

แบบรูปการจัดการสิ่งผิดปรกติสำหรับการปรับแต่งส่วนต่อประสานในการปรับปรุงซอฟต์แวร์เกม

นายฐิติพงศ์ เต็งไตรรัตน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)

are the thesis authors' files submitted through the Graduate School.

EXCEPTION HANDLING PATTERNS FOR USER INTERFACE CUSTOMIZATION IN  
SOFTWARE GAMES MODIFICATION

Mr. Thitipong Tengtrirat

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science Program in Software Engineering  
Department of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
Academic Year 2012

|                                 |   |
|---------------------------------|---|
| หัวข้อวิทยานิพนธ์               | แบบรูปการจัดการสิ่งผิดปรกติสำหรับการปรับแต่งส่วนต่อ |
|                                 | ประสานในการปรับปรุงซอฟต์แวร์เกม                     |
| โดย                             | นายฐิติพงศ์ เต็งไตรรัตน์                            |
| สาขาวิชา                        | วิศวกรรมซอฟต์แวร์                                   |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล                |

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร.บุญสม เลิศธีรวัฒน์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ธนรัตน์ ชลิตาพงศ์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล)

.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.พิษณุ คนองชัยยศ)

.....กรรมการภายนอกมหาวิทยาลัย  
(ผู้ช่วยศาสตราจารย์ ดร.เบญจพร ลีธรรมาภรณ์)

ฐิติพงศ์ เต็งไตรรัตน์ : แบบรูปการจัดการสิ่งผิดปกติสำหรับการปรับแต่งส่วนต่อ  
 ประสานในการปรับปรุงซอฟต์แวร์เกม (EXCEPTION HANDLING PATTERNS FOR  
 USER INTERFACE CUSTOMIZATION IN SOFTWARE GAMES MODIFICATION)  
 อ. ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ.นครทิพย์ พร้อมพูล, 182 หน้า

การมอดเกมหรือการปรับแต่งเกมโดยผู้ใช้นั้นกลายเป็นที่นิยมแพร่หลายในปัจจุบัน  
 เนื่องจากทางผู้ใช้นั้นนิยมใช้มอดเป็นเสมือนการแสดงผลงาน แนวคิด นวัตกรรมใหม่ซึ่งต่อ  
 ยอดจากเกมที่ผู้พัฒนาปล่อยออกมา ทางผู้พัฒนาเกมก็อาศัยมอดเป็นตัวช่วยสนับสนุนการต่อ  
 ยอดอายุของเกม รวมไปถึงการนำแนวคิด คุณลักษณะของมอดนั้นๆ ไปปรับใช้กับเกมของตน  
 ต่อไป แต่การมอดเกมอันเกิดจากผู้ใช้นั้นมีพื้นฐานที่หลากหลาย ทำให้สามารถเกิด  
 ข้อผิดพลาดต่างๆ ส่งผลในการปรับแต่งบางรายการที่มีแนวคิด คุณลักษณะที่ดี ไม่ได้นำไปผนวก  
 รวมเป็นส่วนหนึ่งของคุณลักษณะของเกม

ดังนั้นงานวิจัยฉบับจึงนำเสนอแบบรูปเพื่อจัดการปัญหาที่เกิดขึ้นจากการปรับแต่ง  
 ซอฟต์แวร์เกมโดยผู้ใช้นั้น โดยการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติเป็นแกนหลักใน  
 การจัดการปัญหาที่เกิดขึ้น โดยมุ่งเน้นในกรณีการปรับแต่งส่วนต่อประสานซึ่งถือเป็นส่วนแรกสุด  
 ที่ผู้ใช้จะมีปฏิสัมพันธ์ด้วย จำนวน 6 แบบรูปของการปรับแต่งส่วนต่อประสาน เพื่อให้เกิดความ  
 สอดคล้องกับวัตถุประสงค์การปรับแต่งของผู้ใช้ที่หลากหลาย โดยการประยุกต์ใช้แบบรูปผ่าน  
 แผนภาพลิตเติลจิล แผนภาพคลาส และแผนลำดับ อีกทั้งได้ทดสอบแบบรูปที่นำเสนอ จากเกม  
 ตัวอย่างจำนวน 3 เกม ที่พัฒนาจากโอเพนซอร์ส ซึ่งครอบคลุมทุกแบบรูปที่นำเสนอ ผลการ  
 ทดสอบพบว่าแบบรูปที่นำเสนอสามารถจัดการกับปัญหาได้อย่างถูกต้อง

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....  
 สาขาวิชา.....วิศวกรรมซอฟต์แวร์.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....  
 ปีการศึกษา.....2555.....

# # 5470929321 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : END USER SOFTWARE ENGINEER, GAME MOD, DESIGN PATTERN

THITIPONG TENGTRIRAT: EXCEPTION HANDLING PATTERNS FOR USER  
INTERFACE CUSTOMIZATION IN SOFTWARE GAMES MODIFICATION.

MASTER PROJECT ADVISOR: ASST.PROF. NAKORNTHIP PROMPOON, 182 pp

Mod game or game customization by end-users is widely popular among end-user because end-user uses mod to show the new innovative concept, which capped off from game released. The game developer relies on mod to support the extension of game lifetime and includes the introduction of the new concept and the feature of the mod to incorporate for the new game. However, the mod game arising from the end-users have varieties of backgrounds may lead to cause errors easily. The result in some customization has great concept, great feature is finally integrated as a part of game features because of the error occurrence

This research presents the six patterns for software game modification problem handling from end-user. The proposed patterns are based on the exception handling patterns and focused on user interface customization. They are developed with an aim to cover varieties of end-user modification needs. The application of the proposed patterns is shown by using Little-Jill diagrams, class diagrams and sequence diagrams. In addition, the three sample open source games are selected to test the six proposed patterns. The testing result confirms that they can handle problems correctly.

Department : ~~Computer Engineer~~

Student's Signature .....

Field of Study : ~~Software Engineer~~

Advisor's Signature .....

Academic Year : ~~2012~~

### กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความเมตตาและความช่วยเหลือเป็นอย่างยิ่งจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล ที่ให้คำแนะนำทางด้านการศึกษา แนวทางการค้นคว้า คุณธรรม จริยธรรม มารยาทและข้อเสนอแนะต่างๆตลอดจนคอยดูแลให้ทำวิทยานิพนธ์นี้ลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ธนรัตน์ ชลิตาพงศ์ เป็นประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. พิษณุ คนองชัยยศและผู้ช่วยศาสตราจารย์ ดร. เบญจพร ลิ้มธรรมภรณ์ เป็นกรรมการสอบวิทยานิพนธ์ ซึ่งได้กรุณาสละเวลาและให้คำแนะนำต่างๆทั้งในการสอบ โครงร่างและการสอบวิทยานิพนธ์ของข้าพเจ้าเป็นอย่างยิ่ง

ขอขอบคุณเพื่อนๆที่ๆและน้องๆ ทั้งภาคในและภาคนอกเวลาที่ห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ของจุฬาลงกรณ์มหาวิทยาลัยและทุกคนที่ผ่านเข้ามาให้คำแนะนำ แลกเปลี่ยนความเห็น แก่ข้าพเจ้า และอยากขอบพระคุณอย่างยิ่งกับกลุ่มเพื่อนสนิทที่คอยสนับสนุนยามมีปัญหา เต็มกำลังใจ ยามท้อให้กันเสมอ คอยช่วยเหลือทุกๆด้านจนทำให้ผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วง

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดาและน้องสาวทั้งสองคนของข้าพเจ้า ที่ให้การสนับสนุนในทุกๆด้าน และเป็นกำลังใจให้กันตลอดมา

## สารบัญ

|   | หน้า      |
|---|-----------|
| บทคัดย่อภาษาไทย.....  | ง         |
| บทคัดย่อภาษาอังกฤษ.....   | จ         |
| กิตติกรรมประกาศ.....  | ฉ         |
| สารบัญ.....   | ช         |
| สารบัญตาราง.....  | ฅ         |
| สารบัญภาพ.....  | ฎ         |
| บทที่   |           |
| <b>1 บทนำ.....</b>  | <b>1</b>  |
| 1.1 ความเป็นมาและความสำคัญของปัญหา.....                                     | 1         |
| 1.2 วัตถุประสงค์ของการวิจัย.....  | 2         |
| 1.3 ขอบเขตของการวิจัย.....  | 2         |
| 1.4 ขั้นตอนและวิธีการดำเนินงานวิจัย.....                                    | 3         |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ.....  | 3         |
| 1.6 บทความวิชาการที่ได้รับการพิมพ์.....                                     | 3         |
| <b>2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....</b>                                 | <b>5</b>  |
| 2.1 ทฤษฎีที่เกี่ยวข้อง.....   | 5         |
| 2.1.1 เกมเอ็นจิน.....   | 5         |
| 2.1.2 มอดหรือการปรับแต่ง.....   | 6         |
| 2.1.3 วิศวกรรมผู้ใช้ขั้นปลาย.....   | 8         |
| 2.1.4 แบบรูปโมเดล-วิว-พรีเซ็นเตอร์หรือเอ็มวีพี.....                         | 9         |
| 2.2 งานวิจัยที่เกี่ยวข้อง.....  | 12        |
| 2.2.1 เป้าหมายการออกแบบและพื้นฐานการออกแบบแบบรูป<br>การพัฒนาเกมเอ็นจิน..... | 12        |
| 2.2.2 แผนภาพลิตเติล-จิล.....  | 15        |
| 2.2.3 แบบรูปจัดการสิ่งผิดปรกติ.....   | 16        |
| 2.2.4 การออกแบบส่วนต่อประสานของเกมโดเจสซิส.....                             | 22        |
| <b>3. การศึกษา วิเคราะห์ปัญหาของการปรับแต่งและระบุแนว ทางแก้ไข.....</b>     | <b>26</b> |

|           |   |           |
|-----------|---|-----------|
| 3.1       | วิเคราะห์รูปแบบและปัญหาของการมอดประเภทการปรับแต่งส่วนต่อ<br>ประสาน.....   | 27        |
| 3.2       | พัฒนาเกมเอ็นจินเพื่อใช้สำหรับการมอดหรือการขยายซอฟต์แวร์.....  | 32        |
| 3.3       | ผนวกเกมเอ็นจินที่ออกแบบร่วมกับกรณีการปรับแต่งส่วนต่อประสาน.....   | 38        |
| <b>4.</b> | <b>การออกแบบแบบรูปและการประยุกต์ใช้เพื่อการแก้ปัญหาของการมอดด้วย<br/>วิธีการปรับปรุงส่วนต่อประสาน.....</b>                    | <b>41</b> |
| 4.1       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>นำเสนอข้อผิดพลาดจาก การปรับปรุงส่วนต่อประสาน.....               | 43        |
| 4.2       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีเพื่อความพึงพอใจ.....                 | 47        |
| 4.3       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับแต่งเพื่อการใช้งานเฉพาะ.....   | 55        |
| 4.4       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน..... | 62        |
| 4.5       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับแต่งเพื่อเพิ่มประสิทธิภาพ..... | 70        |
| 4.6       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับแต่งเพื่อการปรับตัว.....       | 77        |
| 4.7       | การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับแต่งเพื่อการบำรุงรักษา.....    | 85        |
| 4.8       | สรุปการประยุกต์ใช้.....   | 92        |
| 4.9       | วิเคราะห์ความสัมพันธ์ของแบบรูปจัดการสิ่งผิดปรกติและหลักการการ<br>ออกแบบส่วนต่อประสานเกมไคเจสซิส.....                          | 94        |
| <b>5</b>  | <b>การทดสอบและการประเมินผลการออกแบบ.....</b>  | <b>95</b> |
| 5.1       | การสร้างกรณีศึกษาเพื่อกำหนดกรณีทดสอบ จากการศึกษาการปรับแต่ง<br>ส่วนต่อประสานตามกรณี.....                                      | 96        |
| 5.2       | ออกแบบและกำหนดรายละเอียดกรณีทดสอบ จากกรณีศึกษา.....   | 98        |
| 5.3       | ออกแบบแบบรูปเพื่อจัดการสิ่งผิดปรกติตามทฤษฎีอ้างอิงตาม<br>กรณีทดสอบ.....   | 99        |
| 5.4       | วิเคราะห์ความสัมพันธ์ของแบบรูปและทฤษฎีการออกแบบส่วนต่อ<br>ประสานเกมไคเจสซิส.....  | 104       |



| บทที่                           | หน้า   |
|---------------------------------|--|
| 5.4                             | สรุปผลการทดสอบและวิเคราะห์จากผลการทดสอบ..... 105   |
| 5.5                             | การทดลองด้วยซอฟต์แวร์เกมตัวอื่นๆ..... 105  |
| <b>6.</b>                       | <b>สรุปผลการวิจัย..... 108</b>   |
| 6.1                             | บทสรุปของผลงานวิจัย..... 108   |
| 6.2                             | ข้อจำกัดของงานวิจัย..... 109   |
| 6.3                             | งานวิจัยในอนาคต..... 109   |
| <b>รายการอ้างอิง..... 110</b>   |  |
| <b>ภาคผนวก</b>                  | <b>..... 113</b>   |
| ภาคผนวก ก                       | อภิธานศัพท์..... 114   |
| ภาคผนวก ข                       | โครงสร้างแบบรูปการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติเพื่อจัดการการนำเสนอล้อผิดพลาดจากการปรับปรุงส่วนต่อประสาน..... 115 |
| ภาคผนวก ค                       | เกมสล็อตแมชชีน..... 122  |
| ภาคผนวก ง                       | กรณีทดสอบทดสอบระบบต้นฉบับ..... 126   |
| ภาคผนวก จ                       | กรณีทดสอบของแบบรูป..... 143  |
| ภาคผนวก ฉ                       | เกม งู..... 165  |
| ภาคผนวก ช                       | เกม แข่งรถ..... 168  |
| ภาคผนวก ซ                       | การทดสอบแบบรูปการปรับแต่งส่วนต่อประสานกรณีเพื่อความพึงพอใจกับเกมงู..... 171  |
| ภาคผนวก ฌ                       | การทดสอบแบบรูปการปรับแต่งส่วนต่อประสานกรณีเพื่อการเพิ่มฟังก์ชันกับเกมแข่งรถ..... 176                                   |
| ประวัติผู้เขียนวิทยานิพนธ์..... | 182  |

