมดของ IM เป็นหมดที่ใช้งานยากแต่มีความยืดหยุ่นสูง เป็น API ในระดับล่างสำหรับใช้งานได้รวดเร็วและมีประสิทธิภาพเท่าที่จะเป็นไปได้บนระบบ ในโหมด RM เป็นโหมดที่สร้างขึ้นมาเป็น Layer ที่อยู่บนสุดของโหมด IM โดยจะจัดเตรียมบริการต่างๆ เช่นการจัดการ Texture, การโหลด Object File, การจัดลำดับเฟรม และการทำObject เคลื่อนไหว การศึกษาและการใช้งานในโหมด RM นั้นง่ายกว่าเมื่อเทียบกับโหมด IM แต่การทำงานในโหมด RM ได้หยุดลงใน DirectX 6 และมุ่งพัฒนาการทำงานในโหมด IM ให้มีความสามารถและใช้งานง่ายใน version ต่อมา ด้วยเหตุนี้การทำงานในโหมด RM จึงไม่สนับสนุนเทคโนโลยีใหม่ เช่น Multitexturing, Bump Mapping, Hardware Transformation และ Lighting ดังนั้นความสามารถทั้งหมดของโปรแกรมสามมิติ ควรเขียนขึ้นมาด้วยการใช้โหมด IM

2.2.5 การทำงานของ Direct3D

API ทั้งหมดของ DirectX ได้รับการออกแบบมาให้มีความเข้ากันได้ทั้งหมดของ Version ที่ผ่านมา การทำงานร่วมกับฮาร์ดแวร์เร่งความเร็วกราฟิกของแอพพลิเคชั่น Direct3D ทั้งโหมด RM และ IM ทำงานคล้ายคลึงกัน Direct3D จะติดต่อฮาร์ดแวร์ผ่านทาง HAL แต่ถ้าฟีเจอร์นั้นไม่มีในฮาร์ดแวร์ Direct3D จะใช้ HEL แทน ในโมเดลที่จะแสดงให้เห็นต่อไป Direct3D เป็นเหมือนอินเตอร์เฟสที่เชื่อมต่อไปยัง Object DirectDraw ดังนั้น Microsoft จึงอ้างอิงส่วนที่เป็น HAL ด้วย DirectDraw/Direct3D HAL แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 ฟังก์ชั่นของ DirectX ที่นำมาใช้จัดการเกี่ยวกับกล้อง

IDirect3DRMViewport2::SetCamera เป็น method สำหรับกำหนดตำแหน่ง ทิศทาง และการหันเห ของ Viewport ไปให้กับ camera frame.วิวจะถูกหันเหตามแกน z ที่เป็นบวกของcamera frame, กับ up direction ที่เป็นไปในทิศทางของ แกน y ที่เป็นบวก.

2.3 เวอร์เท็กซ์ของ Direct3D กับกระบวนการแปรพิกัดและให้แสง

งานในกระบวนการเรนเดอร์อ็อบเจ็กต์สามมิติแบ่งเป็นสองขั้นตอน ขั้นตอนแรกคือ กระบวนการแปรพิกัดและให้แสง (เรียกย่อว่า T&L หรือ Translation and Lighting) ขั้นตอนนี้เป็นกระบวนการแปรพิกัดของเวอร์เท็กซ์จากพิกัดที่มีรูปแบบทศนิยมไปสู่พิกัดแบบพิเซลตามมุมมองของกล้องจำลองภายในซีน นอกจากนี้เอฟเฟ็กต์ของแสงแบบต่างๆสามารถนำมาใช้ในขั้นตอนนี้ (รวมทั้งกระบวนการขริบ (Clipping) และปรับขนาดวิวพอร์ต(ขอบเขตการมอง) ด้วย) ขั้นตอนที่สองคือ กระบวนการแปรมาสู่พิเซล(Resterization) เป็นขั้นตอนลงจุด เส้น และรูป สามเหลี่ยมด้วยชิ้นงานกราฟิกที่ผ่านกระบวนการ T&L ออกมาแล้ว รูปร่างชิ้นงานกราฟฟิกสุดท้ายที่ผ่านกระบวนการเป็นพิเซลแล้วจะถูกวาดลงพื้นผิวของ DirectDraw ผืนที่เป็นเป้าหมาย ในการแปะ Texture และทำกระบวนการ Interpolate สีระหว่างเวอร์เท็กซ์อยู่ หลังจากนั้นจะนำข้อมูลความลึกใน Depth Buffer เพื่อกำหนดว่าพิกเซลในชิ้นงานสุดท้ายพิกเซลใดจะปรากฏพิกเซลใดจะถูกบังจากพิเซลของชิ้นงานอื่น รูปภาพ 2-6 แสดงงานในแต่ละขั้นตอนการเรนเดอร์ชิ้นงานกราฟิกสามมิติ

รูปที่ 2-6 ขั้นตอนทั้งสองในกระบวนการเรนเดอร์อ็อบเจ็กต์สามมิติ

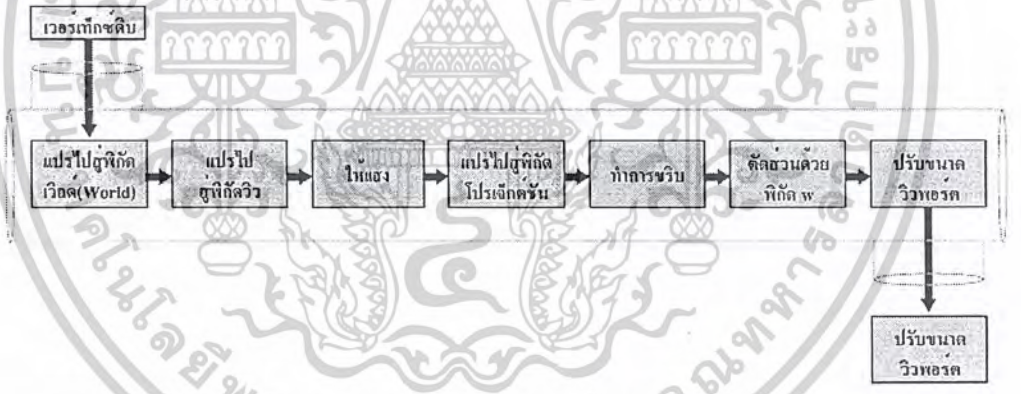
Direct3D มีความสามารถทำงานได้สองขั้นตอน คือขั้นตอน T&L และขั้นตอนแปรเป็นพิกเซลได้ด้วยตนเอง แต่อนุญาตให้คุณเลือกได้ว่าจะให้ Direct3D ทำขั้นตอนสองแยกจากกันหรือว่าทำพร้อมกันด้วยการเรียกใช้ฟังก์ชันในครั้งเดียวกัน และยังอนุญาตให้คุณสามารถสร้างขั้นตอน T&L ของคุณขึ้นมาเอง และนำเวอร์เท็กซ์ที่ผ่านกระบวนการ T&L เรียบร้อยแล้วจัดส่งให้กับกระบวนการแปรเป็นพิกเซลโดยตรง โดยไม่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการ T&L เดิมของ Direct3D ที่มีอยู่ บทนี้จะอธิบายเนื้อหาเกี่ยวกับเวอร์เท็กซ์ของ Direct3D และขั้นตอนการแปรพิกัดและให้แสง

2.4 ภาพรวมของ T&L Pipeline

หากเราจะเปรียบเทียบขั้นตอนการทำงานของกระบวนการแปรพิกัดและให้แสงก็เปรียบได้กับท่อส่งน้ำหรือ Pipeline (หรือคล้ายกับสายงานการประกอบชิ้นส่วนในโรงงาน) การทำงานเริ่มต้นโดยการส่งเวอร์เท็กซ์ดิบเข้าสู่สายการผลิตทางด้านหนึ่ง จากนั้นเวอร์เท็กซ์ดิบดังกล่าวจะผ่านกระบวนการแปรพิกัดและให้แสงจนจบสิ้นกระบวนการเมื่อเวอร์เท็กซ์โผล่ออกมาที่ปลายอีกด้านหนึ่ง กระบวนการทั้งหมดนี้คือ T&L Pipeline ซึ่งแอพพลิเคชันของคุณต้องติดตั้งโดยใช้เมทริกซ์, วิวพอร์ต และแสงจำนวนหนึ่งตามแบบที่คุณต้องการ หลังจากนั้นเมื่อเวอร์เท็กซ์ป้อนเข้าสู่สายงาน มันจะถูกแปรพิกัด , ให้แสงขริบขอบที่ไม่แสดงผล, วางจุดหมายอ้างอิงลงพื้นที่สกรีน และปรับขนาดมุมมองตามวิวพอร์ต เสร็จสิ้นแล้วเวอร์เท็กซ์ที่สำเร็จจากกระบวนการดังกล่าวก็พร้อมที่จะส่งมอบให้กับกระบวนการแปรพิกัดพิเซลต่อไป ภาพต่อไปนี้แสดงกระบวนการภายใน T&L Pipeline



รูปที่ 2-7 กระบวนการ T&L Pipeline

เราสามารถข้ามการทำงานบางงานหรือทุกงานในขั้นตอน T&L Pipeline ที่นำมาแสดงอยู่ที่นี่ได้ใครก็ตามที่มีกระบวนการแปรพิกัดและให้แสงของตนเอง สามารถยกเลิกการทำงานในขั้นตอน T&L Pipeline จากนั้นก็ส่งเวอร์เท็กซ์ที่ผ่านกระบวนการของตนเองไปให้กระบวนการแปรเป็นพิกเซลของ Direct3D ได้แต่ทางเลือกที่นิยมส่วนใหญ่และเป็นวิธีที่ดีที่สุดคือใช้กระบวนการทั้งหมดที่มีอยู่ใน T&L Pipeline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

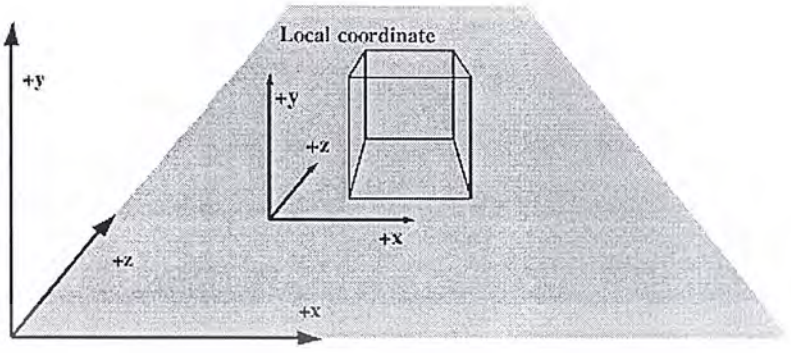
2.4.1 การแปรไปสู่พิกัดเวิลด์

ในระหว่างกระบวนการแปรพิกัดและให้แสง พิกัดของอ็อปเจ็กต์ทั้งหมดที่นำมาเรนเดอร์ต้องแปรให้มาอยู่ในระบบพิกัดเดียวกันนั่นคือแปรให้มาอยู่ใน พิกัดเวิลด์(World Transformation) แต่เพื่อความสะดวกแก่โปรแกรมของคุณ อ็อบเจ็กต์ทุกตัวจึงมีพิกัดพิเศษของตนเองระบบพิกัดนี้เรียกว่าพิกัดท้องถิ่น(Local Space Coordinate) ดังนั้นก่อนการเรนเดอร์คุณจึงต้องแปรพิกัดก่อนโดยใช้เมทริกซ์ที่มีชื่อว่า เมทริกซ์ World (World Transformation Matrix) การใช้พิกัดท้องถิ่นช่วยคุณทำงานได้ง่ายกว่าเดิมมาก เช่นการเคลื่อนย้ายอ็อบเจ็กต์โดยการแปรอ็อบเจ็กต์จากพิกัดท้องถิ่นไปสู่พิกัดเวิลด์ทำได้ง่ายและเร็วกว่าการแปรพิกัดทั้งหมดของอ็อบเจ็กต์เข้าสู่พิกัดเวิลด์ในคราวเดียว นอกจากนี้ยังอนุญาตให้คุณสร้างอินสแตนซ์ได้ด้วย นั่นคือคุณสามารถวาดอ็อบเจ็กต์ เช่นรูปทรงกลมไว้ที่ตำแหน่งใดตำแหน่งหนึ่ง และยังสามารถวาดรูปทรงกลมนี้ได้อีกในตำแหน่งอื่นเพียงแต่จะมีพิกัดของตนเองไม่ใช่พิกัดเดียวกัน ดังนั้นกระบวนการแปรจากพิกัดท้องถิ่นไปสู่พิกัดเวิลด์จึงต่างกัน การแปรเปลี่ยนอ็อบเจ็กต์ภายในพิกัดท้องถิ่นทำได้ง่ายดาย อย่างเป็นธรรมชาติ เช่นการหมุนทรงกลมรอบจุดศูนย์กลางในพิกัดท้องถิ่นจะสามารถทำได้ง่ายดาย โดยไม่ต้องคำนึงพิกัดจริงในพิกัดเวิลด์

