# บทที่ 5
# หลักการออกแบบและการสร้างเกมเอนจินประเภท GUI

### 5.1 นิยามของเกมเอนจินประเภท GUI

เกมเอนจิน ประเภท GUI (Graphic User Interface) คือ เครื่องมือที่ช่วยในการสร้างเกมที่มีส่วนติดต่อกับผู้ใช้ลักษณะเป็นกราฟฟิก ทำให้ผู้ใช้สามารถใช้งานได้ง่ายและไม่จำเป็นต้องมีความรู้ความชำนาญด้านการเขียนโปรแกรม ผู้ใช้จะสามารถทำการกำหนดประเภทของเกมที่จะสร้าง, เนื้อเรื่อง แล้วจึงเลือกเกมเอนจินประเภท GUI ที่สามารถสร้างเกมประเภทเดียวกับที่ผู้ใช้ต้องการจะสร้างเท่านั้น

### 5.2 นิยามของเกมประเภท RPG

เกมประเภท RPG (Role Playing Game) เป็นเกมประเภทที่ผู้เล่นต้องสวมบทบาทเป็นตัวละครภายในเกมและดำเนินตามเนื้อเรื่องของเกมที่มีการออกแบบไว้ ซึ่งภายในเกมจะมีเหตุการณ์ต่างๆ เช่น มีการพูดคุยกับ NPC (Non Player Control) ซึ่งเป็นตัวละครอื่นๆเพื่อหาข่าวสารที่เป็นประโยชน์ในการดำเนินตามเนื้อเรื่อง มีการต่อสู้ กับ NPC ประเภทศัตรู เช่น สัตว์ประหลาด (Monster) ตัวละครที่ผู้เล่นสวมบทบาทอาจมีการพัฒนาตัวเองขึ้นได้ ซึ่งอาจมาจากการที่ได้สู้กับศัตรู หรือเหตุการณ์อื่นๆตามแต่เกมนั้นจะกำหนด มีการแก้ปริศนาต่างๆจากคำบอกเล่าของ NPC ในเกม ซึ่งจะทำให้ผู้เล่นรู้สึกสนุกสนาน ไปกับการดำเนินเนื้อเรื่องและความสมจริงของเกมแนวนี้

### 5.3 แนวคิดของการออกแบบเกมเอนจินประเภท GUI

เกมเอนจินประเภท GUI นี้มีขอบเขตในการสร้างจำเพาะแนวเกมเดียว เช่นหากต้องการสร้างเกมเอนจินประเภท GUI ที่ใช้สร้างเกมแนวไล่ยิงกันแบบมุมมองบุคคลที่หนึ่ง เกมเอนจินที่สร้างออกมาจะไม่สามารถนำไปใช้สร้างเป็นเกมแนวอื่นๆได้ ในการออกแบบเกมเอนจินจึงควรคิดหาแนวเกมที่เหมาะสมในการที่จะนำมาทำเกมเอนจินประเภท GUI เป็นอันดับแรก

เกมเอนจินประเภท GUI ที่ถูกสร้างขึ้นในปริญญานิพนธ์เล่มนี้ เป็นเกมเอนจินประเภท GUI เพื่อใช้ในการสร้างเกมประเภทเกม RPG เนื่องจากเกมแนวนี้กำลังเป็นที่นิยมในปัจจุบันไม่ว่าในประเทศไทยหรือในต่างประเทศ และความสนุกของเกมประเภทนี้ คือการที่ผู้เล่นได้สวมบทบาทเข้าไปเป็น ตัวละครภายในเกมเพื่อดำเนินเนื้อเรื่อง ดังนั้นหากเกมมีเนื้อเรื่องที่ดีก็จะทำให้มีความสนุกมากยิ่งขึ้น แต่ผู้ที่ใฝ่ฝันจะสร้างเกมและมีความสามารถในการแต่งเนื้อเรื่องที่ดีก็อาจจะไม่มีความรู้ในการเขียนโปรแกรม ทำให้ไม่สามารถสร้างสรรค์เกมที่ดีออกมา ดังนั้นเกมเอนจินประเภท GUI ที่ใช้พัฒนาเกมแนวนี้น่าจะสามารถทำให้ความฝันของคนกลุ่มนี้เป็นจริงขึ้นมาได้

เกมเอนจินประเภท GUI ที่ได้ออกแบบ จะแบ่งออกได้เป็นสองส่วนใหญ่ๆ คือ

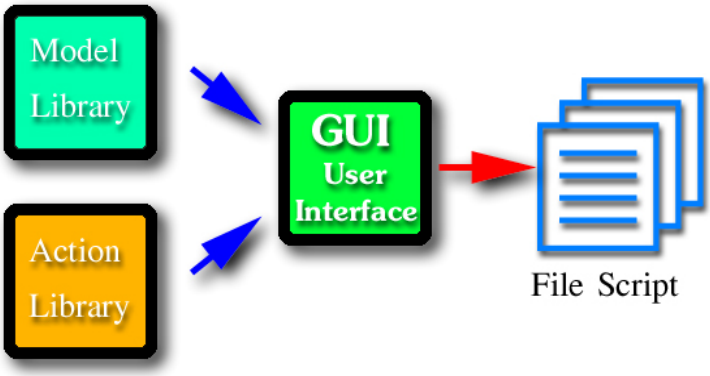
- ส่วนที่เป็นหน้าต่างติดต่อกับผู้ใช้ในการสร้างเกม (GUI User Interface) ส่วนนี้จะเป็นส่วนที่ผู้ใช้ตัว GUI ทำการสร้างฉากของเกม ตามโครงเรื่องที่ได้วางเอาไว้และจะสามารถแก้ไขค่าต่างๆ ภายในเกมได้ที่เกมเอนจินส่วนนี้เท่านั้น
- ส่วนที่ใช้เล่นเกม (GUI Game Player) ส่วนนี้จะเป็นส่วนที่ผู้ใช้นำเกมที่ได้สร้างขึ้นมาแล้วมาเล่นได้ และจะสามารถเล่นได้โดยผ่านเกมเอนจินส่วนนี้เท่านั้น

จะเห็นได้ว่า ทั้งสองส่วนแยกออกจากกัน แต่จะมีตัวกลางที่ทำให้เกมเอนจินทั้งสองส่วนสามารถติดต่อกันได้ ส่วนนี้คือ File Script โดยในส่วนของ File Script นี้จะมีหน้าที่เก็บข้อมูลเกมตามที่ผู้ใช้ได้สร้างขึ้นมา โดยเมื่อผู้ใช้ทำการสร้างเกมผ่าน GUI User Interface แล้วจะได้ File Script นี้มา และเมื่อผู้ใช้ทำการเล่นเกมที่ได้สร้างขึ้นมาผ่าน GUI Game Player ตัว GUI Game Player จะทำการเรียกข้อมูลเกม จาก File Script เพื่อประมวลแสดงผลออกมาในรูปของเกมตามที่สร้างไว้ ลักษณะการทำงานของเกมเอนจินโดยรวมดังที่ได้กล่าวมาแล้วข้างต้นจะสามารถแสดงได้ดังรูป 5-1

***ภาพที่ 5-1 แสดงการทำงานโดยรวมของเกมเอนจินประเภท GUI***

### 5.3.1 เกมเอนจินประเภท GUI ในส่วนที่เป็นหน้าต่างติดต่อผู้เล่น

ในส่วนนี้จะมีหน้าที่ รับ Input จาก Library ที่ได้กำหนดไว้ในตัวของเกมเอนจิน เพื่อมาสร้างให้เป็นเกมประเภท RPG ตามเนื้อเรื่องที่ได้กำหนดไว้ โดยจะสามารถแสดงหลักการทำงานได้ดัง รูปที่ 5-2




*รูปที่ 5-2 แสดงการทำงานของเกมเอนจินประเภท GUI ส่วนที่เป็นหน้าต่างติดต่อกับผู้เล่น*

การทำงานคือ โปรแกรมจะรับ Input มาจาก Library สองส่วน คือ ส่วน Model Library และ Action Library ผ่านตัว GUI User Interface แล้วจึงสร้างออกมาเป็น File Script ซึ่ง Library ทั้ง 2 ส่วนมีรายละเอียดดังนี้