## สารบัญตาราง

| ตารางที่ |   | หน้า |
|----------|---|------|
| 2.1      | ประเภทการออกแบบเป้าหมาย กลยุทธ์ และการประเมินการออกแบบ.....                       | 13   |
| 3.1      | รูปแบบ จุดประสงค์ และตัวอย่างการปรับแต่งส่วนต่อประสาน.....                        | 27   |
| 3.2      | รายละเอียดการปรับแต่ง.....  | 28   |
| 3.3      | ตัวอย่างการทำงาน การมอดปรับแต่งส่วนต่อประสานเพื่อความพึงพอใจของผู้ใช้...          | 29   |
| 3.4      | ตัวอย่างการกรณีแก้ไขข้อผิดพลาดด้วยแบบรูป.....                                     | 31   |
| 3.5      | ตารางแนวทางแก้ไขปัญหาและแบบรูปจัดการสิ่งผิดปกติที่นำมาประยุกต์ใช้.....            | 32   |
| 4.1      | ตัวอย่างการทำงานของแบบรูป.....  | 41   |
| 4.2      | ตัวอย่างการปรับแต่งเพื่อความพึงพอใจ.....  | 47   |
| 4.3      | ตัวอย่างการปรับแต่งเพื่อการใช้งานเฉพาะ.....                                       | 55   |
| 4.4      | ตัวอย่างการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน.....                                     | 62   |
| 4.5      | ตัวอย่างการปรับแต่งเพื่อประสิทธิภาพ.....  | 70   |
| 4.6      | ตัวอย่างการปรับแต่งเพื่อการปรับตัว.....   | 77   |
| 4.7      | ตัวอย่างการปรับแต่งเพื่อการปรับตัว.....   | 85   |
| 4.8      | ตารางรายการสรุปการประยุกต์ใช้.....  | 92   |
| 4.9      | ตารางความสัมพันธ์แบบรูปและหลักการการออกแบบส่วนต่อประสานเกมไคเจสซิส.....           | 94   |
| 5.1      | การปรับแต่งเกมสล็อตแมชชีนให้สอดคล้องกับกรณีการปรับแต่งส่วนต่อประสานที่ออกแบบ..... | 97   |
| 5.2      | แบบฟอร์มผลการกรณีศึกษา.....   | 98   |
| 5.3      | กรณีทดสอบ.....  | 99   |
| 5.4      | รายละเอียดกรณีทดสอบ การเปลี่ยนภาพ.....  | 103  |
| 5.5      | แบบฟอร์มการประเมินรายการตรวจสอบ.....  | 104  |
| 5.6      | ตารางความสัมพันธ์แบบรูปและทฤษฎีการออกแบบส่วนต่อประสานเกมไคเจสซิส                  | 105  |
| ง.1      | กรณีทดสอบเปลี่ยนภาพในช่องสล็อต (T01) .....  | 126  |
| ง.2      | กรณีทดสอบเพิ่มข้อความนำเสนอข้อมูลเมื่อชนะเกม (T02) .....                          | 128  |
| ง.3      | กรณีทดสอบเพิ่มกราฟฟิคแสดงข้อมูลขณะใช้งาน (T03) .....                              | 131  |
| ง.4      | กรณีทดสอบการรวมปุ่มการทำงาน (T04) .....   | 134  |
| ง.5      | กรณีทดสอบการเพิ่มฟังก์ชันใหม่ในระบบ (T05) .....                                   | 136  |

| ตารางที่ | หน้า  |
|----------|---|
| ง.6      | กรณีทดสอบการบำรุงรักษาระบบเมื่อมีการปรับแต่ง (T06) ..... 141  |
| จ.1      | รายละเอียดกรณีทดสอบ แบบรูปกรณีการใช้งานเฉพาะโดยเพิ่มข้อความ<br>นำเสนอข้อมูลเมื่อขณะเกม (T012) ..... 143     |
| จ.2      | รายละเอียดกรณีทดสอบ แบบรูปกรณีการเพิ่มฟังก์ชัน โดยทดสอบเพิ่มกราฟฟิก<br>แสดงข้อมูลขณะใช้งาน (T013) ..... 147 |
| จ.3      | รายละเอียดกรณีทดสอบแบบรูปกรณีการเพิ่มฟังก์ชัน โดยทดสอบเพิ่มกราฟฟิก<br>แสดงข้อมูลขณะใช้งาน (T014) ..... 151  |
| จ.4      | รายละเอียดกรณีทดสอบ แบบรูปกรณีการปรับตัว โดยเพิ่มฟังก์ชันใหม่ใน<br>ระบบ (T015) ..... 154                    |
| จ.5      | รายละเอียดกรณีทดสอบ แบบรูปกรณีการบำรุงรักษา โดยการบำรุงรักษาระบบ<br>เมื่อมีการปรับแต่ง (T016) ..... 159     |
| ช.1      | รายละเอียดกรณีทดสอบ (T020)..... 171   |
| ช.2      | รายละเอียดกรณีทดสอบ การเปลี่ยนภาพ เมื่อประยุกต์ใช้แบบรูป (T020)..... 173                                    |
| ฉ.1      | กรณีทดสอบการเพิ่มฟังก์ชันใหม่ในระบบ (OT30) ..... 176  |
| ฉ.2      | รายละเอียดกรณีทดสอบ แบบรูปกรณีการปรับตัว โดยเพิ่มฟังก์ชันใหม่ใน<br>ระบบ (T031) ..... 178                    |

## สารบัญภาพ

| ภาพที่ |  | หน้า |
|--------|--|------|
| 2.1    | แผนภาพต้นไม้ ตัวอย่างโครงสร้างของเกมเอนจิน.....  | 5    |
| 2.2    | ส่วนต่อประสานเกม world of warcraft.....  | 6    |
| 2.3    | ส่วนต่อประสาน world of warcraft กับมอด nUI.....  | 7    |
| 2.4    | แผนภาพวัตถุ (Object Diagram) ความสัมพันธ์ของแบบรูป เอ็มวีซี.....                                     | 10   |
| 2.5    | แผนภาพลำดับ (Sequence Diagram) แสดงการทำงานของแบบรูปเอ็มวีซี.....                                    | 10   |
| 2.6    | แผนภาพ การทำงานของแบบรูป เอ็มวีพี.....   | 11   |
| 2.7    | แผนภาพลำดับ (Sequence Diagram) การทำงานของเอ็มวีพี.....  | 11   |
| 2.8    | ตัวอย่างแผนภาพ ลิตเติ้ล-จิล.....   | 15   |
| 2.9    | สัญลักษณ์ขั้นตอนของลิตเติ้ล-จิล.....   | 16   |
| 2.10   | สัญลักษณ์กลไกเงื่อนไขของลิตเติ้ล-จิล.....  | 16   |
| 2.11   | แผนภาพ การทำงานแบบทางเลือกมีลำดับ.....   | 18   |
| 2.12   | แผนภาพ การทำงานแบบทางเลือกไม่มีลำดับ.....  | 18   |
| 2.13   | แผนภาพการทำงานของการแก้ไขแบบทันที.....   | 19   |
| 2.14   | แผนภาพการทำงานของการแก้ไขแบบเชื่อมต่อ.....   | 19   |
| 2.15   | แผนภาพแสดงการทำงานแบบทดลองใหม่.....  | 20   |
| 2.16   | แผนภาพการทำงานแบบเงื่อนไขการทำงานใหม่.....   | 20   |
| 2.17   | แผนภาพการทำงานของแบบปฏิเสธการทำงาน.....  | 21   |
| 2.18   | แผนภาพการทำงานของแบบขดเชยการกระทำ.....   | 21   |
| 2.19   | กราฟ การใช้แบบรูปจัดการสิ่งผิดปกติในกระบวนการทำเคมีบำบัดและระบบ<br>ใกล้เคียงข้อพิพาทบนเครือข่าย..... | 22   |
| 2.20   | ตัวละครเปิดแผนในโลกของเกม.....   | 24   |
| 2.21   | ส่วนต่อประสานแสดงจำนวนกระสุนปืนในเกม.....  | 24   |
| 2.22   | ส่วนต่อประสานแสดงตัวละครที่ผู้ใช้ควบคุมได้.....  | 25   |
| 2.23   | ส่วนต่อประสานแสดงภาพหน้าจอเมื่อโดนโจมตี.....   | 25   |
| 3.1    | แผนภาพกิจกรรมขั้นตอนการดำเนินงานวิจัย.....   | 26   |
| 3.2    | องค์ประกอบคุณลักษณะของเอนจิน.....  | 34   |
| 3.3    | องค์ประกอบภายนอกของคุณลักษณะส่วนขยายของเอนจิน.....   | 35   |
| 3.4    | แผนภาพคลาสโครงสร้างของเอนจิน.....  | 37   |

| ภาพที่ |  | หน้า |
|--------|--|------|
| 3.5    | แผนภาพคลาสความสัมพันธ์ของ เอ็มวีพีส่วนขยายและ เอ็มวีพีของเอนจิน.....   | 38   |
| 3.6    | แผนภาพการไหลของข้อมูลของการทำงานส่วนขยายในเอนจิน.....  | 39   |
| 3.7    | แผนภาพลำดับ (Sequence Diagram) การทำงานของส่วนขยายภายในเอนจิน...   | 40   |
| 4.1    | แผนภาพกิจกรรมการทำงานของส่วนขยาย เพื่อจัดการการทำงานของมอด.....  | 43   |
| 4.2    | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติในการจัดการมอดเกม.....                                      | 45   |
| 4.3    | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการนำเสนอดีพลาตจากการปรับปรุงส่วนต่อประสาน.....       | 46   |
| 4.4    | แผนภาพ ลิตเติล-จิล ตัวอย่างของการจัดการมอดเกม จากการทำมอดเกม.....  | 46   |
| 4.5    | ภาพตัวอย่างการเปลี่ยนขนาดภาพ.....  | 50   |
| 4.6    | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติในการทำงานกรณีการปรับแต่งเพื่อความพึงพอใจของผู้ใช้.....     | 52   |
| 4.7    | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีเพื่อความพึงพอใจ.....    | 53   |
| 4.8    | แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อความพึงพอใจของผู้ใช้.....                             | 54   |
| 4.9    | รูปภาพการเพิ่มข้อความแสดงความยินดี.....  | 58   |
| 4.10   | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อการใช้งานเฉพาะ.....                     | 59   |
| 4.11   | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีเพื่อการใช้งานเฉพาะ..... | 60   |
| 4.12   | แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการใช้งานเฉพาะ.....                                   | 61   |
| 4.13   | รูปภาพการเพิ่มข้อความแจ้งตัวเลขเพิ่มเติม เพื่อแสดงแต้มที่รับ.....  | 65   |
| 4.14   | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อการเพิ่มฟังก์ชันงาน.....                | 67   |
| 4.15   | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีเพิ่มฟังก์ชันงาน.....    | 68   |
| 4.16   | แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน.....                                 | 69   |

| ภาพที่ | หน้า  |
|--------|---|
| 4.17   | รูปภาพการรวมปุ่มการทำงาน..... 73  |
| 4.18   | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปรกติกรณีการ<br>ปรับแต่งเพื่อประสิทธิภาพ..... 74                     |
| 4.19   | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีเพื่อประสิทธิภาพ..... 75 |
| 4.20   | แผนภาพ ลิตเติ้ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อเพิ่ม<br>ประสิทธิภาพ..... 76                              |
| 4.21   | รูปภาพการปรับตัวจากฟังก์ชันการทำงานใหม่..... 80   |
| 4.22   | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปรกติกรณีการ<br>ปรับแต่งเพื่อการปรับตัว..... 82                      |
| 4.23   | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการปรับตัว..... 83       |
| 4.24   | แผนภาพลิตเติ้ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการ<br>ปรับตัว..... 84                                     |
| 4.25   | รูปภาพการเพิ่มตุ้เกม..... 88  |
| 4.26   | แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปรกติกรณีการ<br>ปรับแต่งเพื่อการบำรุงรักษา..... 90                   |
| 4.27   | แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการ<br>ปรับปรุงส่วนต่อประสาน กรณีการบำรุงรักษา..... 91    |
| 4.28   | แผนภาพลิตเติ้ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการ<br>บำรุงรักษา..... 92                                  |
| 5.1    | แผนภาพกิจกรรมขั้นตอนการทดสอบและการประเมินผลการออกแบบ..... 95  |
| 5.2    | การทำงานปกติก่อนทดสอบ..... 97   |
| 5.3    | ผลการทดสอบกรณีเพื่อความพึงพอใจของผู้ใช้ โดยการเปลี่ยนภาพในช่องสล็อต. 98   |
| 5.4    | แผนภาพ ลิตเติ้ล-จิล โครงสร้างของแบบรูปการปรับแต่งเพื่อความพึงพอใจของ<br>ผู้ใช้ สำหรับจัดการขนาดภาพช่องสล็อต..... 100      |
| 5.5    | ซอร์สโค้ดในการเพิ่มรูปภาพใหม่แก่ระบบ..... 100   |
| 5.6    | ซอร์สโค้ดการคัดกรองกรณีการปรับแต่งเกี่ยวกับรูปภาพ..... 101  |
| 5.7    | ซอร์สโค้ด Reject แจ้งผลการทำงานกรณีที่ไม่สามารถปรับแต่งได้..... 102   |

| ภาพที่ | หน้า  |
|--------|---|
| 5.8    | ภาพตัวอย่างส่วนต่อประสานที่ถูกแก้ไข ภาพเซอร์รี่ในช่องสล็อตถูกแทนที่ด้วย<br>สี่ปะระรด..... 102                       |
| 5.9    | แผนภาพคลาส แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่งเพื่อการบำรุงรักษา<br>แก้ไขภาพเซอร์รี่..... 103                   |
| 5.10   | ภาพที่ใช้สำหรับการปรับแต่ง..... 104   |
| ข.1    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อความพึงพอใจ..... 116               |
| ข.2    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อความพึงพอใจการใช้งานเฉพาะ..... 117 |
| ข.3    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อเพิ่มฟังก์ชันงาน..... 118          |
| ข.4    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อเพิ่มประสิทธิภาพ..... 119          |
| ข.5    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อการปรับตัว..... 120                |
| ข.6    | แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง<br>เพื่อการบำรุงรักษา..... 121             |
| ค.1    | ส่วนต่อประสานของเกม..... 123  |
| ค.2    | แผนภาพคลาสโครงสร้างของเกมสล็อตแมชชีน..... 124   |
| ง.1    | ภาพที่ใช้สำหรับการปรับแต่ง..... 127   |
| ง.2    | ผลการทำงานก่อนทำการทดสอบ T01..... 127   |
| ง.3    | ผลกรณีทดสอบ T01..... 128  |
| ง.4    | ซอร์สโค้ดข้อความยินดี..... 129  |
| ง.5    | ข้อความแสดงความยินดี..... 129   |
| ง.6    | ผลการทำงานก่อนทำการทดสอบ T02..... 130   |
| ง.7    | ผลกรณีทดสอบ T02..... 130  |
| ง.8    | ซอร์สโค้ดการเพิ่มกราฟฟิกเหรียญวิ่ง..... 132   |
| ง.9    | ผลการทำงานก่อนทำการทดสอบ T03..... 133   |
| ง.10   | ผลกรณีทดสอบ T03..... 133  |
| ง.11   | ซอร์สโค้ดการรวมปุ่ม..... 134  |

