อภินันทนาการ

สำนักหอสมุด

รายงานการวิจัย

ระบบค้นหาสถานที่แบบ 3 มิติ กรณีศึกษา อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

Searching Location System using 3D Style in Faculty of Science Building, Naresaun University

โดย

นายอดิเรก รุ่งรังษี
นายสัญญา เครือหงษ์

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 23 ส.ค. 2554
เลขทะเบียน 1560779
เลขเรียกหนังสือ ว TK 5105 .T47 อ1275 2551

ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร
ทุนอุดหนุนการวิจัยคณะวิทยาศาสตร์
ประจำปีงบประมาณ 2551

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลุล่วงได้อย่างดี จากคณะวิทยาศาสตร์ที่ได้มอบทุนในการทำงานวิจัย สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ และขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ฯ ที่ให้ความช่วยเหลือในการทำวิจัยนี้ในเรื่องต่างๆ

คุณค่าและประโยชน์อันพึงมีจากงานวิจัยฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

อดิเรก รุ่งรังษี

สัญญา เครือหงษ์

มหาวิทยาลัยนเรศวร

บทคัดย่อ

รูปแบบ สามมิติ จัดว่าได้รับความนิยม และเข้าถึงผู้ใช้ได้ง่ายในปัจจุบัน ไม่ว่าจะเป็นในด้านเกม หรือ แอนนิเมชั่นก็นำรูปแบบสามมิติมาใช้เป็นจำนวนมาก เพราะมีความเสมือนจริงในการแสดงผล คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร เป็นคณะที่มีอาณาบริเวณกว้างขวาง และมีห้องเป็นจำนวนมาก เมื่อมีผู้มาติดต่อกับทางคณะ ถ้าไม่คุ้นเคยกับสถานที่ ก็จะทำให้สับสน ล่าช้าต่อการติดต่อธุระต่างๆ และถ้าเป็นแผนที่แบบ 2 มิติก็อาจทำให้ดูไม่เข้าใจได้ ดังนั้น ผู้จัดทำโครงการจึงมีแนวคิดจัดสร้าง ระบบค้นหาสถานที่ภายในคณะวิทยาศาสตร์มหาวิทยาลัยนเรศวร แบบสามมิติผ่านเครือข่ายอินเตอร์เน็ตขึ้น เพื่ออำนวยความสะดวกสำหรับผู้ที่มาติดต่อทางคณะ โดยระบบที่จัดทำขึ้นสามารถ ค้นหาสถานที่ได้โดยการเลือกชื่อสถานที่ของสถานเริ่มต้นและปลายทาง จากนั้นระบบจะทำการจำลองการเดินทางในรูปแบบสามมิติให้ผู้ใช้ และจากผลการประเมินประสิทธิภาพของระบบคิดเป็นร้อยละ 90.5 ซึ่งอยู่ในระดับดี

คำสำคัญ 3 มิติ ระบบค้นหาสถานที่

Abstract

In the present, 3D is popular and easy to access for users, included game or animation because it seem to be real in displaying. Faculty of science is one of extensive area in Naresuan University that has several rooms. The visitors may be confused and waste time when they would like to contact the officer in the faculty. In addition, 2D map still difficult to understand. Therefore, we create this project, Searching Location in Faculty of Science using 3D Style via Internet, to offer the convenience for visitors. The system will discover the location by choosing the name of beginning and destination places. After that, the system will be display as traveling sample in 3D pattern. The efficiency output and using this system equal 90.5 %(good levels).

Keywords: 3D, Location Searching System

สารบัญเรื่อง

บทที่ **หน้า**

กิตติกรรมประกาศ.....I

บทคัดย่อภาษาไทย.....II

บทคัดย่อภาษาอังกฤษ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญรูป.....VII

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์.....1

1.3 ขอบเขตงานวิจัย.....2

1.4 แนวคิดและทฤษฎีที่เกี่ยวข้อง.....2

1.5 แผนการดำเนินงาน.....6

บทที่ 2 การดำเนินงาน.....7

2.1 รูปแบบของโปรแกรม.....7

2.2 การพัฒนาโปรแกรม.....7

2.3 โครงสร้าง x File.....10

บทที่ 3 ผลการวิจัย.....16

3.1 ผลการของการพัฒนาโปรแกรม.....16

3.2 การออกแบบการประเมินผล.....17

3.3 ผลการประเมินผล.....17

บทที่ 4 บทสรุปและข้อเสนอแนะ.....19

4.1 บทสรุป.....19

4.2 ข้อเสนอแนะ.....19

สารบัญ (ต่อ)

บทที่ หน้า

บรรณานุกรม.....20
ภาคผนวก ก.....21
ภาคผนวก ข.....25
ภาคผนวก ค.....26

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 1.1 ตารางแสดงระยะเวลาในการดำเนินการ.....6

ตารางที่ 2.1 x File DATA Type.....11

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัญหาหนึ่งที่ผู้มาติดต่อสถานที่ราชการหรือหน่วยงานต่างๆก็คือการหาเส้นทางที่จะไปยังสถานที่เป้าหมาย ถ้าสถานที่แห่งนั้นไม่คุ้นเคย ซับซ้อนแล้ว ก็จะทำให้ผู้มาเยือนเสียเวลามาก โดยเฉพาะคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร ซึ่งเป็นคณะที่มีอาณาบริเวณ กว้างขวางและมีห้องเป็นจำนวนมาก เพราะถือว่าเป็นคณะที่มีพื้นที่กว้างขวางที่สุด มีห้องเรียน ห้องพัก ห้องสำนักงานมากที่สุดในมหาวิทยาลัยนเรศวร ประกอบกับเป็นคณะที่มีการจัดกิจกรรมที่มีบุคคลภายนอกมหาวิทยาลัยจะมีโอกาสมาร่วมกิจกรรมบ่อยครั้ง ในทุกครั้งที่ผู้มาเยือนติดต่อธุระกับฝ่ายต่างๆ ของคณะ ก็จะเกิดความสับสน การติดป้ายบอกเส้นทางหรือแผนที่แบบ 2 มิติ นั้นก็อาจจะไม่เพียงพอซึ่งดูยาก จากการศึกษารูปแบบการแสดงผลในคอมพิวเตอร์ในปัจจุบันที่มีการนำการแสดงภาพแบบ 3 มิติ มาใช้งานที่หลากหลายจำนวนมาก ไม่ว่าจะเป็นเกมหรือแอนนิเมชันต่างๆ และในปัจจุบันก็สามารถที่จะพัฒนาได้ง่าย จะเห็นจากการพัฒนาที่มีออกมาอย่างต่อเนื่อง ตัวอย่างเกมที่มีชื่อเสียงก็คือเคาน์เตอร์สไตรค์ (Counter-Strike) ที่อาศัยทักษะการควบคุมเมาส์และคีย์บอร์ดในการเล่นเกมเพื่อเดินไปยังสถานที่ต่างๆในมุมมองบุคคลที่หนึ่ง ซึ่งมีความเสมือนจริง ทำให้รู้สึกว่าได้อยู่ในสถานที่นั้นจริงๆอีกด้วยและรู้สึกมีส่วนร่วมในเกม

จากปัญหาและประกอบกับเทคนิคดังกล่าว ผู้วิจัยจึงมีแนวคิดที่จะนำเทคโนโลยีทางด้าน 3 มิติ และการเขียนโปรแกรมเพื่อควบคุมเมาส์ แป้นพิมพ์ของเกม[1] ที่มีลักษณะคล้ายเกมที่อาจจะดูว่ามีความรุนแรง นำมาพัฒนาเป็นโปรแกรมที่เป็นประโยชน์ โดยได้จัดทำโปรแกรมค้นหาสถานที่แบบ 3 มิติขึ้น เพราะการนำเสนอในรูปแบบ 3 มิตินั้นจะสามารถให้รายละเอียดที่ชัดเจนเข้าใจง่ายขึ้น ภาพที่เสมือนจริงมากกว่าการนำเสนอแบบ 2 มิติ ประกอบกับในปัจจุบันแต่ละตึกของคณะวิทยาศาสตร์ก็จะมีคอมพิวเตอร์ตั้งบริการอยู่แล้ว เมื่อนำไปไว้ในจุดต่างๆของคณะวิทยาศาสตร์ก็น่าจะเป็นประโยชน์มาก เพื่อให้ผู้ที่มาเยือนสามารถจำลองเส้นทางการเดินของตัวเองก่อนมาคณะวิทยาศาสตร์ ก่อนได้

1.2 วัตถุประสงค์

เพื่อพัฒนาระบบค้นหาสถานที่แบบ 3 มิติ อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

1.3 ขอบเขตของงานวิจัย

เพื่อพัฒนาระบบค้นหาสถานที่แบบ 3 มิติ อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร โดยโปรแกรมดังกล่าวสามารถค้นหาห้องอัตโนมัติและสามารถควบคุมด้วยแป้นพิมพ์ได้

1. ค้นหาห้องอัตโนมัติ ผู้ใช้สามารถเลือกห้องที่กำหนดไว้ในรายการแล้วระบบจะนำเสนอเป็นภาพเคลื่อนไหวนำท่านไปสู่ห้องที่กำหนดไว้ได้

2. การควบคุมด้วยแป้นพิมพ์ ผู้ใช้สามารถควบคุมการทำงานด้วยแป้นพิมพ์โดยใช้คีย์ลูกศรมาควบคุมการทำงานเสมือนเดินเข้าไปภายในตึกจริง และแสดงจุดที่ท่านอยู่ด้วยว่าท่านอยู่ ณ ตำแหน่งใดของตึก

1.4 แนวคิดและทฤษฎีที่เกี่ยวข้อง

1.4.1 API ของ DirectX 9.0 SDK[1]

DirectX 9.0 SDK ถูกพัฒนาให้มีความสามารถเพิ่มขึ้นจาก เวอร์ชันก่อนหน้านี้โดยสามารถแบ่งออกเป็นส่วนประกอบหลักๆได้ดังนี้

DirectX Graphics เป็นการรวมกันระหว่าง DirectDraw และ direct 3D โดยมีจุดเด่นตรงที่การประมวลผล กราฟิกแบบขนานและสามารถจัดการกับกราฟิกให้มีความสมจริงกว่าเวอร์ชันที่ผ่านมา ทำให้ผู้พัฒนาเกม สามารถพัฒนาเกม 3มิติ ได้ง่ายขึ้น

DirectInput เป็นอีกองค์ประกอบหนึ่งของ Direct X ที่ช่วยจัดการในส่วนของการจัดการอุปกรณ์แป้นพิมพ์ (Keyboard), เมาส์ (Mouse) และ จอยสติก (Joy Stick) ซึ่งเป็นส่วนที่ช่วยให้โปรแกรมเมอร์ สามารถที่จะติดต่อกับอุปกรณ์ดังที่ได้กล่าวมาได้โดยง่ายโดยในการเขียนโปรแกรมติดต่อกับอุปกรณ์ Keyboard หรือ Mouse ก็จะมีขั้นตอนของการเขียนโปรแกรมดังนี้ คือ

- สร้าง Keyboard หรือ Mouse Device
- Set Cooperation level ให้กับ Keyboard เพื่อกำหนดรูปแบบการใช้งานอุปกรณ์
- Set รูปแบบข้อมูลให้กับ Keyboard Device หรือ Mouse Device
- Acquire Keyboard Device หรือ Mouse Device

ในเวอร์ชันนี้ได้มีการเพิ่มประสิทธิภาพใหม่เข้าไป คือสามารถใช้ Action mapping ได้ซึ่งใช้กำหนดค่าต่างๆให้กับอุปกรณ์ควบคุมและยังสามารถจัดการเชื่อมต่อระหว่าง Input Action กับอุปกรณ์ควบคุม (Input Device) ได้และการเปลี่ยนแปลงค่าทำได้ง่ายขึ้น โดยจะขึ้นอยู่กับชนิดของอุปกรณ์ควบคุม (Input Device) นั้นๆ นอกจากนี้ยังได้เพิ่มการสนับสนุนอุปกรณ์ต่างๆได้มากขึ้น

DirectPlay ถูกเพิ่มประสิทธิภาพในการสนับสนุนการทำงานในรูปแบบเนตเวิร์กหรือระบบเครือข่ายให้มีความหลายหลายในการใช้งาน อีกทั้งยังเพิ่มการป้องกันความผิดพลาดที่อาจเกิดขึ้น

ได้กับระบบนอกจากนี้DirectPlay ยังเพิ่มการสนับสนุนการสื่อสารด้วยเสียงระหว่าผู้เล่น (Voice Communication)