#### 5.3.1.1 Model Library

เป็น Library ที่เก็บโมเดลต่างๆที่มีอยู่ในเกม โดยจะแบ่งออกเป็น 2 ประเภท คือ

- Library ที่รวบรวมแผนที่ภายในเกม (Map Library) จะเก็บโมเดลในส่วนที่เป็นแผนที่ ซึ่งจะสามารถนำไปใช้งานในเกมได้ทั้งหมด
- Library ที่รวมรวมตัวละครต่างๆภายในเกม (Character Library) ในส่วนนี้จะเก็บโมเดลในส่วนที่เป็นตัวละครต่างๆ รวมไปถึงค่าพลังของตัวละคร ซึ่งจะสามารถนำไปใช้งานในเกมได้ทั้งหมด
- Library ที่รวบรวมสิ่งก่อสร้างต่างๆ (Environment Library) ในส่วนนี้จะเก็บโมเดลในส่วนที่เป็นสิ่งก่อสร้าง เช่น บ้าน ต้นไม้ เก้าอี้ เตียง เป็นต้น ซึ่งจะสามารถนำไปใช้งานประกอบฉากตกแต่งภายในเกมได้ทั้งหมด

#### 5.3.1.2 Action Library

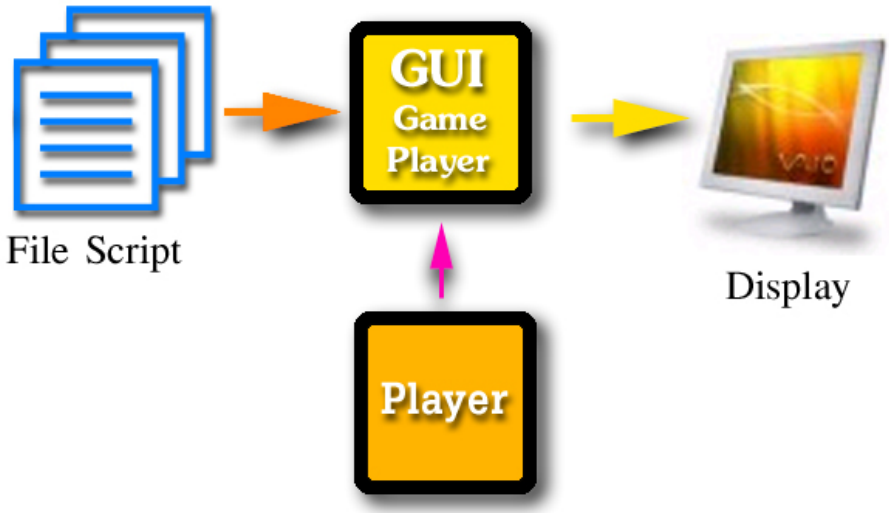
เป็น Library ที่รวบรวมคำสั่งที่เป็น Action ที่จะต้องใช้ภายในเกมส์เอาไว้เพื่อให้ผู้ใช้สามารถเลือกมาใช้กับเหตุการณ์ภายในเกมได้ เช่น Action ที่ใช้ในการพูด, Action ที่ใช้ในการฟื้นพลัง เป็นต้น

**5.3.1.3 GUI User Interface**

ส่วนนี้จะเป็นส่วนที่ประมวลผลจาก Input ที่ผู้ใช้ได้ทำการเลือกมาจาก Library มาจัดเก็บเอาไว้ตามโครงสร้างของแต่ละส่วนซึ่งจะแสดงที่ Class Diagram และเมื่อผู้ใช้ได้ทำการจัดเก็บ (Save) ส่วนนี้จะทำการเขียนข้อมูลลงใน File Script

**5.3.2 เกมเอนจินประเภท GUI ในส่วนที่ใช้เล่นเกม**

ในส่วนนี้จะทำหน้าที่ติดต่อกับ File Script ที่ได้มาจาก GUI User Interface เพื่อนำมาประมวลผลและจัดวางสิ่งต่างๆภายในเกม ที่ได้ออกแบบไว้ในเกมเอนจินในส่วนแรกแล้ว และเป็นส่วนที่ตอบสนอง Input จากผู้เล่นเกม ซึ่งได้ออกแบบให้ใช้ Mouse เป็นตัวควบคุมภายในเกมเสียเป็นส่วนใหญ่ แล้วจึงนำมาแสดงผล โดยการทำงานดังที่กล่าวมานี้จะแสดงได้ดังรูปที่ 5-3



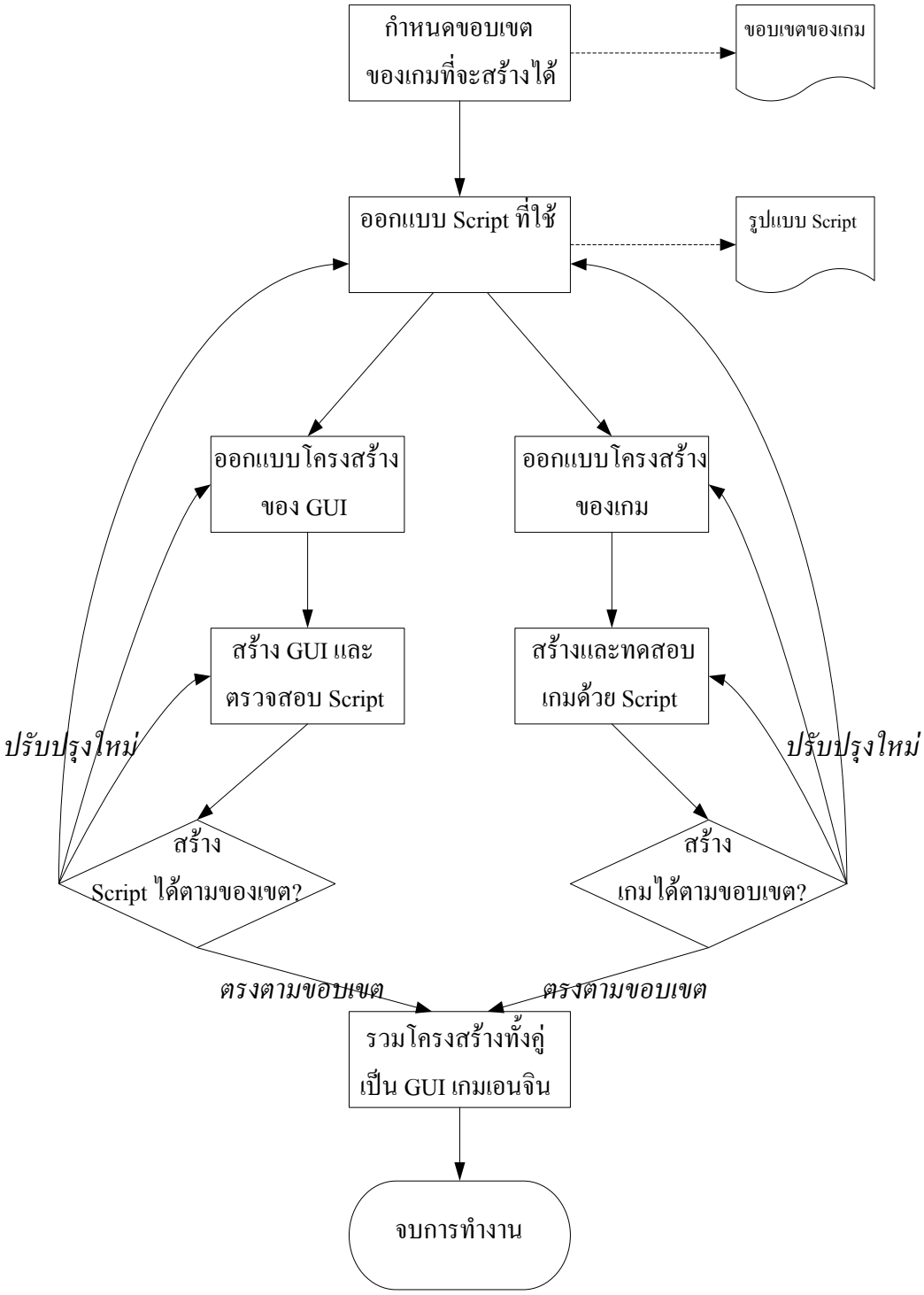

***รูปที่ 5-3 แสดงการทำงานของเกมเอนจินประเภท GUI ในส่วนของ GUI Game* Player**

**5.3.3 File Script**

ส่วนนี้เป็นส่วนที่ใช้เก็บข้อมูลที่ใช้ในการควบคุมสิ่งต่างๆภายในเกมส์ในส่วนที่ผู้ใช้ได้ทำการกำหนดไว้โดย ซึ่งในรายละเอียดของ File Script จะกล่าวถึงในหัวข้อต่อๆไป