| ภาพที่ | หน้า  |
|--------|---|
| ง.12   | ผลการทำงานก่อนทำการทดสอบ T04..... 135   |
| ง.13   | ผลกรณีทดสอบ T04..... 135  |
| ง.14   | ซอร์สโค้ดการเพิ่มฟังก์ชันใหม่ในระบบ ..... 137   |
| ง.15   | ผลการทำงานก่อนทำการทดสอบ T05..... 140   |
| ง.16   | ผลกรณีทดสอบ T05..... 140  |
| ง.17   | ผลการทำงานก่อนทำการทดสอบ T06..... 141   |
| ง.18   | ผลกรณีทดสอบ T06..... 142  |
| จ.1    | ซอร์สโค้ดการเพิ่มข้อความภาพแสดงความยินดี..... 144   |
| จ.2    | ซอร์สโค้ดการแก้ไขการปรับแต่ง..... 145   |
| จ.3    | ผลกรณีทดสอบ T012..... 146   |
| จ.4    | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการใช้งานเฉพาะ โดยการเพิ่ม<br>ข้อความแสดงความยินดีเมื่อชนะเกม..... 146  |
| จ.5    | แผนภาพคลาส การทำงานของแบบรูปกรณีเพื่อการใช้งานเฉพาะ โดยการเพิ่ม<br>ข้อความแสดงความยินดีเมื่อชนะเกม..... 147   |
| จ.6    | ซอร์สโค้ดการเพิ่มกราฟฟิคเหรียญวง แก้ไขให้เรียกใช้ แบบรูปได้..... 148  |
| จ.7    | ซอร์สโค้ดสำหรับตรวจสอบการแสดงผลของกราฟฟิค ให้แสดงผลได้ปกติ..... 149   |
| จ.8    | ซอร์สโค้ดสำหรับตรวจสอบการทำงานของฟังก์ชัน ซึ่งทำให้กับการแสดงผลให้<br>ถูกต้องตามเงื่อนไข..... 149             |
| จ.9    | ผลกรณีทดสอบ T013..... 150   |
| จ.10   | แผนภาพลำดับ การทำงานของแบบรูปกรณีการเพิ่มฟังก์ชัน โดยการเพิ่มกราฟฟิค<br>แสดงข้อมูลขณะใช้งาน..... 150          |
| จ.11   | แผนภาพคลาส การทำงานของแบบรูปกรณีการเพิ่มฟังก์ชัน โดยการเพิ่มกราฟฟิค<br>แสดงข้อมูลขณะใช้งาน..... 151           |
| จ.12   | ซอร์สโค้ดการรวมปุ่มการทำงานโดยมีโครงสร้างเป็นไปตามแบบรูปการจัดการการ<br>ปรับแต่งกรณีเพื่อประสิทธิภาพ..... 152 |
| จ.13   | ซอร์สโค้ดการตรวจสอบลำดับการทำงาน..... 152   |
| จ.14   | ซอร์สโค้ดตรวจสอบความสมบูรณ์ของการทำงาน..... 152   |
| จ.15   | ผลกรณีทดสอบ T014..... 153   |
| จ.16   | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อประสิทธิภาพ โดยการรวมปุ่มการ<br>ทำงาน..... 153                          |



| ภาพที่ | หน้า  |
|--------|---|
| จ.17   | แผนภาพคลาส การทำงานของแบบรูปกรณีเพื่อประสิทธิภาพ โดยการรวมปุมการทำงาน..... 154                      |
| จ.18   | ซอร์สโค้ดส่วนการนำเสนอส่วนต่อประสาน..... 155  |
| จ.19   | ซอร์สโค้ดการจัดการฟังก์ชันงาน..... 156  |
| จ.20   | ซอร์สโค้ดการจัดการผลกระทบส่วนต่อประสาน..... 157   |
| จ.21   | ผลกรณีทดสอบ T015..... 158   |
| จ.22   | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการปรับตัว โดยการเพิ่มฟังก์ชันใหม่..... 158                   |
| จ.23   | ซอร์สโค้ดการบำรุงรักษาระบบโดยการตรวจสอบความสอดคล้องของส่วนต่อประสาน..... 159                        |
| จ.24   | ซอร์สโค้ดการบำรุงรักษาระบบโดยการแยกการทำงานของส่วนต่อประสาน..... 160                                |
| จ.25   | ซอร์สโค้ดการบำรุงรักษาระบบโดยการแยกการทำงานของส่วนต่อประสาน..... 162                                |
| จ.26   | ผลกรณีทดสอบ T016..... 163   |
| จ.27   | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการบำรุงรักษา โดยการบำรุงรักษาระบบเมื่อมีการปรับแต่ง..... 163 |
| จ.28   | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการบำรุงรักษา โดยการบำรุงรักษาระบบเมื่อมีการปรับแต่ง..... 164 |
| ฉ.1    | ส่วนต่อประสานของเกมงู..... 166  |
| ฉ.2    | แผนภาพคลาสโครงสร้างของเกมงู..... 167  |
| ช.2    | ส่วนต่อประสานของเกมแข่งรถ..... 169  |
| ช.2    | โครงสร้างของระบบเกมแข่งรถ..... 169  |
| ช.1    | ภาพงูกินจู..... 171   |
| ช.2    | ซอร์สโค้ดในการเปลี่ยนรูปภาพใหม่แก่ระบบ..... 172   |
| ช.3    | ผลการทำงานปกติก่อนทดสอบ T020..... 172   |
| ช.4    | ผลการทดสอบกรณีเพื่อความพึงพอใจของผู้ใช้ โดยการภาพงู T020..... 172                                   |
| ช.5    | ซอร์สโค้ดในการตรวจสอบขนาดภาพ..... 174   |
| ช.6    | ซอร์สโค้ดในการปรับขนาดภาพ..... 174  |
| ช.7    | ผลกรณีทดสอบ T021..... 175   |
| ช.8    | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อความพึงพอใจ โดยการเปลี่ยนภาพ..... 175                         |

| ภาพที่ |  | หน้า |
|--------|--|------|
| ช.9    | แผนภาพคลาส การทำงานของแบบรูปกรณีเพื่อความพึงพอใจ โดยการเปลี่ยน<br>ภาพ..... | 175  |
| ฅ.1    | ซอร์ซโค้ดในการเพิ่มฟังก์ชันเร่งความเร็ว.....                               | 177  |
| ฅ.2    | ผลการทำงานปกติก่อนทดสอบ (T030).....  | 177  |
| ฅ.3    | ผลการทดสอบกรณี (T030).....   | 177  |
| ฅ.4    | ซอร์ซโค้ดในการเพิ่มฟังก์ชันเร่งความเร็ว (ใหม่).....                        | 178  |
| ฅ.5    | ซอร์ซโค้ดจัดการฟังก์ชันเร่งความเร็ว .....                                  | 179  |
| ฅ.6    | ซอร์ซโค้ดจัดการการนำเสนอข้อมูล .....                                       | 179  |
| ฅ.7    | ผลกรณีทดสอบ T031.....  | 180  |
| ฅ.8    | แผนภาพลำดับ การทำงานของแบบรูปกรณีเพิ่มฟังก์ชัน ปุ่มเร่งความเร็ว.....       | 180  |
| ฅ.9    | แผนภาพคลาส การทำงานของแบบรูปกรณีเพิ่มฟังก์ชัน ปุ่มเร่งความเร็ว.....        | 181  |

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการพัฒนาซอฟต์แวร์ในทุกๆ โดเมนจำเป็นต้องมีการปรับปรุง เปลี่ยนแปลง ซอฟต์แวร์อยู่อย่างสม่ำเสมอ เพื่อให้สอดคล้องกับเทคโนโลยีหรือกระบวนการทางธุรกิจที่มีการพัฒนาอยู่อย่างต่อเนื่อง จากเหตุดังกล่าวจึงเกิดการพัฒนารายย่อย (Extension) ส่วนเพิ่ม (add-on) หรือปลั๊กอิน (plug-in) เข้ามาที่บทบาทในการพัฒนาซอฟต์แวร์ โดยวิธีข้างต้นพัฒนาเพื่อตอบสนองต่อความต้องการของผู้ใช้ในการปรับปรุง เปลี่ยนแปลง ซอฟต์แวร์เดิมแทนการพัฒนาซอฟต์แวร์ตัวใหม่ ในบางกรณีนั้นความต้องการข้างต้นอาจเป็นความต้องการของผู้ใช้เพียงบางกลุ่มเท่านั้น เป็นสาเหตุให้ความต้องการเหล่านั้นอาจจะถูกเพิกเฉยจากทีมผู้พัฒนา ผู้ใช้ชั้นปลายจึงพัฒนาส่วนขยายขึ้นมาเองเพื่อตอบสนองต่อความต้องการเฉพาะกลุ่มนั้นๆ หนึ่งการพัฒนาปรับปรุงซอฟต์แวร์โดยตัวผู้ใช้ชั้นปลายเรียกว่า การเขียนโปรแกรมโดยผู้ใช้ชั้นปลาย (End-User Programming) ในอุตสาหกรรมเกมเป็นหนึ่งในอุตสาหกรรมที่การพัฒนาในลักษณะดังกล่าวเช่นกัน เรียกว่าวิธีการมอด (mod) หรือ การปรับแต่งเกม (Game Modification) ปัจจุบันวิธีการมอดเป็นที่นิยมแพร่หลายเนื่องจากเป็นการยืดอายุของตัวเกมและการปรับปรุงเปลี่ยนแปลงที่สร้างขึ้นจากนวัตกรรมทางความคิดของผู้ใช้ชั้นปลาย โดยแนวคิดบางส่วนอาจถูกต่อยอดกลายเป็นองค์ประกอบสำคัญในเกมประเภทนั้นต่อไปหรืออาจจะกลายเป็นเกมใหม่ แต่จากการที่พัฒนาโดยผู้ใช้ชั้นปลายนั้นมีลักษณะพื้นฐานที่หลากหลายทำให้สามารถมีข้อผิดพลาดเกิดขึ้นได้ง่าย ดังนั้นจึงนำไปสู่แนวทางในการแก้ไขปัญหาที่เกิดขึ้นต่อไป

การมอดคือการปรับปรุงซอฟต์แวร์ที่มีความใกล้เคียงกับการแก้ไขซอฟต์แวร์ (Patching) อันเป็นการพัฒนา ปรับปรุงซอฟต์แวร์เช่นเดียวกัน แต่การแก้ไขซอฟต์แวร์ถูกพัฒนาโดยผู้พัฒนาแตกต่างจากการมอดที่ถูกพัฒนาโดยผู้ใช้ชั้นปลาย โดยการมอดนั้นผู้ใช้ชั้นปลายต้องมีการศึกษาทำความเข้าใจการพัฒนาด้วยตัวเอง สามารถเรียนรู้ผ่านเครื่องมือ คู่มือ และสังคมออนไลน์โดยอาจจะได้รับการสนับสนุนจากผู้พัฒนาหรือกลุ่มผู้ใช้ชั้นปลายด้วยตนเอง ผู้พัฒนา (Developer) จึงพัฒนาโครงสร้างของระบบเกมหรือที่เรียกว่า เกมเอ็นจิน (Game Engine) ให้สามารถรองรับการปรับปรุง แก้ไขเกมของตนเองได้ง่าย ทั้งนี้ยังช่วยให้เกิดความสะดวก รวดเร็ว ง่ายแก่การทำความเข้าใจซึ่งผู้ใช้ชั้นปลายก็ได้ประโยชน์ร่วมจากความสามารถดังกล่าวของเกมเอ็นจินด้วย อย่างไรก็ตามการออกแบบของระบบมักไม่เพียงพอที่จะรองรับแนวคิดการพัฒนาและพื้นฐานที่หลากหลายของผู้ใช้ชั้นปลายได้ การพัฒนา มอดจึงสามารถเกิดข้อผิดพลาดได้ง่ายส่งผลให้มอดขาดความน่าเชื่อถือเมื่อมีการปล่อยออก (release) อันเป็นผลกระทบสำคัญทำให้มอดโดยผู้ใช้ชั้นปลายที่มีแนวคิด (idea) น่าสนใจขาดความน่าเชื่อถือ

จากสาเหตุข้อผิดพลาดที่แสดงผลออกมา อาจจะเป็นการปิดโอกาสของผู้ใช้ชั้นปลายบางกลุ่มในการนำเสนอผลงาน

จากข้อผิดพลาดจากพัฒนาซอฟต์แวร์โดยผู้ใช้ชั้นปลายดังที่กล่าวมานำไปสู่การใช้วิศวกรรมซอฟต์แวร์มาช่วยในการจัดการสิ่งผิดปกติที่เกิดขึ้นเรียกว่า วิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลาย (End-User Software Engineer) มุ่งเน้นในการจัดการสิ่งผิดปกติที่เกิดจากการทำส่วนขยาย แก้ไขปรับปรุงซอฟต์แวร์มีผู้ใช้ชั้นปลายเป็นเป้าหมายเป็นสำคัญ มีหลัก 3 ประการได้แก่จัดการสิ่งผิดปกติ ยืนยันความถูกต้อง และการรองรับการขยายขอบเขตสนับสนุนการพัฒนาโดยผู้ใช้ชั้นปลาย [1] มีจุดประสงค์เพื่อให้โครงสร้างของระบบรองรับการพัฒนาโดยผู้ใช้ชั้นปลายและเพื่อให้ผู้ใช้ชั้นปลายสามารถทราบข้อผิดพลาดของตัวเองได้