ขั้นตอนแรกในกระบวนการ T&L Pipeline คือใช้เมทริกซ์ World ซึ่งเป็นเทริกซ์สำหรับใช้แปรพิกัดตำแหน่งของเวอร์เท็กซ์ในอ็อบเจ็กต์จากพิกัดท้องถิ่นไปสู่พิกัดเวิลด์ นอกจากนี้ยังนำมาใช้ในการหมุน (หมุนอ็อบเจ็กต์รอบแกน x ,แกน y หรือแกน z), การเคลื่อนย้าย (ย้ายอ็อบเจ็กต์ตามแกน x, แกน y หรือแกน z) และใช้ปรับขนาดร่วมกันอีกด้วย รูป 2-8 แสดงความสัมพันธ์ของระบบกับเวิลด์และระบบท้องถิ่น

หน้าที่สำคัญที่สุดของระบบพิกัดเวิลด์คือการจัดเตรียมระบบพิกัดสามัญที่สามารถใช้ร่วมกันได้กับอ็อบเจ็กต์สามมิติที่ต้องการใช้ระบบพิกัดเฉพาะของตนเอง หากมีพิกัดของเวอร์เท็กซ์ใดที่ถูกกำหนดในพิกัดเวิลด์แล้ว Direct3D ก็จะจดจำระบบพิกัดท้องถิ่น หรือแม้แต่กระบวนการแปรพิกัดจากท้องถิ่นสู่เวิลด์อีกต่อไป กระบวนการแปรระบบพิกัดนี้ถ้าจะเปรียบไปแล้วก็เหมือนกับการเปลี่ยนหน่วยวัดให้เป็นมาตรฐานเดียวกัน เช่นการเปลี่ยนหน่อยปอนด์, กิโลกรัม และตัน ให้เป็นหน่วยกรัมเช่นเดียวกัน แต่ถ้าคุณไม่ต้องการใช้ระบบพิกัดท้องถิ่นแต่ละพิกัดของตนเองแยกจากกัน โดยกำหนดพิกัดของเวอร์เท็กซ์ให้อยู่ในพิกัดเวิลด์เสียแต่เริ่มแรก เมทริกซ์ World ของคุณจะต้องเป็นเมทริกซ์เอกลักษณ์ เพื่อการกำหนดว่าพิกัดท้องถิ่นเป็นพิกัดเวิลด์อยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 การแปรไปสู่พิกัดเวิลด์(ระบบพิกัดเวิลด์ และ Local)

2.4.2 การแปรไปสู่พิกัดวิว

ระบบเวิลด์ไม่มีตัวตนให้คุณได้สัมผัสได้ ไม่ว่าจะเป็นตำแหน่งของจุดกำเนิดและแกนภายในพื้นที่ของระบบพิกัด ล้วนถูกควบคุมตามความต้องการของโปรแกรมเมอร์โดยไม่ได้เกี่ยวข้องกับ Direct3D เลย งานที่สองของขั้นตอน T&L Pipeline เป็นงานแปรพิกัดเวิลด์ไปสู่พิกัดที่มองออกมาจากกล้อง(พิกัดมุมมองกล้อง) กล้องที่ว่านี้คือกล้องจำลองซึ่งอยู่ ณ ตำแหน่งจุดกำเนิดและสามารถให้อยู่ตามแนวแกน z เรื่อยลงมาตามแนวแกนด้านบวกได้ นอกจากนี้ในขั้นตอนการให้แสงทั้งหมดก็จะถูกแปรเข้ามาระบบพิกัดมุมมองกล้องด้วย รูปภาพ 2-9 จะแสดงสิ่งได้อธิบายนี้ให้เข้าใจง่ายขึ้น

รูปที่ 2-9 การแปรไปสู่พิกัดวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การให้แสง

ในขั้นตอนนี้ เอฟเฟ็กต์ของแสงแต่ละแบบจะถูกคำนวณผลที่มีต่อเวอร์เท็กซ์ทั้งหมด โดยโค้ดในกระบวนของการให้แสงจะพิจารณาเวกเตอร์ normal (เวกเตอร์ตั้งฉากที่ออกมาจากเวอร์เท็กซ์ในโพลีกอน), สี และคุณสมบัติของวัสดุที่แปะอยู่บนเวอร์เท็กซ์ ซึ่งเป็นปัจจัยร่วมกับคุณสมบัติของแสงสำหรับคำนวณผลของเอฟเฟ็กต์ที่มีต่อเวอร์เท็กซ์ หลังจากนั้นงานนี้จะเสร็จสิ้นเมื่อจัดสีของเวอร์เท็กซ์ หลังจากนั้นงานนี้จะเสร็จสิ้นเมื่อจัดการเก็บสีของเวอร์เท็กซ์แต่ละตัวที่สำเร็จกระบวนการเหล่านี้แล้วลงบนโครงสร้างข้อมูลของตนเอง ในงานต่อไปของขั้นตอน T&L นี้ Direct3D จะไม่เข้าไปยุ่งเกี่ยวกับแสงและวัสดุของเวอร์เท็กซ์อีก

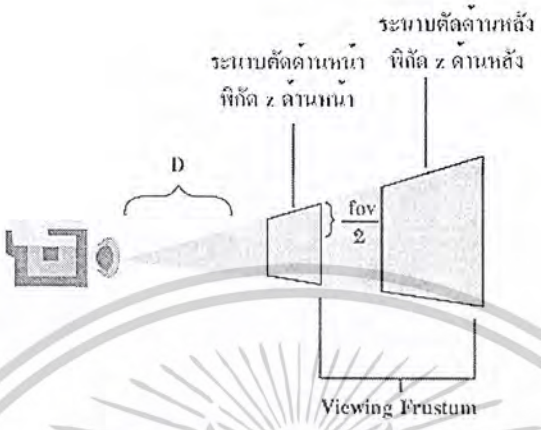
2.4.4 การแปรไปสู่พิกัดโปรเจ็กต์ชันและ Viewing Frustum

งานต่อไปเป็นขั้นตอน T&L Pipeline จะทำการปรับขนาดอ็อบเจ็กต์ในซีนตามความห่างจากจุดมอง ซึ่งเราเรียกว่า การแปรไปสู่โปรเจ็กต์ชัน (Projection Transformation) กระบวนการนี้ทำให้อ็อบเจ็กต์ปรากฏตามความลึกภายในซีน โดยการทำให้อ็อบเจ็กต์ที่อยู่ห่างไกลมีขนาดเล็กว่าอ็อบเจ็กต์ที่อยู่ใกล้จุดมองมากกว่า หลังจากเสร็จสิ้นกระบวนการแปรนี้แล้ว เวอร์เท็กซ์ทั้งหมดจะถูกแปรมาอยู่ในพิกัด พิกัดโปรเจ็กต์ชัน

การที่เราทราบว่ากระบวนการแปรไปสู่โปรเจ็กต์ชันมีกระบวนการอย่างไร จะช่วยให้เข้าใจเกี่ยวกับสิ่งที่เรียกว่า Viewing Frustum ได้โดยง่าย ในศาสตร์แห่งการคำนวณแล้วสิ่งที่เรียกว่า Frustum คือเรขาคณิตของการเคลื่อนย้ายจุดยอดของปิรามิด แต่สำหรับในคอมพิวเตอร์แล้ว Viewing Frustum เป็นการก่อรูปปิรามิดทางแนวความคิดโดยวางจุดยอดลงบนจุดมุมมองของกล้องเมื่อเคลื่อนจุดมุมกล้องจากจุดยอดเข้าไปในใจกลางปิรามิด "กำแพง" ทั้ง 4 ด้านของปิรามิดจะเป็นด้านทั้งสี่ของสกรีนและเปิดได้เห็นด้านหน้า(ส่วนยอดปิรามิด) และด้านหลัง(ส่วนฐานของปิรามิด) ตรงตำแหน่งของระนาบตัดด้านหน้าและระนาบด้านหลัง ปริมาตรที่อยู่ระหว่างระนาบตัดทั้งสองด้านกับกำแพงทั้งสี่ด้านนี้เองคือ Viewing Frustrum ซึ่งเป็นปริมาตรแสดงวิวที่เรนเดอร์แล้วในระบบพิกัดมุมมองกล้อง แม้ว่าคุณจะสามารถใช้โปรเจกต์ได้หลายชนิดในการอ้างอิงโมเดลสามมิติจากพิกัดมุมมองกล้องให้มาอยู่บนสกรีน แต่ในที่นี้ขออธิบายชนิดที่เป็นสามัญและนิยมใช้กันแพร่หลายคือ โปรเจ็กต์ชันแบบ Perspective ซึ่งมีลักษณะที่ว่าอ็อบเจ็กต์ที่อยู่ห่างไกลจากกล้องจะมีขนาดเล็กกว่าอ็อบเจ็กต์ที่อยู่ใกล้ชิดกล้องมากกว่า และอีกชนิดหนึ่งคือ โปรเจ็กต์ชันแบบ Orthogonal ชนิดหลังนี้จะไม่ปรับขนาดอ็อบเจ็กต์ ซึ่งมีแอพพลิเคชันบางตัวเท่านั้นที่ใช้โปรเจ็กต์ชันชนิดนี้ แต่เป็นไปได้ที่คุณจะมีโอกาสได้ใช้ในการปรับขนาดอ็อบเจ็กต์ตามระยะทางมุมมองของผู้เล่นภายในเกม

การแปรไปสู่โปรเจ็กต์ชันแบบ Perspective จะเปลี่ยน Viewing Frustum มาเป็น Cuboid ซึ่งมีลักษณะคล้ายลูกบาศก์ แต่จะมีด้านไม่เท่ากัน เนื่องจากด้านที่ใกล้กว่าจะมีขนาดเล็กกว่าด้านที่ไกลกว่า ส่งผลให้อ็อบเจ็กต์ที่อยู่ระหว่างใกล้กว่ามีขนาดใหญ่กว่าอ็อบเจ็กต์ที่อยู่ระยะไกลซึ่งทำให้การสร้างมุมมองไปด้านภายในซีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-10 แสดงองค์ประกอบของ Viewing Frustum