DirectSound เป็นชุด API ที่มีเครื่องมือในการใช้จัดการกับระบบเสียงแบบสเตริโอและแบบ3 มิติ ได้อย่างมีประสิทธิภาพประกอบด้วยความสามารถในการจัดการหน่วยความจำ การผสมเสียงของฮาร์ดแวร์ DirectSound ได้รับการออกแบบมาเพื่อดึงความสามารถของฮาร์ดแวร์ระบบ การรวมเสียงแบบ 3 มิติเข้าไปในเกมหรือการซิมูเลชัน ทำให้แอพพลิเคชันสามารถให้เสียงได้สมจริงสมจัง การได้ยินเสียงจากทางซ้าย ทางขวา หรือจากทางด้านบน โดยเฉพาะอย่างยิ่งเสียงที่มาจากรอบตัวจะช่วยสร้างสภาพแวดล้อมให้ตื่นเต้นมากขึ้น

DirectMusic เป็นชุด API ที่ทำงานกับข้อมูลประเภท Message-base Musicial Data ซึ่งเป็นข้อมูลที่แปลงมาจาก Wave Sample ด้วยซินธีไซเซอร์(Synthesizer) ทั้งแบบฮาร์ดแวร์หรือซอฟแวร์โดยปกติถ้าเป็นซอฟแวร์จะใช้โปรแกรม Microsoft Software Synthesizer เพื่อสร้าง Wave Sample ให้กับDirectSoundการใช้เสียงดนตรีที่ได้จากซินธีไซเซอร์จะเป็นไปตามมาตรฐาน DLS(Downloadable Sound) นอกจากนี้ DirectMusic ยังมีกลไกที่ใช้สร้างเพลงได้ตามที่เรากำหนด *DirectShow* ได้ถูกเพิ่มเข้ามาตั้งแต่ Version DirectX 8.0 นี้โดยในเวอร์ชันก่อนหน้านี้ DirectShow จะถูกแยกต่างหากจาก DirectX โดยจะทำหน้าที่ในการเล่นไฟล์มัลติมีเดีย ประเภทภาพวีดีโอเป็นหลัก แต่เมื่อรวมเข้ากับ DirectX แล้ว ทำให้การเล่นไฟล์มัลติมีเดียประเภทภาพวีดีโอสามารถควบคุมได้ง่ายและมีประสิทธิภาพมากขึ้น

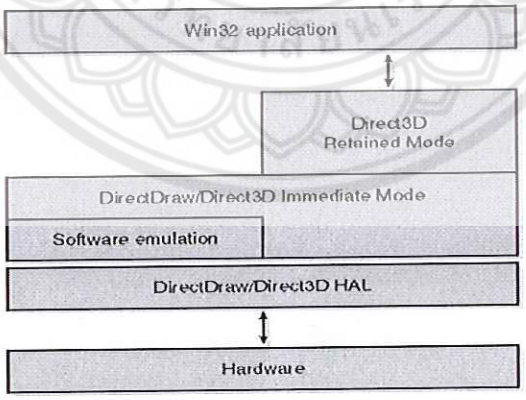
DirectSetup API ชุดนี้จะทำหน้าที่เกี่ยวกับการติดตั้งคอมโพเนนต์ DirectX Runtime ให้กับระบบโดยอัตโนมัติถ้าหากระบบดังกล่าวยังไม่ได้รับการติดตั้ง แม้ในปัจจุบันมักจะมีการติดตั้งคอมโพเนนต์มากับระบบปฏิบัติการแล้วก็ตาม แต่สำหรับ DirectX แล้วไมโครซอฟท์ได้อนุญาตให้ใช้โดยที่ไม่ต้องเสียค่าใช้จ่าย จึงไม่ต้องกังวลเรื่องของลิขสิทธิ์ โดยสามารถทำการติดตั้งกี่ครั้งก็ได้

Direct Media แม้ว่าโปรแกรมเมอร์สามารถเขียนโปรแกรมติดต่อกับ ActiveX Control โดยตรงได้โดยใช้ฟังก์ชันของ COM Interface แต่มันก็ออกจะยุ่งยากไปบ้าง ดังนั้นเพื่อความง่ายในการใช้งาน โปรแกรมเมอร์สามารถเลือกใช้ DirectX Media SDK ซึ่งมีคลาสของ MFC ที่มีชื่อว่า CMediaPlayer ให้โปรแกรมเลือกใช้แทนที่จะเรียก COM Interface โดยตรง ทำให้การใช้งาน Media Player Control ได้ง่ายมากขึ้น

1.4.2 การทำงานของ Direct3D[3]

Direct3D มีขึ้นครั้งแรกใน DirectX 2.0 ประมาณช่วงปี ค.ศ. 1996 หลังจากนั้นภายในระยะเวลาอันสั้น Direct3D ได้กลายเป็นชุด API สำหรับการทำงานที่ได้รับความนิยมอย่างสูงในท้องตลาดที่เกมและซิมมูเลชันส่วนใหญ่เลือกใช้รวมทั้งได้รับการสนับสนุนจากผลิตภัณฑ์เร่งความเร็ว 3 มิติ ปัจจุบันฮาร์ดแวร์เร่งความเร็วกราฟิกพัฒนาไปจนสามารถแสดงโพลีก้อนได้หลายล้านโพลีก้อนต่อวินาที ฮาร์ดแวร์เร่งความเร็วที่นำมาเชื่อมการทำงานกับ Direct3D มีความเร็วมากขึ้น และช่วยลดต้นทุนของแอบพลิเคชัน 3 มิติ ทำให้ไม่ต้องทุ่มทุนสูงนักเพื่อแลกกับประสิทธิภาพในการทำงานที่สูงขึ้น เช่นเดียวกับ API ทั้งหมดของ DirectX ชุด API ของ Direct3D ไม่ได้ออกแบบมาให้เป็นเพียง API สำหรับการพัฒนาเกมเพื่อรองรับความสามารถที่จะมีขึ้นในอนาคตเท่านั้น แต่ยังได้รับการออกแบบมาให้มี ความเข้ากันได้ทั้งหมดกับเวอร์ชันที่ผ่านมา ดังนั้นเกมที่พัฒนาด้วย DirectX เวอร์ชันเก่าก็จะสามารถรันได้ ใน DirectX เวอร์ชันที่จะมีขึ้นในอนาคต

Direct3D ในยุคแรกมี API อยู่สองโหมดคือ โหมด Immediate(IM) และโหมด Retained(RM) ในโหมด IM (การเรนเดอร์อ็อบเจ็กต์สามารถทำได้โดยฉับพลัน ได้ตามความต้องการของโปรแกรมเมอร์) เป็นโหมดที่ใช้งานยากแต่จะมีความยืดหยุ่นสูง เป็น API ในระดับล่างสำหรับใช้เขียนเกมที่ทำงานได้เร็วและมีประสิทธิภาพเท่าที่เครื่องจะเป็นไปได้บนระบบ ในโหมด RM (API จะเก็บซีนลงในฐานข้อมูลแล้วนำมาเรนเดอร์ทั้งหมดในคราวเดียวกัน) เป็นโหมดที่สร้างขึ้นมาเป็นเลเยอร์ ที่อยู่บนสุดของโหมด IM โดยในโหมดนี้จะจัดเตรียมบริการต่าง ๆ เช่น การจัดการ Texture, การโหลด Object File, การจัดลำดับเฟรม และการทำ อ็อบเจกต์เคลื่อนไหว



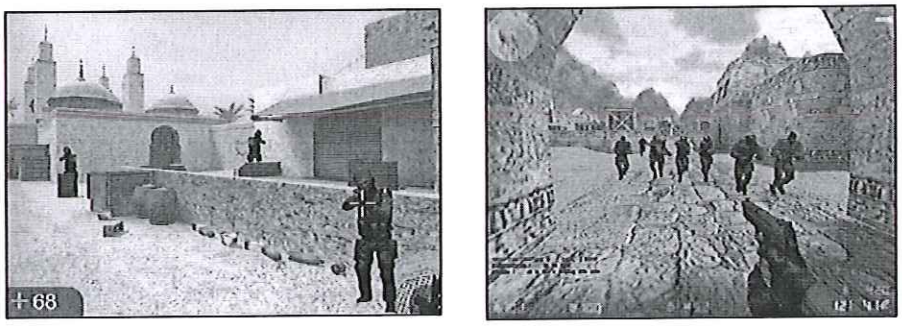
รูปที่ 1.1 แสดงการทำงานของ Direct3D ภายใต้สภาพแวดล้อม Win32 และตัวเร่งความเร็วฮาร์ดแวร์

การศึกษาและการใช้งานโหมด RM นั้นจะง่ายกว่าเมื่อเทียบกับการใช้งานในโหมด IM แต่ถ้าหากโปรแกรมเมอร์คนใดต้องการประสิทธิภาพและความยืดหยุ่นในการทำงานก็มักจะเลือกใช้

การทำงานในโหมด IM ดังนั้นการทำงานในโหมด RM จึงได้หยุดพัฒนาลงใน Direct X version 6.0 และหันไปมุ่งพัฒนาโหมด IM ให้มีความสามารถและใช้งานง่ายในเวอร์ชันต่อมา ด้วยเหตุนี้ การทำงานใน โหมด RM จึงไม่สนับสนุนเทคโนโลยีใหม่ เช่น Multitexturing, Bump Mapping, Hardware Transformation และ Lighting ดังนั้นความสามารถทั้งหมดของโปรแกรม 3 มิติ ควรเขียนขึ้นมาโดยใช้โหมด IM และในการทำงานร่วมกับฮาร์ดแวร์เร่งความเร็วกราฟิกของแอบพลิเคชัน Direct3D จะใช้การติดต่อกับฮาร์ดแวร์ผ่านทาง HAL (Hardware Abstraction Layer) แต่ถ้าในฟีเจอร์นั้นไม่มีในฮาร์ดแวร์Direct3D จะใช้ HEL(Hardware Emulation Layer) แทน

1.4.3. ลักษณะการแสดงแบบมุมมองบุคคลที่หนึ่ง

งานวิจัยนี้ผู้วิจัยมีแรงจูงใจมาจากลักษณะของโปรแกรมเกมเคาน์เตอร์สไตร์ค(Counter-Strike) ซึ่งมีลักษณะการทำงานเป็นรูปแบบ 3 มิติ การแสดงที่เป็นแบบมุมมองบุคคลที่หนึ่ง[9] (First Person Shooter : FPS) ซึ่งจะพบบ่อยในเกมประเภทยิงที่ใช้มุมมองของบุคคลที่หนึ่ง เป็นเกมที่จะต้องปฏิบัติภารกิจตามกำหนด ที่มีจุดเด่นตรงที่สามารถสนุกกับการต่อสู้อย่างน่าตื่นตาด้วยการใช้มุมมองที่ใกล้เคียงกับความเป็นจริง โดยลักษณะภาพที่เห็นนั้นจะถูกจำลองภาพให้เห็นบนหน้าจอมอนิเตอร์เสมือนหนึ่งว่าผู้ใช้กำลังเดินทางไปยังสถานที่นั้นจริงๆ ซึ่งมุมมองที่เห็นก็ถูกกำหนดขึ้นมาตามหลักของแสงและเงาที่ใกล้เคียงกับความเป็นจริงอีกด้วย สามารถหมุนภาพหรือมุมมองได้ 360 องศา กำหนดมุมมองไว้หลายแบบเพื่อไม่ให้ผู้เล่นสับสน โดยใช้การควบคุมจากเมาส์และคีย์บอร์ด ซึ่งในแถบยุโรปกับอเมริกาได้รับความนิยมเป็นอย่างมากด้วย เพื่อให้ทำภารกิจที่ได้รับมอบหมาย การเคลื่อนที่ในเกมจะมีการกำหนดระยะในการกดคีย์แต่ละครั้งว่าจะให้การเคลื่อนที่ได้เท่าไหร่ การประมวลผลจะดีหรือไม่ขึ้นอยู่กับประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้ด้วย เนื่องจากโปรแกรมที่พัฒนาจะมีการใช้พื้นที่ทำงานใน หน่วยประมวลผลกลาง ฮาร์ดดิสค์ หน่วยความจำค่อนข้างสูงมาก ดังนั้นปัญหาของการทำงานในลักษณะนี้ก็คงหนีไม่พ้นในสิ่งเหล่านี้ ถ้ามีไม่เพียงพอก็อาจจะทำงานได้ไม่ดีเท่าที่ควร