งานวิจัยฉบับนี้ได้อ้างอิงหลักการวิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลาย เป็นแกนหลักในการควบคุมข้อผิดพลาดควบคู่ไปกับหลักการออกแบบเกมเอ็นจินซึ่งเป็นการออกแบบโครงสร้างของระบบเกมโดยการนำการออกแบบแบบรูปเป็นแกนหลักตามหลักวิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลายมุ่งเน้นการทำจัดการสิ่งผิดปกติที่นำเสนอแก่ผู้ใช้ โดยใช้แบบรูปจัดการสิ่งผิดปกติ (Exception Handling Pattern) อันเป็นแบบรูปที่นำเสนอแนวทางในการควบคุมเงื่อนไขที่ไม่พึงประสงค์เป็นสิ่งอ้างอิง

## 1.2 วัตถุประสงค์ของการวิจัย

นำเสนอการออกแบบแบบรูปเพื่อจัดการปัญหา ข้อผิดพลาดที่เกิดจากการปรับปรุงซอฟต์แวร์โดยผู้ใช้ชั้นปลายหรือการมอดในซอฟต์แวร์เกม โดยการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ มุ่งเน้นในการมอดประเภทการปรับแต่งส่วนต่อประสาน

## 1.3 ขอบเขตงานวิจัย

- 1) ประยุกต์ใช้การออกแบบแบบรูป สำหรับการมอดในอุตสาหกรรมเกม
- 2) ศึกษาและออกแบบกรณีทดสอบการมอดด้วยวิธีการปรับปรุงส่วนต่อประสาน และข้อผิดพลาดที่เกิดขึ้นในแต่ละกรณีโดยแบ่งประเภทข้อผิดพลาดเป็น 3 กลุ่มดังต่อไปนี้
  1. ข้อผิดพลาดนำเสนอผ่านการแสดงผล
  2. ข้อผิดพลาดที่ส่งผลกระทบต่อการทำงานส่วนอื่น
  3. ข้อผิดพลาดที่ส่งผลให้ระบบไม่สามารถทำงานได้
- 3) ประเมินความถูกต้องของการออกแบบแบบรูป ด้วยระบบต้นแบบซึ่งพัฒนามาจากโอเพนซอร์สเพื่อใช้ในการทดสอบ
- 4) ข้อผิดพลาดที่เกิดขึ้นอ้างอิงตามขอบเขตที่เอ็นจินนั้นๆ รองรับเท่านั้น

- 5) ทดสอบจากไอเฟนเซอร์สของเกม 3 ประเภท ดังนี้
  - เกมกระดาน (Board Game)
  - เกมพีซเซิล (Puzzle Game)
  - เกมแข่งรถ (Racing Game)
- 6) การออกแบบแบบเบบรูป รองรับการปรับปรุงซอฟต์แวร์สำหรับอุตสาหกรรมเกมรองรับ ด้วยวิธีการ ปรับปรุงส่วนต่อประสานกับผู้ใช้ เท่านั้น

#### 1.4 วิธีดำเนินงานวิจัย

- 1) ศึกษาและทำความเข้าใจลักษณะทั่วไป และกระบวนการการปรับปรุงซอฟต์แวร์ในอุตสาหกรรม
- 2) ศึกษาและวิเคราะห์ประเด็นปัญหาที่ต้องพิจารณาในการปรับปรุงซอฟต์แวร์ ด้วยการมอดในอุตสาหกรรมเกม
- 3) ศึกษาและทำความเข้าใจ วิธีการออกแบบเกมเอ็นจิน
- 4) การออกแบบรูปแบบ และศึกษาหลักการอื่นๆที่เกี่ยวข้องกับออกแบบ กำหนดลักษณะเชื่อมโยง กรณีปัญหากับการออกแบบรูปแบบ นำมาวิเคราะห์ถึงผลการแก้ปัญหา
- 5) เลือกรุ่นมือ และพัฒนาระบบต้นแบบตามทีออกแบบไว้
- 6) ทดลองและประเมินผลวิธีวิจัยและระบบที่พัฒนาขึ้นเพื่อสนับสนุนแนวคิดในงานวิจัย
- 7) จัดทำบทความวิชาการ
- 8) สรุปผลแนวทางการวิจัย ข้อเสนอแนะและจัดทำเล่มโครงการ

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้การออกแบบแบบรูปใช้ในการปรับปรุงซอฟต์แวร์เกม เพื่อลดข้อผิดพลาดที่เกิดขึ้นจากการทำมอด ที่เกิดจากกรณีการปรับปรุงส่วนต่อประสาน
- 2) ได้วิธีการนำไปใช้ และตัวอย่างของการประยุกต์ใช้แบบรูปเพื่อใช้ในการปรับปรุงซอฟต์แวร์เกม ในกรณีปรับปรุงส่วนต่อประสาน เพื่อลดข้อผิดพลาด

#### 1.6 บทความวิชาการที่ได้รับการตีพิมพ์

ในงานวิจัยนี้ ผู้วิจัยมีผลงานวิชาการร่วมกับคณะผู้วิจัย ซึ่งบทความวิชาการระดับนานาชาติรวม 1 บทความ ได้แก่ บทความวิชาการเรื่อง “การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติสำหรับการปรับแต่งส่วนต่อประสานในการปรับปรุงซอฟต์แวร์เกม (Applying Exception Handling

Pattern for User Interface Customization in Software Games Modification)” ซึ่งได้รับการคัดเลือกนำเสนอและตีพิมพ์ในงาน”งานประชุมวิชาการนานาชาติของวิศวกรรมและวิทยาการคอมพิวเตอร์ (The International MultiConference of Engineers and Computer Scientists 2013: IAENG 2013)” ระหว่างวันที่ 13-15 มีนาคม 2556 ณ โรงแรมรอยัลการ์เดน เกาลูน เขตปกครองพิเศษฮ่องกง

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

ในงานวิจัยมีทฤษฎีที่เกี่ยวข้องได้แก่ เกมเอ็นจิน (Game Engine), มอดหรือการปรับแต่ง (Modding or Modification), วิศวกรรมผู้ใช้ชั้นปลาย (End-User Software Engineer), รูปแบบเดล-วิว-พรีเซ็นเตอร์หรือเอ็มวีพี (Model-View-Present Pattern : MVP) โดยมีรายละเอียดดังต่อไปนี้

##### 2.1.1 เกมเอ็นจิน (Game Engine)

เกมเอ็นจินเป็นสถาปัตยกรรมพื้นฐานของระบบเกม ที่ออกแบบระบบเพื่อช่วยเหลือในการพัฒนาซอฟต์แวร์เกม โดยลักษณะพื้นฐานของเกมเอ็นจินจะสามารถแยกองค์ประกอบเป็นส่วนๆ เพื่อนำไปปรับปรุง ใช้ซ้ำได้ รวมถึงการสร้างองค์ประกอบใหม่เพิ่มเติม การออกแบบเกมเอ็นจินเป็นการออกแบบโครงสร้างของระบบหรือ ขอบเขตของระบบ เพื่อง่ายแก่นำไปใช้พัฒนา ปรับปรุงพัฒนาต่อได้ อย่างสะดวกและง่ายยิ่งขึ้น การพัฒนาเกมเอ็นจินเป็นเสมือนแก่นหลักของการพัฒนาเกม [2,3,4] เป็นการพัฒนาสถาปัตยกรรมพื้นฐานระบบที่มีคุณลักษณะ (Feature) เป็นตัวกำหนดลักษณะการทำงานที่เป็นเอกลักษณ์ของเกมเอ็นจินแต่ละตัว เกมเอ็นจินจึงมีการกำหนดคุณลักษณะเฉพาะของเกมเอ็นจิน เพื่อให้สร้างความแตกต่างหรือจุดเด่นในการใช้งานซึ่งคุณสมบัติที่กล่าวมาจะสืบทอดไปสู่เกมอื่นๆ ที่ใช้เกมเอ็นจินเดียวกันพัฒนาต่อไป คุณลักษณะเฉพาะดังกล่าวประกอบด้วย ฟังก์ชันความพึงพอใจ (Satisfies Function) และ ความต้องการเชิงคุณภาพ (Non-function Requirement) ที่ผ่านการออกแบบตามจุดประสงค์ของเอ็นจินแต่ละตัว[5] การออกแบบเกมเอ็นจินซึ่งมีการพัฒนาหลากหลายแบบตาม จุดมุ่งหมายหรือความต้องการของการพัฒนาเกมเอ็นจิน มีลักษณะดังภาพที่ 2.1

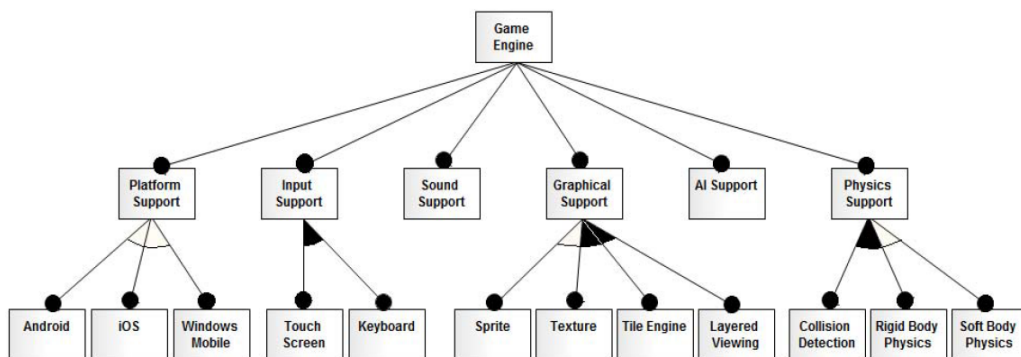


Figure 1.Feature Diagram For A Typical Game Engine

ภาพที่ 2.1 แผนภาพต้นไม้ ตัวอย่างโครงสร้างของเกมเอ็นจิน [1]

### 2.1.2 มอดหรือการปรับแต่ง (Modding or Modification)

นักพัฒนาเกมพัฒนาองค์ประกอบต่างๆ (เช่น กฎของเกม พฤติกรรม ตัวละคร ) โดยใช้พื้นฐานจากเกมเอ็นจิน โดยตัวเกมเอ็นจินอนุญาตให้ผู้พัฒนาสามารถสร้าง กฎ องค์ประกอบใหม่ๆ ของเกมได้จากการปรับเปลี่ยน การทำส่วนขยายซอฟต์แวร์ (software extension) [6,7] หรือนำองค์ประกอบเดิมๆมาใช้ใหม่ เกมเอ็นจินที่เป็นที่นิยมหลายตัวมักมีการออกแบบช่องทางหรือวิธีการเพื่อให้ง่ายแก่การใช้งาน ตัวอย่างเช่น ภาษาสคริปต์ โปรแกรมสนับสนุนในลักษณะของเครื่องมือให้สามารถทำงานง่ายขึ้น ซึ่งการใช้งานสิ่งเหล่านี้เพื่อนำไปสร้างเป็นเกมใหม่ หรือแก้ไขให้ต่างไปจากเดิม เรียกว่า มอด (mod) โดยมักใช้เรียกสำหรับการพัฒนาเกมใหม่จากเอ็นจินเกมเดิมโดยผู้ใช้ชั้นปลาย

การมอดเป็นนวัตกรรมทางความคิดที่เกิดจากแนวคิดของผู้ใช้ชั้นปลาย หรือประสบการณ์ในการเล่นเกมาถึงส่วนที่ควรมีจนกลายเป็นความต้องการในการพัฒนา ผลที่ได้จากการมอดอาจจะไม่ใช่สิ่งใหม่ เป็นเพียงสิ่งที่พัฒนาต่อยอดจากระบบเกมเดิมหรืออาจจะสามารถทำงานโดยลำพังได้ (Stand-alone) ยังคงต้องพึ่งพาระบบต้นฉบับของเกมเป็นพื้นฐานในการทำงาน นอกจากนี้ที่มาของความต้องการจากประสบการณ์หรือแนวคิดใหม่จากผู้ใช้ชั้นปลายแล้ว ยังรวมถึงการแลกเปลี่ยนความเห็นจากสังคมเครือข่ายของผู้ใช้ชั้นปลายด้วยกันเอง อนึ่งการมอดนำไปสู่การรวมตัวเป็นทีมพัฒนาของผู้ใช้ชั้นปลายเอง ทำให้เกิดการเรียนรู้ใหม่จากเหล่าผู้ใช้ชั้นปลายในกลุ่มที่มีวัฒนธรรมสังคมที่ต่างกัน ในบางกรณีการมอดอาจเป็นต้นแบบของเกมในรุ่นต่อไป (next version)

ภาพที่ 2.2 นำเสนอตัวอย่างส่วนต่อประสานเกมปกติของเกม เวิร์ลออฟวอร์คราฟท์ และ ภาพที่ 3 นำเสนอตัวอย่างการปรับปรุงส่วนต่อประสานเกมเดียวกัน



ภาพที่ 2.2 ส่วนต่อประสานเกม world of warcraft [8]





ภาพที่ 2.3 ส่วนต่อประสาน world of warcraft กับมอด nUI [9]

การมอด ถูกแบ่งเป็น 4 ประเภทได้แก่

#### 1) การปรับแต่งส่วนต่อประสาน (User Interface Customizations)

การแก้ไขปรับแต่งส่วนต่อประสาน เป็นการเพิ่ม แก้ไขส่วนหนึ่งส่วนใดแก่ส่วนต่อประสาน ทั้งสิ่ง que ผู้ใช้สามารถมีปฏิสัมพันธ์ได้เช่น ปุ่มต่างๆ หรือสิ่งที่ผู้ใช้ไม่สามารถมีปฏิสัมพันธ์ได้โดยตรงเช่น ภาพพื้นหลัง ข้อความบรรยาย เป็นต้น โดยเป็นการปรับแต่งเพื่อตอบสนองความต้องการของผู้ใช้ 3 ประเภทได้แก่

##### (1) ปรับแต่งเพื่อสร้าง ส่วนต่อประสานเป็นเอกลักษณ์ของตัวเอง

การสร้างส่วนต่อประสานเพื่อความพึงพอใจส่วนตัว หรือเพื่อสร้างความต่าง ความโดดเด่น แตกต่างจากระบบเดิม

##### (2) การปรับแต่งการแสดงผล

การปรับแต่งเพื่อให้เกิดประสิทธิภาพในการทำงาน การใช้งาน มากยิ่งขึ้น และรวมถึงการปรับแต่งส่วนหนึ่งส่วนใดเพื่อให้ง่ายแก่การทำความเข้าใจ เช่นทำสีให้เด่นชัด การทำกรอบเน้นข้อความ การเพิ่มเติมเสียง เป็นต้น