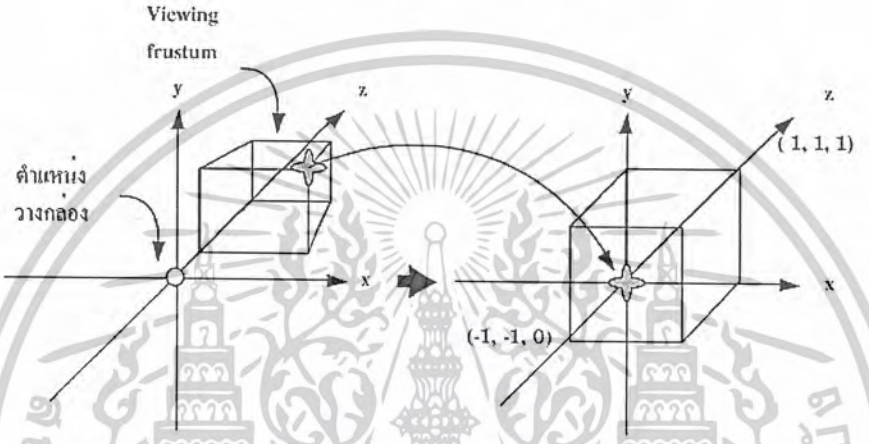
ในรูปภาพที่ 2-10 คุณจะเห็นระนาบตัดด้านหน้าและด้านหลังแสดงอยู่ ระนาบตัดทั้งสองด้านใช้กำหนดสิ่งที่จะปรากฏแก่ผู้มองเมื่อเราเรนเดอร์ซีน โดยระนาบด้านหน้าเป็นตัวกำหนดระยะทางใกล้สุดที่อ็อบเจ็กต์จะถูกรวบเข้ามาเรนเดอร์ในซีน และระนาบด้านหลังจะเป็นระยะทางไกลสุด อ็อบเจ็กต์ใดๆที่อยู่นอกพื้นที่ Viewing Frustum จะไม่นำมาเรนเดอร์ ดังนั้นระนาบตัดทั้งสองจึงเป็นสิ่งจำเป็นเพราะเรานำมาใช้กำหนดค่าน้อยที่สุดและค่ามากที่สุดให้แก่ Z-Buffer ซึ่งหากไม่มีระนาบในระยะไกลแล้วอุปกรณ์เรนเดอร์จะไม่สามารถรู้ค่าที่จะแมปเข้ากับค่า z ค่าสูงที่สุดจำไว้ว่าระยะห่างที่คุณกำหนดให้กับระยะไกล(ระนาบตัดด้านหลัง) จะเป็นสิ่งที่มีผลต่อความเร็วและคุณภาพของซีน ถ้าคุณกำหนดระยะห่างไว้ใกล้เกินไป คุณจะพบอาการสะดุด ทำให้อ็อบเจ็กต์ดังกล่าวกำลังเคลื่อนที่เข้ามาหาจากระยะไกล ซึ่งคุณจะเห็นเช่นนี้ก็ต่อเมื่ออ็อบเจ็กต์ระยะไกลมองดูเล็กจริง และคุณไม่สามารถสังเกตความเปลี่ยนแปลงได้ แต่ถ้าคุณกำหนดระยะไกลมากเกินไป จะทำให้เกิดการขริบอ็อบเจ็กต์ส่วนเกินออกไปอย่างมีประสิทธิภาพ ซึ่งนั่นก็เป็นสาเหตุที่เรนเดอร์ใช้เวลานานเกินไปเช่นกัน

เราสามารถอธิบาย Viewing Frustum เพิ่มเติมได้ว่าเป็นพื้นที่ที่เรามองเห็น (fov : field of view) ซึ่งเป็นพื้นที่ที่เป็นมุมอันเกิดจากมุมมองที่มองออกจากกล้องเข้าหาระนาบที่ใกล้ที่สุด และเกิดขึ้นจากระยะทางตั้งแต่จุดมองไปจนถึงระนาบตัดด้านหน้าและระนาบตัดด้านหลัง ระยะทางดังกล่าวนั้นคือพิกัด z ของระนาบตัดด้านหน้าและด้านหลัง ตัวแปร D คือระยะทางจากกล้องเข้าหาระนาบตัดด้านหน้า (จุดกำเนิดของระบบพิกัดที่กำหนดขึ้นจากกระบวนการแปรไปสู่วิว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปภาพที่ 2-11 แสดงกระบวนการแปรไปสู่โปรเจ็กต์ชันที่ใช้ในการแปร Viewing Frustum ไปเป็นระบบพิกัดใหม่ เห็นได้ว่า Frustum กลายรูปเป็น Cuboid ที่เป็นเช่นนี้เพราะโปรเจ็กต์ชันที่นำมาใช้คือชนิด Perspective หลังจากกระบวนการแปรนี้เสร็จสิ้นลง ขอบเขตของด้านทางแกน x จะมีค่าเท่ากับ -1 สำหรับระนาบด้านซ้ายและ 1 สำหรับระนาบด้านขวา, ขอบเขตของด้านทางแกน y จะมีค่าเท่ากับ -1 สำหรับระนาบด้านล่างและ 1 สำหรับด้านล่างและ 1 สำหรับระนาบด้านบน และขอบเขตของด้านทางแกน z จะมีค่าเท่ากับ 0 สำหรับระนาบด้านหน้าและ 1 สำหรับระนาบด้านหลัง



รูปที่ 2-11 โปรเจ็กต์ชันแบบ Perspective

2.4.5 การขริบ

งานนี้เป็น การขริบ (Clipping) ซึ่งเป็นขั้นตอนที่จะสร้างความมั่นใจว่าอ็อปเจ็กต์ที่อยู่นอกขอบเขตของ Viewing Frustum จะไม่ถูกเรนเดอร์ และอ็อบเจ็กต์ที่อยู่ภายใน Viewing Frustum จะถูกวาดขึ้นโดยไม่มีพิเซลใดเล็ดลอดออกจากพื้นที่สี่เหลี่ยมที่กำหนดขึ้นโดยวิวพอร์ต(ขอบเขตการมอง) การขริบนี้เป็นงานหนึ่งในขั้นตอน T&L Pipeline ซึ่งต้องการข้อมูลของชิ้นงานกราฟฟิกที่ต่อเชื่อมกับเวอร์เท็กซ์ เช่น ถ้ามีเวอร์เท็กซ์จุดหนึ่งของรูปสามเหลี่ยมหลุดลอดขอบเขตของ Viewing Frustum ออกมา โค้ดที่ใช้ในการขริบจะต้องสามารถกำหนดจุด 2 จุด ซึ่งเป็นขอบของรูปสามเหลี่ยมที่ Intersect (การนำเฉพาะค่าที่ซ้ำกันมาใช้) อยู่กับ Viewing Frustum และตัดแบ่งรูปสามเหลี่ยมออกเป็นสามเหลี่ยมสองรูป เพราะว่าเส้นร่างรูปสามเหลี่ยมที่ถูกตัดขณะนี้มีด้านอยู่ 4 ด้าน ดังนั้นถ้าหากมีข้อมูลชิ้นส่วนอิมเมจป้อนให้แก่ T&L Pipeline การขริบก็จะสามารถกระทำได้โดยสมบูรณ์ ด้วยเหตุนี้ถ้า T&L Pipeline แปรพิกัดและให้แสงเวอร์เท็กซ์ใน Vetex Buffer ด้วยการเรียกใช้ Process Vertices จะทำให้ T&L Pipeline สามารถระบุและบันทึกเวอร์เท็กซ์ภายนอกระบบของ Viewing Frustum จากนั้นก็จะนำข้อมูลนี้มาใช้ในภายหลังจากการเรียกใช้ฟังชัน DrawPrimitive เพื่อใช้ขริบชิ้นส่วนอิมเมจร่วมกับการใช้ Vertex Buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61372

2.4.6 การตัดหารด้วยพิกัด w หรือกระบวนการทำ Nonhomogeneous

งานต่อไปนี้เป็นงานที่ Direct3D จะทำการแปรเวอร์เท็กซ์ที่อยู่ในรูปแบบ Homogeneous ไปเป็น Nonhomogeneous เสียก่อน เมื่อคุณกำหนดรายละเอียดของเวอร์เท็กซ์ คุณจะต้องกำหนดพิกัด x, y และ z ให้กับเวอร์เท็กซ์แต่ละจุด แต่ว่าในกระบวนการแปรพิกัดและให้แสงนั้นจะมีพิกัดหนึ่งเพิ่มเติมขึ้นมา พิกัดนี้เรียกว่าพิกัด w ซึ่งกำหนดค่าเริ่มต้นเป็น 1.0 เมื่อป้อนเวอร์เท็กซ์เข้าสู่กระบวนการแปรมันจะถูกแปรด้วยเมทริกซ์ต่างๆ ซึ่งผลลัพธ์จะถูกเปลี่ยนแปลงโดยพิกัด w เราเรียกเวอร์เท็กซ์ที่รวมกับค่า w อื่นที่ไม่ใช่ 1.0 ว่าเป็นเวอร์เท็กซ์แบบ Homogeneous หลังจากนั้นเมื่อกระบวนการแปรเสร็จสิ้นแล้วเวอร์เท็กซ์จะถูกแปรกลับมาอยู่ในรูปแบบ Nonhomogeneous ดังเดิม โดยการตัดหารพิกัด x, y และ z หรือที่จะเรียกว่าอยู่ในรูปแบบ RHW(Reciprocal-of-homogeneous-w)ได้เท่านั้น

2.4.7 การปรับขนาดวิวพอร์ต

งานนี้เป็นงานสุดท้ายในขั้นตอน T&L Pipeline เป็นงานปรับเวอร์เท็กซ์ให้มีขนาดเหมาะสมกับวิวพอร์ต (ขอบเขตการมอง) ที่คุณต้องการ วิวพอร์ตอนุญาตให้คุณสามารถกำหนดวิธีที่ใช้ในการแมปภาพอิมเมจลงพื้นผิวเป้าหมายได้ โดยสามารถกำหนดให้มีการโยกย้ายและปรับขนาด แต่โดยทั่วๆไปแล้วมักจะเป็นการเติมภาพอิมเมจลงบนสิ่งที่คุณต้องการเรนเดอร์ ดังนั้นจึงกำหนดให้ไม่มีการโยกย้ายใดๆ จากนั้นก็ปรับขนาดพิกัดขนาดของเวอร์เท็กซ์ให้ค่า -1 ของพิกัด x แมปอยู่กับขอบด้านซ้ายและค่า 1 แมปอยู่กับขอบด้านขวา, ค่า -1 ของพิกัด y แมปอยู่กับขอบด้านบนสุดและค่า 1 แมปอยู่กับขอบด้านล่างสุด, ค่า -1 ของพิกัด z แมปอยู่กับขอบด้านหน้าและค่า 1 แมปอยู่กับขอบด้านหลัง แต่ว่าคุณสามารถปรับขนาดให้พิกัด z ได้ถ้าต้องการเรนเดอร์ให้มีระยะความลึกคุณสามารถกำหนดใช้ค่าเต็มช่วงในพิกัดนี้ได้จากการตั้งค่าวิวพอร์ต โดยใช้ค่า z น้อยที่สุดเท่ากับ 0.0 และมากกว่าเท่ากับ 1.0

2.4.8 เวอร์เท็กซ์

ชิ้นงานกราฟฟิกทุกชิ้นซึ่ง DirectX สามารถเรนเดอร์ขึ้นรูปมาจากเวอร์เท็กซ์ ในทางด้านเรขาคณิตเวอร์เท็กซ์คือจุดในพื้นที่ว่างซึ่งกำหนดมุมของโพลีกอน แต่ใน Direct3D มันคือจุดที่ยึดติดกับพิกัดของตำแหน่งที่อยู่ของตัวเองและสามารถจักเก็บข้อมูลอย่างเช่น เวกเตอร์ normal , พิกัดของ Texture และข้อมูลสี เป็นต้น Direct3D จะเป็นผู้แปรเวกเท็กซ์จากรูปแบบที่กำหนดโดยผู้ใช้ให้ไปอยู่ในรูปแบบที่พร้อมที่จะป้อนเข้ากระบวนการแปรพิเซลซึ่ง DirectX3 มีรูปแบบเวอร์เท็กซ์ชนิด Flexivle ซึ่งคุณสามารถกำหนดได้หลายวิธี ก่อนที่เราจะเข้าเนื้อหานี้ เราจะมาดูชนิดของเวอร์เท็กซ์ที่นำมาแสดงในรายการต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8.1 D3DVERTEX (ไม่แปรพิกัดและไม่ให้แสงเวอร์เท็กซ์)

หากไม่ต้องการแปรพิกัดและให้แสงเวอร์เท็กซ์ ก่อนส่งผ่านให้ Direct3D ให้กำหนดใช้ชนิดเวอร์เท็กซ์ชนิดนี้ในการใช้งานคุณต้องประกาศพารามิเตอร์ที่ใช้ในการให้แสง และเมทริกซ์ที่ใช้ในการแปรพิกัดสำหรับใช้กับเวอร์เท็กซ์ชนิดนี้ โดย Direct3D ยังคงคำนวณการให้แสงและแปรพิกัดเวอร์เท็กซ์ให้แก่คุณ

2.4.8.2 D3DLVERTEX (ไม่แปรพิกัดแต่ให้แสงเวอร์เท็กซ์)

หากต้องการให้แสงแต่ไม่ต้องการแปรพิกัดเวกเตอร์ของคุณก่อนส่งผ่านให้ Direct3D หรือหากต้องการสร้างการคำนวณให้แสงของคุณขึ้นเอง ควรกำหนดใช้เวอร์เท็กซ์ชนิดนี้