## 5.4 โครงสร้างและการออกแบบ GUI 3D RPG Game Engine

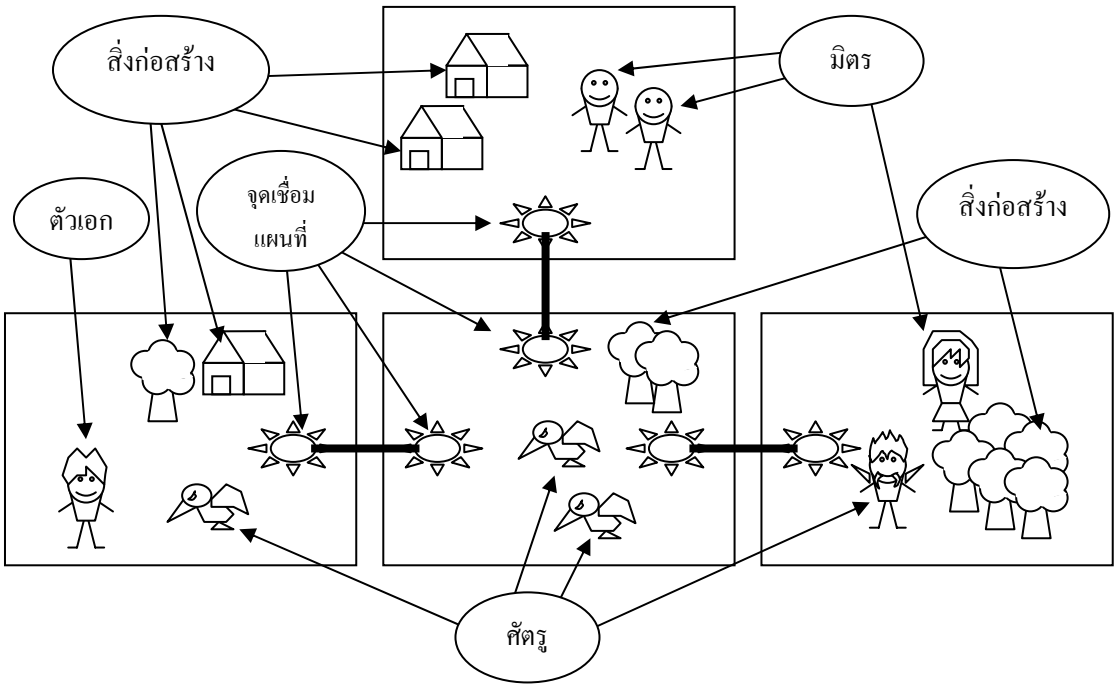
จากแนวคิดของการออกแบบที่กล่าวมาได้แบ่งโครงสร้างของเกมเอนจินประเภท GUI ออกเป็น 3 ส่วน ทำให้สามารถกำหนดกระบวนการออกแบบได้ดังรูปที่ 5 -4




***รูปที่ 5-4 แสดงกระบวนการออกแบบและสร้าง GUI 3D RPG Game Engine***

### 5.4.1 ขั้นตอนการกำหนดขอบเขตของเกมที่ GUI Game Engine สามารถสร้างได้

ขั้นตอนนี้จะทำการค้นคว้ารูปแบบของเกมที่เหมาะสม และสรุปออกมาเป็นผลลัพธ์ คือ ขอบเขตของเกมที่ GUI สามารถสร้างได้ ซึ่งข้อสรุปของโครงงานนี้คือ เกมในแนว RPG ซึ่งผู้เล่นต้องสวมบทบาทเป็นตัวละครตัวใดตัวหนึ่งซึ่งเป็นตัวเอก แล้วควบคุมตัวเอกนั้นดำเนินไปตามเนื้อเรื่องจนกว่าจะจบเกม โดยรูปแบบของเนื้อเรื่อง ก็มีได้หลายรูปแบบแล้วแต่ผู้สร้างเกมจะออกแบบมา แต่ลักษณะโดยพื้นฐาน ที่มีเช่น มีการพบศัตรูแล้วกำจัดเพื่อแก้ไขปริศนา หรือเพิ่มความสามารถของตัวเอก หรือมีการไปคุยกับผู้คนให้แก้ไขปริศนาหรือบอกปริศนา หรือแก้ไขปริศนาได้หมด แล้วจบเกม โดยปริศนาในแต่ละขั้นนั้น อาจจะอยู่ในฉากหรือแผนที่ซึ่งแตกต่างกันไป จะเห็นได้ว่า การออกแบบเกมแนว RPG นี้สามารถกำหนดเนื้อเรื่องได้หลากหลาย จึงต้องกำหนดขอบเขตต่างๆ แสดงให้เห็นจากภาพรวมของเกม ดังนี้




***รูปที่ 5-4 แสดงความสัมพันธ์ของแผนที่และสิ่งต่างๆในเกม***

ความสัมพันธ์ของสิ่งต่างๆภายในเกมจะถูกกำหนดดังนี้

- ลักษณะแผนที่ในเกมจะเป็นแผนที่ย่อยต่อๆกัน
- จุดเชื่อมต่อระหว่างแผนที่จะมีหรือไม่ก็ได้และมีเฉพาะแผนที่ที่อยู่ติดกัน

- **ตัวละคร** ในเกมที่สามารถเคลื่อนไหวและโต้ตอบกับผู้เล่นได้ สามารถนำมาประกอบใน โมเดลแผนที่เพื่อให้ฉากแต่ละฉาก มีตัวละครแตกต่างกันไป โดยแบ่งเป็นประเภทย่อยๆตามลักษณะเกมแนว RPG ดังนี้

**ตัวเอก** ตัวละครประเภทนี้เป็นตัวเอกในการเล่นเกม คือเป็นตัวละครที่มีได้ตัวเดียว และ เป็นตัวละครที่ผู้เล่นเกมจะใช้ บังคับเพื่อดำเนินเกม โดยตัวเอกนั้นจะมีความสามารถต่างๆติดตัว และมีค่าพลังชีวิต

ตามแต่ผู้สร้างเกมกำหนดว่าให้มีเท่าใด หรือกำหนดว่า มีอาวุธอะไรติดตัวอยู่บ้าง มีพลังในการโจมตีศัตรู ความว่องไวในการเคลื่อน ความฉลาดในการหลบหนี พลังชีวิตสามารถเพิ่มได้หรือไม่ โดยตัวเอกนี้จะไม่สร้างไว้ในเกมไม่ได้ เพราะเกมจะไม่สามารถดำเนินได้ และตัวเอกนี้หากเล่นแล้วพลังชีวิตหมด หรือโดนฆ่าตายก็คือ เกมจบ โดยหากต้องการสร้างเกมให้เนื้อเรื่องหลากหลาย ต้องกำหนดให้เลือกตัวพระเอกได้หลากหลายด้วย

**ศัตรู** ตัวละครประเภทนี้เป็นศัตรูกับตัวเอก คือเมื่อมีการเจอกับตัวเอกต้องมีการต่อสู้กัน โดยตัวศัตรูนั้นมีความสามารถแตกต่างกันไป แล้วแต่ผู้สร้างเกมจะกำหนดว่ามีความสามารถอย่างไร และมีการผูกปริศนาไว้กับตัวละครตัวนี้หรือไม่ เพื่อให้เนื้อเรื่องเปลี่ยนแปลงไปตามต้องการ เช่นเมื่อมีการสร้างตัวศัตรูแล้วกำหนดให้ปริศนาเกมสิ้นสุดเมื่อฆ่ามันตาย โดยตัวศัตรูนั้นก็มีพลังชีวิตเช่นเดียวกับตัวเอก และสามารถกำหนดได้ว่าเป็นเท่าใด เพื่อให้เกิดความหลากหลายในเกม เช่น เมื่อตัวพระเอกมีพลังชีวิตน้อยๆ ก็ต้องเลือกต่อสู้กับตัวศัตรูที่มีพลังชีวิตใกล้เคียงกัน โดยตัวศัตรูนั้นก็สามารถกำหนดความสามารถแต่ละอย่างได้ต่างกัน เช่น ความว่องไวในการเคลื่อนไหว ค่าพลังชีวิต เป็นต้น