รูปที่ 1.2 ตัวอย่างภาพจากเกมเคาน์เตอร์สไตร์ค

1.5 แผนการดำเนินการ

1. ศึกษาการทำงานของ Dark basic professional และสร้างโปรแกรมควบคุมโมเดล 3 มิติอย่างง่าย

ทำการติดตั้ง Dark basic professional โดยศึกษาตัวอย่างการใช้งาน Dark basic professional ที่ให้มาพร้อมกับโปรแกรม ซึ่งเราจะศึกษาในส่วนที่ทำงานโมเดล 3 มิตินามสกุล. X และ นามสกุล .dbo ในส่วนของการสร้างโมเดลอย่างง่าย จะใช้โปรแกรม Google SketchUP6 Pro และ 3D World Studio 5 ในการสร้าง

2. ศึกษาโครงสร้างอาคารคณะวิทยาศาสตร์

ทำการศึกษาโครงสร้างอาคารของคณะวิทยาศาสตร์ รวมทั้งรวมทั้งส่วนประกอบภายนอกต่างๆ เพื่อให้ได้โมเดลที่เสมือนจริง

3. สร้างระบบและพัฒนาระบบ

หลังจากที่ได้ศึกษาการทำงานของ Dark basic professional มาพอสมควร และได้สร้างโมเดลขึ้นมา จาก google sketchup 6 pro, 3dsMax9, 3dworld studio แล้ว เริ่มทำการโปรแกรม Dark basic professional ขึ้นมา โดยเราจะทำการเขียนโปรแกรมระบบการควบคุมดังต่อไปนี้ คือ

- ระบบการค้นหาห้องโดยเดินทางในแนว 3 มิติ
- ระบบการเคลื่อนที่ในแนว 3 มิติ

4. ทดสอบและดำเนินการแก้ไข

หลังจากเขียนระบบเสร็จแล้ว ก็จะทำการทดสอบระบบ เพื่อตรวจหาข้อผิดพลาดและแก้ไขให้ดีขึ้น

5. ทำเอกสารประกอบ

จัดทำคู่มือวิธีการเล่น โดยจัดทำรูปเล่มที่มีความน่าสนใจ และแสดงรายละเอียดของระบบวิธีการใช้งานที่ทำให้ผู้ที่มีความสนใจอ่านเข้าใจง่าย

ตารางที่1.1 ตารางแสดงระยะเวลาแผนการดำเนินการ

ขั้นตอน	ปี2551																							
	เดือน1				เดือน2				เดือน3				เดือน4				เดือน5				เดือน6			
	1	2	3	4	1	2	3	4	1	3	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1																								
2																								
3																								
4																								
5																								

บทที่ 2
การดำเนินงาน

การออกแบบโปรแกรมที่จะสามารถบอดเส้นทางได้นั้น จะต้องมีการออกแบบที่คล้ายสถานที่จริง จึงจะส่งผลให้การใช้งานได้สมจริงมากไปด้วย ดังนั้นโปรแกรมที่จะพัฒนาขึ้นนี้จะมีอาคารบริหารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร เป็นโมเดลสำหรับพัฒนา ซึ่งออกแบบให้เป็น 3 มิติ (3 Dimensions: 3D) สำหรับภายในอาคารก็จะมีการออกแบบให้เสมือนจริงด้วย เพื่อเพิ่มความน่าสนใจให้กับผู้ใช้งาน โดยผู้ใช้งานสามารถเลือกการทำงานได้ 2 แบบคือ แบบแรกคือผู้ใช้เลือกชื่อสถานที่หรือชื่อห้อง เพื่อค้นหาเส้นทางการเดินทางไปยังไปยังสถานที่นั้นๆ แบบที่สองคือผู้ใช้งานสามารถควบคุมการเดินทางไปยังสถานที่ต่างๆได้ด้วยตนเองได้ด้วย โดยผู้วิจัยตจะได้นำเสนอการทำงานต่อไป

2.1 รูปแบบของโปรแกรม

รูปแบบของโปรแกรมที่ผู้วิจัยจะพัฒนาแบ่งเป็น 2 ลักษณะคือ

1. ค้นหาห้องอัตโนมัติ ผู้ใช้สามารถเลือกห้องที่กำหนดไว้ในรายการแล้วระบบจะนำเสนอเป็นภาพเคลื่อนไหวนำท่านไปสู่ห้องที่กำหนดไว้ได้ ในรูปแบบ 3 มิติ

2. การควบคุมด้วยแป้นพิมพ์ ผู้ใช้สามารถควบคุมการทำงานด้วยแป้นพิมพ์โดยใช้คีย์ลูกศรหรือคีย์ W, A, S, D มาควบคุมทิศทาง การทำงานเสมือนเดินเข้าไปภายในตึกจริง และแสดงจุดที่ท่านอยู่ด้วยว่าท่านอยู่ ณ ตำแหน่งใดของตึก และใช้เมาส์ในการบังคับการหมุนหน้าจอของระบบ

2.2 การพัฒนาโปรแกรม

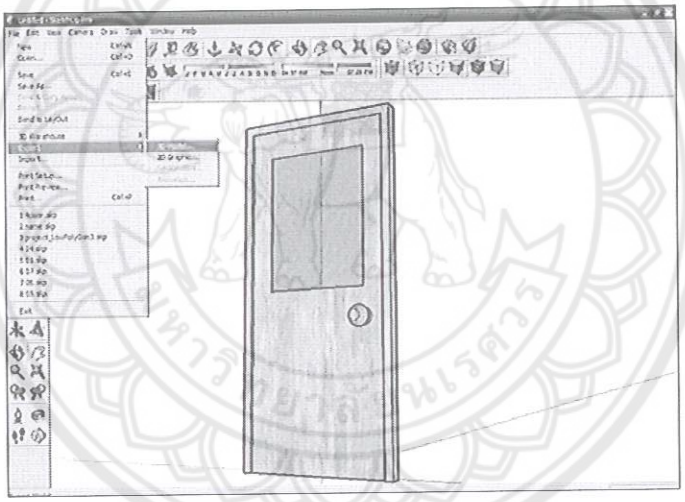
ในการออกแบบและพัฒนาโปรแกรมนั้นยังสามารถแบ่งเป็นทั้งหมด 4 ขั้นตอนเพื่อแสดงให้เห็นถึงวิธีการออกแบบพัฒนาระบบ โดยผู้วิจัยเริ่มจากการออกแบบโมเดลหรือวัตถุต่างๆ การใส่แสงเงา การรวมวัตถุทุกอย่างเข้าด้วยกัน และการเขียนโปรแกรมควบคุมการทำงานจนเป็นโปรแกรมที่นำไปใช้งานได้ต่อไป

รูปที่ 2.1 ขั้นตอนการออกแบบโปรแกรม

2.2.1 **สร้างโมเดล**

1. ในการออกแบบโมเดลอาคารบริหารคณะวิทยาศาสตร์นั้น จะใช้โปรแกรม Sketchup6 Proทำการสร้างส่วนประกอบย่อยๆของตัวอาคารก่อน เช่น ประตู หน้าต่าง ฯลฯ

รูปที่ 2.2 โมเดลที่สร้างด้วยโปรแกรม Sketchup6 Pro



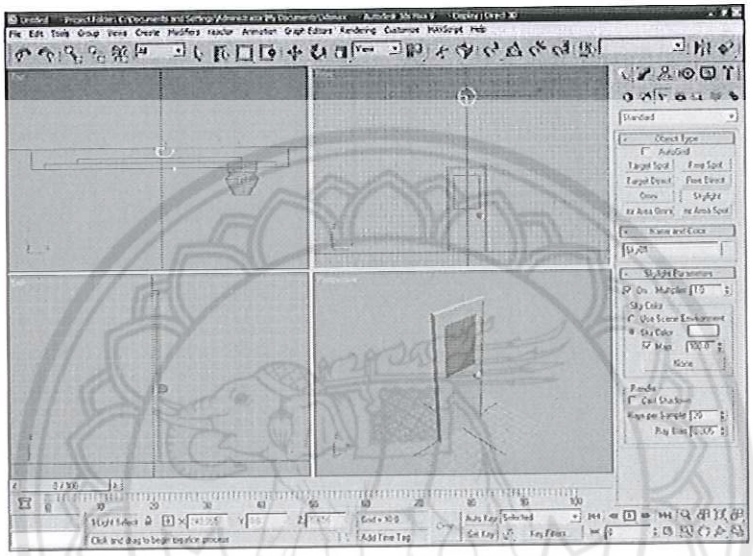
รูปที่ 2.3 หน้าจอโปรแกรม Sketchup6 Pro

2. ทำการ Export โมเดลที่สร้างออกมาโดยใช้นามสกุล .3ds โดยเลือกเฉพาะวัตถุที่สำคัญต่อการมองเห็นเท่านั้น โดยที่สเกลที่ใช้การพัฒนานี้ใกล้เคียงกับของจริง เพื่อให้เหมาะสมกับการมองเห็นในการใช้โปรแกรม

2.2.2 **กำหนดแสงเงา**

1. เมื่อออกแบบวัตถุต่างๆออกมาเป็นชิ้นแล้ว ก็จะทำการใส่แสงเงาด้วยโปรแกรม 3Ds Max 6.0 เพื่อให้โมเดลดูเป็นมิติและสวยงามสมจริง ซึ่งแสงเงาที่กำหนดนั้นก็สามารถปรับได้ตามต้องการเพื่อให้สอดคล้องกับการทำงาน

รูปที่ 2.4 โมเดลที่สร้างแสงและเงาด้วยโปรแกรม 3Ds Max 6.0



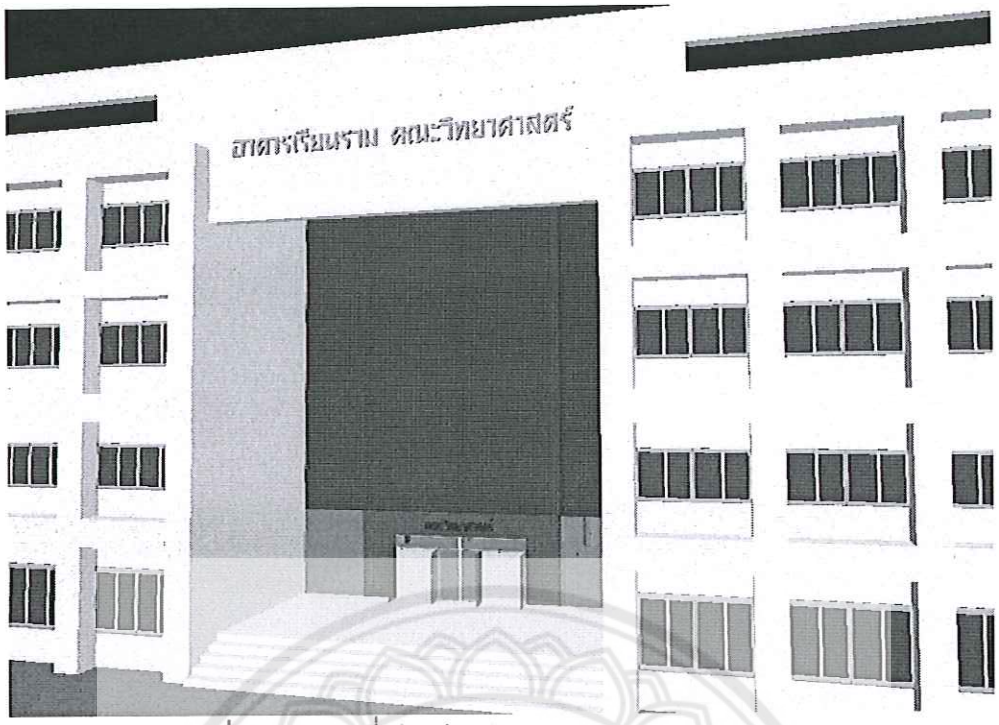
รูปที่ 2.5 หน้าจอโปรแกรม 3Ds Max 6.0

2. จากนั้นทำการ Export โมเดลที่ได้ออกมาโดยใช้นามสกุล .x

2.2.3 รวมวัตถุทั้งหมดเข้าด้วยกัน

1. ทำการสร้างอาคารบริหารคณะวิทยาศาสตร์ ด้วย 3D World Studio 5

2. นำไฟล์โมเดล นามสกุล.x ที่ได้จากส่วนของ "แสงเงา" มาใส่ในโมเดลที่สร้างขึ้นในโปรแกรม 3D world Studio 5



รูปที่ 2.6 โมเดลที่สร้างด้วยโปรแกรม 3D World Studio 5

3. จากนั้นทำการ Export โมเดลที่ได้ออกมาโดยใช้นามสกุล.DBO เพื่อใช้ในโปรแกรม Dark basic Professional[2,3,6] ต่อไป

2.2.4 การเขียนโปรแกรมควบคุม

เขียนโปรแกรมคำสั่งต่างๆ โมเดลนามสกุล.DBO ที่ได้จากส่วนของ"รวม" ในโปรแกรม Dark basic Professional เพื่อใช้เป็นตัวควบคุมบังคับโมเดลให้เป็นไปตามที่ต้องการ (ภาคผนวก ค)

2.3 โครงสร้างของ X File

การนำข้อมูลของ Model มาแสดงในโปรแกรมนั้น เราได้นำ X File มาใช้สำหรับการเก็บ ข้อมูลเหล่านั้น รูปแบบการเก็บข้อมูลของ X File จะแบ่งออกเป็น Template ต่าง ๆ

1. X File Template

Template จะนำมาใช้ในการอธิบายการจัดเก็บกลุ่มข้อมูลเพื่อช่วยในการแปลงไปใช้ในโปรแกรม

Template Form, Name, and UUID

template <template-name> {

<UUID>

<member 1>;

...