2.4.8.3 D3DTLVERTEX (แปรพิกัดและให้แสงเวอร์เท็กซ์)

หากต้องการแปรพิกัดและให้แสงเวอร์เท็กซ์ก่อนส่งผ่านให้ Direct3D ให้กำหนดใช้เวอร์เท็กซ์ชนิดนี้ กระบวนการทำงานภายใน T&L Pipeline จะข้ามขั้นตอนการแปรพิกัดและให้แสง แต่จะไม่ข้ามขั้นตอนการขริบหากไม่จำเป็น คุณสามารถบอก Direct3D ได้หากต้องการจะทำกระบวนการขริบเอง

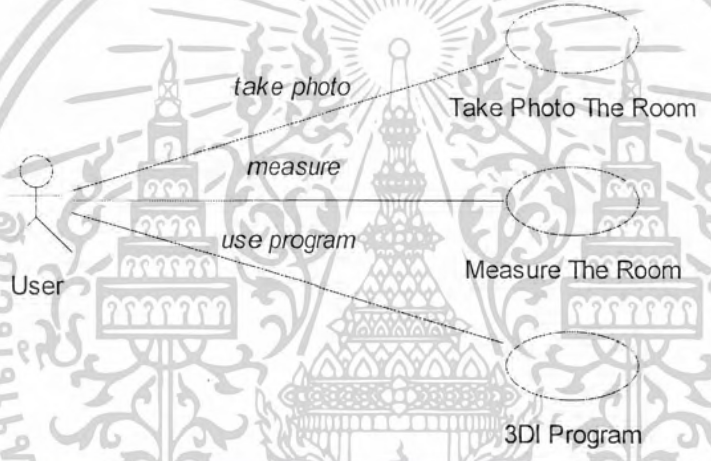
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 Use-Case Diagram

เป็นการสร้าง Use Case ของระบบขึ้นมาเพื่อช่วยให้เห็นภาพรวมของระบบว่าจะต้องทำอะไรในส่วนใดได้บ้างและหาฟังก์ชั่นการทำงานเบื้องต้นของระบบ

3.1.1 Use case :: System

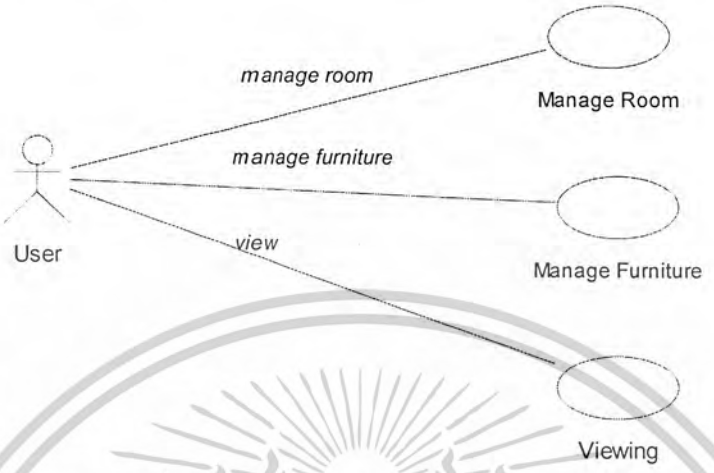


รูปที่ 3-1 แสดง Use-Case Diagram โดยรวมของระบบ

จากรูป 3-1 แสดง Use-Case Diagram ของระบบโดยรวม โดยก่อนเริ่มต้นการสร้างห้องสามมิตินั้น User จะต้องทำการวัดขนาดของห้องจริงที่ต้องการสร้าง ซึ่งประกอบด้วย ความกว้าง ความยาว ความสูง ของห้อง นอกจากค่าทั้งสามค่านี้แล้ว ถ้า User ต้องการให้โปรแกรมแสดงพื้นผิวของห้อง User ก็จะต้องทำการถ่ายภาพของมุมต่างๆของห้องที่ต้องการดูประกอบในการจัดวางเฟอร์นิเจอร์ เพื่อที่จะได้เห็นภาพที่ใกล้เคียงกับห้องจริง และUser จะต้องวัดค่าตำแหน่งของกล้องในการถ่ายภาพจากมุมต่างๆด้วย โดยค่าทั้งหมดนี้ User จะเป็นผู้นำไปป้อนเป็นค่าอินพุตให้กับ 3DI Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

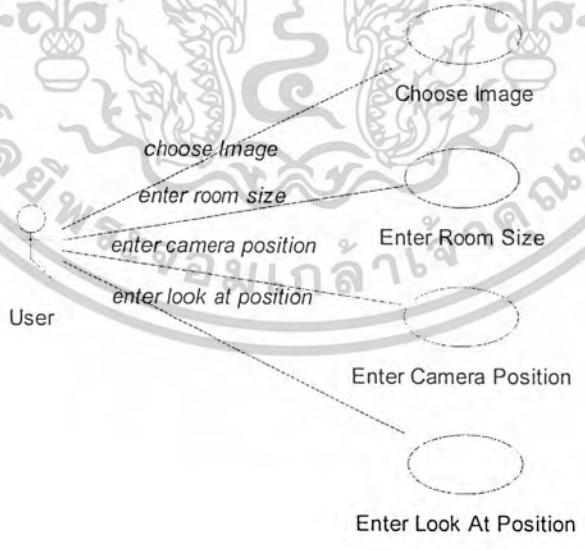
3.1.2 Use case :: 3DI Program



รูปที่ 3-2 แสดง Use-Case Diagram โดยรวมของโปรแกรม

จากรูปที่ 3-2 แสดง Use-Case Diagram โดยรวมของระบบ ซึ่งอธิบายว่า User สามารถใช้งานโปรแกรมได้อย่างไรบ้าง โดยหลังจากที่ User เสร็จสิ้นขั้นตอนในการวัดขนาดห้องแล้ว User ก็จะสามารถทำการควบคุมการสร้างห้อง และสามารถนำเฟอร์นิเจอร์ที่ต้องการมาแสดงและควบคุมการจัดวางเฟอร์นิเจอร์ในโปรแกรมได้นอกจากนั้น User ยังมีความสามารถในการควบคุมมุมมองภายในโปรแกรมได้อีกด้วย

3.1.3 Use Case :: Create Room

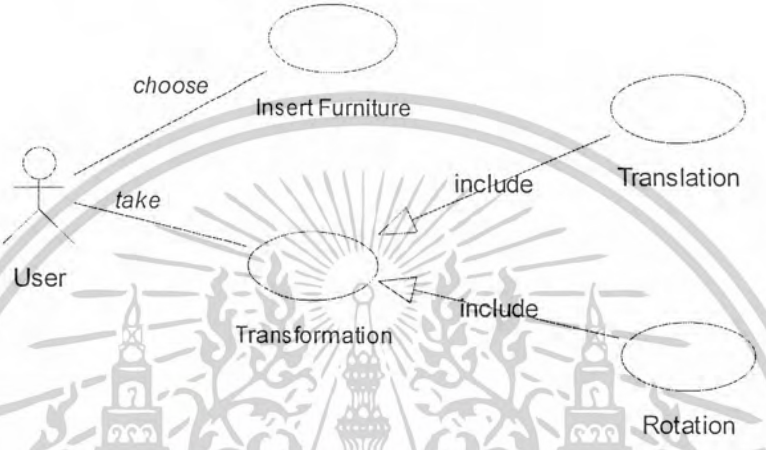


รูปที่ 3-3 แสดง Use-Case Diagram ในส่วนการ Create Room

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-3 แสดง Use-Case Diagram ในส่วนของการสร้างห้องสามมิติ โดย User สามารถป้อนค่าพารามิเตอร์คือค่าขนาดของห้อง และสามารถเลือกรูปที่ต้องการแสดงในโปรแกรม และ User ยังสามารถป้อนค่าตำแหน่งกล้อง และจุด Look At Point เพื่อใช้ในการแสดงภาพในแต่ละมุม

3.1.4 Use Case :: Furniture Manipulation.

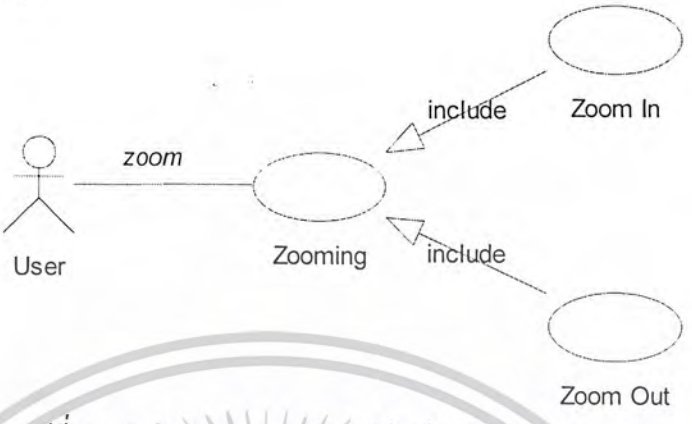


รูปที่ 3-4 แสดง Use-Case Diagram ในส่วนการ Furniture Manipulation

รูปที่ 3-4 แสดง Use-Case Diagram ในส่วนของ Furniture Manipulation โดย User สามารถเพิ่มเฟอร์นิเจอร์ที่ต้องการเข้าไปจัดวางภายในโปรแกรมได้ โดย User สามารถทำการเลื่อนหรือเคลื่อนย้ายเฟอร์นิเจอร์ไปยังตำแหน่งที่ต้องการภายในห้องสามมิติที่สร้างขึ้นได้ นอกจากนั้น User ยังสามารถหมุนเฟอร์นิเจอร์ไปในทิศทางตามที่ต้องการเพื่อใช้ในการจัดวางเฟอร์นิเจอร์ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

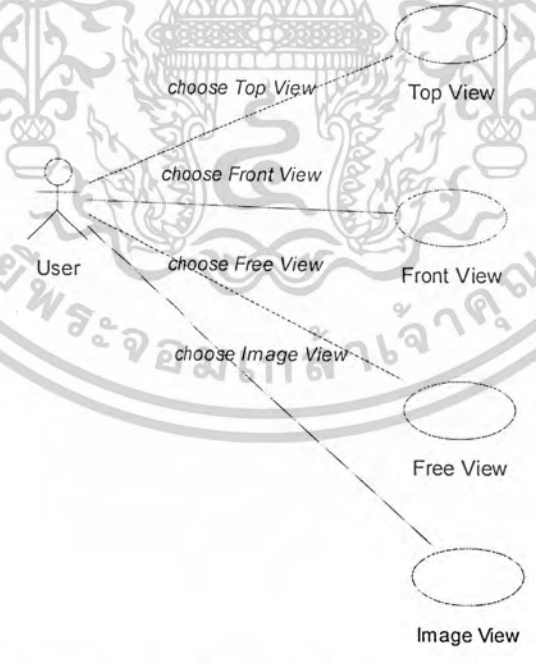
3.1.5 Use Case :: Viewing



รูปที่ 3-5 แสดง Use-Case Diagram ในส่วนการ Viewing

จากรูปที่ 3-5 แสดง Use-Case ในส่วนของการ Viewing ในการจัดวางเฟอร์นิเจอร์นั้นระหว่างที่ User กำลังจัดวางเฟอร์นิเจอร์หรือหลังจากจัดวางเฟอร์นิเจอร์เรียบร้อยแล้ว User สามารถที่จะขยายมุมมองการดูการจัดวางเฟอร์นิเจอร์ภายในห้องให้ใกล้ หรือ ไกลได้ เพื่อให้สามารถมองเห็นทัศนียภาพภายในห้องและช่วยให้เพิ่มประสิทธิภาพในการจัดวางเฟอร์นิเจอร์ได้อีกด้วย

3.1.6 Use Case :: Viewing Manipulation.



รูปที่ 3-6 แสดง Use-Case Diagram ในส่วนการ Viewing Manipulation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