##### (3) เพิ่มส่วนต่อประสานสืบเนื่องมาจากการเพิ่มโมดูล

การปรับแต่งส่วนต่อประสานเมื่อเกิดการปรับปรุงโมดูลหรือการฟังก์ชันงานใดๆแล้วส่งผลต่อส่วนต่อประสาน

#### 2) การแปลงเกม (Game Conversions)

การปรับเปลี่ยน ฟังก์ชันการทำงานของเกม ให้มีลักษณะการทำงานที่แตกต่างจากเดิม

- การปรับแต่งการควบคุม ความสามารถของผู้ใช้ และแก้ไข ปรับปรุง เพิ่มเติม ปัญหาประติษฐ์
- การจัดการกับวัตถุ
- การจัดการกับสภาพแวดล้อม โชน หรือแผนที่

- การจัดการกฎของเกม
- การแก้ไขข้อบกพร่องของเกม นอกเหนือจากที่กล่าวมา ซึ่งอาจจะเป็นนวัตกรรมใหม่
- การแก้ไขทั้งระบบ หรือการแก้ไขหลายๆประเภทข้างต้นร่วมกัน

### 3) การเปลี่ยนรูปแบบ (Mechinima)

การปรับแต่งระบบเกมเพื่อใช้งานในรูปแบบอื่นๆ เช่น ใช้เกมเป็นเครื่องมือสำหรับทำภาพยนตร์ เครื่องมือสำหรับทำภาพคอมพิวเตอร์กราฟิก

### 4) การเจาะระบบเกมปิด (Hacking Closed Game Systems)

การแก้ไขระบบปิดของซอฟต์แวร์เกมที่ผิดต่อสัญญาการติดตั้งและใช้งานของเกม โดยมีจุดประสงค์ใดๆ ซึ่งถือเป็นการประสงค์ร้ายต่อผู้ใช้อื่นๆภายในระบบ หรือการสร้างข้อได้เปรียบให้แก่ตนที่อยู่นอกเหนือจากกฎที่ผู้พัฒนากำหนด เช่น การปรับแต่งฟังก์ชันงานในซอฟต์แวร์เกมเพื่อสร้างความได้เปรียบในการแข่งขัน การเจาะระบบฐานข้อมูลของเกมเพื่อขโมยข้อมูลสำคัญของผู้เล่น เป็นต้น

#### 2.1.3 วิศวกรรมผู้ใช้ชั้นปลาย (End-User Software Engineer)

ในการพัฒนาโครงการพัฒนาเกมทุกๆโครงการ ภายหลังปล่อยตัวเกมแก่ผู้ใช้งาน ตัวผู้ใช้มักมีความต้องการใหม่ๆจากแนวคิด ประสบการณ์ ทักษะการใช้งานที่แตกต่างกัน หรือต่อยอดจากคุณสมบัติของซอฟต์แวร์เดิม (Feature) ผู้พัฒนาเกมจะนำความต้องการเหล่านั้นไปพัฒนา ปรับปรุง แก้ไขต่อไป (patching) แต่ในบางกรณีอาจจะเป็นความต้องการเพื่อตอบสนองต่อเป้าหมายเฉพาะกลุ่ม ซึ่งความต้องการกลุ่มดังกล่าวมักถูกเพิกเฉยจากผู้พัฒนา และถูกนำไปพัฒนา ปรับปรุง ต่อโดยกลุ่มผู้ใช้เอง ซึ่งการพัฒนาโดยผู้ใช้ชั้นปลายนี้ เรียกกันว่า การเขียนโปรแกรมโดยผู้ใช้ชั้นปลาย (End-User Programming) [10, 11] หรือ การพัฒนาซอฟต์แวร์โดยผู้ใช้ชั้นปลาย (End-User Development)

การเขียนโปรแกรมโดยผู้ใช้ชั้นปลายเป็นการเขียนโปรแกรมซึ่งพัฒนาซอฟต์แวร์จากผู้ใช้ชั้นปลาย เริ่มมีการใช้งานกันแพร่หลาย มีเครื่องมือที่ใช้งานง่ายสะดวก และตรงตามความต้องการของผู้ใช้ ตัวอย่างเช่น โปรแกรมประเภทตารางจัดการ (Spreadsheet) เป็นต้น แต่การพัฒนาด้วยวิธีดังกล่าว นั้นจะมีปัญหาตามมา อันเนื่องความสามารถของผู้ใช้ ความซับซ้อนของระบบ ส่งผลให้ซอฟต์แวร์เกิดความผิดพลาดได้ง่าย อันเนื่องมาจากความสามารถ ทักษะ ของตัวผู้ใช้ชั้นปลายเอง ความไม่สมบูรณ์ดังกล่าวจึงนำไปสู่การพัฒนาวิธีการในการจัดการปัญหา หรือป้องกันปัญหาที่เกิดขึ้น เรียกว่า วิศวกรรมซอฟต์แวร์เพื่อผู้ใช้ชั้นปลาย

การทำวิศวกรรมซอฟต์แวร์เพื่อผู้ใช้ชั้นปลาย มีหลักการอ้างอิงจากหลักทางวิศวกรรมซอฟต์แวร์แต่ไม่ครอบคลุมตามวงจรชีวิตซอฟต์แวร์ (Software Life Cycle) โดยมุ่งเน้นในจัดการสิ่งผิดปกติจาก

การเขียนโปรแกรมโดยผู้ใช้งานปลายทางเป็นสิ่งสำคัญ เพื่อให้ผู้ใช้งานมีส่วนร่วมในการตรวจสอบ เสนอแนวทาง ในการพัฒนาซอฟต์แวร์จนสำเร็จ ลดข้อผิดพลาดที่เกิดขึ้น ทำให้ผู้ใช้งานปลายทางสามารถเรียนรู้หรือ ทำความเข้าใจข้อผิดพลาดที่เกิดขึ้นได้ หลักการวิศวกรรมซอฟต์แวร์จากผู้ใช้งานปลายทางที่กล่าวไว้มีดังนี้

- ผลจากการทำวิศวกรรมซอฟต์แวร์จะช่วยเสริมในส่วนของการตรวจสอบข้อผิดพลาด แสดง ถึงข้อผิดพลาดให้ผู้ใช้งานทราบถึงผลของข้อผิดพลาดและสาเหตุของข้อผิดพลาดได้
- ยืนยันความถูกต้อง แสดงให้เห็นถึงสิ่งที่สำคัญ นำเสนอแนวทางเพื่อช่วยเหลือในวิธีการใช้การ ทดสอบ
- ขยายขอบเขต แสดงขอบเขต แนะนำสิ่งที่ทำต่อยอด สร้างแรงบันดาลใจหรือทักษะในการ พัฒนา นำสู่การปรับปรุงวิศวกรรมซอฟต์แวร์ต่อไป

โดยทั่วไปการเขียนโปรแกรมโดยผู้ใช้งานปลายทางเป็นการพัฒนา ปรับปรุง ซอฟต์แวร์โดยตัวผู้ใช้งาน ซึ่งการพัฒนาดังกล่าวเป็นสิ่งจำเป็นในวงการซอฟต์แวร์เพื่อให้สามารถพัฒนาปรับปรุงซอฟต์แวร์ให้ รองรับการใช้งานได้สอดคล้องความต้องการของผู้ใช้ได้อย่างแท้จริง จึงนำไปสู่การทำวิศวกรรมซอฟต์แวร์ เพื่อผู้ใช้งานซึ่งช่วยในการสนับสนุนรองรับการจัดการปัญหาข้อผิดพลาดในการพัฒนาโดยมี สาเหตุมาจากพื้นฐานผู้ใช้ที่หลากหลาย เพื่อให้ผู้ใช้งานมีความเข้าใจ สามารถแก้ไข ข้อผิดพลาดที่เกิดขึ้นได้ ด้วยตัวเอง หรือช่วยในการจัดการสิ่งผิดปกติที่เกิดขึ้น ทำให้ผลผลิตจากพัฒนาซอฟต์แวร์โดยผู้ใช้งาน ปลายทางนั้นมีความน่าเชื่อถือ มีความสมบูรณ์เป็นที่น่าพอใจ

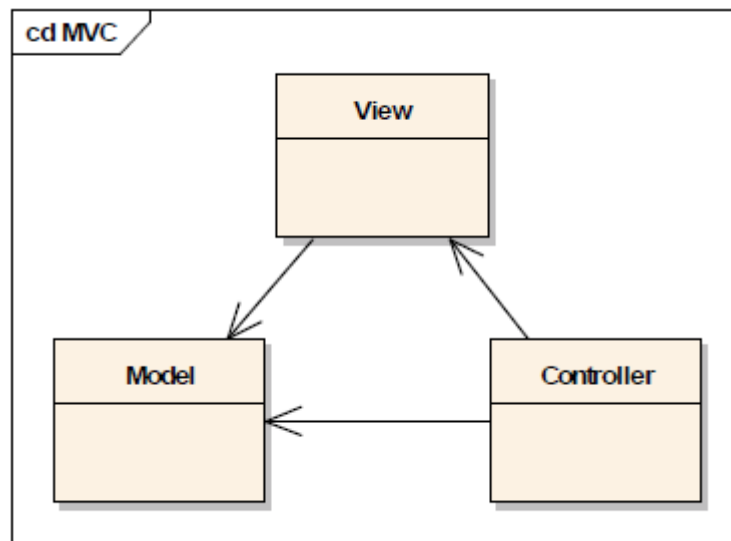
ทั้งนี้จากงานวิจัยกำหนดนิยามผู้พัฒนาและผู้ใช้งานดังต่อไปนี้

**ผู้พัฒนา (Developer)** ในโครงการวิจัย กล่าวถึงผู้พัฒนา เอนจินเกม หรือระบบเกมต้นฉบับ เป็น ผู้นำผลของงานวิจัยได้แก่ แบบรูป ไปประยุกต์ใช้ในการพัฒนาเอนจิน

**ผู้ใช้งานปลายทาง (End-User)** ในโครงการวิจัย คือ ผู้ใช้ซอฟต์แวร์เกม ผู้พัฒนามอดเกม (Modder) เป็น ผู้ได้ประโยชน์จากผลของงานวิจัย

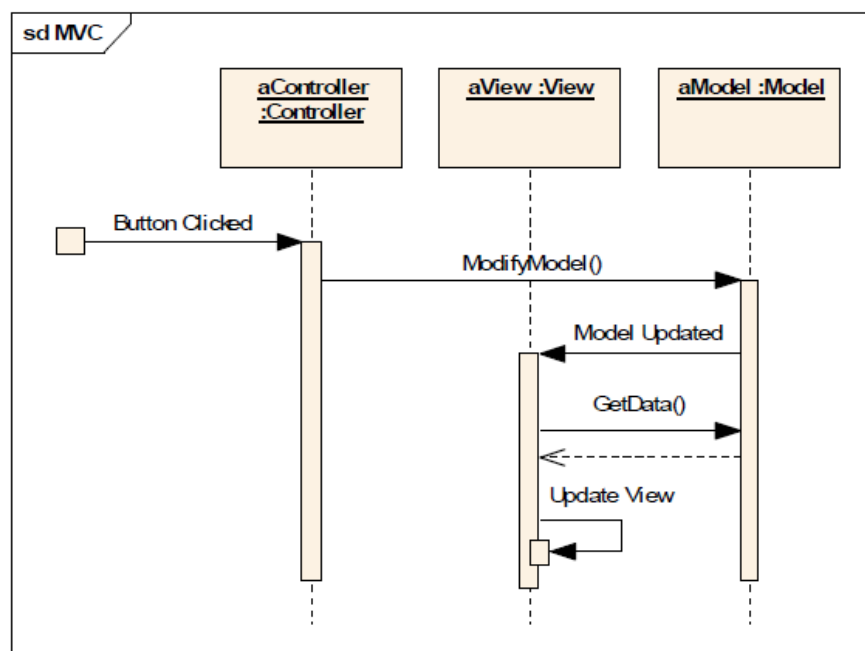
#### 2.1.4 แบบรูปโมเดล-วิว-พรีเซ็นเตอร์หรือเอ็มวีพี (Model-View-Present Pattern : MVP)

เอ็มวีพีเป็นแบบรูปที่พัฒนาจาก โมเดล-วิว-คอนโทรลเลอร์หรือเอ็มวีซี (Model-View-Controller : MVC) [12] เนื่องจากแบบรูปเอ็มวีซีสามารถให้ วิว เข้าถึงการใช้งานโมเดลได้ แม้จะต้อง มีการสั่งงานจากคอนโทรลเลอร์ ดังภาพที่ 2.4 และ 2.5



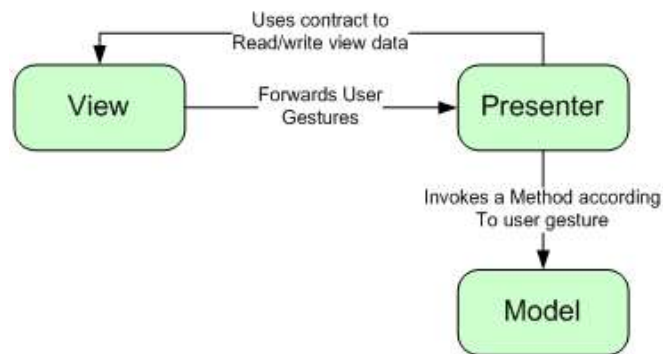
ภาพที่ 2.4 แผนภาพวัตถุ (Object Diagram) ความสัมพันธ์ของแบบรูป เอ็มวีซี

ซึ่งการทำงานของแบบรูปลักษณะนี้ทำให้ วิวต้องมีตรรกะ (logic) เป็นองค์ประกอบ ทำให้วิวมีคำสั่ง (code) มากขึ้นตามไปด้วย นอกจากนี้ผลที่วิวมีคำสั่งแทรกอยู่ภายในทำให้เมื่อมีการเปลี่ยนแปลงแก้ไข สถานะต่างๆของคำสั่งในวิวจะส่งผลต่อการทำงานของวิวที่นำเสนอแก่ผู้ใช้หรืออาจจะส่งผลต่อส่วนคอนโทรลเลอร์ได้ รูปแบบดังกล่าวทำให้เกิดการขึ้นต่อกัน (Dependency) ของการทำงานขึ้นระบบจึงมีความซับซ้อน ยากในการพัฒนา ทดสอบ และบำรุงในภายหลัง