<member n>;

[restrictions]

}

template name – เป็นชื่อของ Template สามารมีอักขระที่เป็นได้ทั้งตัวอักษร,ตัวเลขหรือเครื่องหมาย "_" แต่ไม่สารถเริ่มต้นด้วยตัวเลข

UUID (universally unique identifier) – มีรูปแบบเป็น Open Software Foundation's Distributed Computing Environment ปิดด้วยเครื่องหมาย "< >"

template members - ประกอบด้วย data type หรือ array ของ data type. โดยมี data type แสดงดังตารางที่ 2.1

ตารางที่ 2.1 X file DATA TYPE

Type	Size
WORD	16 bits
DWORD	32 bits
FLOAT	IEEE float
DOUBLE	64 bits
CHAR	8 bits
UCHAR	8 bits
BYTE	8 bits
STRING	NULL terminated string

ตัวอย่าง :

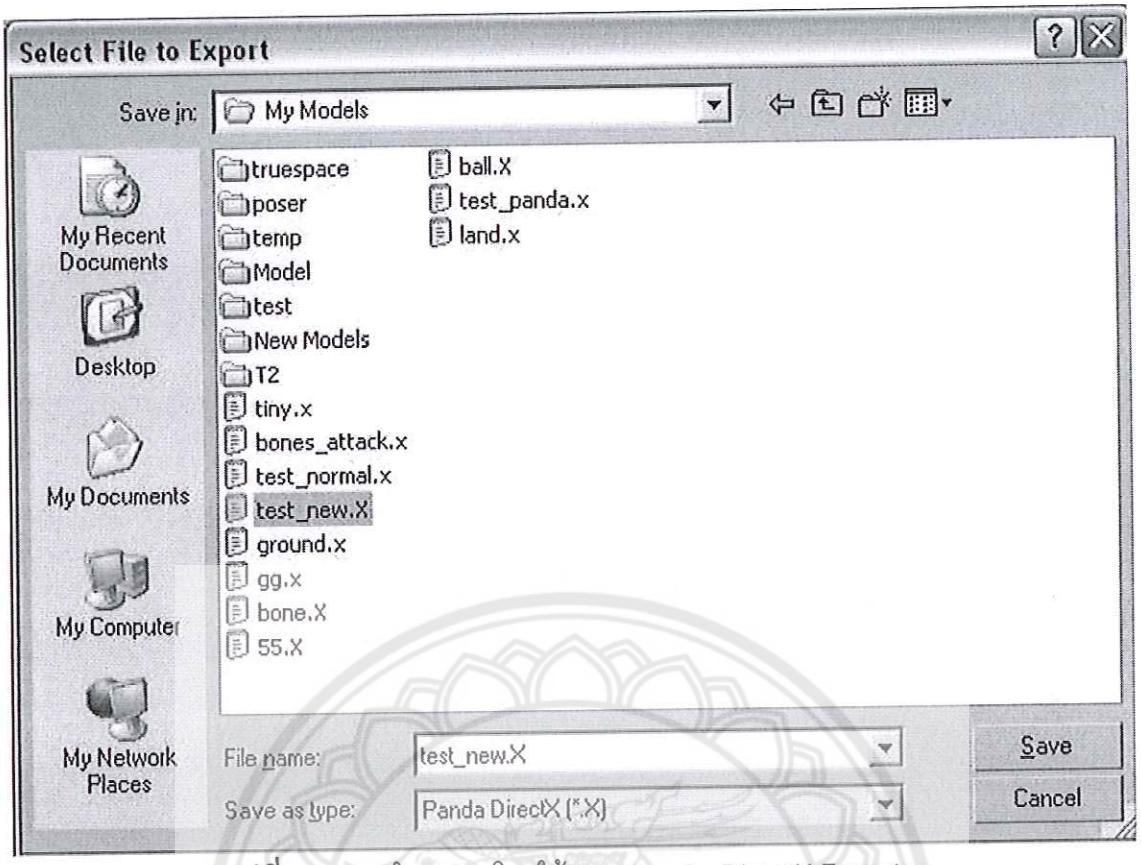
```
template Matrix4x4 {
< F6F23F45-7686-11cf-8F52-0040333594A3 >
array float matrix[16];
}
template SkinWeights {
<6f0d123b-bad2-4167-a0d0-80224f25fabb>
STRING transformNodeName;
DWORD nWeights;
array DWORD vertexIndices[nWeights];
array FLOAT weights[nWeights];
```

```
Matrix4x4 matrixOffset;
}
template FrameTransformMatrix {
< F6F23F41-7686-11cf-8F52-0040333594A3 >
Matrix4x4 frameMatrix;
}
template Frame{
< 3D82AB46-62DA-11cf-AB39-0020AF71E433 >
FrameTransformMatrix frameTransformMatrix;
Mesh mesh;
}
```

2. การแปลงไฟล์โมเดลเป็นไฟล์ .x

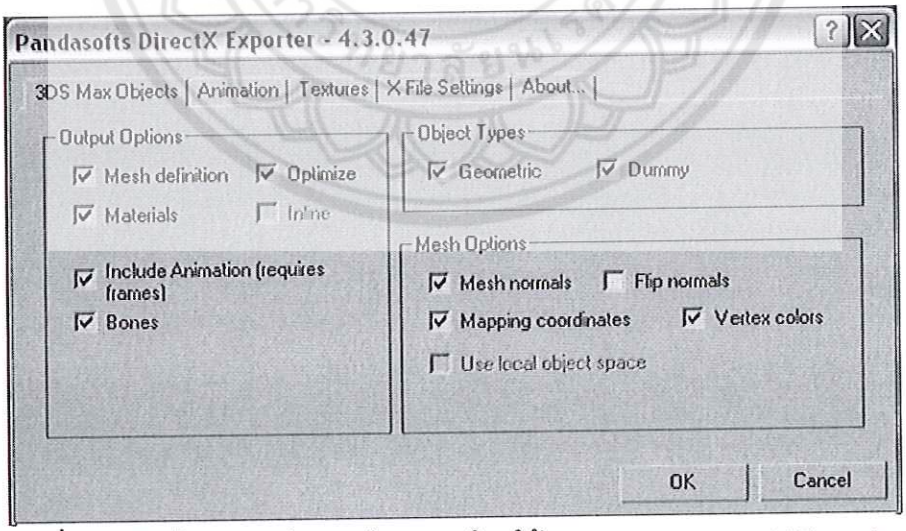
หลังจากทำความเข้าใจเกี่ยวกับ X File Template แล้วต่อไปจะเป็นการแปลงไฟล์โมเดลที่สร้างด้วยโปรแกรม 3D Studio MAX ให้เป็นไฟล์ .x โดยวิธีการแปลงไฟล์นั้นสามารถทำได้หลายวิธีด้วยกัน เครื่องมือที่ทาง Microsoft ให้มาพร้อมกับ DirectX คือ conv3ds.exe และ XSkinExp.dll โดยเครื่องมือทั้ง 2 นี้สามารถศึกษาได้จาก DirectX SDK ส่วนเครื่องมือที่เราได้นำมาใช้คือ Pandasofts DirectX Exporter–4.3.0.47 ซึ่งเป็นปลั๊กอินใน 3D Studio MAX R4.2-5.0 เครื่องมือตัวนี้เป็น Freeware สามารถเข้าไปโหลดได้ที่ http://www.pandasoft.demon.co.uk/directx.htm

การใช้งานเครื่องมือนี้จะต้องนำไฟล์ PandaDXExport5.dle ที่โหลดมานำไปไว้ที่ไดเรกเทอรี pluginsของโปรแกรม 3D Studio MAX 9.0 การใช้งาน Pandasofts DirectX Exporter หลังจากเราสร้างโมเดลเสร็จแล้วให้เข้าไปที่เมนู File เลือกคำสั่ง Export... จะปรากฏหน้าต่าง Select File to Export แล้วชื่อไฟล์ที่จะเซฟ โดยเลือก type เป็น Panda DirectX(*.x) หลังจากเลือกเสร็จแล้วคลิกที่ปุ่ม save

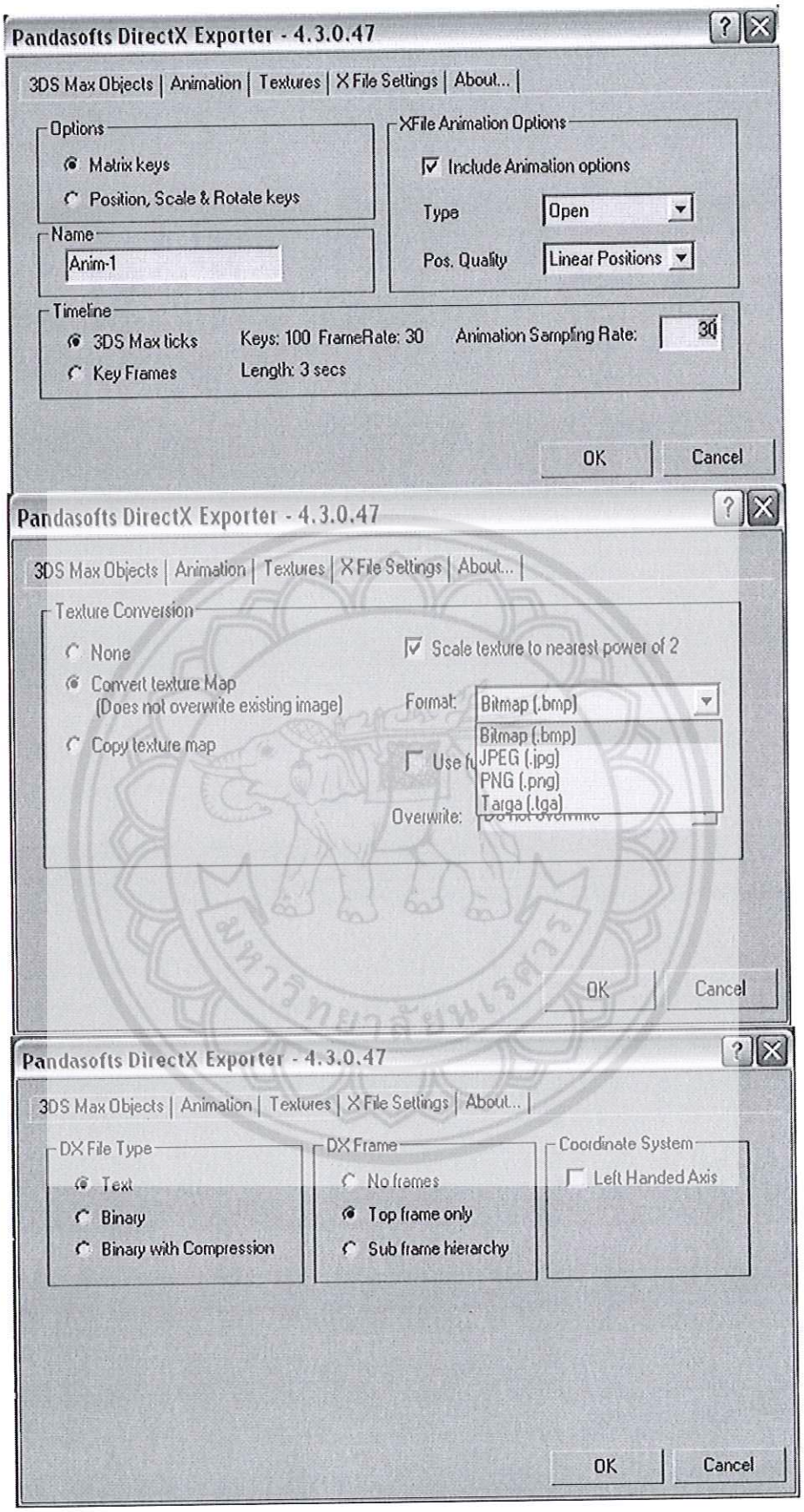


รูปที่ 2.7 การทำ X file โดยใช้ Pandasofts DirectX Exporter

ที่หน้าต่าง Pandasofts DirectX Exporter ก็จะมีออปชันให้เลือกเพื่อกำหนดรูปแบบที่ต้องการ ดังรูปที่ 2.8