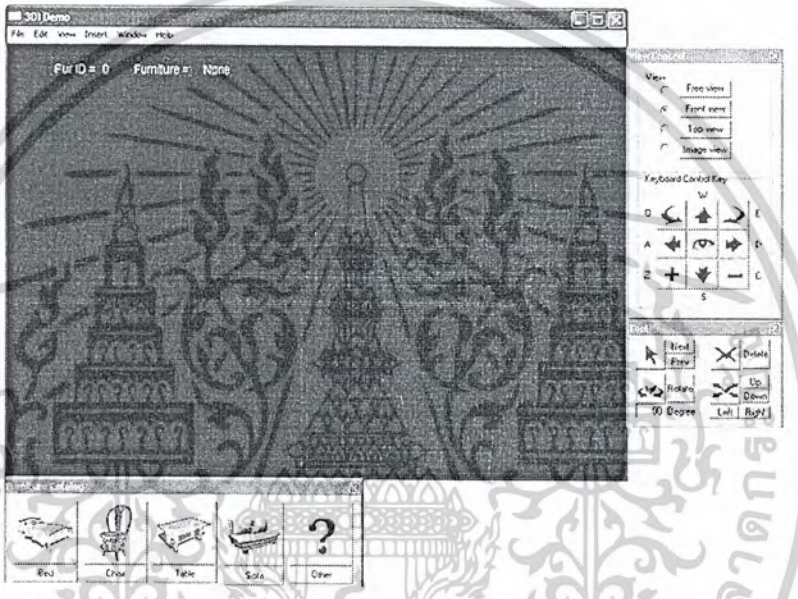
จากรูปที่ 3-6 แสดง Use-Case Diagram :: Viewing Manipulation ซึ่งมุมมองที่โปรแกรมรองรับได้นั้นประกอบด้วย

- **Top View** ซึ่งเป็นมุมมองจากทางด้านบนของห้องซึ่งมุมมองนี้ทำให้ผู้ใช้สะดวกในการจัดวางเฟอร์นิเจอร์
- **Front View** ซึ่งเป็นมุมมองจากทางด้านหน้าของห้องโดยเทียบจากตำแหน่งมุมมองของกล้อง
- **Free View** ซึ่งเป็นมุมมองอิสระ ที่ User สามารถควบคุมได้
- **Image View** ซึ่งเป็นมุมมองจากภาพถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 User Interface

ในหัวข้อนี้จะอธิบาย เครื่องมือ(Tool) ภายในโปรแกรม โดยจากรูปที่ 3-7 เป็นภาพแสดงหน้าจอทั้งหมดของโปรแกรม 3DI โดยจะประกอบด้วย หน้าจอหลัก และ เครื่องมือที่ใช้ควบคุมการใช้งานโปรแกรม โดยในส่วนของเครื่องมือนี้จะอยู่ในส่วนของทางด้านขวาของหน้าจอหลัก และทางด้านล่างของหน้าจอหลัก โดยอัตโนมัติเมื่อผู้ใช้ทำการเปิดโปรแกรม 3DI ขึ้นมาใช้ครั้งแรก และนอกจากนั้น ผู้ใช้สามารถทำการเคลื่อนย้าย เครื่องมือไปยังตำแหน่งที่ผู้ใช้ต้องการได้อีกด้วย โดยผู้เขียนจะขออธิบายการใช้งานเครื่องมือเป็นส่วนๆในหัวข้อต่อไป

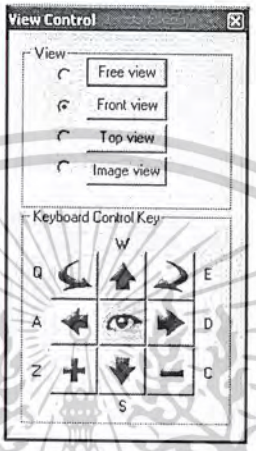


รูปที่ 3-7 แสดงหน้าจอทั้งหมดของโปรแกรม 3DI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 View Control

ในส่วนนี้จะเป็นเครื่องมือที่ใช้ในการควบคุมมุมมองของโปรแกรมเพื่อช่วยอำนวยความสะดวกในการจัดวางเฟอร์นิเจอร์ ดังรูปที่ 3-8 แสดงส่วนของ View Control ซึ่งประกอบด้วย 2 ส่วนคือ View และ Keyboard Control Key



รูปที่ 3-8 แสดงส่วนของ View Control

3.2.1.1 View เป็นเครื่องมือที่ใช้ในการเลือกมุมของการใช้งานภายในโปรแกรมโดยในส่วนของ view ประกอบด้วย 4 ส่วนดังนี้

- **Free View** เป็นมุมมองอิสระที่อนุญาตผู้ใช้สามารถดูมุมมองต่างๆภายในโปรแกรมโดยใช้ Keyboard Control Key เป็นตัวช่วยในการควบคุมและบังคับมุมมอง โดยในส่วนของ Keyboard Control Key จะขออธิบายในส่วนต่อไป
- **Front View** เป็นมุมมองที่แสดงมุมมองจากทางด้านหน้าของห้องสามมิติโดยอ้างอิงจากตำแหน่งของกล้องและจุด Look AT Point ที่ได้กำหนดไว้ในขั้นตอนการสร้างห้อง
- **Top View** เป็นมุมมองที่แสดงมุมมองจากทางด้านบนของห้องสามมิติ ซึ่ง Top View นี้เป็นที่นิยมใช้ในการจัดวางเฟอร์นิเจอร์เนื่องจากสามารถเห็นมุมมองจากทางด้านบนทำให้ง่ายในการจัดวางเฟอร์นิเจอร์
- **Image View** เป็นส่วนของมุมมองจากภาพถ่ายจริงที่ผู้ใช้ต้องการเห็น โดยมุมมองนี้สามารถช่วยเพิ่มให้โปรแกรมสามารถแสดงภาพของห้องในมุมมองของผู้ใช้ที่ต้องการเห็นได้สมจริงมากยิ่งขึ้น

3.2.1.2 Keyboard Control Key เป็นส่วนของคีย์บนคีย์บอร์ดที่ใช้ในการควบคุมการจัดวางเฟอร์นิเจอร์และควบคุมการบังคับมุมมองแบบ Free View โดย Keyboard Control Key ประกอบด้วยส่วนต่างๆดังนี้

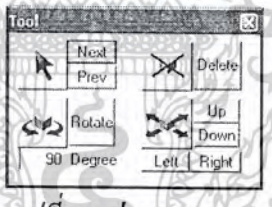
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **W** เป็นคีย์ที่ใช้ในการเคลื่อนย้ายเฟอร์นิเจอร์ ไปทางตำแหน่งไกลจากตัวผู้ใช้เมื่อเทียบกับ Front View
- **D** เป็นคีย์ที่ใช้ในการเคลื่อนย้ายเฟอร์นิเจอร์ไปทางขวา
- **S** เป็นคีย์ที่ใช้ในการเคลื่อนย้ายเฟอร์นิเจอร์ไปทางตำแหน่งเข้าหาตัวผู้ใช้เมื่อเทียบกับ Front View
- **A** เป็นคีย์ที่ใช้ในการเคลื่อนย้ายเฟอร์นิเจอร์ไปทางซ้าย
- **Q** เป็นคีย์ที่ใช้ในการหมุนเฟอร์นิเจอร์ไปทางด้านขวา
- **E** เป็นคีย์ที่ใช้ในการหมุนเฟอร์นิเจอร์ไปทางด้านซ้าย
- **Z** เป็นคีย์ที่ใช้ในการเพิ่มความสว่างของโปรแกรม
- **I** เป็นคีย์ที่ใช้ในการลดความสว่างของโปรแกรม

ในการควบคุมมองมองของ Free View นั้น ผู้ใช้จะใช้ Keyboard Control Key ในการบังคับทิศทางมุมมองด้วยเช่นกัน

3.2.2 Tool

เป็นเครื่องมือที่ใช้ในการควบคุมการจัดการเฟอร์นิเจอร์ที่เพิ่มเติมขึ้นมาเพื่ออำนวยความสะดวกในการใช้โปรแกรม จากรูปที่ 3-9 เป็นส่วนของ Tool โดยจะขออธิบายได้ในแต่ละส่วนดังต่อไปนี้



รูปที่ 3-9 ส่วนของ Tool

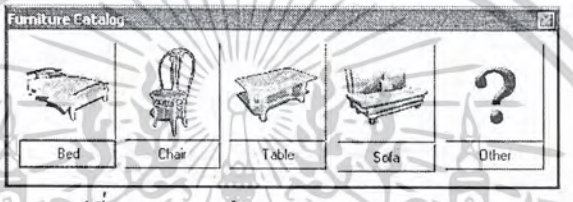
- **Next** เป็นปุ่มที่ใช้ในการเลือกไปยังเฟอร์นิเจอร์ตัวถัดไปเมื่อเทียบกับเฟอร์นิเจอร์ที่ผู้ใช้ทำการควบคุมอยู่ในปัจจุบันในกรณีที่ในโปรแกรมมีเฟอร์นิเจอร์หลายตัว
- **Prev** เป็นปุ่มที่ใช้ในการเลือกไปยังเฟอร์นิเจอร์ตัวก่อนหน้าไปเมื่อเทียบกับเฟอร์นิเจอร์ที่ผู้ใช้ทำการควบคุมอยู่ในปัจจุบันในกรณีที่ในโปรแกรมมีเฟอร์นิเจอร์หลายตัว
- **Delete** เป็นปุ่มที่ใช้ในการลบเฟอร์นิเจอร์ที่ไม่ต้องการออกจากห้องสามมิติ
- **Rotate** เป็นปุ่มที่เพิ่มเข้ามาเพื่อช่วยอำนวยความสะดวกในการหมุนเฟอร์นิเจอร์โดยผู้ใช้สามารถทำการป้อนค่าองศาที่ต้องการหมุนเฟอร์นิเจอร์ได้ตามต้องการแล้วคลิกที่ปุ่มนี้ เฟอร์นิเจอร์ก็จะถูกหมุนโดยอัตโนมัติตามค่าองศาที่ถูกป้อนเข้าไป
- **Up** เป็นปุ่มที่เคลื่อนย้ายตำแหน่งเฟอร์นิเจอร์ไปทิศทางด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

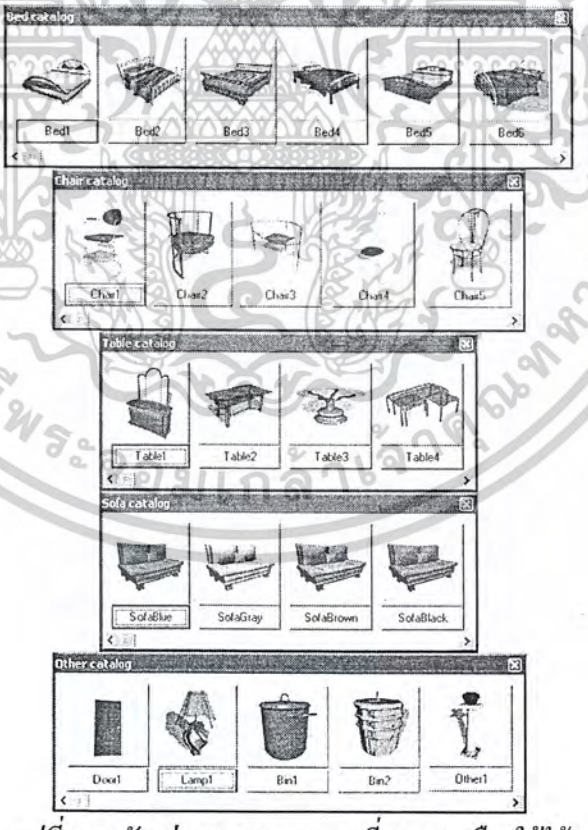
- **Down** เป็นปุ่มที่เคลื่อนย้ายตำแหน่งเฟอร์นิเจอร์ไปทิศทางด้านล่าง
- **Right** เป็นปุ่มที่เคลื่อนย้ายตำแหน่งเฟอร์นิเจอร์ไปทิศทางขวา
- **Left** เป็นปุ่มที่เคลื่อนย้ายตำแหน่งเฟอร์นิเจอร์ไปทิศทางซ้าย

3.2.3 Furniture Catalog

ในส่วนนี้จะเป็นส่วนของเฟอร์นิเจอร์ต่างๆที่ผู้ใช้สามารถเลือกนำมาวางภายในห้องได้ โดยดังรูปที่ 3-10 เป็นส่วนที่แสดง Furniture Catalog โดยจะมีการแบ่งแยกประเภทของเฟอร์นิเจอร์ประเภทต่างๆเป็นหมวดหมู่ เช่น เตียงนอน เก้าอี้ โต๊ะ โซฟา เป็นต้น



รูปที่ 3-10 แสดงส่วนของ Furniture Catalog



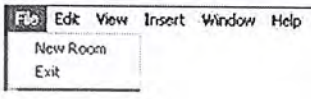
รูปที่ 3-11 ตัวอย่างของ Furniture ที่สามารถเลือกใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3-11 เป็นตัวอย่างของเฟอร์นิเจอร์ที่ผู้ใช้สามารถคลิกเพื่อเลือกไปวางภายในโปรแกรมซึ่งจะทำการแยกย่อยตามแต่ละประเภทของเฟอร์นิเจอร์