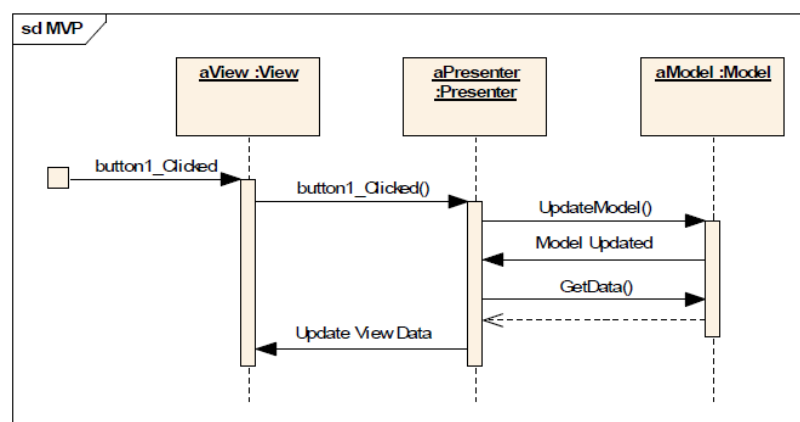
ภาพที่ 2.5 แผนภาพลำดับ (Sequence Diagram) แสดงการทำงานของแบบรูปเอ็มวีซี

เอ็มวีพีเป็นโมเดลที่แยกส่วน คอนโทรลหรือตรรกะเป็นอิสระออกจาก วิวเรียกว่า 프리เซ็นเตอร์ ทำให้วิวเป็นส่วนที่นำเสนอข้อมูลแก่ผู้ใช้ และเป็นส่วนต่อประสานให้ผู้ใช้มีปฏิสัมพันธ์ด้วยเท่านั้น ส่วน 프리เซ็นเตอร์ทำหน้าที่ในการรับข้อมูลนำเข้า(input) จากวิว มาจัดการเป็นศูนย์กลางของวิว ในการเข้าถึงโมเดล ดังภาพที่ 2.6



ภาพที่ 2.6 แผนภาพ การทำงานของแบบรูป เอ็มวีพี [13]

โดยความสัมพันธ์ระหว่าง วิวและฟรีเซ็นเตอร์เป็นไปในรูปแบบของ แบบรูป ดีโคเรเตอร์ (Decorator) อนึ่ง วิวมีความสัมพันธ์กับ ฟรีเซ็นเตอร์เป็นในรูปแบบของส่วนประกอบ ที่หนึ่งวิวอาจจะประกอบด้วยฟรีเซ็นเตอร์หลายตัว แทนการมีฟรีเซ็นเตอร์เฉพาะของตัวเองแบบหนึ่งต่อหนึ่งหรือหนึ่งวิวต่อหนึ่งฟรีเซ็นเตอร์ รูปแบบของเอ็มวีพีนั้นฟรีเซ็นเตอร์จะทำงานและส่งผลของแต่ละฟรีเซ็นเตอร์กลับไปให้วิวที่เรียกใช้ฟรีเซ็นเตอร์นั้นๆ การไม่ขึ้นต่อกันของวิวและฟรีเซ็นเตอร์ข้างต้น ทำให้ฟรีเซ็นเตอร์และวิวมีความซับซ้อนน้อยลง มีความสะดวกมากขึ้นในการทดสอบ ตรวจสอบตรรกะของคำสั่ง และทำให้สามารถนำฟรีเซ็นเตอร์ไปใช้งานใหม่หรือใช้งานซ้ำในวิวอื่นๆ ต่อไปได้ รูปแบบการทำงานของ เอ็มวีพีเป็นดังภาพที่ 2.7



ภาพที่ 2.7 แผนภาพลำดับ (Sequence Diagram) การทำงานของเอ็มวีพี

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 เป้าหมายการออกแบบและพื้นฐานการออกแบบแบบรูปการพัฒนาเกมเอ็นจิน

(A Design Goal and Design Pattern Based Approach for Development of Game Engine)

งานวิจัยนำเสนอขั้นตอนวิธีการออกแบบเกมเอ็นจินเพื่อตอบสนองต่อความต้องการในการพัฒนาระบบ โดยมีรายละเอียดเบื้องต้นประกอบด้วย การออกแบบคุณลักษณะเด่นของเอ็นจิน ส่วนประกอบของการทำงานภายใน เช่น ภาพ เสียง การควบคุม เป็นต้น และการผนวกองค์ประกอบเข้าด้วยกัน โดยแบ่งขั้นตอนการออกแบบมีดังต่อไปนี้ [14]

#### 1) การกำหนดคุณลักษณะของเกมเอ็นจิน (Defining Features of the Game Engine)

เป็นขั้นตอนในการกำหนด รายละเอียดเฉพาะใดๆ ตัวอย่างเช่น โดเมน ผ่านการกำหนดคุณลักษณะให้เป็นไปตามความต้องการ แบ่งออกเป็นฟังก์ชันงานและฟังก์ชันเชิงคุณภาพ การกำหนดคุณลักษณะควรกำหนดจากแพลตฟอร์มเป้าหมายและคุณลักษณะเด่นที่มีการนำเสนอชัดเจน ในบางกรณีสามารถรองรับคุณลักษณะได้หลากหลายประเภทพร้อมกันได้ เช่น การออกแบบเพื่อรองรับการทำงานหลากหลายแพลตฟอร์ม (Multiplatform)

#### 2) กำหนดชุดเป้าหมายการออกแบบและกลยุทธ์ (Defining a Set of Design Goals and Strategies)

การกำหนดเป้าหมาย กำหนดกฎและทฤษฎีให้สอดคล้องกับเป้าหมาย กลยุทธ์ในการออกแบบมีจุดมุ่งหมายแบ่งเป็น 4 จุดมุ่งหมายหลัก ปรากฏดังตารางที่ 2.1

ตารางที่ 2.1 ประเภทการออกแบบเป้าหมาย กลยุทธ์ และการประเมินการออกแบบ

| ประเภทการออกแบบเป้าหมาย          | กลยุทธ์การออกแบบ   | การประเมิน  |
|----------------------------------|--|---|
| (1) การใช้งาน (Usability)        | มีความสามารถศึกษา ทำความเข้าใจได้ง่าย<br>เป้าหมายหลักเพื่อลดการเรียนรู้เอ็นจินและ<br>เพิ่มความสามารถในการผลิต ตรงตามกลุ่มเป้าหมาย  | ผู้ใช้งานสามารถคาดเดาได้ ทำความเข้าใจได้<br>มีการใช้แผนภาพหรือเครื่องมือประกอบ<br>เพื่ออำนวยความสะดวก   |
| (2) ประสิทธิภาพ (Efficiency)     | จัดสรรทรัพยากร ในการใช้งานให้ได้คุ้มค่าที่สุด<br>ที่สุดได้ผลิตผลมากที่สุด  | <ul style="list-style-type: none"> <li>• การกำหนดข้อจำกัดในการใช้งาน</li> <li>• การกำหนดทรัพยากร</li> <li>• การแยกส่วนการทำงาน</li> <li>• การกำหนดการใช้งานในการสร้างทรัพยากรบางประเภท ตัวอย่างเช่น ตัวแปร</li> </ul> |
| (3) การเคลื่อนย้าย (Portability) | รองรับการเคลื่อนย้ายในหลากหลายแพลตฟอร์ม โดยมีระดับการเคลื่อนย้าย 3 ระดับได้แก่<br><ul style="list-style-type: none"> <li>• กลยุทธ์ในการบำรุงรักษาให้ทำงานเป็นปกติ</li> </ul> ขณะปฏิบัติงานในทรัพยากรที่แตกต่างกัน<br><ul style="list-style-type: none"> <li>• กลยุทธ์ในการบำรุงรักษาให้ทำงานเป็นปกติ หรือใกล้เคียงจากการใช้งานบนระบบที่มีองค์ประกอบแตกต่างกัน</li> <li>• กลยุทธ์ในการปรับตัวของโปรแกรมบนระบบที่มีทรัพยากรตรงตามเป้าหมายที่กำหนด</li> </ul> | ประเมินจากการกำหนดแพลตฟอร์มเป้าหมาย<br>ในการเคลื่อนย้ายชัดเจนและระบุระดับการเคลื่อนย้าย   |

ตารางที่ 2.1 ประเภทการออกแบบเป้าหมาย กลยุทธ์ และการประเมินการออกแบบ (ต่อ)

| ประเภทการออกแบบเป้าหมาย       | กลยุทธ์การออกแบบ  | การประเมิน  |
|-------------------------------|---|---|
| (4) การปรับตัว (Adaptability) | รองรับการปรับตัวของระบบในการปรับแต่ง ปรับปรุง บำรุง เพื่อให้สามารถทำงานได้ตามปกติ | การประเมินตามจุดประสงค์ของเอ็นจินที่ปรับตัวให้เป็นไปตามความต้องการอย่างไร |

### 3) สกัดเป้าหมายการออกแบบจากคุณลักษณะ (Extracting Design Goal from Features)

ในเป้าหมายการออกแบบจะมีความสัมพันธ์กับคุณลักษณะ ขั้นตอนต่อไปนี้เป็นขั้นตอนการแยก เป้าหมายที่มีความสัมพันธ์กับแต่ละคุณลักษณะ เพื่อแยกความสัมพันธ์แบบทางตรงและทางอ้อมออกจากกัน แสดงถึงคุณลักษณะหลักที่ตอบสนองต่อเป้าหมายกับคุณลักษณะเสริมที่หมายถึงคุณลักษณะที่มีการเรียกใช้งานในบางกรณีหรือไม่มีการเรียกใช้งานก็ได้

### 4) กำหนดแบบรูปการออกแบบ (Defining Design Pattern)

ขั้นตอนในการนำการออกแบบแบบรูปมาประยุกต์ใช้เพื่อให้สามารถทำงานตามเป้าหมายที่วางไว้ อาจจะเป็นเสริม หรือการกำหนดขอบเขต เพื่อแสดงให้เห็นว่ามีคุณลักษณะใดทำงานในขอบเขตที่กำหนดและสามารถควบคุมได้ การกำหนดการออกแบบแบบรูปตามคุณลักษณะที่สนใจ โดยนำองค์ประกอบที่กล่าวมาทั้งหมดนำเสนอในรูปแบบของแผนภาพหรือเอกสาร เพื่อให้เห็นถึงการทำงาน ความสัมพันธ์ของแต่ละองค์ประกอบในลักษณะของแบบรูป

### 5) การผนวกรวมแบบรูปและการกำหนดเป็นการออกแบบที่สมบูรณ์สำหรับเกมเอ็นจิน (Integrating all Defined Pattern and Forming the Complete Design of Game Engine)

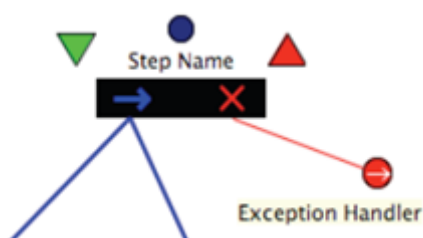
ขั้นตอนในการออกแบบเอ็นจินตามเป้าหมาย คุณลักษณะ ตามแบบรูปที่กำหนดไว้ ผลผลิตการออกแบบจากขั้นตอนก่อนหน้ามาผนวกรวมเป็นโครงสร้างของเอ็นจิน

นอกจากการนำเสนอขั้นตอนการออกแบบเกมเอ็นจินแล้วงานวิจัยดังกล่าวยังกล่าวถึงนำเสนอการนำทฤษฎีข้างต้นไปประเมินเกมเอ็นจินที่นิยมใช้ในอุตสาหกรรม โดยใช้การประเมินตามทฤษฎีเพื่อนำไปวิเคราะห์ ถึงเป้าหมายในการพัฒนาเอ็นจินแต่ละตัว ว่าเป็นไปตามเป้าหมายของลักษณะการใช้ของเอ็นจินนั้นๆ เพียงใด ทางผู้วิจัยนำขั้นตอนการออกแบบเอ็นจินจากงานวิจัยดังกล่าวเป็นอ้างอิงในการกำหนดเป้าหมายของเกมเอ็นจินภายในงานวิจัย



## 2.2.2 แผนภาพลิตเติล-จิล (Little-JIL Diagram)

เป็นแผนภาพแสดงลำดับ ขอบเขตของกระบวนการทำงานประกอบด้วยการนำเสนอการกระทำผ่านไวยากรณ์ภาพ มีประโยชน์ในการนำเสนอการทำงานในจัดการสิ่งผิดปกติช่วยในการลดรูปให้สามารถเข้าใจง่ายและกระชับสำหรับกระบวนการที่มีความซับซ้อนของเงื่อนไขมากมายในรูปแบบรันไทม์ (runtime) ลิตเติล-จิลนำเสนอขั้นตอนการทำงานผ่านความสัมพันธ์ของขั้นตอน (Step) ซึ่งเป็นหน่วยพื้นฐานของ ลิตเติล-จิล ขั้นตอนนำเสนอด้วยภาพสัญลักษณ์เป็นแถบสีดำดังภาพที่ 2.8 [15, 16] มีลักษณะการทำงานตามสัญลักษณ์ภายในของขั้นตอน โดยมีเส้นแสดงความสัมพันธ์กับการทำงานรูปแบบของสัญลักษณ์ภายในมี 2 รูปแบบคือขั้นตอนรอง (Substeps) และกลไกจัดการสิ่งผิดปกติ (Exception Handling Mechanisms) โดยมีรายละเอียดดังนี้

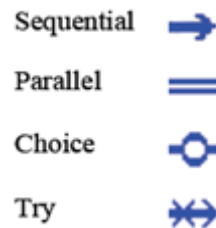


ภาพที่ 2.8 ตัวอย่างแผนภาพ ลิตเติล-จิล

### 1) ขั้นตอนรอง (Substeps)

เป็นรูปแบบการทำงานปกติเป็นขั้นตอนถัดไปจากขั้นตอนหลักหรือที่เรียกว่าขั้นตอนแม่ (Parent Step) ซึ่งเป็นแถบก่อนหน้า ขั้นตอนรองเป็นแถบขั้นตอนที่เชื่อมด้วยเส้นเชื่อมจากทางด้านซ้ายของแถบขั้นตอนแม่ โดยมีลักษณะการทำงานตามสัญลักษณ์ภายในแถบ 4 รูปแบบ ตามภาพที่ 2.9 มีรายละเอียดดังต่อไปนี้