**มิตร** ตัวละครประเภทนี้เป็นตัวละครทั่วๆไป ที่ไม่ทำร้ายตัวเอก โดยลักษณะของมิตร คือช่วยตัวเอกในการแก้ไขปริศนา เพิ่มความสามารถของตัวเอกเพื่อประโยชน์ในการแก้ไขปริศนาในขั้นต่อไป ผู้สร้างเกมสามารถนำ GUI RPG ไปกำหนดเนื้อเรื่องจาก ความหลากหลายของ ตัวละครนี้ได้

- ตัวละครอื่นๆที่ไม่ใช่ตัวเอก ซึ่งผู้เล่นไม่สามารถควบคุมได้เรียกอีกอย่างว่า NPC (None Player Control)
- ตัวละครพวก NPC สามารถเดินไปมาในฉากเองได้
- ตัวละครจะมีการระยะเวลาคอยการโจมตี ขึ้นอยู่กับค่าความว่องไวของตัวละครแต่ละตัวเมื่อโจมตีไปครั้งหนึ่งหากมีความว่องไวสูงกว่าก็จะสามารถโจมตีครั้งต่อไปได้เร็วกว่า
- ตัวเอกจะต่อสู้กับศัตรูได้และเกมจะจบถ้า ค่าพลังชีวิตเหลือ 0
- ค่าพลังของตัวละครแต่ละตัวสามารถปรับแต่งได้ สามารถกำหนดตำแหน่งและการกระทำได้

- **สิ่งแวดล้อม** ที่นำมาประกอบใน โมเดลแผนที่ เพื่อให้มีภูมิประเทศแตกต่างกัน แม้ว่าจะมีแผนที่คล้ายๆกัน เช่น โมเดลแผนที่ ที่เป็นทะเล อาจจะมีความแตกต่างของ โมเดลสิ่งแวดล้อมเช่น ทะเลแห่งแรกมีแต่โขดหิน ส่วนทะเลอีกแห่งอาจประกอบด้วยไม้ เป็นป่าโกงกาง เป็นต้น โดย แบ่งเป็น 3 ประเภท ดังนี้

**สิ่งก่อสร้าง** คือสิ่งก่อสร้างโดยทั่วไป ที่มีความใหญ่โต เมื่อนำมาประกอบกับ โมเดลแผนที่ แล้วทำให้ฉากในแต่ละฉากเปลี่ยนแปลงไปมาก เช่น โมเดลแผนที่ ถนน นั้นอาจจะมี ถนนสายหนึ่งประกอบด้วย ตึกสูงๆหลายๆตึกรวมกันเป็น ถนนเศรษฐกิจ ส่วนถนนอีกสายหนึ่งอาจจะประกอบด้วยบ้านคนกลายเป็นถนนในหมู่บ้าน เป็นต้น จากความหลากหลายของสิ่งก่อนสร้าง นี่เอง ทำให้เกม RPG นั้นมีภูมิประเทศที่แตกต่างกันโดยสิ้นเชิง

**ต้นไม้** คือต้นไม้ เป็นส่วนประกอบฉากหนึ่งๆ ที่สร้างมาเพื่อให้ฉากนั้นๆ สวยงามขึ้นและเปลี่ยนภูมิประเทศ เป็นแนวป่าไม้ แทนเมือง ในแบบแรก เป็นต้น ทำให้ฉากแต่ละฉากในเกม ดูหลากหลาย ผู้เล่น

สามารถกำหนดเนื้อเรื่องเป็นอีกประเภทได้ เช่นจากเมืองที่มีตึกมีตัวศัตรูโจร เป็น ฉากป่าไม้ที่มีตัวศัตรูเป็นสัตว์ เป็นต้น

**สิ่งแวดล้อมทั่วไป** คือสิ่งก่อสร้างเล็กๆ หรือ สิ่งของจุกจิกโดยทั่วไป เช่นโต๊ะ เก้าอี้ ก้อนหิน

- **เหตุการณ์** จะเป็นสิ่งที่กำหนดเนื้อเรื่องของเกม แบ่งออกเป็น 2 ประเภท ดังนี้

**เหตุการณ์ของแผนที่** เป็น Script ที่กำหนดเหตุการณ์ให้กับฉากโดยตรง เช่น เมื่อเดินเข้าไปในฉากแต่ละฉากแล้วกำหนดให้เสียงดนตรี เป็นอย่างไร หากมีเสียงจังหวะระทึกแสดงว่าเป็นฉากสำคัญที่มีการต่อสู้กับตัวศัตรูตัวสำคัญ เช่นหัวหน้าโจร หากเป็นเสียงดนตรี หรือเสียงคนมากมาย ก็แสดงว่าเป็นฉากที่มีพ่อค้าแม่ค้า หรือมีพลเมืองมากๆ หรือ หากเป็นเสียงดนตรีธรรมดา ก็อาจจะเป็นการดำเนินเรื่องทั่วไป เป็นต้น หรือกำหนดให้ บางฉากเข้าไปแล้วมีหมอกปกคลุมจนหาปริศนาลำบาก หรือมีหมอกสีแดงเพื่ออำพรางศัตรูตัวสำคัญในฉากการต่อสู้ หรือกำหนดให้ จุดเปลี่ยนฉากไปยังฉากอื่นนั้นทำงานหรือไม่ โดยอาจจะมีเงื่อนไขในการทำงานตัวอย่าง เช่น ฉากชายทะเลจะกำหนดให้จุดเปลี่ยนฉาก ทำงานคือเปลี่ยนฉากไปยัง ฉากบนเรือได้เมื่อตัวพระเอก นั้นมีเรือ หรือไปซื้อเรือมาได้จึงเปลี่ยนฉากได้ เป็นต้น หรือกำหนดว่าจะเข้าไปในฉากบ้านได้เมื่อตัวพระเอกมีสิ่งของคือกุญแจ เป็นต้น จะเห็นได้ว่า ส่วนของ เหตุการณ์แผนที่นี้สามารถนำไปสร้างเป็นปริศนาเพื่อให้เกมเนื้อเรื่องตรงตามต้องการและให้เกมจบได้ตามต้องการ เช่นให้เกมจบเมื่อตัวพระเอกหากุญแจมาเพื่อเปลี่ยนฉากไปยังฉาก ห้องลับเพื่อช่วยตัวพลเมืองได้ เป็นต้น

**เหตุการณ์ของตัวละคร** เป็นการกำหนดกำหนดเหตุการณ์ให้กับตัวละครโดยตรง ว่าเหตุการณ์ในแต่ละอย่างที่เกิดขึ้นมานั้น เกิดขึ้นเพราะอะไร และทำงานอย่างไร ตัวละครที่ได้รับเหตุการณ์ จะเป็นอย่างไร ตัวละครที่สร้างเหตุการณ์สร้างอย่างไร เหตุการณ์ที่เปลี่ยนแปลงความสามารถของตัวละคร เหตุการณ์ที่เกิดขึ้นเพื่อแก้ไขปริศนาของเกม เช่น

เหตุการณ์การพูดคุย ของเกม ตัวอย่างคือตัวพระเอกไปพบกับตัวพลเมืองตำรวจเพื่อสอบถามว่าตัวศัตรูโจรอยู่ที่ใดบ้าง เพื่อไปกำจัด หรือไปพบกับตัวพลเมืองพ่อค้าเพื่อสอบถามว่าสิ่งของที่ต้องการหาเช่นรองเท้าวางอยู่ที่ใดบ้าง เป็นต้น

เหตุการณ์การเก็บสิ่งของ เช่น เมื่อตัวพระเอกพบกับสิ่งของคือยารักษา ก็จะเพิ่มพลังชีวิตให้กับตัวพระเอก เป็นต้น หรือหากตัวพระเอกเก็บสิ่งของคือเสื้อเกราะก็จะเพิ่มค่าในการป้องกันการโจมตีเอาไว้