รูปที่ 2.8 ตัวเลือกต่างๆ ในการทำ X file โดยใช้ Pandasofts DirectX Exporter



รูปที่ 2.9 ตัวเลือกต่างๆ ในการทำ X file โดยใช้ Pandasofts DirectX Exporter

เมื่อเลือกออปชันได้ตามต้องการแล้วคลิกที่ปุ่ม OK ก็จะได้ไฟล์ .x ที่ต้องการ โดยเราได้เลือกให้แสดงออกมาแสดงในรูป text data เพื่อง่ายในการทำความเข้าใจ ตัวอย่างบางส่วนของไฟล์ .x ที่ได้

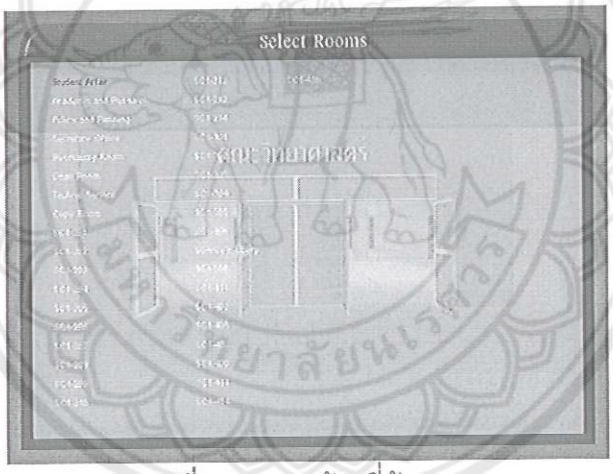
```
Frame Body {
FrameTransformMatrix {
9.088300,0.000000,0.000000,0.000000,
0.000000,9.088300,0.000000,0.000000,
0.000000,0.000000,9.088300,0.000000,
0.000000,0.000000,0.000000,1.000000;;
}
Mesh {
2031;
-1.142503;-4.730632;56.127384;,
-3.024516;-4.518504;55.674118;,
-2.254452;-4.582504;53.072826;,
-4.734245;-3.742142;55.977085;,
-4.568051;-3.930426;55.192978;,
-6.906632;1.809548;54.984364;,
-6.718039;-0.712774;54.880413;,
```

บทที่ 3
ผลการวิจัย

3.1 ผลของการพัฒนาโปรแกรม

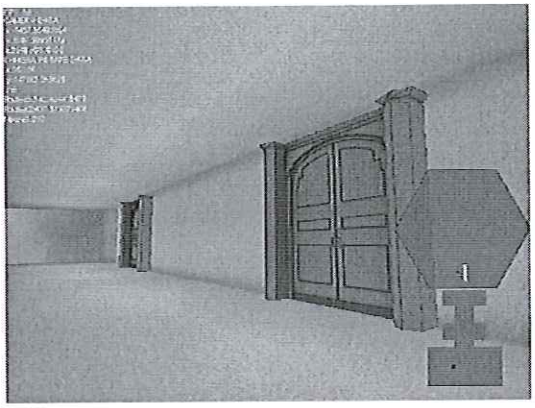
โปรแกรมของผู้วิจัยออกแบบนั้น การทำงานใน 2 ลักษณะคือ การเคลื่อนที่อัตโนมัติ และการควบคุมด้วยแป้นพิมพ์

การทำงานอัตโนมัตินั้นโปรแกรมจะทำการแสดงสถานที่หรือห้องให้เลือกจากเมนู เพื่อให้ผู้ใช้ได้เลือกห้องที่ต้องการ ระบบจะแสดงภาพเคลื่อนที่ไปยังเป้าหมายที่เลือก จากนั้นเมื่อแสดงไปถึงยังเป้าหมายก็จะมีเสียงบอกชื่อห้องเป็นภาษาไทย ด้วย โดยการแสดงเป็นแบบ 3 มิติ ทุกครั้งของการเรียกใช้งานในรูปแบบนี้ จะมีจุดเริ่มต้นอยู่ที่หน้าอาคารบริหารคณะวิทยาศาสตร์เสมอของการเดินทางไปยังห้องเป้าหมายต่างๆ ไม่ว่าจะไปห้องไหนก็ตาม โดยภาพเคลื่อนไหวที่เกิดขึ้นนั้นเป็นการสั่งการทำงานจากโปรแกรมไม่ใช่ไฟล์วิดีโอ ซึ่งจะมีข้อดีตรงที่เมื่อมีการย้ายห้องใหม่ ก็แค่มาปรับทิศทางการเดินใหม่เท่านั้น โปรแกรมก็จะสามารถทำงานได้เหมือนเดิม

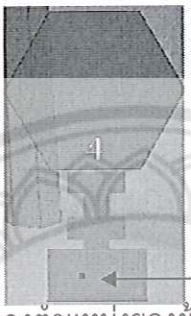


รูปที่ 3.1 แสดงห้องที่ต้องการ

ส่วนการควบคุมด้วยแป้นพิมพ์นั้นโปรแกรมออกแบบมาให้ใช้ปุ่มคีย์บอร์ดโดยใช้คีย์ W, S, A, D หรือ ปุ่มลูกศร ขึ้น-ลง-ซ้าย-ขวา เพื่อใช้ควบคุมหน้าจอและใช้เมาส์เพื่อควบคุมการหมุนของหน้าจอ โดยจะมีภาพแผนที่แบบ 2 มิติ อยู่ตรงมุมขวาด้านล่างตามรูปทรงของอาคารมุมมองจากด้านบนหลังคา ซึ่งจะบอกเลขชั้น จุดที่อยู่ ขณะการเคลื่อนที่ภายในอาคารนั้นด้วย เพื่อไม่ให้ผู้ใช้สับสนตำแหน่งและชั้นที่อยู่ ณ ปัจจุบันของตนเอง คล้ายการเล่นเกมเคาน์เตอร์สไตร์ค ดังรูปที่ 3.2-3.3



รูปที่ 3.2 แสดงการค้นหาเส้นทางด้วยการควบคุมด้วยแป้นพิมพ์



รูปที่ 3.3 แสดงตำแหน่งของผู้ใช้ในแผนที่แบบ 2 มิติ

3.2 การออกแบบการประเมินผล

เมื่อโปรแกรมได้พัฒนาเสร็จแล้ว ผู้วิจัยได้ออกแบบรูปแบบการประเมินการใช้งานของโปรแกรม โดยกำหนดหัวข้อเพื่อประเมินประกอบด้วย ด้านการแสดงภาพที่รวดเร็ว ความสมจริงของการแสดง ความถูกต้องของสถานที่และลักษณะการใช้งานที่เข้าใจง่าย โดยแบ่งความพึงพอใจเป็นเป็น 5 ระดับ มีน้ำหนักคะแนน 5 ถึง 1 คือ ดีมาก ดี ปานกลาง น้อย และควรปรับปรุง มีระดับคะแนนตามลำดับ

3.3 ผลการการประเมิน

จากนั้นได้หาผู้ทดลองใช้ระบบจำนวน 50 โดยแบ่งเป็นกลุ่มคนเป็น 2 กลุ่มคือ นิสิต บุคคลากรที่อยู่ในคณะวิทยาศาสตร์และนิสิต บุคลากรที่อยู่ต่างคณะหรือผู้ที่ไม่เคยมาที่คณะวิทยาศาสตร์ โดยใช้สมการที่จะหาค่าเฉลี่ยของผู้ประเมินดังนี้

สมการผลประเมิน (1)

$$= \frac{1}{N}\sum_{i=1}^{N} q_i$$

จากสมการตัว N คือ จำนวนผู้ทดลองใช้แล้วประเมินผลทั้งหมด ส่วน q คือผลการประเมินเฉลี่ยทุกหัวข้อของผู้ใช้แต่ละราย จากผลการประเมินได้ 4.05 อยู่ในระดับดี ส่วนเบี่ยงเบนมาตรฐานอยู่ที่ 0.48 โดยสามารถอภิปรายผลได้แต่ละหัวข้อดังนี้

ในหัวข้อด้านการแสดงภาพที่รวดเร็วนั้น ผลที่ได้จากการประเมินอยู่ในระดับดี ผู้ประเมินส่วนใหญ่แสดงความเห็นว่าโปรแกรมสามารถแสดงภาพได้รวดเร็วดี

ในหัวข้อความสมจริงของการแสดงผล ผลที่ได้จากการประเมินอยู่ในระดับดี ผู้ประเมินแสดงความเห็นว่าการแสดงภาพสมจริงดี บอกได้ว่าสิ่งที่เห็นคืออะไร

ในหัวข้อด้านความถูกต้องของสถานที่ ผลที่ได้จากการประเมินอยู่ในระดับดี ผู้ประเมินส่วนใหญ่แสดงความเห็นว่าสถานที่ที่แสดงในโปรแกรมนั้น สามารถบอกสถานที่ได้ถูกต้องตามสถานที่จริง

ในหัวข้อด้านลักษณะการใช้งานที่เข้าใจง่าย ผลที่ได้จากการประเมินอยู่ในระดับดี ผู้ประเมินแสดงความเห็นว่าโปรแกรมสามารถใช้งานได้ เข้าใจหลักการทำงานของเมนู สามารถใช้งานในการควบคุมการทำงานได้ง่าย

บทที่ 4
บทสรุปและข้อเสนอแนะ

4.1 บทสรุป

งานวิจัยที่ได้พัฒนาโปรแกรมค้นหาสถานที่ภายในคณะวิทยาศาสตร์แบบ 3 มิติขึ้น โดยใช้รูปแบบคล้ายเกมมาประยุกต์เป็นโปรแกรมที่มีประโยชน์ โดยสามารถทำการพัฒนาโปรแกรมออกมาได้เป็นผลสำเร็จและจากการทดลองใช้งานพบว่าโปรแกรมสามารถแก้ไขปัญหาการค้นหาสถานที่ได้ดี แสดงรายละเอียดได้ชัดเจน สวยงามเสมือนจริง มีเสียงบอกชื่อสถานที่เมื่อถึงเป้าหมายได้ โดยโปรแกรมสามารถทำงานได้ 2 ลักษณะได้แก่ การทำงานแบบอัตโนมัติ เมื่อเลือกสถานที่เป้าหมายโปรแกรมจะแสดงเส้นทางไปยังเป้าหมายให้แบบอัตโนมัติ ในรูปแบบคล้ายวิดีโอ ส่วนการทำงานอีกแบบหนึ่งคือการควบคุมด้วยมือสามารถควบคุมทิศทางการทำงานด้วยการใช้คีย์บนแป้นพิมพ์และเมาส์ ได้ ซึ่ง ณ ตอนนี้สามารถนำโปรแกรมไปใช้กับระบบปฏิบัติการวินโดว์ได้

ผลการประเมินประสิทธิภาพที่ได้จากผู้ทดลองใช้โปรแกรมจำนวน 50 คน โดยใช้แบบสอบถามแบ่งเป็นระดับการประเมินเป็น 5 ระดับซึ่งได้ผล 4.05 อยู่ในระดับดี ส่วนเบี่ยงเบนมาตรฐานอยู่ที่ 0.48

4.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับผู้ที่สนใจการพัฒนาค้นหาสถานที่แบบ 3 มิตินี้ ถ้าจะให้สมจริงมากขึ้นสเกลของวัตถุต่างๆที่ออกแบบควรจะมีขนาดตามสัดส่วนของจริง และการแสดงผลในรูปแบบ 3มิติ ยังต้องใช้สภาพแวดล้อมสำหรับการทำงานของคอมพิวเตอร์ที่ค่อนข้างสูง ถ้ามีแนวทางในการพัฒนาให้อยู่ในรูปแบบที่สามารถใช้งานในระบบที่ไม่สูงมากนักก็จะเป็นการดี จะทำให้สามารถนำไปใช้กับคอมพิวเตอร์ที่มีข้อจำกัดในด้านการประมวลผลที่ต่ำได้ง่ายขึ้น