3.2.4 Menu

3.2.4.1 Menu File



รูปที่ 3-12 แสดงฟังก์ชั่นการทำงานใน Menu File

- New Room ใช้ในการกำหนดค่าเริ่มต้นของขนาด และมุมมองห้อง
- Exit ใช้เพื่อการจบการทำงานของโปรแกรม

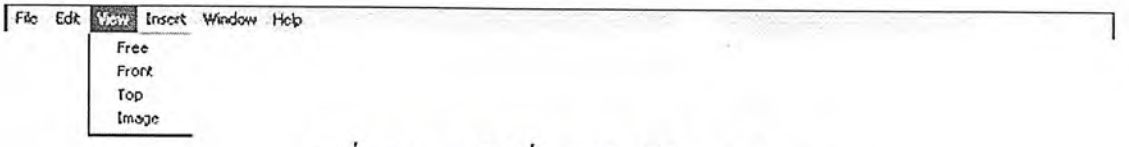
3.2.4.2 Menu Edit

รูปที่ 3-13 แสดงฟังก์ชั่นการทำงานใน Menu Edit

- Next Furniture ใช้เพื่อเลือกเฟอร์นิเจอร์ที่อยู่ใน List อันถัดไป
- Previous Furniture ใช้เพื่อเลือกเฟอร์นิเจอร์ที่อยู่ใน List อันก่อนหน้า
- Delete Furniture ใช้ลบเฟอร์นิเจอร์ที่เลือกอยู่ปัจจุบัน
- Rotate ใช้ในการหมุนเฟอร์นิเจอร์รามแนวแกน Z ในองศาที่กำหนดไว้ได้แก่ 45, 90, และ 180 องศา
- Set Collision กำหนดให้โปรแกรมมีการเช็ค Collision Detection ระหว่างเฟอร์นิเจอร์กับผนังห้อง
- Show Wall กำหนดให้โปรแกรมแสดงหรือ ไม่แสดงผนังห้อง
- Show Debug กำหนดให้โปรแกรม แสดง Debug ข้อมูลขาดห้องและ ตำแหน่งของเฟอร์นิเจอร์

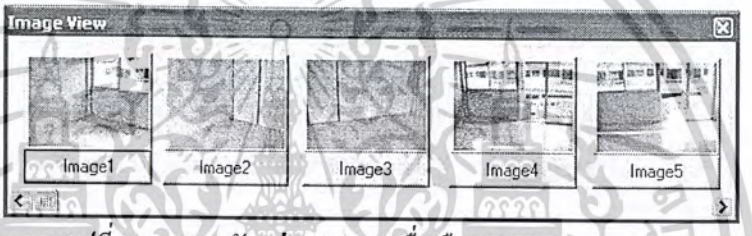
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.3 Menu View



รูปที่ 3-14 แสดงฟังก์ชั่นการทำงานใน Menu View

- Free เลือกการแสดงผลเป็นแบบมุมมองอิสระ สามารถควบคุมดูส่วนต่างๆได้เอง
- Front เลือกการแสดงผลเป็นแบบมุมมองจากด้านหน้า
- Top เลือกการแสดงผลเป็นแบบมุมมองจากด้านบน
- Image เลือกการแสดงผลเป็นมุมมองจากภาพถ่าย

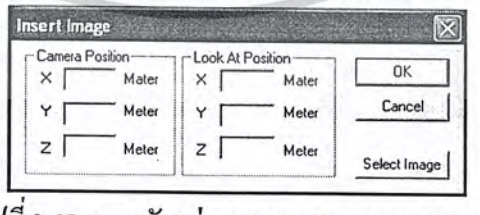


รูปที่ 3-15 แสดงตัวอย่าง Dialog เมื่อเลือก Menu Image View

3.2.4.4 Menu Insert

รูปที่ 3-16 แสดงฟังก์ชั่นการทำงานใน Menu Insert

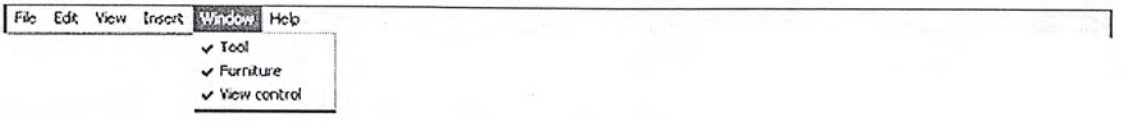
- Image ใช้เพื่อเลือกภาพถ่ายที่จะใช้ใน Image View โดยจะต้องกำหนดค่าการวัดตำแหน่งของกล้อง และตำแหน่ง Look at ด้วย



รูปที่ 3-17 แสดงตัวอย่าง Dialog ของ Insert Image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

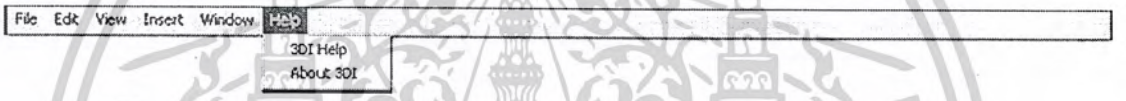
3.2.4.5 Menu Window



รูปที่ 3-18 แสดงฟังก์ชั่นการทำงานใน Menu Window

- Tool ใช้เปิดหรือปิด Dialog Tool
- Furniture ใช้เปิดหรือปิด Dialog Furniture
- View Control ใช้เปิดหรือปิด Dialog View Control

3.2.4.6 Menu Help



รูปที่ 3-19 แสดงฟังก์ชั่นการทำงานใน Menu Help

- 3DI Help ใช้งาน Help ของโปรแกรม
- About 3DI รายละเอียดของผู้จัดทำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลอง

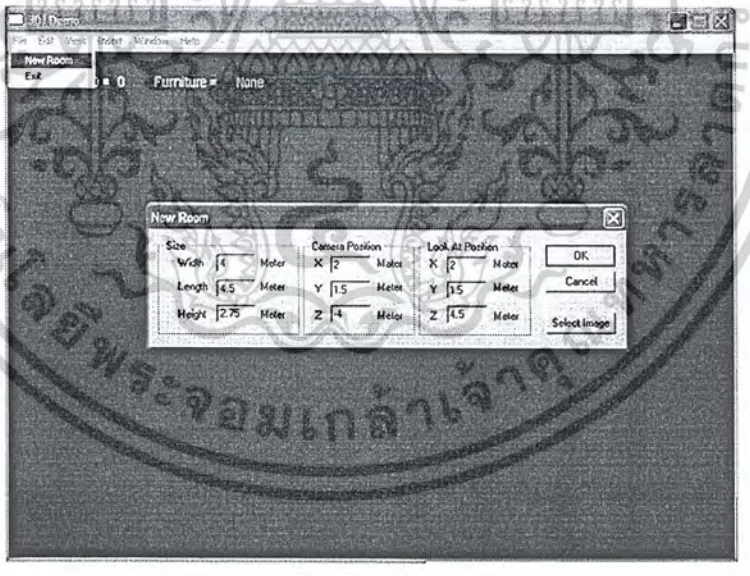
ทำการสร้างโปรแกรมเพื่อทดสอบการทำงานทั้งหมด โดยใช้ภาพถ่ายจากห้องโล่ง ไม่มีการตกแต่งเฟอร์นิเจอร์ และข้อมูลตำแหน่งต่างๆที่วัดได้ เป็นอินพุตให้กับโปรแกรม และจำลองจัดวางเฟอร์นิเจอร์

โปรแกรมที่นำมาใช้พัฒนาในโครงงานนี้ได้แก่

- Microsoft Visual C++ 6.0
- Microsoft DirectX 8 SDK
- 3D Studio Max
- Panda DirectX (Plug-In)

4.1 การทดลองการสร้างห้อง

- ทำการกำหนดค่าความกว้าง ความยาว ความสูงของห้อง และตำแหน่งมุมมองที่ต้องการ ที่ได้จากการวัดใส่ให้กับโปรแกรม

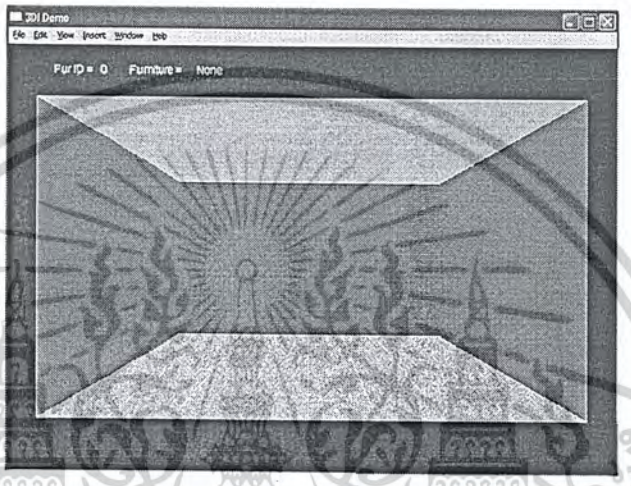


รูปที่ 4-1 แสดงการกำหนดขนาดห้องและมุมมองเริ่มแรกของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-1 เป็นการแสดงการป้อนค่าพารามิเตอร์ที่ได้จากการวัดห้องจริงเพื่อทำการสร้างห้องจำลองสามมิติขึ้นภายในโปรแกรม โดยป้อนค่า ความกว้าง ความยาว และความสูง เพื่อกำหนดขนาดของห้อง นอกจากนั้นยังมีค่าตำแหน่งของกล้อง และค่าจุด Look At เพื่อกำหนดมุมมองของกล้อง

- โปรแกรมจะสร้างห้องจำลองขึ้น เพื่อทำการจัดวางเฟอร์นิเจอร์ต่อไป

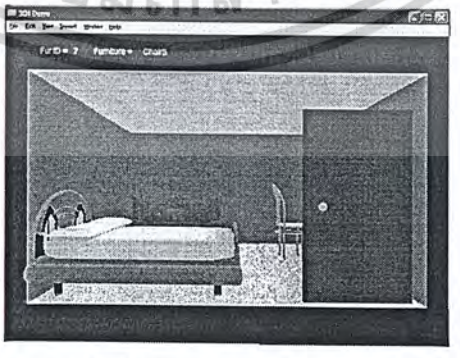


รูปที่ 4-2 แสดงห้องที่โปรแกรมสร้างขึ้น

จากรูปที่ 4-2 แสดงห้องที่โปรแกรมสร้างขึ้นมา ตามค่าพารามิเตอร์ที่ได้ป้อนเข้าไปหัวข้อที่หนึ่ง โดยมีขนาดของห้อง และมุมมองตามค่าพารามิเตอร์ ต่อจากนี้ผู้ใช้สามารถนำเฟอร์นิเจอร์เข้ามาจัดวางภายในห้องที่โปรแกรมสร้างขึ้นได้

4.2 การทดลองการจัดวางเฟอร์นิเจอร์และการแสดงห้องในมุมมองแบบต่างๆ

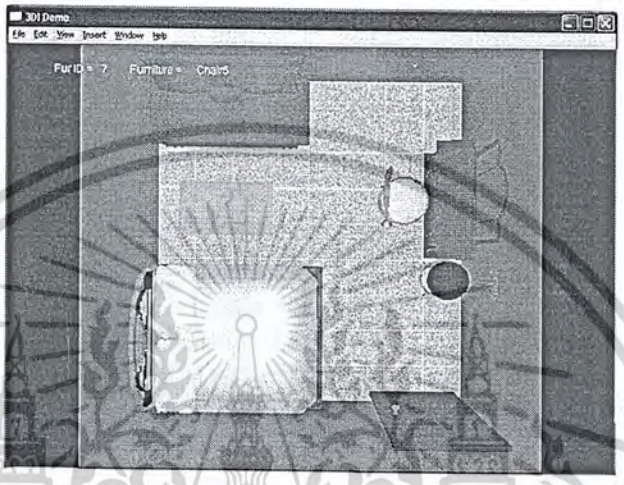
- เลือกเฟอร์นิเจอร์จาก Furniture Catalog ของโปรแกรมแล้วนำไปจัดวางภายในห้อง