เหตุการณ์การให้สิ่งของ เช่น เมื่อตัวพระเอกมีสิ่งของคือแผนที่ แล้วตัวพระเอกนำไปให้กับตัวพลเมืองคือตำรวจ แล้วตัวตำรวจก็จะแก้ไขปริศนาให้อีกขั้น เป็นต้น

เหตุการณ์การเปลี่ยนตำแหน่งตัวละคร เช่น เมื่อตัวพระเอกได้พบกับตัวพลเมืองคนนำทาง คนนำทางก็จะพาตัวพระเอกไปยังฉาก หรือไปยังตำแหน่งที่ กำหนดเงื่อนไขเอาไว้ เป็นต้น

**5.4.2 ขั้นตอนการออกแบบ Script**

ดังที่ได้กล่าวมาแล้วว่าเกมเอนจินประเภท GUI ที่ได้ออกแบบมาจะมีส่วนที่ทำหน้าที่เป็นเสมือนตัวกลางเชื่อมการทำงานระหว่าง ส่วนที่เป็นหน้าต่างติดต่อกับผู้ใช้ และ ส่วนที่ใช้เล่นเกม โดยจะเป็นส่วนที่กำหนดสิ่งต่างๆภายในเกม ซึ่ง File script ที่ออกแบบมาใช้ภายในเกมเอนจิน มีทั้งหมด 9 ประเภท คือ

1. MFS (Map File Script) ทำหน้าที่เก็บข้อมูลเกี่ยวกับฉากที่ผู้ใช้ได้ทำการเลือกออกมาใช้ เช่น ได้เลือกใช้ฉากใด ใช้ Model ชื่ออะไร ใช้ Texture ชื่ออะไร เป็นต้น และเป็น File Script แรกที่จะถูกอ่าน ในเกมเอนจินส่วนที่ใช้เล่นเกม

2. Cdef (Character Definition) ทำหน้าที่เก็บข้อมูลค่าพลังต่างๆของตัวละครทั้งหมดที่ผู้ใช้เลือกมาใช้ภายในเกมที่สร้างขึ้น เช่น ค่าพลังชีวิต ค่าพลังโจมตี ของตัวละคร

3. Ch (Character) ทำหน้าที่เก็บข้อมูลในส่วนที่เกี่ยวกับตัวละครภายในแต่ละฉาก เช่น รหัสตัวละครทั้งหมดในฉาก ตำแหน่งของตัวละครภายในฉาก และตัวละครจะอ้างอิงค่าพลัง File Script ประเภท Cdef และ อ้างอิงถึง Model ที่ใช้ จาก File Script ประเภท Cmod

4. Cmod (Character Model) ทำหน้าที่เก็บข้อมูลในส่วนของ Model ของตัวละครต่างๆที่ใช้ภายในฉากต่างๆของเกม เช่น ชื่อ File Model ชื่อ File Texture

5. Env (Environment) ทำหน้าที่เก็บข้อมูลของสิ่งแวดล้อมต่างๆ ภายในฉาก และจะอ้างอิงถึง File Script ประเภท Emod ซึ่งเก็บข้อมูล Model ที่สิ่งแวดล้อมในฉากนั้นๆต้องใช้

6. Emod (Environment Model) ทำหน้าที่เก็บข้อมูลในส่วนของ Model ของสิ่งแวดล้อมต่างๆที่ใช้ภายในฉากต่างๆของเกม เช่น ชื่อ File Model ชื่อ File Texture

7. Trp (Transport) เก็บข้อมูลของจุดที่ใช้บอกตำแหน่งที่เชื่อมกันระหว่างฉากแต่ละฉาก เพื่อใช้ในการกำหนดตำแหน่งเมื่อตัวละครมีการเดินทางข้ามฉากภายในเกม

8. Scp (Script) จะเป็น File ที่ใช้เก็บคำสั่งที่ใช้ควบคุมเหตุการณ์ต่างๆภายในเกมโดยเหตุการณ์ภายในเกมจะแบ่งส่วนออกเป็น 2 ส่วน คือ

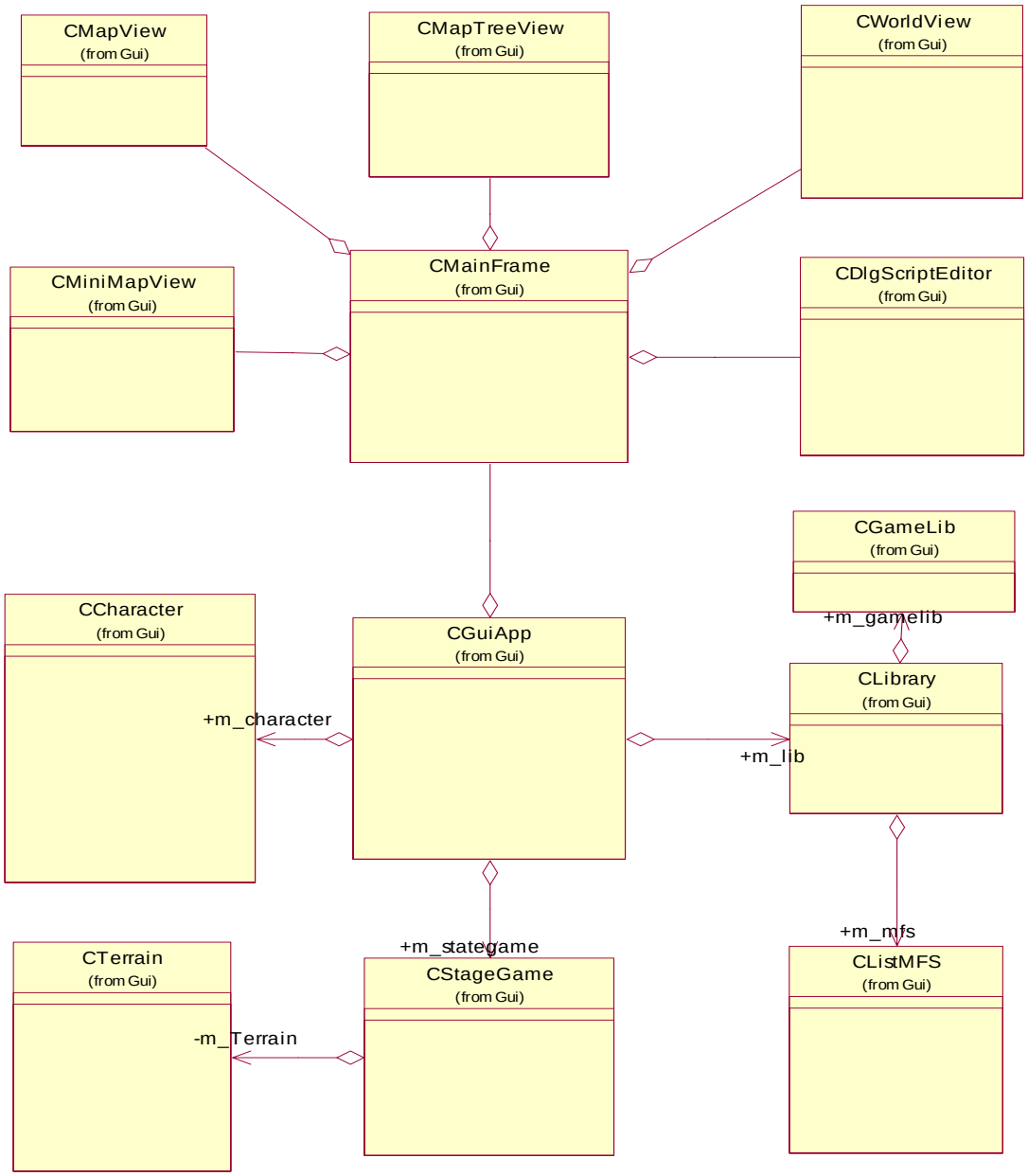
- เหตุการณ์ที่เกิดขึ้นกับแผนที่ (Map Script) จะใช้ควบคุมเหตุการณ์ที่เกิดขึ้นเมื่อเข้ามาในแผนที่นี้ เช่นเมื่อมาที่แผนที่นี้แล้ว เปลี่ยนเพลง มีหมอก เป็นต้น
- เหตุการณ์ที่เกิดขึ้นกับตัวละคร (Character Script) จะใช้ควบคุมเหตุการณ์ที่เกิดขึ้นจากวัตถุต่างๆเช่น เมื่อสัมผัสกับตัวละครในเกม เป็นต้น