- (1) ต่อเนื่อง (Sequential) การทำงานต่อเนื่องเรียงลำดับการทำงานจากซ้ายไปขวา โดยทำสำเร็จเพียง 1 ขั้นตอนจะสามารถเข้าสู่ขั้นตอนต่อไป
- (2) คู่ขนาน (Parallel) ทำงาน พร้อมกันทุกขั้นตอนภายใต้สัญลักษณ์
- (3) ตัวเลือก (Choice) เลือกทำเพียงขั้นตอนเดียวโดยไม่มีลำดับ
- (4) ทดสอบ (Try) ทำงานจากซ้ายไปขวาจจนกว่าจะสำเร็จทั้งหมด



ภาพที่ 2.9 สัญลักษณ์ขั้นตอนของลิตเติล-จิล

## 2) กลไกจัดการสิ่งผิดปกติ (Exception Handling Mechanisms)

เป็นรูปแบบขั้นตอนการทำงานถัดไปเมื่อเกิดความผิดพลาดที่ไม่สามารถทำตามขั้นตอนรองที่สืบทอดจากขั้นตอนแม่ได้ เป็นแถบขั้นตอนที่เชื่อมต่ออยู่ในส่วนด้านขวามีเครื่องหมาย X โดยผ่านเส้นเชื่อม มีลักษณะการทำงานตามสัญลักษณ์ภายในแถบ 4 รูปแบบ ตามภาพที่ 2.10 มีรายละเอียดดังต่อไปนี้

- (1) สมบูรณ์ (Completion) เมื่อเข้าสู่ส่วนควบคุมความผิดพลาด จะจบการทำงานของขั้นตอนรอง
- (2) ต่อเนื่อง (Continuation) เมื่อเข้าสู่เงื่อนไขจะทำงานต่อตามกระบวนการต่อไป
- (3) เริ่มใหม่ (Start) เมื่อเข้าสู่เงื่อนไขจะเริ่มต้นการทำงานของกระบวนการใหม่
- (4) การโยนซ้ำ (Rethrow) เมื่อเข้าสู่เงื่อนไข จะเลือกชดเชยไปทำกระบวนการอื่นๆตามที่กำหนด



ภาพที่ 2.10 สัญลักษณ์กลไกเงื่อนไขของลิตเติล-จิล

### 2.2.3 แบบรูปจัดการสิ่งผิดปกติ (Exception Handling Pattern)

**แบบรูป(Pattern)** คือวิธีการแบบแผน หรือแนวทางในการจัดการปัญหาใดๆที่เกิดขึ้นอย่างสม่ำเสมอ และไม่มีรูปแบบตายตัวในการนำไปใช้ แบบรูปจะอธิบายถึงลักษณะโครงสร้างหรือแนวทางการนำไปประยุกต์ใช้ในลักษณะงานต่างๆ [15, 16]

**แบบรูปการออกแบบ (Design Pattern)** เป็นการนำแบบรูปไปประยุกต์ใช้โดยไม่เฉพาะเจาะจงการนำไปใช้งาน โดยจะแสดงความสัมพันธ์ระหว่าง คลาส (class) หรือ วัตถุ (object)

**แบบรูปจัดการสิ่งผิดปกติ (Exception Handling Pattern)** แบบรูปที่ออกแบบมาเพื่อใช้ในการควบคุมการทำงานเมื่อเกิดการผิดปกติในการทำงาน เพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพ หรืออยู่ภายใต้การควบคุม โดยมีลักษณะการควบคุมการทำงาน 3 รูปแบบดังนี้

- การนำเสนอ จัดการสิ่งผิดปกติโดยการนำเสนอผลการทำงานอื่นๆทดแทนการทำงานเดิม
- การเพิ่ม จัดการสิ่งผิดปกติจากการแทรกการทำงานเข้าไป โดยนิยามว่างานที่เพิ่มไปจะสามารถทำให้ผลการทำงานเป็นไปตามปกติ
- การยกเลิก การยกเลิกการทำงานโดย เพิ่มการทำงานใดๆเข้าไปเพื่อควบคุมการทำงานก่อนยกเลิกการทำงาน

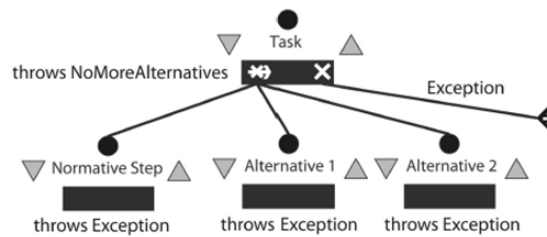
แบบรูปของจัดการสิ่งผิดปกติเป็นการออกแบบแบบรูปเพื่อควบคุมการเกิดกรณีใดๆที่ไม่พึงประสงค์อันส่งผลให้ระบบไม่สามารถทำงานจนบรรลุผลได้ แบบรูปจัดการสิ่งผิดปกติจะทำหน้าที่จัดการเงื่อนไขดังกล่าวให้ผลการทำงานมีความถูกต้องแม่นยำยิ่งขึ้น มีความยืดหยุ่นในกระบวนการทำงาน จุดประสงค์การออกแบบเพื่อแก้ปัญหากรณีเกิดการทำงานไม่พึงประสงค์ โดยควบคุมการทำงานตามแบบรูปเมื่อเกิดเงื่อนไขดังกล่าว โดยมีแบ่งประเภทการทำงานเป็น 3 ประเภทใหญ่ๆ [17, 18] ดังนี้

1) การนำเสนอ หรือการทำงานด้วยทางเลือกอื่นๆ (Presenter: Trying Other Alternatives)

การจัดการเงื่อนไขเมื่อเกิดปัญหาให้สามารถทำงานต่อไปได้โดยการใช้ทางเลือกอื่นๆในการทำงาน ดังนี้

(1) ทางเลือกแบบมีลำดับ (Ordered Alternatives)

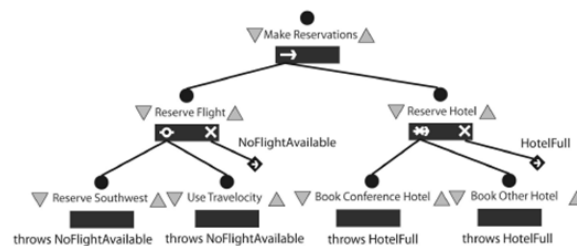
มีการจัดการเงื่อนไขโดยการใช้ทางเลือกอื่นๆ แทน โดยจะมีรูปแบบการทำงานในแต่ละทางเลือกจะมีลักษณะเป็นไปในรูปแบบเดียวกันกับการทำงานปกติในทุกๆทางเลือก ตัวอย่าง ในการท่องเที่ยวจังหวัดใด ต้องมีการจองโรงแรมเมื่อโรงแรมเต็ม ต้องเปลี่ยนไปจองโรงแรมอื่นๆต่อไป โดยมีโครงสร้างตามภาพที่ 2.11



ภาพที่ 2.11 แผนภาพ การทำงานแบบทางเลือกมีลำดับ

## (2) ทางเลือกแบบไม่มีลำดับ (Unordered Alternatives)

การจัดการเงื่อนไขโดยใช้ทางเลือกอื่นๆ แทน โดยจะมีรูปแบบการทำงานในแค่ทางเลือก จะมีลักษณะเป็นรูปแบบที่แตกต่างจากการทำงานปกติในแต่ละทางเลือก ตัวอย่าง ในการท่องเที่ยวต่างจังหวัด เดิมมีแผนในการเดินทางโดยเครื่องบิน แต่เนื่องจากสนามบินจังหวัดปลายทางปิดปรับปรุง จึงหาการเดินทางรูปแบบอื่นเช่น รถทัวร์ รถไฟ ทางเรือ หรือขับรถไปเอง เป็นต้น โดยที่ขั้นตอนในการจองเส้นทางแต่ละแบบจะมีลักษณะที่ต่างกันได้ ตามภาพที่ 2.12



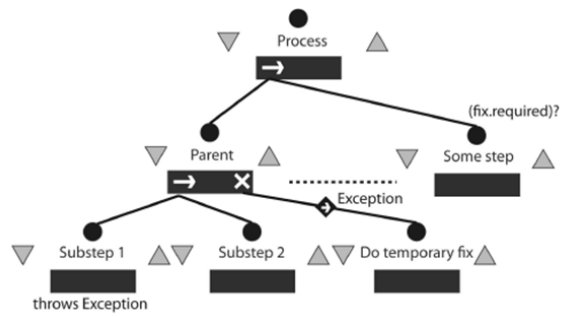
ภาพที่ 2.12 แผนภาพ การทำงานแบบทางเลือกไม่มีลำดับ

## 2) การแทรกหรือการแทรกขั้นตอน (Insert: Inserting Behavior)

การจัดการเงื่อนไขโดยการเพิ่มขั้นตอนเพื่อให้การทำงานสามารถทำงานต่อไปได้ตามปกติ มีลักษณะการใช้งาน 4 รูปแบบ มีรายละเอียดดังนี้

### (1) การแก้ไขแบบทันทีทันใด (Immediate Fixing)

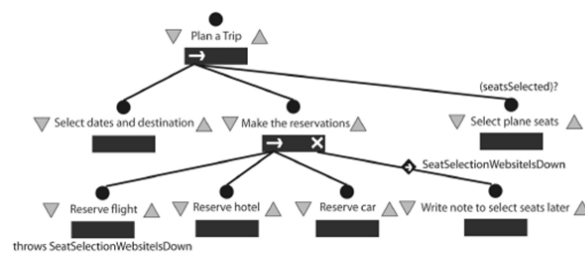
การเพิ่มการทำงานโดยตรวจสอบงานและเพิ่มขั้นตอนแก้ไขกระบวนการทำงานนั้นๆ โดยผลการทำงานที่เพิ่มจะนิยามว่าจะทำให้สามารถทำงานต่อตามเดิมได้ตามภาพที่ 2.13



ภาพที่ 2.13 แผนภาพการทำงานของการทำงานของการแก้ไขแบบทันที

### (2) การแก้ไขแบบเชื่อมโยง (Deferred Fixing)

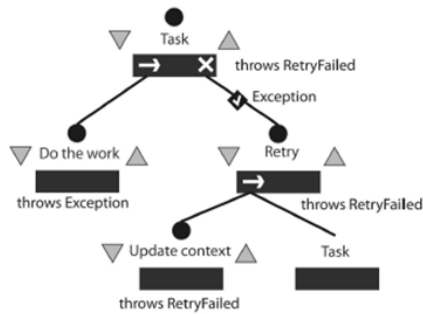
การเพิ่มการทำงานโดยตรวจสอบงานและเพิ่มพฤติกรรมแก้ไขกระบวนการทำงานนั้นๆ เช่นเดียวกับการแก้ไขแบบทันทีทันใด แต่มีเพิ่มการจดบันทึกรายการการแก้ไข เพื่อนำรายการดังกล่าวไปประยุกต์ใช้งานต่อไปได้ตามภาพที่ 2.14



ภาพที่ 2.14 แผนภาพการทำงานของการทำงานของการแก้ไขแบบเชื่อมโยง

### (3) ทดลองใหม่ (Retry)

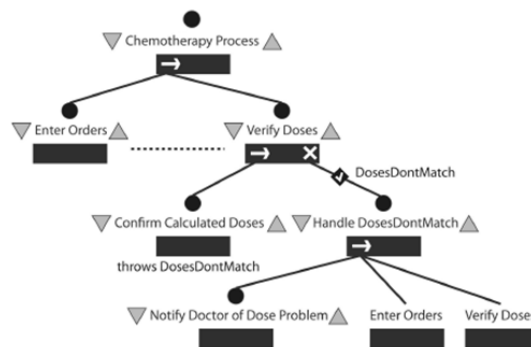
การเพิ่มการทำงานโดยตรวจสอบงานและเพิ่มพฤติกรรมแก้ไขกระบวนการทำงานนั้นๆ เมื่อแก้ไขแล้วจะชดเชยไปเริ่มการทำงานในขั้นตอนแม่ของการทำงานนั้นหรือการทำงานที่เป็นเงื่อนไข เพื่อเป็นการเริ่มต้นการทำงานในส่วนองค์ประกอบนี้ใหม่อีกครั้งตามภาพที่ 2.15



ภาพที่ 2.15 แผนภาพแสดงการทำงานแบบทดลองใหม่

#### (4) เงื่อนไขในการทำงานใหม่ (Exception-Driven Rework)

การเพิ่มการทำงานโดยตรวจสอบงานและเพิ่มพฤติกรรมแก้ไขกระบวนการทำงานนั้นๆ และมีการตรวจสอบ ความครบถ้วนของกระบวนการ เพื่อรองรับการแก้ไขที่อาจจะมีได้มีเพียงกระบวนการเพียงกระบวนการเดียวในการทำให้การทำงานนั้นสมบูรณ์หรือดำเนินการตามปกติ แต่อาจจะมีหลายกระบวนการทำงาน ซึ่งการจะทำงานต่อไปได้ตามปกติได้เมื่อทำงานหรือทำตามเงื่อนไขนั้นครบแล้วเท่านั้นตามภาพที่ 2.16



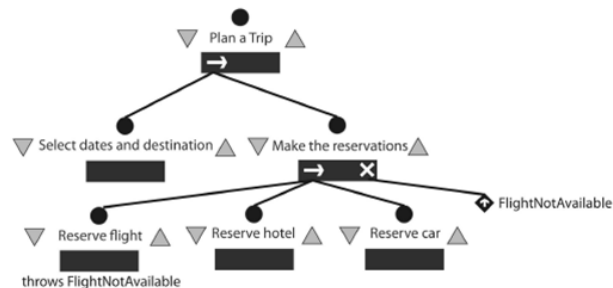
ภาพที่ 2.16 แผนภาพการทำงานแบบเงื่อนไขการทำงานใหม่

#### 3) การล้มเหลว หรือการยกเลิกขั้นตอน (Aborting: Cancelling Behavior)

การจัดการเงื่อนไขเพื่อยกเลิกการทำงานโดยที่ไม่ส่งผลกระทบต่อระบบ ประกอบด้วย 2 รูปแบบ มีรายละเอียดดังนี้