บรรณานุกรม

[1] สัญญา เครือหงษ์ สุรางค์รัตน์ เชาว์โคกสูง, "การพัฒนาเกมแนวต่อสู้โดยใช้ชุดคำสั่งซีดีเอ็กซ์เอนจินมหาสารคามวิจัยครั้งที่ 3, 6-7 ก.ย. 2550

[2] Jonathan S. Harbour, Joshua R. Smith 2003 Premier – Beginner's Guide to Dark BASIC Game Programming, Stacy L. Hiquet

[3] http://www.interior3d.th.gs/

[4] http://www.tek-tips.com/

[5] http://www.gggraphics.co.uk/

[6] http://www.exteen.com/tag/darkbasic/

[7] http://www.thai3d.net/

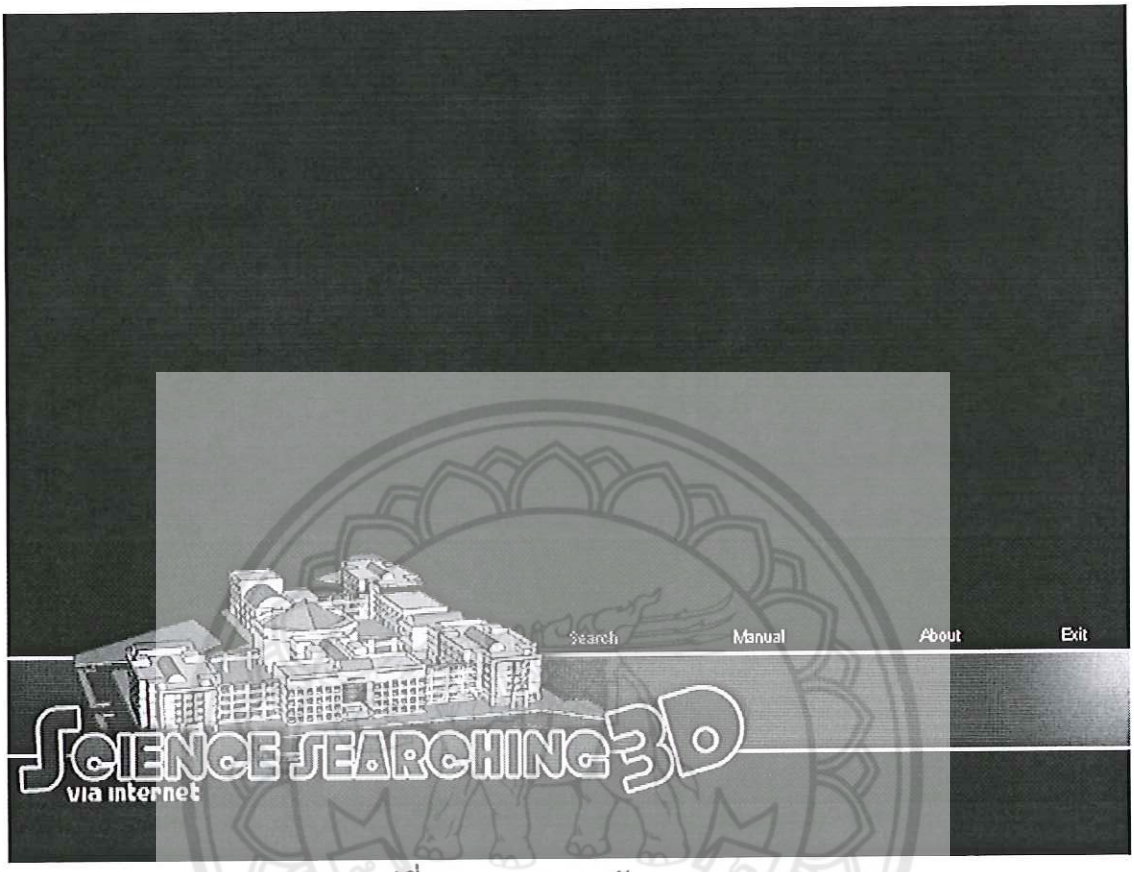
[8] http://www.thaigameadvance.com/

[9] http://th.wikipedia.org/

ภาคผนวก ก

การใช้งานโปรแกรม

เริ่มต้นโปรแกรมก็จะพบกับหน้าจอดังรูปที่ ก.1

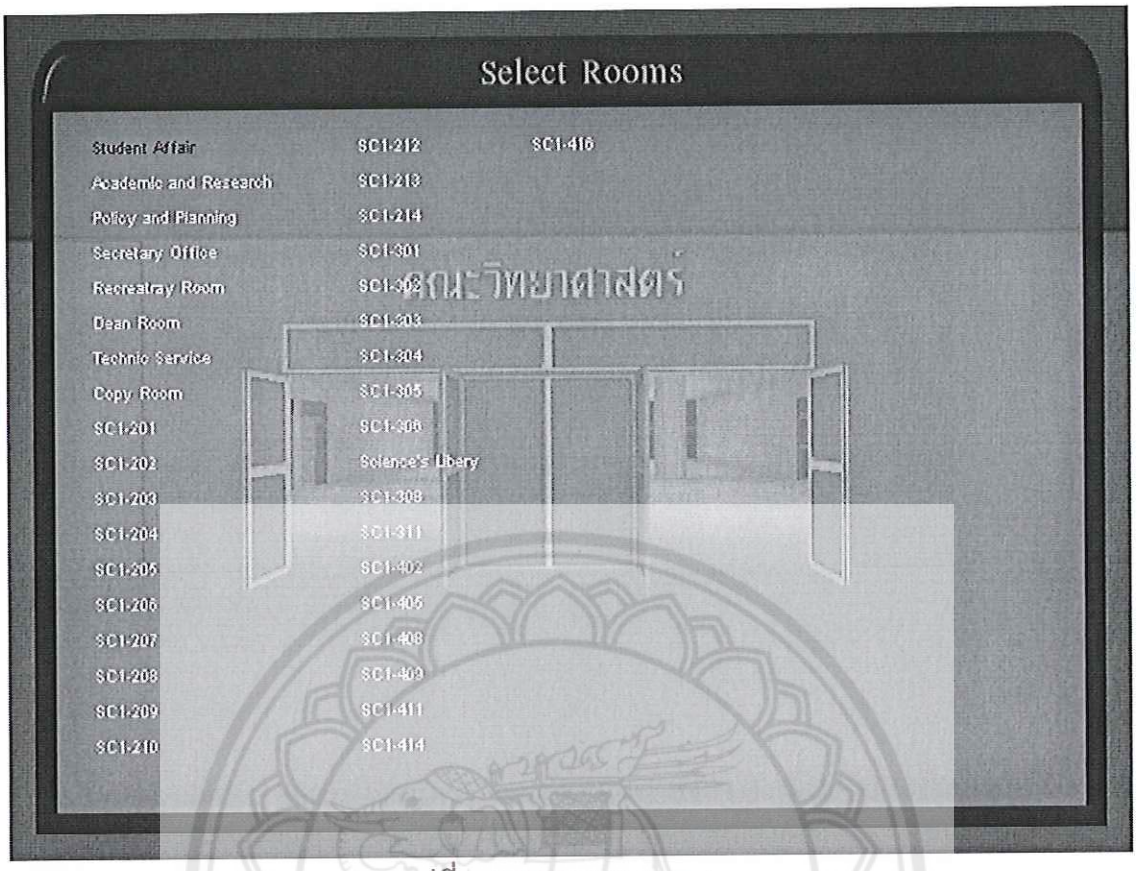


รูปที่ ก.1 แสดงเมนูหลักของระบบ

การเลือกเมนู ทำได้โดยการเลื่อนลูกศร ซ้าย-ขวาจากแป้นพิมพ์ กดปุ่ม Enter เพื่อทำการตกลงเลือกเมนูๆ มีทั้งหมด 4 ตัวเลือกคือ

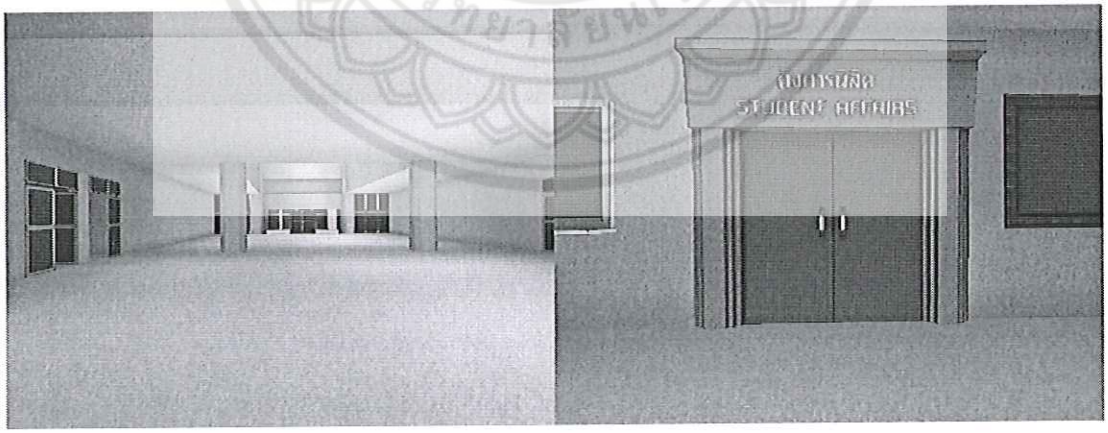
1. Search : ค้นหาสถานที่โดยเลือกจากรายชื่อสถานที่
2. Manual : ค้นหาสถานที่ต่างๆ ด้วยตนเอง
3. About : เกี่ยวกับโปรแกรม
4. Exit : ออกจากโปรแกรม

1. Menu search



รูปที่ ก.2 แสดงเมนู Search

ใช้ลูกศรบนแป้นพิมพ์ ขึ้น-ลง เพื่อเลือกชื่อห้องของอาคารบริหารคณะวิทยาศาสตร์ หลังจากนั้น กด Enter เพื่อทำการตกลงเลือกห้องจากนั้นระบบจะทำการแสดงเส้นทางเดินไปยังห้องที่เลือกดังรูปที่ ก.3