9. Rt (route point) เป็นไฟล์ที่ใช้ควบคุมทิศทางการเดินของตัวละครที่ไม่ใช่ตัวเอก หรือ NPC โดยจะให้ผู้ใช้ได้กำหนดทิศทางและระยะทาง เพื่อให้ตัวละครเหล่านี้เคลื่อนที่ได้

### 5.4.3 ขั้นตอนการออกแบบโครงสร้างของ GUI และ เกม

ในขั้นตอนนี้จะสามารถแยกการทำงานออกเป็น 2 ส่วน คือ การออกแบบโครงสร้างของ GUI ที่ใช้ในการสร้าง File Script กับ การออกแบบโครงสร้าง ตัวเกม RPG 3 มิติ ที่สามารถอ่าน File Script นั้นและนำไปแสดงผลออกมาเป็นเกมได้

ในส่วนของการออกแบบโครงสร้าง GUI นั้นแบ่งการออกแบบหลักได้เป็น 2 ส่วนใหญ่ๆ คือ ส่วนที่เป็นการสร้างคลาสเพื่อการสร้างหน้าต่างติดต่อกับผู้ใช้งาน กับอีกส่วนคือ ส่วนในการทำงานด้านสามมิติ ดังรูปข้างล่าง




*รูปที่ 5-5 แสดงแผนภูมิคลาสของส่วน GUI*

**- ส่วนที่เป็นคลาสเพื่อการสร้างหน้าต่างติดต่อกับผู้ใช้งาน**

CGuiApp เป็นคลาสหลักที่สืบทอดมาจากคลาส CWinApp ซึ่งทำหน้าที่หลักคือเรียกโปรแกรมขึ้นมาทำงานซึ่งภายในมีการสร้าง อินแสตนท์ ออบเจ็กต์ของ CMainFrame ไว้โดย CGuiApp มีหน้าที่ในการเริ่มต้นและปิดโปรแกรม โดยมีฟังก์ชัน InitInstance() ทำหน้าที่กำหนดรูปแบบโปรแกรมว่าเป็นโปรแกรมแบบชนิดใดโดยได้กำหนดให้เป็นโปรแกรมแบบเปิดได้หลายหน้าต่างพร้อมกัน และมีฟังก์ชันหลักคือ OnIdle() ซึ่งเป็นฟังก์ชันที่ถูกเรียกใช้งานเมื่อโปรแกรมไม่มีการรับอินพุต โดยทำหน้าที่ในกรณีที่โปรแกรมถูกเรียกใช้โหมดการแสดงผลภาพสามมิติด้วยการเรียกใช้คลาส CWorldView

CMainFrame เป็นคลาสที่สืบทอดมาจากคลาส CMDIFrameWnd ของ MFC ซึ่งถูกเรียกใช้งานหลังจากเปิดโปรแกรมทำหน้าที่เป็นตัวสร้างกรอบหน้าต่างหลักว่าจะมี Frame ย่อยๆอะไรบ้าง ซึ่งประกอบด้วยคลาสย่อยๆ 5 คลาส คือ CDlgScriptEditor , CMapView , CWorldView , CMiniMapView , CMapTreeView ซึ่งทั้ง 5 คลาสนี้ เป็นคลาสในการสร้าง Frame ย่อยๆ ที่ทำหน้าที่ในการแสดงผลต่างกันไป และคลาส CMainFrame นี้ถูกสร้างอินแสตนท์ขึ้นมาหลังจากสร้างหน้าต่างหลักที่ทำหน้าที่ติดต่อกับผู้ใช้ดังนั้นหน้าที่หลักคือเลือกสร้าง โครงงานเกมใหม่ เปิดโครงงานเกมที่สร้างไว้แล้ว และ บันทึกข้อมูลโครงงานเกมที่กำลังจัดทำอยู่ โดยจะมีการสร้างเมนูในการทำงานหลายรูปแบบ และมี ไดอะล็อคติดต่อกับผู้ใช้เพื่อรายงานผลข้อมูลของ GUI RPG ทั้งหมดไม่ว่าจะเป็นไลบรารี ที่เก็บโมเดล หรือ เรียกหน้าต่างย่อยๆเพื่อการทำงานในแต่ละส่วน

CMapTreeView เป็นคลาสที่สืบทอดมาจาก CTreeView ของ MFC ทำหน้าที่แสดงรายละเอียดทั้งหมดของเกมที่ผู้ใช้ได้สร้างขึ้นมาในรูปแบบของ แผนภูมิต้นไม้ เพื่อให้ผู้สร้าง สามารถตรวจสอบรายละเอียดของเกมได้โดยสะดวก โดยคลาสจะเริ่มทำงานเมื่อมีการเปิดโครงงานเก่า ขึ้นมาโดยจะนำแผนที่ทั้งหมดในโครงงานนั้นมาแสดง และจะนำทั้ง ตัวละคร, สิ่งแวดล้อมและจุดเชื่อมแผนที่ทั้งหมดที่อยู่ในแต่ละแผนที่ออกมาแสดงในรูปแบบเป็นรากของแต่ละแผนที่นั้น

CMapView เป็นคลาสที่สืบทอดมาจาก CView ของ MFC ทำหน้าที่ในการเลือกแสดงผลภาพแผนที่ซึ่งกำลังถูกเลือกใช้งานอยู่ หรือเป็นแผนที่ ที่มีตัวละครที่กำลังเรียกใช้งานอยู่ ฟังก์ชันหลัก คือ OnDraw() ทำหน้าที่วาดรูปแผนที่บนวิวที่กำหนด โดยมี คลาส CDIBSectionLite m_bitmap ทำหน้าที่ในช่วยการแสดงผลรูปภาพแผนที่เพิ่มเข้ามาโดยสามารถอ่านไฟล์ในรูปแบบ Bitmap ที่มีขนาดแตกต่างกันได้

CWorldView เป็นคลาสที่สืบทอดมาจาก CScrollView ของ MFC ทำหน้าที่ 2 รูปแบบการทำงาน คือ การทำงานในการแสดงการเชื่อมต่อกันของแผนที่ทั้งหมด กับ แสดงผลแผนที่เดียวในรูปแบบ 3 มิติ ในส่วนรูปแบบการทำงานในการแสดงการเชื่อมต่อกันของแผนที่ทั้งหมดนั้น จะมีคลาส CDIBSectionLite *m_pBitmap[] ทำหน้าที่ในการแสดงผลรูปแผนที่ทั้งหมด ออกมาเป็นรูปแบบคล้ายแผนที่โลก และมีฟังก์ชัน MouseMapMove() ทำหน้าที่ในการเปลี่ยนตำแหน่งแผนที่ที่เลือกว่าต่อกันอย่างไรโดยมีการรับ Input จากเมาส์โดยฟังก์ชัน OnMouseMove() และฟังก์ชัน OnInitialUpdate() ทำหน้าที่ในการเริ่มต้นวาด