รูปที่ 4-3 แสดงตัวอย่างการจัดวางเฟอร์นิเจอร์ใน Front View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-3 เป็นการแสดงตัวอย่างการจัดวางเฟอร์นิเจอร์ภายในห้อง ซึ่งหลังจากที่ผู้ใช้ได้ทำการออกแบบการจัดวางห้องเสร็จสิ้นแล้ว โดยในรูปที่ 4-3 จะเป็นตัวอย่างของการแสดงห้องในมุมมองแบบ Front View และรูปที่ 4-4 เป็นการเปลี่ยนมุมมองเป็นแบบ Top view ซึ่งมุมมองนี้สามารถแสดงภาพจากมุมด้านบนของห้องซึ่งทำให้ผู้ใช้สามารถเห็นทัศนียภาพจากมุมมองทางด้านบนภายในห้อง



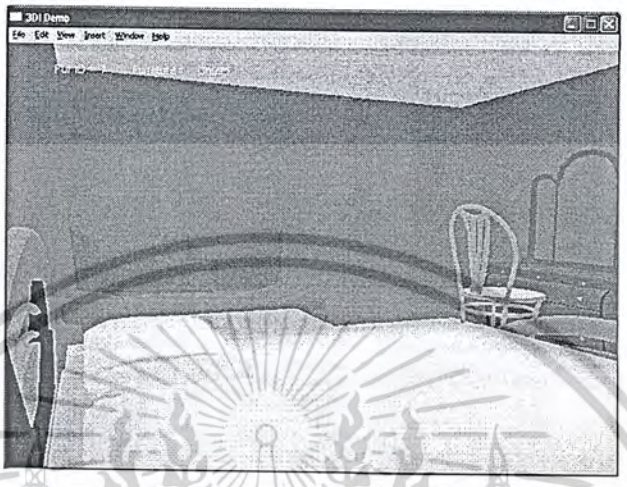
รูปที่ 4-4 แสดงตัวอย่างการจัดวางเฟอร์นิเจอร์ใน Top View

- โปรแกรมสามารถให้ผู้ควบคุมมุมมองที่ต้องการดูเองได้ใน Free View สำหรับดูการจัดวางต่างๆ ในมุมมองที่แปลกออกไป

รูปที่ 4-5 แสดงตัวอย่างที่1 สำหรับมุมมองแบบ Free View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-5 แสดงตัวอย่างที่1 สำหรับมุมมองแบบ Free View ซึ่งในมุมมองแบบนี้ผู้ใช้สามารถควบคุมการมองได้ตามความต้องการ ส่วนในรูปที่ 4-6 เป็นตัวอย่างที่ 2 สำหรับมุมมองแบบ Free View



รูปที่ 4-6 แสดงตัวอย่างที่2 สำหรับมุมมองแบบ Free View

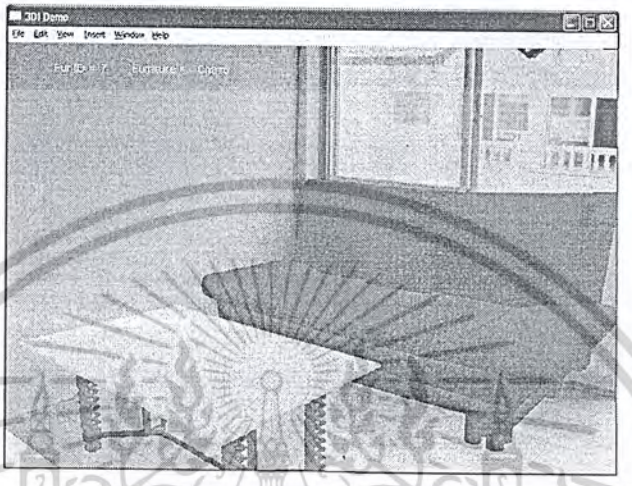
4.3 การทดลองการนำภาพถ่ายเข้ามาแสดงภายในโปรแกรม

- โปรแกรมสามารถให้ผู้ใช้ดูมุมมองจากภาพที่ถ่ายจากสถานที่จริง เพื่อให้ได้สภาพแวดล้อมที่สมจริงมากยิ่งขึ้น ใน Image Views โดยผู้ใช้ต้องทำการถ่ายภาพและวัดตำแหน่งต่างๆในมุมมองการถ่าย เพื่อเป็นข้อมูลใส่ให้กับโปรแกรมด้วย

รูปที่ 4-7 รูปถ่ายจากสถานที่จริง มุมมองที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-7 เป็นรูปถ่ายจากสถานที่จริงที่ผู้ใช้ต้องการนำมาแสดงภายในโปรแกรม โดยจากตัวอย่างนี้ โปรแกรมจะสมารถทำการนำภาพถ่ายจากสถานที่จริงมาแสดงในมุมมองของห้องสามมิติที่โปรแกรมสร้างขึ้น ได้อย่างสอดคล้องและสัมพันธ์กัน ดังแสดงในรูปที่ 4-8

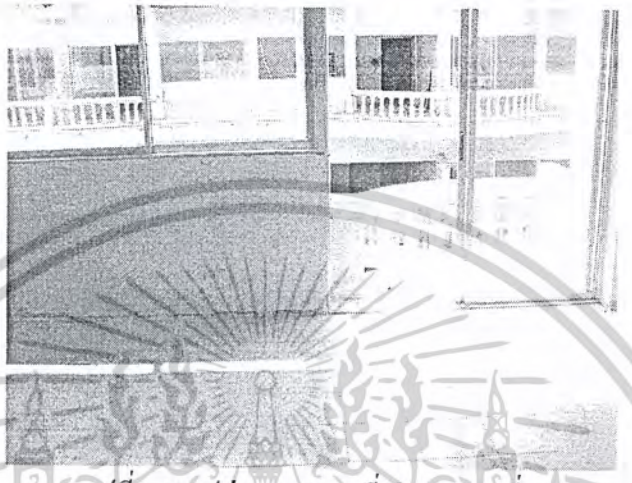


รูปที่ 4-8 แสดงตัวอย่าง Image View มุมมองที่ 1

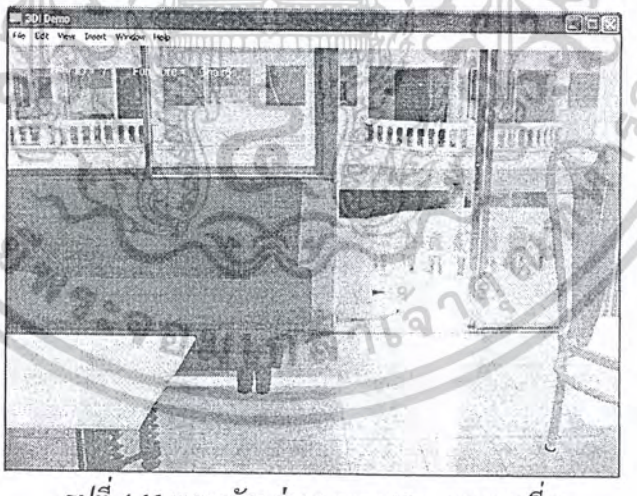
รูปที่ 4-9 แสดงตัวอย่าง มุมมองที่ 1 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-9 เป็นตัวอย่างที่แสดงในมุมมองเดียวกันกับรูปที่ 4-7 และ รูป 4-8 แต่โปรแกรมได้แสดงมุมมองแบบธรรมดาโดยที่ไม่ได้ใช้ภาพถ่ายจากสถานที่จริงนำมาแสดง

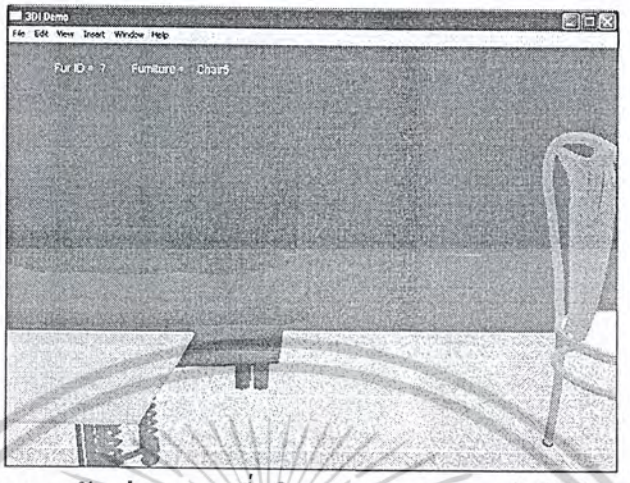


รูปที่ 4-10 รูปถ่ายจากสถานที่จริง มุมมองที่ 2



รูปที่ 4-11 แสดงตัวอย่าง Image View มุมมองที่ 2

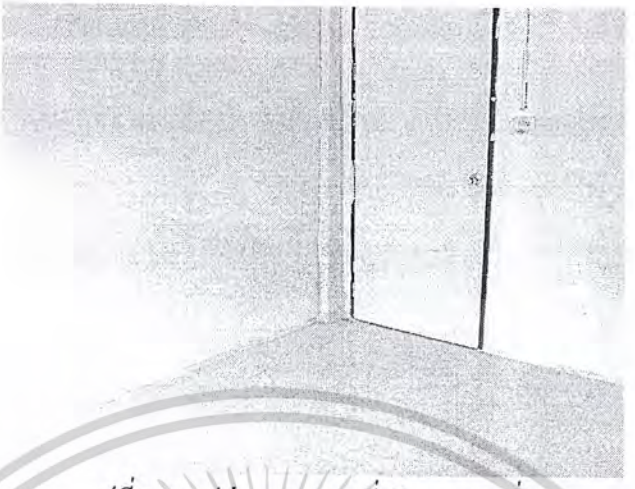
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



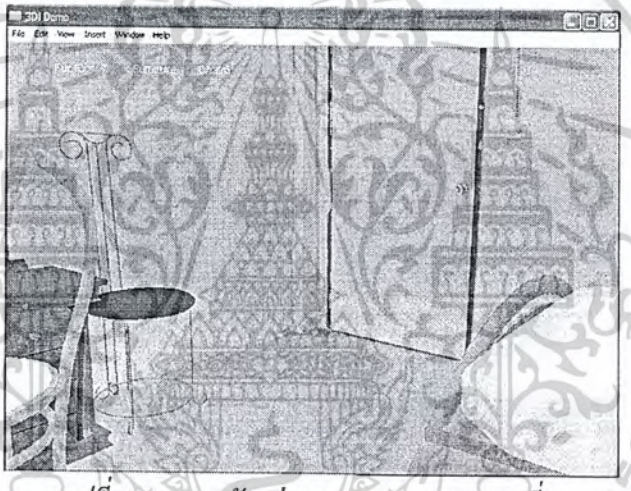
รูปที่ 4-12 แสดงตัวอย่าง มุมมองที่ 2 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย

ส่วนในรูปที่ 4-10 , 4-11 และ 4-12 เป็นอีกตัวอย่างที่แสดงการนำภาพถ่ายจริงแต่เป็นอีกคนละมุมมอง มาแสดงภายในโปรแกรม อีกตัวอย่างหนึ่งแสดงในรูป 4-13 ,4-14 และ ในรูปที่ 4-15 ซึ่งจะเห็นว่าโปรแกรมสามารถแสดงบรรยากาศภายในห้องที่ใกล้เคียงกับสถานที่จริงได้

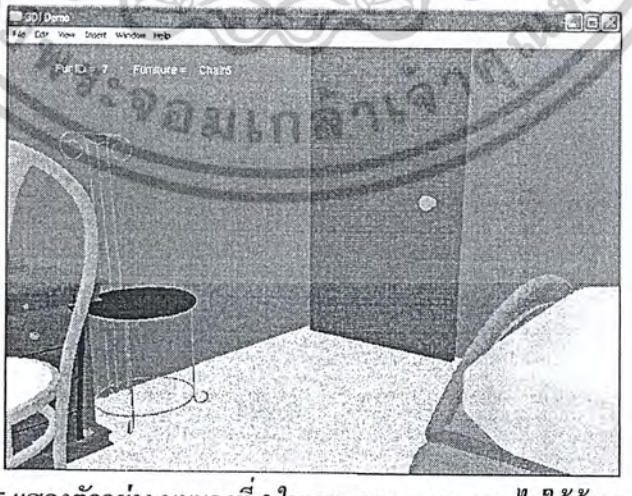
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 รูปถ่ายจากสถานที่จริง มุมมองที่ 3



รูปที่ 4-14 แสดงตัวอย่าง Image View มุมมองที่ 3



รูปที่ 4-15 แสดงตัวอย่าง มุมมองที่ 3 ในมุมมองแบบธรรมดาไม่ใช้ข้อมูลภาพถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุป