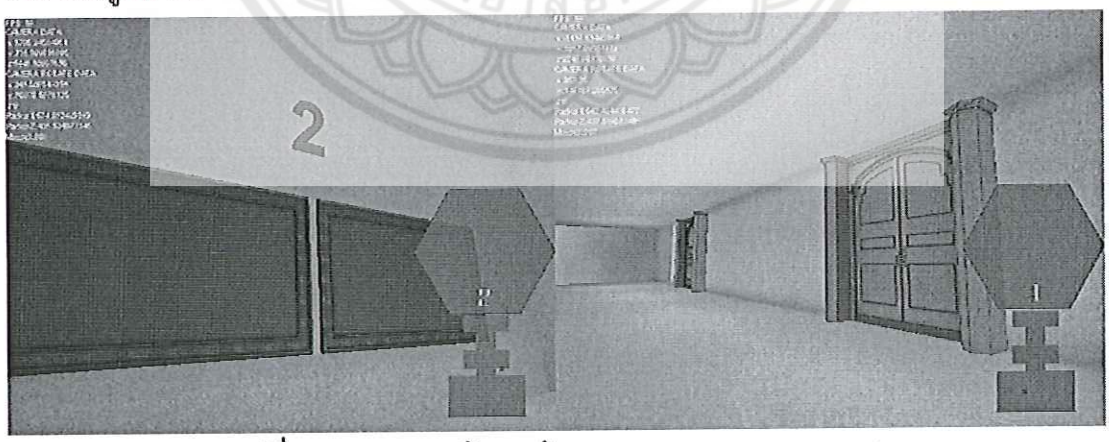


รูปที่ ก.3 แสดงการค้นหาเส้นทางการเดินทางของระบบแบบเลือกชื่อสถานที่

2. Menu Manual

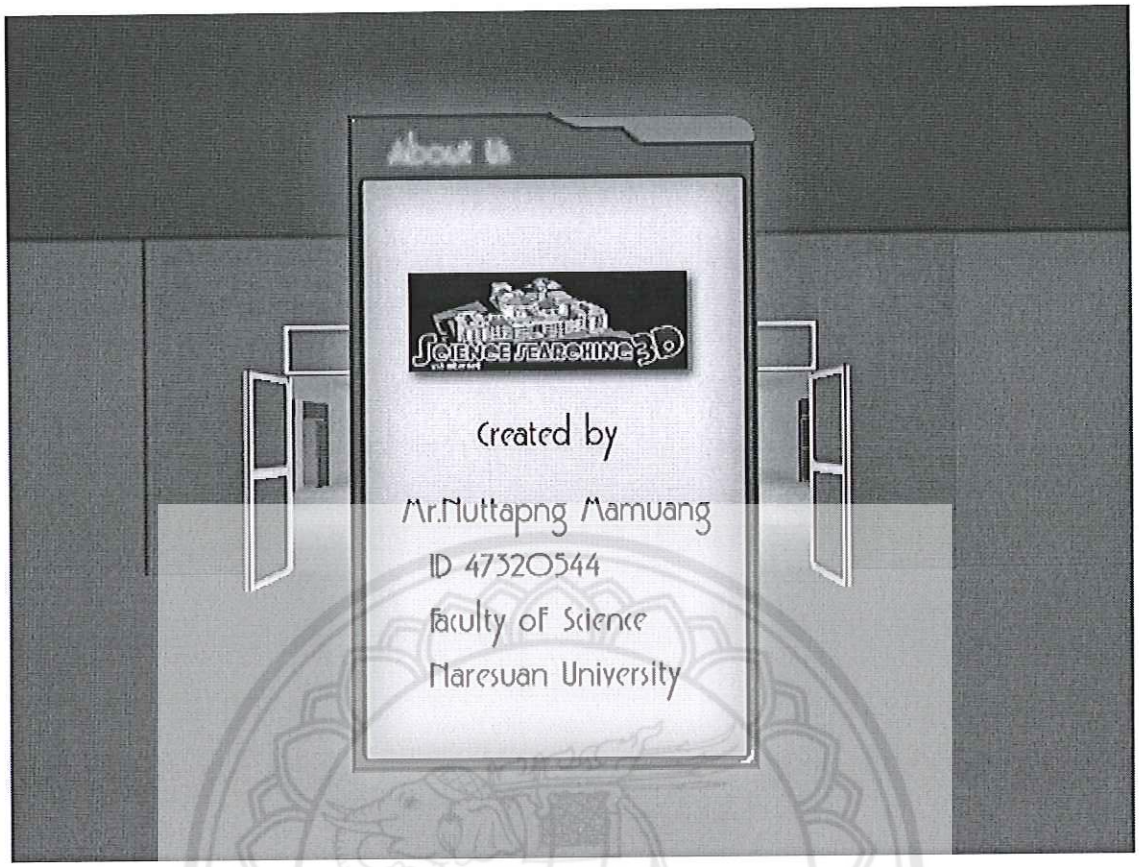
รูปที่ ก.4 แสดงเมนู Manual

ใช้ปุ่ม W, A, S, D หรือ ปุ่มลูกศร ขึ้น-ลง-ซ้าย-ขวา เพื่อควบคุมหน้าขอ และ ใช้เมาส์เพื่อควบคุมการหมุนของหน้าจอ โดยจะมี แผนที่ 2 มิติบอกถึงจุดที่อยู่ และเลขชั้นขณะนั้นภายในอาคารดังรูปที่ ก.5



รูปที่ ก.5 แสดงการค้นหาเส้นทางของระบบแบบ Manual

3. Menu About



รูปที่ ก.6 แสดงเมนู about

แสดงเกี่ยวกับรายละเอียดของโปรแกรม สามารถกลับไปยังเมนูหลักด้วยการกดปุ่ม Escape ที่แป้นพิมพ์

4. Menu Exit

ใช้ในการออกจากโปรแกรม และในทุกเมนู สามารถออกจากเมนูนั้นๆ ไปยัง Main Menu ทำได้โดยการกดปุ่ม Escape

ว TK
7462
.T47
ต1275
2551
1 5606779

สำนักหอสมุด
23 ส.ค. 2554

ภาคผนวก ข

ตัวอย่างแบบสอบถาม
ระบบค้นหาสถานที่แบบ 3 มิติ กรณีศึกษา อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

แบบสอบถามฉบับนี้ เป็นเครื่องมือสำหรับเก็บรวบรวมข้อมูลเกี่ยวกับความพึงพอใจของผู้ทดลองใช้ระบบค้นหาสถานที่แบบ 3 มิติ กรณีศึกษา อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร เพื่อเป็นข้อมูลในการปรับปรุงและพัฒนาระบบให้มีประสิทธิภาพมากขึ้น จึงขอความร่วมมือจากทุกท่านกรุณาตอบความเป็นจริง

❖ ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

1.1 เพศ ☐ชาย ☐หญิง

1.2 อายุ ☐ 16-20 ปี ☐ 21-30 ปี ☐ 30 ปีขึ้นไป

1.3 ตำแหน่ง ☐นิสิต -นักศึกษา ☐อาจารย์ ☐เจ้าหน้าที่ อื่นๆ.....

❖ ตอนที่ 2 ความพึงพอใจในการทดลองใช้ระบบ

คำชี้แจง กรุณาใส่เครื่องหมาย ✓ ลงในช่องระดับความพึงพอใจ

ที่	รายการ	ระดับความพึงพอใจ				
		ดีมาก	ดี	ปานกลาง	น้อย	ควรปรับปรุง
1	ด้านการแสดงภาพที่รวดเร็ว					
2	ความสมจริงของการแสดง					
3	ความถูกต้องของสถานที่					
4	ลักษณะการใช้งานที่เข้าใจง่าย					

❖ ตอนที่ 3 ข้อเสนอแนะ

.....

.....

.....

ขอขอบคุณในความร่วมมือ

ภาคผนวก ค

ตัวอย่างโค้ดโปรแกรมด้วย Dark Basic Professional

ระบบค้นหาสถานที่แบบ 3 มิติ กรณีศึกษา อาคารคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

1. เซตค่าหน้าจอด้วยคำสั่งต่าง ๆดังนี้

```
sync on
sync rate 0
hide mouse
disable escapekey
set gamma 511,511,511
  SET DISPLAY MODE 640,480,32
Autocam off
Set Camera Range 10,10000
color backdrop rgb(0,0,0)
backdrop on
set ambient light 1000
scale listener 100
```

2. ทำการโหลดไฟล์ โมเดล.dbo ที่สร้างจาก 3D world Studio 5.0

```
file$=cl$()
file$=replace$(file$,chr$(34),"")
if file$=""
  file$="search6\search6.dbo"
endif
cls
sync
```

3. สร้าง splash picture

```
splashdelay=3000
if escape=0
  load image "logo_SCI.bmp",1
  cls
  paste image 1,(screen width()-512)/2,(screen height()-152)/2
  sync
  for n=1 to splashdelay
    wait 1
    if escapekey()=1 or mouseclick()=1
      escape=1
      exit
    endif
  next
  delete image 1
endif
if escape=0
  load image "title.bmp",1
  cls
  paste image 1,(screen width()-429)/2,(screen height()-150)/2
  sync
  for n=1 to splashdelay
    wait 1
    if escapekey()=1 or mouseclick()=1
      escape=1
      exit
    endif
  next
  delete image 1
endif
```

4 สร้างข้อความ" Loading...Component"

```
cls
ink rgb(255,255,255),rgb(255,255,255)
text screen width()/2-20,screen height()/2,"Loading...Component"
sync
```

5. สร้างตัวแปลหลักทั้งหมดสำหรับเกมส์(นำมาจากคู่มือ)

```
#Constant TYPE_NGC_ELLIP=1
#Constant TYPE_NGC_MESH=2
#Constant ELLIP_2_ELLIP=1
#Constant ELLIP_2_POLY=2
#Constant RESP_STICK=1
#Constant RESP_SLIDE=2
#Constant RESP_SLIDE_NO_SLOPES=3
#Constant RESP_SLIDE_NO_GRAV=4
#Constant RESP_NONE=5
#Constant DYN_NO_RESP=1
#Constant DYN_RESP=2
#Constant DYN_RESP_LOCK=3
#Constant DYN_RESP_LOCK_NOTURN=4
#Constant TYPE_PLAYER=10
#Constant TYPE_BRUSH=11
#Constant TYPE_TERRAIN=12

global world
global gamespeed#=1.0
global camera=0
global playerobject=1
global player=1
global cam_speed#=6.0
global camerasmoothx#
```

```
global camerasmoothy#
global smoothmove#
global smoothstrafe#
global gravity#=-10.0
global newobjectcounter=1000
global skyenabled
global playerpitch#
global playeryaw#

type flare
object
radius#
color
range#
nextflare
targetalpha#
alpha#
endtype
global dim flares(100) as flare
global countflares
StartCollisionPRO(0,0,0)
SetCollisionsPro(TYPE_PLAYER,TYPE_BRUSH,ELLIP_2_POLY,RESP_SLIDE,DYN_NO_RESP)
SetCollisionsPro(TYPE_PLAYER,TYPE_TERRAIN,ELLIP_2_POLY,RESP_SLIDE,DYN_NO_RESP)
make object sphere playerobject,2
position object playerobject,0,128,0
hide object playerobject
CollisionTypePRO(playerobject,TYPE_PLAYER)
SetObjRadiusPRO(playerobject,32,80,32)
```

6. สร้างท้องฟ้า

```
sky=NewObjectIndex()
if skyenabled=0
  load object "models\skydome.x",sky
  scale object sky,1000000,1000000,1000000
  set object light sky,0
  set object ambient sky,0
  set object fog sky,0
endif
```

7. สร้าง Main menu

```
main:
cls
sync
  position camera 0,150,0
  Rotate camera 0,0,0
  load image "search4\title1.bmp",1000
  set text size 12
  blue=RGB(0,128,255)
  white=RGB(255,255,255)
  selectedItem=1
  do
    cls
    paste image 1000,0,300
    if selectedItem=1 then ink blue,0 else ink white,0
    text 320,350,"Search"

    if selectedItem=2 then ink blue,0 else ink white,0
    text 415,350,"Manual"

    if selectedItem=3 then ink blue,0 else ink white,0
```

```
text 520,350,"About"
if selectedItem=4 then ink blue,0 else ink white,0
text 600,350,"Exit"
 if leftkey()=1 and hold=0 then dec selectedItem : hold=1
if rightkey()=1 and hold=0 then inc selectedItem : hold=1
if scancode()=0 then hold=0
if selectedItem>4 then selectedItem=1
if selectedItem<1 then selectedITem=4

 if returnkey()=1 and hold=0
  hold=1
  if selectedItem=1 then goto choice1
  if selectedItem=2 then goto choice2
  if selectedItem=3 then goto about
 if selectedItem=4
    cls
    end
  endif

 endif
 sync
loop
return
```