หน้าจอภาพ สำหรับส่วนรูปแบบการแสดงผลแผนที่เดียวในรูปแบบ 3 มิตินั้น จะการแสดงผลข้อมูลในฉากแต่ละฉาก และเพิ่มหรือเปลี่ยนแปลงตำแหน่ง ของ โมเดลวัตถุ ต่างๆในฉาก ดังนั้นต้องมีฟังก์ชันในการควบคุมการเคลื่อนไหวของกล้อง การแสดงผลการเคลื่อนไหวของโมเดล รวมทั้งแสดงคุณสมบัติต่างๆของโมเดลวัตถุ เช่น ตำแหน่งตัวละคร ค่าพลังชีวิตต่างๆ ค่าของสิ่งของแต่ละอย่าง เป็นต้น ฟังก์ชันที่จำเป็นต่อการแสดงผลภาพสามมิติ คือ ฟังก์ชัน Render() ซึ่งในฟังก์ชันนี้มีการเรียกใช้ส่วนที่เป็นกราฟฟิก เช่น CStageGame , CCharacter เป็นต้น ฟังก์ชันหลักที่ใช้ในการควบคุมกล้องให้เคลื่อนที่ตามเมาส์มี สี่ส่วนคือ MouseCameraMove() ใช้ในการเคลื่อนที่กล้องไปตามที่ที่ลากไป ฟังก์ชัน MouseCamerraZoom() ใช้ในการเลื่อนกล้องเข้าออกจากจุดโฟกัส ฟังก์ชัน MouseCameraYaw() เป็นฟังก์ชันที่ใช้ในการหมุนกล้องตามแนวนอน และฟังก์ชัน MouseCameraPtich() เป็นฟังก์ชันที่ใช้ในการหมุนกล้องตามแนวตั้ง ซึ่งทั้ง 3 ฟังก์ชันมีประโยชน์ในการกำหนดตำแหน่งที่จะวางวัตถุโมเดลลงไปบนแผนที่ เพราะหากแผนที่มีขนาดใหญ่ก็จำเป็นต้องเลื่อนกล้อง ไปตามจุดต่างๆตามต้องการ ฟังก์ชันถัดมาอีกสี่ฟังก์ชันเป็นฟังก์ชันที่ใช้ในการเพิ่มวัตถุโมเดลลงไปในแผนที่ รวมทั้งเปลี่ยนแปลงแก้ไข ตำแหน่งของวัตถุโมเดลต่างๆในแผนที่ตามต้องการ คือ ฟังก์ชัน MouseTransportMove() ทำหน้าที่ในการเลือกจุดเชื่อมแผนที่ใหม่มาวางบนแผนที่รวมทั้งเคลื่อนย้ายตำแหน่งจุดเชื่อมแผนที่ไปตามต้องการ ฟังก์ชัน MouseCharacterMove() ทำหน้าที่ในการเลือกตัวละครมาวางบนแผนที่ รวมทั้งเปลี่ยนแปลงตำแหน่งตัวละครในแผนที่ตามต้องการ ฟังก์ชัน MouseEnvironmentMove ทำหน้าที่ในการเลือกสิ่งแวดล้อมมาวางบนแผนที่ รวมทั้งแก้ไขเปลี่ยนแปลงตำแหน่งสิ่งแวดล้อมในแผนที่ ตามต้องการ

CMiniMapView เป็นคลาสที่สืบทอดมาจาก CFormView ของ MFC ทำหน้าที่ในการแสดงผลเมนูคำสั่งการจัดการ แผนที่ , ตัวละคร , สิ่งแวดล้อม , จุดเชื่อมแผนที่ที่กำลังเลือกทำงานอยู่ คลาสนี้จึงเป็นคลาสหลักในการสร้างภูมิประเทศของแผนที่ หรือ เพิ่มและแก้ไขตัวละครในแต่ละฉากได้ตามต้องการ โดยจะส่วนใหญ่จะปืนทำงานร่วมกับคลาส CWorldView ที่ใช้แสดงการเชื่อมต่อกันของแผนที่ทั้งหมด และ แสดงรายละเอียดแผนที่แบบ 3 มิติ ซึ่งจะทำให้การจัดการเปลี่ยนแปลงรายละเอียดต่างๆเป็นไปโดยสะดวก

จากคลาสที่อธิบายมาทั้งหมดตัว GUI RPG สามารถสร้างแผนที่ต่อเนื่องกันได้ สามารถวางตัวละครและสิ่งแวดล้อม ได้แล้วแต่ยังไม่สามารถกำหนดเหตุการณ์ ของตัวละคร ปริศนาเงื่อนไขของเกมได้ ดังนั้นจึงต้องมีส่วนของเหตุการณ์ที่เกิดกับวัตถุประเภทต่างๆ เช่น การพูดคุยกันระหว่างตัวละคร จากส่วนที่กล่าวมาทั้งการเปลี่ยนแปลงแก้ไขแผนที่ และการวางแผนที่แล้ว ต่อไปคือการกำหนด เงื่อนไข ปริศนา และเหตุการณ์ต่างๆที่จะเกิดกับตัวละคร นั่นคือการกำหนดเนื้อเรื่องให้กับตัวเกม ซึ่งได้ออกแบบคลาสหลักๆ ดังนี้

CDlgScriptEditor เป็นคลาสที่สืบทอดมาจาก CDialog ของ MFC ซึ่งมีหน้าที่ในการกำหนดเหตุการณ์ให้กับวัตถุโมเดลต่างๆเช่น ตัวละคร แผนที่ เป็นต้น เมื่อผ่านขั้นตอนสองขั้นแรกคือ สร้างแผนที่ทั้งหมดและแก้ไขแผนที่แต่ละแผนที่แล้ว ก็สามารถเลือกกำหนดเนื้อเรื่องและปริศนาให้กับเกม ได้ โดย

เรียกฟังก์ชันแรกในการทำงานคือ OnInitDialog() เพื่อสร้างไดอะล็อกขึ้นมา จากนั้นโปรแกรม จะเรียกฟังก์ชัน LoadAction() ไว้สำหรับเปิด ไฟล์ แอคชันซึ่งจะมีการเรียกแอคชันต้นแบบที่ได้สร้างไว้แล้วขึ้นมาทั้งหมด จากนั้นจะเรียกใช้ LoadScript() เป็นการโหลดแอคชันสคริปที่ผู้ใช้เคยสร้างไว้แล้ว จากนั้นก็เป็นขั้นตอนการสร้างเหตุการณ์ต่างๆตามต้องการ โดยเหตุการณ์ต่างๆมีได้หลายรูปแบบ และมีฟังก์ชันที่สนับสนุนการทำงานดังนี้

การเกิดขึ้นของเหตุการณ์ต่างๆไม่ว่าจะเป็นการพูดคุย การมีเสียงดนตรี นั้น เหตุการณ์จะแบ่งได้เป็น 2 ประเภทหลักๆด้วยกัน คือเหตุการณ์ที่เกิดกับแผนที่ เหตุการณ์ที่เกิดกับตัว คลาส CDlgScriptEditor จึงสร้างเหตุการณ์พื้นฐานขึ้นมาสนับสนุนการสร้างเหตุการณ์ดังนี้

ฟังก์ชัน OnPlayMusic() ใช้สำหรับสร้างสคริปสำหรับเล่นเสียงเพลงโดยการทำงานคือ เมื่อเลือกแอคชัน Play Music จะเกิดไดอะล็อกให้เลือก ไฟล์ Midi และ ให้ใส่ค่า Volumn ผลลัพธ์คือจะได้ สคริปที่อธิบายว่า “Play Music Midi (ชื่อไฟล์เพลง) Volumn( ความดัง )” เป็นต้น

ฟังก์ชัน OnSetflag() ใช้สำหรับกำหนดค่าความจริงให้กับ Flag ซึ่งเป็นตัวแปรที่ใช้อ้างถึงในบางเงื่อนไข ลักษณะของ Script คือ “Set flag (หมายเลข ) is ( จริงหรือเท็จ )”

ฟังก์ชัน OnIfflag() ไว้สำหรับกำหนดค่าว่าจริงหรือเท็จในเพื่อประโยชน์ในการสร้างเงื่อนไขของเกม โดยฟังก์ชันนี้มีค่าที่ต้องกำหนดสองค่าคือ หมายเลขแฟค และ ค่าจริงหรือเท็จ โดยลักษณะสคริปคือ “if flag( หมายเลข ) is ( จริงหรือเท็จ ) then” ฟังก์ชันนี้ต้องใช้คู่กับฟังก์ชัน OnSetflag() ในการกำหนดหมายเลขแฟคเริ่มต้นว่าให้จริงหรือเท็จ ตัวอย่างการใช้งานเช่น เมื่อเริ่มต้นเกมเรากำหนด ตัวโจร มีแฟคหมายเลข 1 เป็นจริงและกำหนดให้เมื่อตัวโจรตายแฟคหมายเลข 1 เป็นเท็จเป็นต้น

ฟังก์ชัน OnElse() เป็นการเขียนเงื่อนไขของสคริปในลักษณะ ที่ว่า “ถ้า .... แล้ว “ เป็นต้น ดังนั้นฟังก์ชันนี้จริงเป็นการกำหนดเงื่อนไขขั้นที่สอง หรือขั้นถัดไปดังนั้น ฟังก์ชันนี้ต้องใช้คู่กับฟังก์ชันที่กำหนดเงื่อนไขเริ่มต้น อย่าง OnIfflag() ลักษณะสคริปคือ “else”