โปรแกรมที่สร้างขึ้นสามารถแสดงมุมมองและการจัดวางเฟอร์นิเจอร์ได้ ทั้งแบบที่เป็นคอมพิวเตอร์กราฟฟิกและแบบที่เป็นภาพถ่าย

ส่วนของภาพที่เป็นแบบคอมพิวเตอร์กราฟฟิก ผู้ใช้สามารถเลือกมุมมองที่การดูเองได้อย่างอิสระ ในการจัดวางเฟอร์นิเจอร์ต่างๆ สามารถตรวจดูได้ตามที่ต้องการได้

ส่วนที่เป็นมุมมองจากภาพถ่าย ภาพที่โปรแกรมสร้างขึ้นจากอินพุตที่ป้อนให้ มีมุมมองเดียวกับภาพจริง ทำให้เราสามารถนำห้องที่สร้างขึ้นในโปรแกรมนี้มาใช้ในการจัดวางเฟอร์นิเจอร์ต่อไป ซึ่งจะทำให้วางเฟอร์นิเจอร์ได้ในมุมมองที่ถูกต้องตรงกับความเป็นจริง

5.2 ปัญหาที่เกิดขึ้น

1 ปัญหาการนำภาพถ่ายจากสถานที่จริงที่ถ่ายจากกล้องดิจิตอลที่ต่างกันและวัดระยะเพื่อนำมาใช้กับโปรแกรม ซึ่งกล้องแต่ละอันจะมีอัตราขยาย ของเลนส์แตกต่างกันไป โดยในการทดลองนี้ใช้กล้องเพียงตัวเดียวในการถ่ายภาพทำการทดลองและตั้งอัตราการขยายไว้ต่ำที่สุด

2 ปัญหาการทำ Collision Detection ไม่สมบูรณ์เมื่อทำการหมุนเฟอร์นิเจอร์ ทำให้การเช็ค Collision เกิดการผิดพลาดในบางครั้ง

3 ปัญหาการใช้งานในส่วนของการจัดการเฟอร์นิเจอร์ยังใช้งานได้ไม่สะดวกนัก

4 ยังไม่สามารถทำให้การเพิ่มภาพถ่ายหรือเฟอร์นิเจอร์เข้าสู่โปรแกรม เป็นแบบ Dynamic ได้ ยังต้องมีการ Compile โปรแกรมใหม่

5.3 แนวทางการพัฒนาต่อ

- ในส่วนของการจัดการเฟอร์นิเจอร์ สามารถทำให้การควบคุมง่ายและสะดวกมากขึ้นได้ เช่น สามารถเลือกเฟอร์นิเจอร์ที่จะแก้ไขได้จากหน้าจอแสดงผล
- เพิ่มจำนวนเฟอร์นิเจอร์ในโปรแกรมให้หลากหลายมากยิ่งขึ้น
- พัฒนาความสวยงานของเฟอร์นิเจอร์ให้มีรายละเอียดมากขึ้น รวมถึงการแสดงผลภาพที่ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

การติดตั้งโปรแกรม 3DI

1 โปรแกรมถูกเขียนขึ้นด้วยโปรแกรม VC++ 6.0 รันได้บน ระบบปฏิบัติการ Microsoft Window

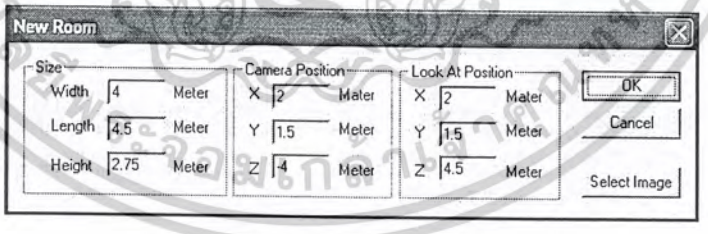
2. ต้องติดตั้ง Microsoft DirectX Runtime ตั้งแต่ Version 8.0 ขึ้นไป

3. ไม่ต้องมีการ Install โปรแกรม สามารถ copy ใช้งานได้ทันที

การใช้งานโปรแกรม 3DI

1. เปิดโปรแกรม 3DI จากนั้น เลือกเมนู File ->New Room จากนั้นกำหนดค่า ความกว้าง ความยาว ความสูง ของห้อง รวมถึง ค่าตำแหน่งของกล้อง และ ตำแหน่ง Look At ใน New Room Dialog

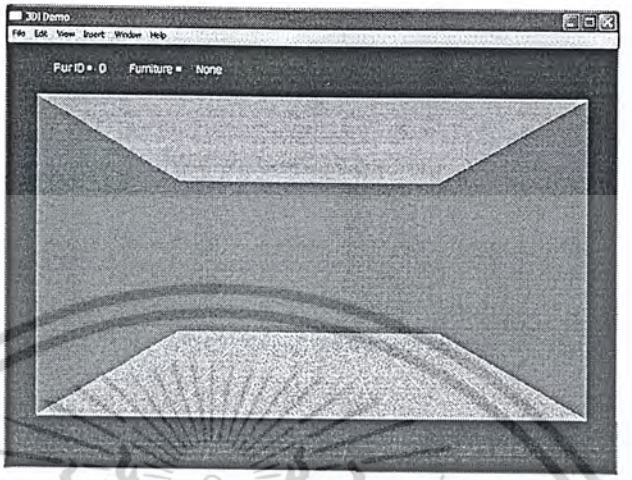
รูปแสดงการเลือกเมนู New Room เพื่อสร้างห้องใหม่



รูปแสดง New Room Dialog และการใส่ค่าต่างๆ

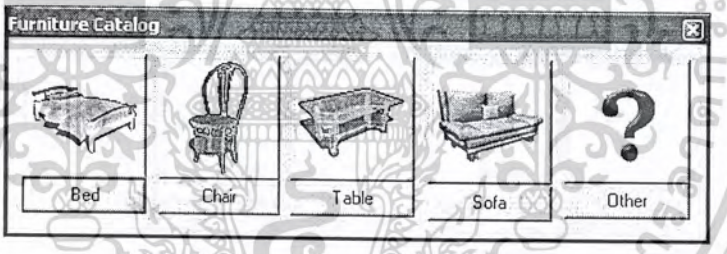
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากนั้นโปรแกรมจะแสดงภาพห้องที่มีขนาดตามที่กำหนด



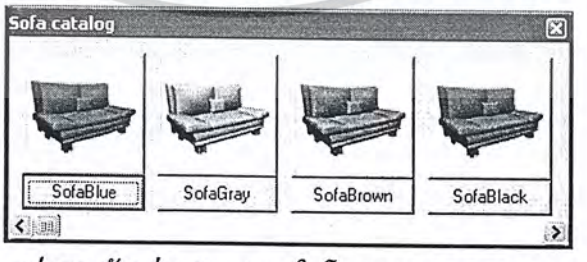
รูปแสดงห้องที่โปรแกรมสร้างขึ้น

3. ทำการเลือกโหมดเฟอร์นิเจอร์ที่ต้องที่จะใส่ในห้อง โดยเลือกจาก Furniture Catalog



รูปแสดง Furniture Catalog เลือกโหมด Furniture

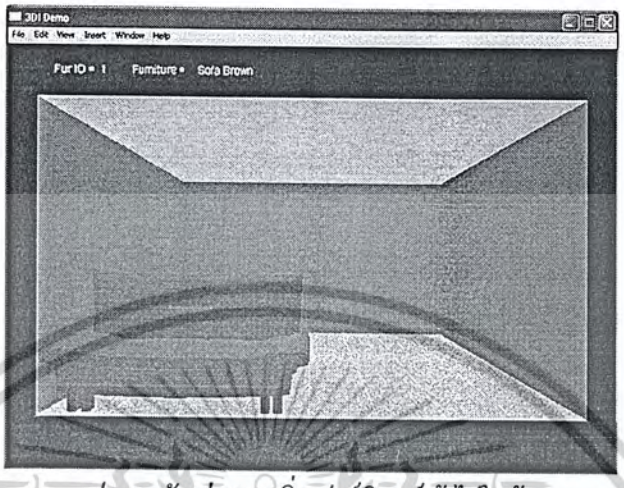
4. จากนั้นเลือกเฟอร์นิเจอร์ที่ต้องการ



รูปแสดงตัวอย่าง Furniture ในโหมดของ Sofa catalog

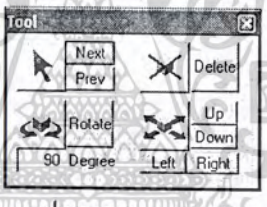
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โปรแกรมจะแสดงเฟอร์นิเจอร์ดังกล่าวขึ้น ภายในห้อง



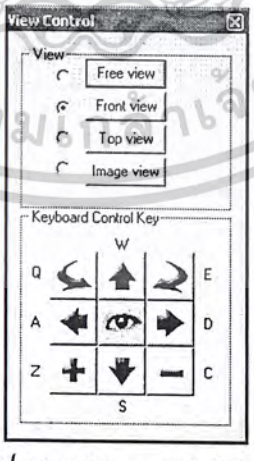
รูปแสดงตัวอย่างการเพิ่มเฟอร์นิเจอร์เข้าไปในห้อง

6. เราสามารถปรับตำแหน่งหรือหมุนเฟอร์นิเจอร์ได้จาก Tool Dialog



รูปแสดง Tool dialog

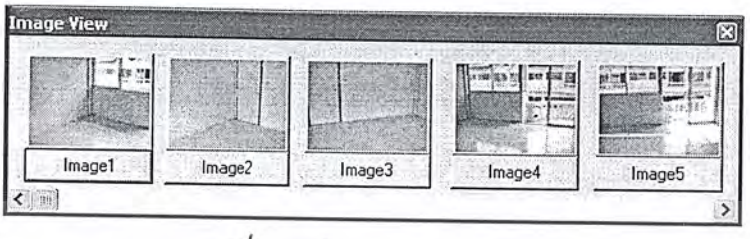
7. โปรแกรมสามารถแสดงมุมมองได้ 4 ลักษณะ ได้แก่ Free View , Front View , Top View,และ Image View



รูปแสดง View control dialog

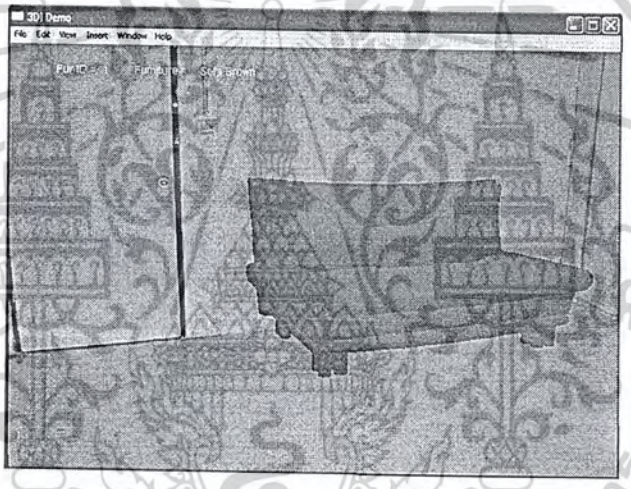
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อเลือก Image View แล้วจะปรากฏ Image View Catalog ให้เลือก มุมมองภาพถ่ายที่ต้องการ



รูปแสดง Image view dialog

9. โปรแกรมจะแสดง Image View ซึ่งจะทำให้ได้ภาพเฟอร์นิเจอร์ที่วางอยู่ในห้องในมุมมองภาพถ่ายที่สมจริงขึ้น เมื่อเทียบกับการใช้คอมพิวเตอร์กราฟฟิกทั้งหมด



รูปแสดงภาพตัวอย่างในมุมมองจาก Image view

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Jim Adams : "Programming Role Playing Game With DirectX" , Stacy L.Hiquet , Premier Press Game Development , USA. 2002, pp. 541-553.

[2] Edward Angel : "Interactive Computer Graphics", A Top Down Approach Using OpenGL (Third Edition) , Addison Wesley, 1998 ,pp. 218-225.

[3] Peter J. Kovach , แปลและเรียบเรียงโดย ชัยดำรงค์ อุทิรัมย์, "Direct3D พลังพัฒนาแห่งเกมสามมิติ" , สำนักพิมพ์สามย่าน.Com , 2544 , pp 52-110.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้