8. สร้าง Searching past

```
choice1:
do
  position camera 0,150,0
  Rotate camera 0,0,0
  position camera camera,object position x(playerobject),object position
y(playerobject)+60,object position z(playerobject)
```

```
  position listener camera position x(),camera position y(),camera position z()
  rotate listener camera angle x(),camera angle y(),camera angle z()
  if object exist(sky)
    position object sky,camera position x(),-100,camera position z()
  endif
framerate=screen fps()
  gamespeed#=60.0/framerate
Rem make radar
  x#=camera position x(0)
  y#=camera position y(0)
  z#=camera position z(0)
  PRX#=(screen width()-83)+(X#/100)
  PRZ#=(screen height()-20)-(Z#/100)
  fastsync
    load image "search4\selected1.bmp",1001
    paste image 1001,0,0
    if selectedItem=1 then ink black,0 else ink white,0
    text 50,70,"Student Affair"
    if selectedItem=2 then ink black,0 else ink white,0
    text 50,90,"Academic and Research"
    if selectedItem=3 then ink black,0 else ink white,0
    text 50,110,"Policy and Planning"
    if selectedItem=4 then ink black,0 else ink white,0
    text 50,130,"Secretary Office"
    if selectedItem=5 then ink black,0 else ink white,0
    text 50,150,"Recreatray Room"
    if selectedItem=6 then ink black,0 else ink white,0
    text 50,170,"Dean Room"
    if selectedItem=7 then ink black,0 else ink white,0
    text 50,190,"Technic Service"
    if selectedItem=8 then ink black,0 else ink white,0
```

```
text 50,210,"Copy Room"
if selectedItem=9 then ink black,0 else ink white,0
text 50,230,"SC1-201"
if selectedItem=10 then ink black,0 else ink white,0
text 50,250,"SC1-202"
if selectedItem=11 then ink black,0 else ink white,0
text 50,270,"SC1-203"
if selectedItem=12 then ink black,0 else ink white,0
text 50,290,"SC1-204"
if selectedItem=13 then ink black,0 else ink white,0
text 50,310,"SC1-205"
if selectedItem=14 then ink black,0 else ink white,0
text 50,330,"SC1-206"
if selectedItem=15 then ink black,0 else ink white,0
text 50,350,"SC1-207"
if selectedItem=16 then ink black,0 else ink white,0
text 50,370,"SC1-208"
if selectedItem=17 then ink black,0 else ink white,0
text 50,390,"SC1-209"
if selectedItem=18 then ink black,0 else ink white,0
text 50,410,"SC1-210"
if selectedItem=19 then ink black,0 else ink white,0
text 200,70,"SC1-212"
if selectedItem=20 then ink black,0 else ink white,0
text 200,90,"SC1-213"
if selectedItem=21 then ink black,0 else ink white,0
text 200,110,"SC1-214"
if selectedItem=22 then ink black,0 else ink white,0
text 200,130,"SC1-301"
if selectedItem=23 then ink black,0 else ink white,0
text 200,150,"SC1-302"
```

```
if selectedItem=24 then ink black,0 else ink white,0
text 200,170,"SC1-303"
if selectedItem=25 then ink black,0 else ink white,0
text 200,190,"SC1-304"
if selectedItem=26 then ink black,0 else ink white,0
text 200,210,"SC1-305"
if selectedItem=27 then ink black,0 else ink white,0
text 200,230,"SC1-306"
if selectedItem=28 then ink black,0 else ink white,0
text 200,250,"Science's Libery"
if selectedItem=29 then ink black,0 else ink white,0
text 200,270,"SC1-308"
if selectedItem=30 then ink black,0 else ink white,0
text 200,290,"SC1-311"
if selectedItem=31 then ink black,0 else ink white,0
text 200,310,"SC1-402"
if selectedItem=32 then ink black,0 else ink white,0
text 200,330,"SC1-405"
if selectedItem=33 then ink black,0 else ink white,0
text 200,350,"SC1-408"
if selectedItem=34 then ink black,0 else ink white,0
text 200,370,"SC1-409"
if selectedItem=35 then ink black,0 else ink white,0
text 200,390,"SC1-411"
if selectedItem=36 then ink black,0 else ink white,0
text 200,410,"SC1-414"
if selectedItem=37 then ink black,0 else ink white,0
text 300,70,"SC1-416"
if upkey()=1 and hold=0 then dec selectedItem : hold=1
if downkey()=1 and hold=0 then inc selectedItem : hold=1
if scancode()=0 then hold=0
```

```
    if selectedItem>37 then selectedItem=1
    if selectedItem<1 then selectedITem=37
     if returnkey()=1 and hold=0
       hold=1
       if selectedItem=1
position camera 0,150,0
  Rotate camera 0,0,0
        BackCamX(260)
        GoCamZ(2600)
        LotL(90)
        BackCamX(1800)
        LotL(90)
         wait key
       endif
       if selectedItem=2
position camera 0,150,0
  Rotate camera 0,0,0

        BackCamX(360)
        GoCamZ(2300)
        LotL(90)
        BackCamX(1800)
        LotR(90)
         wait key
       endif
       if selectedItem=3
position camera 0,150,0
  Rotate camera 0,0,0
        GoCamX(170)
        GoCamZ(2400)
        LotR(90)
```

```
        GoCamX(1925)
        LotL(90)
          wait key
       endif
       if selectedItem=4
position camera 0,150,0
  Rotate camera 0,0,0
       GoCamX(170)
        GoCamZ(2500)
        LotR(90)
        GoCamX(1925)
        LotR(90)
          wait key
       endif
       if selectedItem=5
position camera 0,150,0
  Rotate camera 0,0,0
         GoCamX(170)
         GoCamZ(2400)
        LotR(90)
        GoCamX(3441)
        LotL(90)
          wait key
       endif
       if selectedItem=6
position camera 0,150,0
  Rotate camera 0,0,0
         GoCamX(170)
         GoCamZ(2500)
          LotR(90)
          GoCamX(3441)
```

```
        LotR(90)
        wait key
      endif
if selectedItem=7
position camera 0,150,0
  Rotate camera 0,0,0
        GoCamX(170)
        GoCamZ(2500)
        LotR(90)
        GoCamX(3441)
        LotR(90)
        wait key
      endif
if selectedItem=8
position camera 0,150,0
  Rotate camera 0,0,0
        GoCamX(170)
        GoCamZ(2500)
        LotR(90)
        GoCamX(3441)
        LotR(90)
        wait key
      endif
      if selectedItem=9
position camera 0,150,0
  Rotate camera 0,0,0
        GoCamX(218)
        GoCamZ(5461)
        LotR(90)
        GoCamX(1065)
        LotL(90)
```

```
        Goup1(6200)
        LotR(90)
        GoCamX(1898)
        LotR(90)
        GoUp2(5374)
        LotR(90)
        NBackCamX(716)
        LotL(90)
        BackCamZ(2467)
        LotL(90)
        GoCamX(1275)
        LotL(90)
        wait key
      endif
      if selectedItem=14
position camera 0,150,0
  Rotate camera 0,0,0
        GoCamX(218)
        GoCamZ(5461)
        LotR(90)
        GoCamX(1065)
        LotL(90)
        Goup1(6200)
        LotR(90)
        GoCamX(1898)
        LotR(90)
        GoUp2(5374)
        LotR(90)
        NBackCamX(716)
        LotL(90)
        BackCamZ(2567)
```

```
            LotL(90)
            GoCamX(1275)
            LotR(90)
            wait key
        endif

    endif
if escapekey()=1
        goto main
  endif
loop
return
```

9. สร้าง Manual past

```
choice2:
do
  dx#=mousemovex()*0.25
  dy#=mousemovey()*0.25
  camerasmoothx#=curveangle(dx#,camerasmoothx#,3.0)
  camerasmoothy#=curveangle(dy#,camerasmoothy#,3.0)
  playerpitch#=wrapvalue(playerpitch#+camerasmoothy#)
  if playerpitch#>90.0 and playerpitch#<180.0 then playerpitch#=90.0
  if playerpitch#>180.0 and playerpitch#<270.0 then playerpitch#=270.0
  playeryaw#=playeryaw#+camerasmoothx#
  Rotate Object playerobject,0,playeryaw#,0
  move=0
  strafe=0
  if UpKey()=1 or keystate(17)=1 then move=move+2
  if downkey()=1 or keystate(31)=1 then move=move-2
  if rightkey()=1 or keystate(32)=1 then strafe=strafe+2
  if leftkey()=1 or keystate(30)=1 then strafe=strafe-2
```

```
  movespeed#=cam_speed#
  rem if strafe<>0 and move<>0 then movespeed#=sqrt(cam_speed#)
  movespeed#=movespeed#*gamespeed#
  smoothmove#=curvevalue(move,smoothmove#,6)
  smoothstrafe#=curvevalue(strafe,smoothstrafe#,6)
  move object playerobject,movespeed#*smoothmove#
  move object right playerobject,movespeed#*smoothstrafe#
  Rotate camera camera,playerpitch#,playeryaw#,0
  position object playerobject,object position x(playerobject),object position y(playerobject)-
6.0*gamespeed#,object position z(playerobject)
  RunCollisionPRO()

  position camera camera,object position x(playerobject),object position
y(playerobject)+60,object position z(playerobject)
  position listener camera position x(),camera position y(),camera position z()
  rotate listener camera angle x(),camera angle y(),camera angle z()
  if object exist(sky)
    position object sky,camera position x(),-100,camera position z()
  endif
framerate=screen fps()
  gamespeed#=60.0/framerate
  text 0,0,"FPS: "+str$(framerate)
  x#=camera position x(0)
  y#=camera position y(0)
  z#=camera position z(0)
  PRX#=(screen width()-83)+(X#/100)
  PRZ#=(screen height()-20)-(Z#/100)
  Circle PRX#,PRZ#,1

load image "search4\2dmap.bmp",1
  sprite 1,(screen width()-165),(screen height()-280),1
```

```
  set sprite alpha 1,200
 load image "search4\dot.bmp",2
  sprite 2,PRX#,PRZ#,2

load image "search4\1.bmp",3
load image "search4\2.bmp",4
load image "search4\3.bmp",5
load image "search4\4.bmp",6
load image "search4\T.bmp",7
sprite 3,(screen width()-100),(screen height()-170),3
sprite 4,(screen width()-100),(screen height()-170),4
sprite 5,(screen width()-100),(screen height()-170),5
sprite 6,(screen width()-100),(screen height()-170),6
sprite 7,(screen width()-100),(screen height()-170),7

  show sprite 1
  show sprite 2
  Circle PRX#,PRZ#,1

 if camera position y(0) >0 and camera position y(0) < 715.8
  show sprite 3
  hide sprite 4
  hide sprite 5
  hide sprite 6
  hide sprite 7
endif
if camera position y(0) >715.9 and camera position y(0) < 1291.8
  hide sprite 3
  show sprite 4
  hide sprite 5
  hide sprite 6
```

```
  hide sprite 7
endif
if camera position y(0) >1291.9 and camera position y(0) < 1867.8
  hide sprite 3
  hide sprite 4
  show sprite 5
  hide sprite 6
  hide sprite 7
endif
if camera position y(0) >1867.9 and camera position y(0) < 2443.8
  hide sprite 3
  hide sprite 4
  hide sprite 5
  show sprite 6
  hide sprite 7
endif
if camera position y(0) >2443.9
  hide sprite 3
  hide sprite 4
  hide sprite 5
  hide sprite 6
  show sprite 7

endif
set cursor 0,10
  print "CAMERA DATA"
  print " x:";camera position x(0)
  print " y:";camera position y(0)
  print " z:";camera position z(0)
  print "CAMERA ROTATE DATA"
  print " x:";camera angle x(0)
```

```
  print " y:";camera angle y(0)
  print " z:";camera angle z(0)
  print "RadiusX:";PRX#
  print "RadiusZ:";PRZ#
  print "MouseX:";Mousex()
  fastsync
if escapekey()=1
hide sprite 1
  hide sprite 2
  hide sprite 3
  hide sprite 4
  hide sprite 5
  hide sprite 6
  hide sprite 7
      goto main
  endif
loop
```

10. สร้าง About past

```
about:
cls
sync
  load image "search4\about1.bmp",1002
  do
    cls
    paste image 1002,0,0
    sync
    if escapekey()=1
      goto main
    endif
  loop
```

```
  return
```

11. สร้าง functionต่าง ๆ

```
function GoCamX(PosX#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  X#=(PosX#-CamX#)/10
  for n=1 to (X#)
    CamX#=CamX#+10
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction

function GoCamY(PosY#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  PosZ#=PosZ#/10
  for n=1 to (PosY#)
    CamY#=CamY#+2
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
```

```
    sync
  next n
endfunction

function GoCamZ(PosZ#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  Z#=(PosZ#-CamZ#)/10
  for n=1 to (Z#)
    CamZ#=CamZ#+10
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction

function BackCamX(PosX#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  X#=(PosX#-abs(CamX#))/10
  for n=1 to (X#)
    CamX#=CamX#-10
    position camera CamX#,CamY#,CamZ#
```

```
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction

function NBackCamX(PosX#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  X#=(CamX#-PosX#)/10
  for n=1 to (X#)
    CamX#=CamX#-10
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction
function BackCamY(PosY#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  PosY#=PosY#/10
  for n=1 to (PosY#)
    CamY#=CamY#-10
    position camera CamX#,CamY#,CamZ#
```

```
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction
function BackCamZ(PosZ#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()

  _Z#=(CamZ#-PosZ#)/10
  for n=1 to (_Z#)
    CamZ#=CamZ#-10
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction
function LotR(LotY#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  for n=1 to (LotY#)
    PointY#=PointY#+1
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
```

```
    sync
  next n
endfunction

function LotL(LotY#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  for n=1 to (LotY#)
    PointY#=PointY#-1
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction

function GoUp1(Up#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  Up#=(Up#-CamZ#)/10
  for n=1 to (Up#)
    CamZ#=CamZ#+10
    CamY#=CamY#+4
    position camera CamX#,CamY#,CamZ#
```

```
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction

function GoUp2(Up2#)
  CamX#=camera position X()
  CamY#=camera position Y()
  CamZ#=camera position Z()
  PointX#=camera angle x()
  PointY#=camera angle y()
  PointZ#=camera angle z()
  Up2#=(CamZ#-Up2#)/10
  for n=1 to (Up2#)
    CamZ#=CamZ#-10
    CamY#=CamY#+4
    position camera CamX#,CamY#,CamZ#
    Rotate camera PointX#,PointY#,PointZ#
    sync
  next n
endfunction
```