ฟังก์ชัน OnEndif() เป็นการจบเงื่อนไขสคริปที่สร้างไว้ ในกรณีที่เงื่อนไขยังไม่สิ้นสุด โดยการทำงานคือต้องกำหนดค่าหลังจากวางเงื่อนไขไว้แล้ว เช่น “ ถ้า ... แล้ว ... จบเงื่อนไข” ลักษณะเงื่อนไขคือ “endif” เป็นต้น ฟังก์ชันนี้ต้องใช้คู่กับ OnIfflag() และ OnElse() เพื่อให้เงื่อนไขสมบูรณ์แบบ

ฟังก์ชัน OnEbtransport() เป็นการกำหนดว่า ให้จุดเปลี่ยนฉากทำงานหรือไม่ ลักษณะของสคริปคือ “enable transport ( หมายเลขจุดเปลี่ยนฉาก หรือ ชื่อจุดเปลี่ยนฉาก ) is ( ทำงาน หรือ ไม่ทำงาน ) ” เป็นต้น

ฟังก์ชัน OnVisiblechar() เป็นการกำหนดการมีอยู่ของตัวละคร ลักษณะสคริปคือ “visible character ( หมายเลขตัวละคร หรือ ชื่อตัวละคร ) is ( จริงหรือเท็จ )” จะเห็นได้ว่าฟังก์ชันนี้สามารถนำมาสร้างเงื่อนไข หรือปริศนา ต่อจากฟังก์ชัน OnIfflag() ให้ทำงานได้

ฟังก์ชัน OnSpeak() เป็นฟังก์ชันที่สร้างการสนทนาระหว่างตัวละครให้พูดกันตามที่กำหนด โดยฟังก์ชันนี้ต้องกำหนดค่าสองอย่าง คือตัวละครอะไรและคำพูดที่จะพูด “character ( หมายเลขตัวละคร หรือ ชื่อตัวละคร ) speak ( ประโยคที่จะพูด ) ” จะเห็นได้ว่าฟังก์ชันนี้สามารถนำไปสร้างปริศนา หรือบอกปริศนาให้ผู้เล่นได้รู้

ฟังก์ชัน OnEndscript() เป็นฟังก์ชันการสั่งให้จบการทำงานของสคริป ลักษณะคำสั่งคือ “end script”

ฟังก์ชัน OnInchp() เป็นฟังก์ชันที่ทำงานกับตัวเอกเป็นหลัก คือใช้เพิ่มค่าพลังชีวิตให้กับตัวเอก ลักษณะสคริปคือ “increase hp ( จำนวนที่เพิ่ม ) to hero”

ฟังก์ชัน OnIfchardie() เป็นฟังก์ชันที่ใช้กำหนดเงื่อนไขไว้กับตัวละครกำหนดว่า ถ้าตัวละครที่กำหนดตาย โดยฟังก์ชันนี้ต้องใช้ฟังก์ชันอย่าง OnMoveChar() หรือ OnInchp() ควบคู่เพื่อกำหนดเงื่อนไขต่อไป เช่นหากตัวศัตรูตายแล้ว ตัวพระเอกจะมีพลังเพิ่มขึ้นตามที่กำหนดไว้ เป็นต้น ลักษณะสคริปคือ คือ “if character ( ชื่อตัวละคร หรือหมายเลขตัวละคร ) on map ( ชื่อแผนที่หรือหมายเลขแผนที่ ) die”

ฟังก์ชัน OnShowfog() เป็นฟังก์ชันที่ใช้สร้างหมอกในฉากตามเงื่อนไขที่กำหนด ลักษณะสคริปคือ “fog enable ( ทำงานหรือไม่ทำงาน ) color red ( ค่าสีแดงของหมอก ) green ( ค่าสีเขียวของหมอก ) blue ( ค่าสีน้ำเงินของหมอก ) ” เป็นต้น ตัวอย่างการใช้งานเช่น เมื่อตัวพระเอกกำจัดตัวศัตรูได้แล้วก็ให้เกิดหมอกในฉากเป็นต้น

ฟังก์ชัน OnEndgame() เป็นฟังก์ชันสำคัญที่จะกำหนดว่าเงื่อนไข เกมจะจบเมื่อใด ซึ่งอาจจะควบคู่กับฟังก์ชันที่ผ่านมาได้หลายอย่าง เพื่อกำหนดรูปแบบ ของการจบเกมต่างกันไป เช่น เกมจบเมื่อตัวพระเอกหาสิ่งของที่ต้องการพบ หรือ จบเกมเมื่อตัวพระเอกกำจัดตัวศัตรูได้ตามที่กำหนด ลักษณะของสคริป คือ “ end game”

จะเห็นได้ว่าฟังก์ชันที่กำหนดมาให้เป็นฟังก์ชันพื้นฐานโดยทั่วไปของเกมแนว RPG ทำให้ผู้ใช้สามารถนำไปสร้างเกม ให้มีเนื้อเรื่องตามต้องการได้

**- ส่วนที่เป็นคลาสในการทำงานด้าน 3 มิติ**

เกมแนว RPG นั้นมีข้อมูลที่จำเป็นต้องจัดการหลายส่วน ไม่ว่าจะเป็นตัวละครซึ่งต้องเก็บค่าต่างๆ เช่นตำแหน่งตัวละครบนแผนที่ ทั้งแกน x ,y และ z หรือค่าพลังต่างๆ หรือ จุดเปลี่ยนฉากซึ่งต้องกำหนดว่า จะเปลี่ยนจากฉากใดไปยังฉากใด จุดเปลี่ยนฉากอยู่ที่ตำแหน่งใดของแผนที่ หรือมีการกำหนด ค่าว่าทำงานหรือไม่ทำงาน หรือ ไฟล์หลักที่กำหนดว่าแผนที่รวมนั้นประกอบด้วยแผนที่ย่อยๆกี่แผนที่ และใช้ไฟล์อะไรเป็นตัวแสดงผลเป็นต้น

CStageGame เป็นคลาสที่ทำหน้าที่ช่วยในการแสดงผลข้อมูลด้านกราฟฟิกทั้งหมด เช่น การแสดงผลภาพแผนที่สามมิติ และเป็นคลาสที่จัดการกับข้อมูลไฟล์ env , map , trp

CCharacter เป็นคลาสที่ทำหน้าที่ทำหน้าที่เกี่ยวกับตัวละครโดยเฉพาะเช่น เพิ่มตัวละคร วางตัวละครบนแผนที่ แล้วเก็บตำแหน่ง แสดงผลสามมิติของตัวละคร รวมทั้งกำหนด ค่าต่างๆ เช่นพลังชีวิตเป็นต้น

CLibrary ทำหน้าที่ในการเปิดโปรเจกต์เกมที่มีอยู่แล้ว สร้างโปรเจกต์เกมใหม่ รวมทั้งเก็บข้อมูลโดยภายในคลาสจะมี โดยภายในประกอบด้วย

CListMFS ทำหน้าที่จัดการเรื่องการเพิ่มแผนที่เข้าไปในเกม ไม่ว่าจะเป็นการตั้งชื่อแผนที่ การกำหนดค่าว่าแผนที่ใดต่อกับแผนที่ใด และใช้ไฟล์ชื่อ อะไรในการแสดงผลสามมิติ รวมทั้งเป็นตัวสร้างไฟล์อื่นๆที่จำเป็นต่อการสร้างเกม ไม่ว่าจะเป็น ไฟล์ trp ไฟล์ scp ไฟล์ cmod หรือ ไฟล์ env เป็นต้น

CGameLib ซึ่งทำหน้าที่ในการดึงข้อมูลจากไลบรารีมาแสดงผลไม่ว่าจะเป็น ไฟล์ mmod ไฟล์ cdef ไฟล์ cmod ไฟล์ imod ซึ่งคลาสนี้ทำหน้าที่ในการเปิดไฟล์ขึ้นมาทั้งหมด ตอนเริ่มเปิดโปรเจกต์เกม

หลังจากขั้นตอนการออกแบบโครงสร้างของ GUI เสร็จสิ้นขั้นตอนต่อไปจะเป็นขั้นตอนการทดสอบความถูกต้องของ File Script ที่ GUI สร้างขึ้นมาได้ และนำ File Scriptไปใช้ร่วมกับเกมที่ได้สร้างไว้ ซึ่งจะกล่าวถึงในบทต่อๆไป