##### (1) การปฏิเสธการทำงาน (Reject)

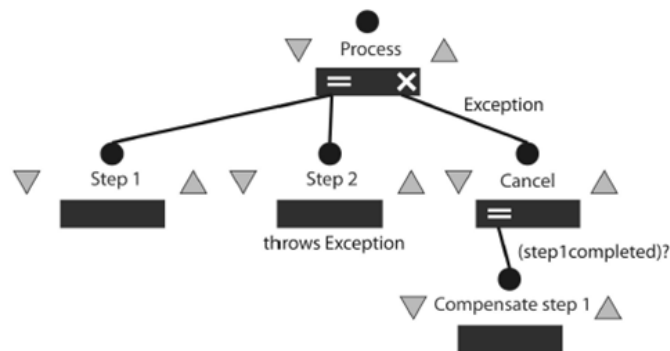
การเพิ่มขั้นตอนสำหรับตรวจสอบการทำงานของขั้นตอนใดๆ เพื่อละเว้นหรือยกเลิกการทำงานนั้น เมื่อเงื่อนไขการทำงานขั้นตอนนี้ล้มเหลวหรือไม่สามารถทำงานได้ตามภาพที่ 2.17



ภาพที่ 2.17 แผนภาพการทำงานของแบบปฏิบัติการทำงาน

## (2) การชดเชยการกระทำ (Compensate)

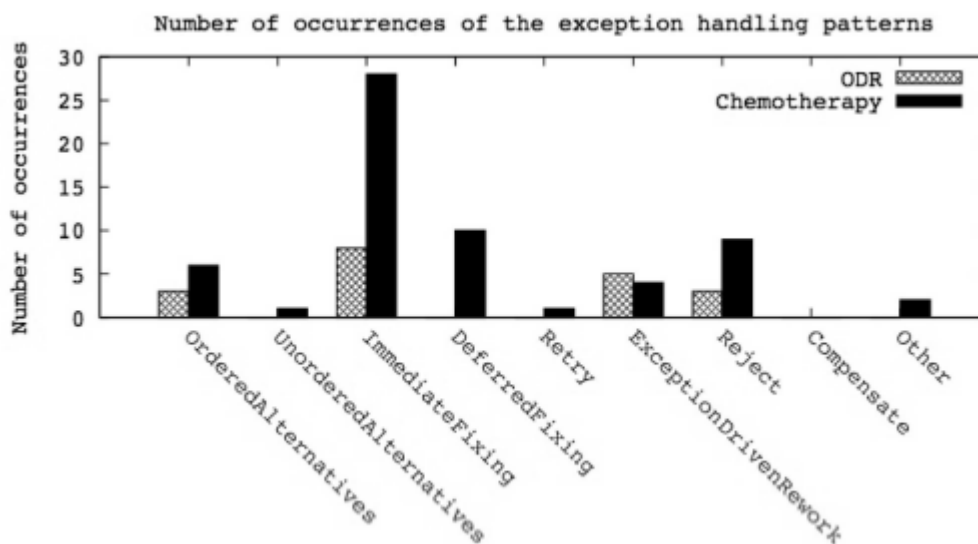
การเพิ่มขั้นตอนสำหรับการตรวจสอบการทำงานของขั้นตอนใดๆ เพื่อย้อนรอยการทำงานกลับไปการกระทำก่อนหน้าเมื่อการขั้นตอนปัจจุบันไม่สามารถทำงานได้ตามภาพที่ 2.18



ภาพที่ 2.18 แผนภาพการทำงานของแบบชดเชยการกระทำ

งานวิจัยอ้างอิงยังได้นำแบบรูปจัดการสิ่งผิดปกติไปประเมินในกระบวนการทางธุรกิจและกระบวนการทำงานเกี่ยวกับซอฟต์แวร์เพื่อประเมินถึงความเหมาะสมในการนำไปใช้งานแก้ปัญหาได้ ได้แก่ การนำไปประเมินในกระบวนการทำเคมีบำบัด (Chemotherapy) ซึ่งเป็นกระบวนการรักษาผู้ป่วยโดยเฉพาะผู้ป่วยโรคมะเร็งโดยใช้เคมี และ ระบบไกล่เกลี่ยความขัดแย้งบนเครือข่าย (Online Dispute Resolution) เป็นระบบที่มีกลไกในการระงับข้อพิพาทโดยใช้ระบบสารสนเทศเข้า

มาช่วย ในกระบวนการเจรจา ติดต่อ สื่อสาร หรือโต้แย้งใช้ในงานเจรจาตัดสินข้อพิพาทคดีความต่างๆ ได้ผลดังภาพที่ 2.19



ภาพที่ 2.19 กราฟแสดง การใช้แบบรูปจัดการสิ่งผิดปกติในกระบวนการทำเคมีบำบัดและระบบไกล่เกลี่ยข้อพิพาทบนเครือข่าย

ผลที่ได้เห็นได้ว่าอ้างอิงจากการนำแบบรูปจัดการสิ่งผิดปกติไปประเมินในกระบวนการทำงานปกติของทั้ง 2 ระบบในแต่ละระบบมีการใช้งานแบบรูปจัดการสิ่งผิดปกติที่แตกต่างกัน กระบวนการทำเคมีบำบัดมีความเสียหายจากผลกระทบค่อนข้างสูงจึงมีการใช้แบบรูปจัดการสิ่งผิดปกติถึง 59 กระบวนการ ในขณะที่ระบบไกล่เกลี่ยข้อพิพาทบนเครือข่ายใช้แบบรูปดังกล่าวเพียง 19 กระบวนการเท่านั้น เห็นได้ว่าแนวทางการแก้ปัญหาในกระบวนการทำงานปกติมีความสอดคล้องตามแบบรูปจัดการสิ่งผิดปกติสอดแทรกอยู่ แต่จากผลการศึกษาเป็นการออกแบบเดิมก่อนที่จะนำแบบรูปไปร่วมวิเคราะห์นั้นขาดความชัดเจน เป็นแนวทางควบคุมการแก้ปัญหาที่ผ่านการออกแบบจากปัญหาที่เกิดขึ้นมากกว่า ทำให้แบบรูปจัดการสิ่งผิดปกติเป็นแบบแผนและแนวทางที่สำคัญในการนำเสนอให้มีความชัดเจนยิ่งขึ้น

#### 2.2.4 การออกแบบส่วนต่อประสานของเกมไดเจสซิส

โลกของเกม(game world) องค์ประกอบที่อยู่ในเกมจริงๆ เช่น อาคารบ้านเรือนในเกม ร่างกายตัวละคร เครื่องประดับหรืออุปกรณ์ที่อยู่ในร่างกายตัวละคร



อ้างอิงตามหลักการออกแบบส่วนต่อประสานเกมหลักการไคเจสซิส (Digesis) [19,20] ซึ่งเป็นหลักการการออกแบบประสานโดยประยุกต์ใช้มาจากการออกแบบภาพยนตร์หรือสื่อบันเทิง การออกแบบส่วนต่อประสานโดยหลักการไคเจสซิสจะความแตกต่างจากส่วนต่อประสานประเภทอื่นๆ เนื่องจากส่วนต่อประสานของเกมจะมีองค์ประกอบของความบันเทิงคดี (Fiction) มาเกี่ยวข้อง ความบันเทิงคดีเป็นองค์ประกอบเพื่อสร้างความเข้าถึงและอารมณ์ร่วมแก่ผู้เล่นหรือผู้ใช้ นอกเหนือจากการออกแบบส่วนต่อประสานปกติซึ่งเน้นการใช้งาน (Usability) ลักษณะของความบันเทิงคดีที่อยู่ในส่วนต่อประสานของเกมคือการสอดแทรกสิ่งหนึ่งสิ่งใดเข้าไปในส่วนต่างๆของเกมเพื่อให้ผู้ใช้เข้าถึงบรรยากาศ เนื้อเรื่อง สภาพแวดล้อม สิ่ง que ผู้เล่นสามารถมีปฏิสัมพันธ์ได้และไม่ได้ หรือส่วนอื่นๆที่ถูกปรับให้เหมาะสมกับเนื้อเรื่องและแนวเกม โดยแบ่งเป็น 2 กลุ่ม

#### 1. การบรรยาย (Narrative)

การบรรยายคือการนำเสนอถึงสถานการณ์ใดๆ โดยอาจจะผ่านข้อความ รูปภาพ ภาพยนตร์ จำลองของเกม ซึ่งการนำเสนอลักษณะนี้เป็นการนำเสนอที่เล่นไม่สามารถมีปฏิสัมพันธ์ด้วยได้ ตัวอย่างเช่น คู่มือ ส่วนช่วยเหลือในเกม (Help) ภาพยนตร์ในเกม

#### 2. การออกแบบผนังสี่ด้าน (The fourth wall)

เป็นส่วนต่อประสานประเภทบรรยายด้วยภาพหรือตัวอักษรที่ทำหน้าที่นำเสนอข้อมูลหรือทำหน้าที่ปฏิสัมพันธ์กับผู้ใช้ โดยมีการแยกประเภทด้วยโดยแบ่งเป็นตัวละครเกม และโลกของเกม มี 4 รูปแบบย่อยดังนี้

##### 2.1 เดียเจติก (Diegetic)

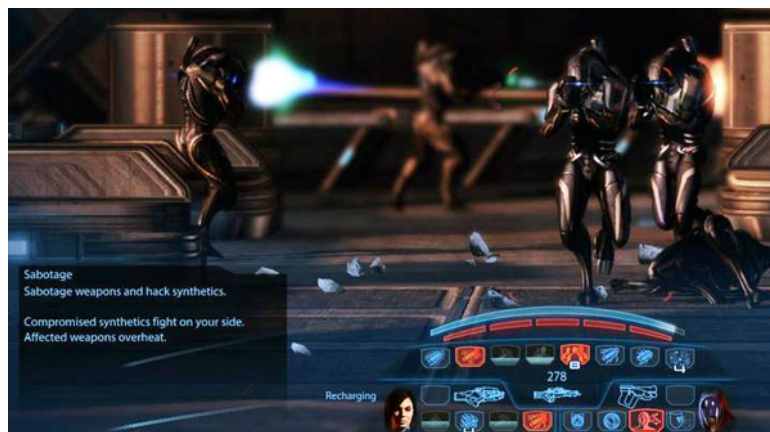
ส่วนต่อประสานซึ่งบรรจุอยู่ในโลกของเกม การปฏิสัมพันธ์กับส่วนต่อประสานประเภทนี้ต้องการทำผ่านตัวละครในเกมซึ่งผู้เล่นบังคับที่ต่อหนึ่ง โดยทั่วไปส่วนต่อประสานประเภทนี้จะสอดแทรกอยู่ในวัตถุของเกมซึ่งส่วนต่อประสานลักษณะนี้จะสร้างความสมจริงในการเล่นเพื่อผู้เล่นเข้าถึงอารมณ์ร่วมกับโลกของเกม เช่น ตัวละครเปิดเพื่อดูแผนที่ในเกมจะต้องเปิดแผนที่โดยการให้ตัวละครหยิบแผนที่ออกมาเปิดดู ดังภาพที่ 2.20



ภาพที่ 2.20 ตัวละครเปิดแผนที่ในโลกของเกม [20]

## 2.2 นอน-เดียเจติก (Non-Diegetic)

ส่วนต่อประสานที่นำเสนออยู่ในมิติของระบบเกม เป็นส่วนต่อประสานที่วางไว้ในหน้าต่าง แสดงผลของเกมแต่อยู่นอกโลกของเกม เพื่ออำนวยความสะดวกแก่ความเข้าใจ ส่วนต่อประสานลักษณะนี้จะสร้างความเข้าใจให้แก่ผู้เล่นทำให้ผู้เล่นเข้าถึงเกมได้ง่ายขึ้น แต่จะลดความสมจริงลง เช่น เกมยิงมุมมองบุคคลที่หนึ่ง (First Person Shooting) จะมีหน้าต่างสำหรับแจ้งจำนวนกระสุนที่เหลืออยู่ในปืนซึ่งปกติผู้เล่นต้องจดจำด้วยตัวเองเรียนรู้ด้วยเอง การยิงปืนแต่ละนัดเหลือกระสุนเท่าไรไม่ได้บอกในลักษณะของโลกความเป็นจริง ดังภาพที่ ภาพที่ 2.21



ภาพที่ 2.21 ส่วนต่อประสานแสดงจำนวนกระสุนปืนในเกม [19]

## 2.3 สเปเชียล (Spatial)

ส่วนต่อประสานที่นิยมใช้ในเกมส์สามมิติ ส่วนต่อประสานประเภทนี้จะปรากฏในโลกของเกม แต่ไม่ใช่องค์ประกอบที่ควรอยู่ในโลกเกม ถูกสร้างมาเพื่อเน้นให้ผู้ใช้ทราบถึงการมีอยู่ของสิ่งใดสิ่งหนึ่ง

ที่สามารถมีปฏิสัมพันธ์ด้วยได้ โดยจะเป็นการสร้างสีหรือปรับความสว่างเน้นในส่วนโลกของเกมเพื่อให้ผู้ใช้แยกแยะได้ถึงสิ่งที่ระบบต้องการนำเสนอแก่ผู้เล่น ตัวอย่างเช่น เกมมุมมองบุคคลที่หนึ่งจะมีการเน้นสีเหนือของที่ผู้ใช้สามารถมีปฏิสัมพันธ์ได้ด้วย หรือ เกมวางแผนการรบจะมีกราฟิกบอกว่าเราเลือกตัวละครตัวใดอยู่จะมีวงกลมเน้นสีอยู่รอบๆตัวละครที่เราเลือก ดังภาพที่ 2.22



ภาพที่ 2.22 ส่วนต่อประสานแสดงตัวละครที่ผู้ใช้ควบคุมได้ [20]

#### 2.4 เมตา(Meta)

ส่วนต่อประสานแสดงถึงสิ่งที่เกิดขึ้นในโลกของเกมแต่ไม่ได้อยู่ในโลกของเกมและไม่ได้อยู่ในส่วนต่อประสานในระบบเกม ไม่ได้ถูกสร้างมาเพื่อบอกหรือเน้นตำแหน่งใดๆ เฉพาะเจาะจงดังแบบสเปเชียล แต่แสดงผลออกมา ณ หน้าจอแสดงผลทั้งจอ โดยผลที่ออกมานี้ไม่ได้อยู่ในโลกของเกม ตัวอย่างเช่น หน้าจอมีขอบสีแดงเมื่อตัวละครโดนยิง ดังภาพที่ 2.23 ซึ่งการแสดงผลลักษณะนี้มักแสดงผลในเกมที่มีเป็นสามมิติ

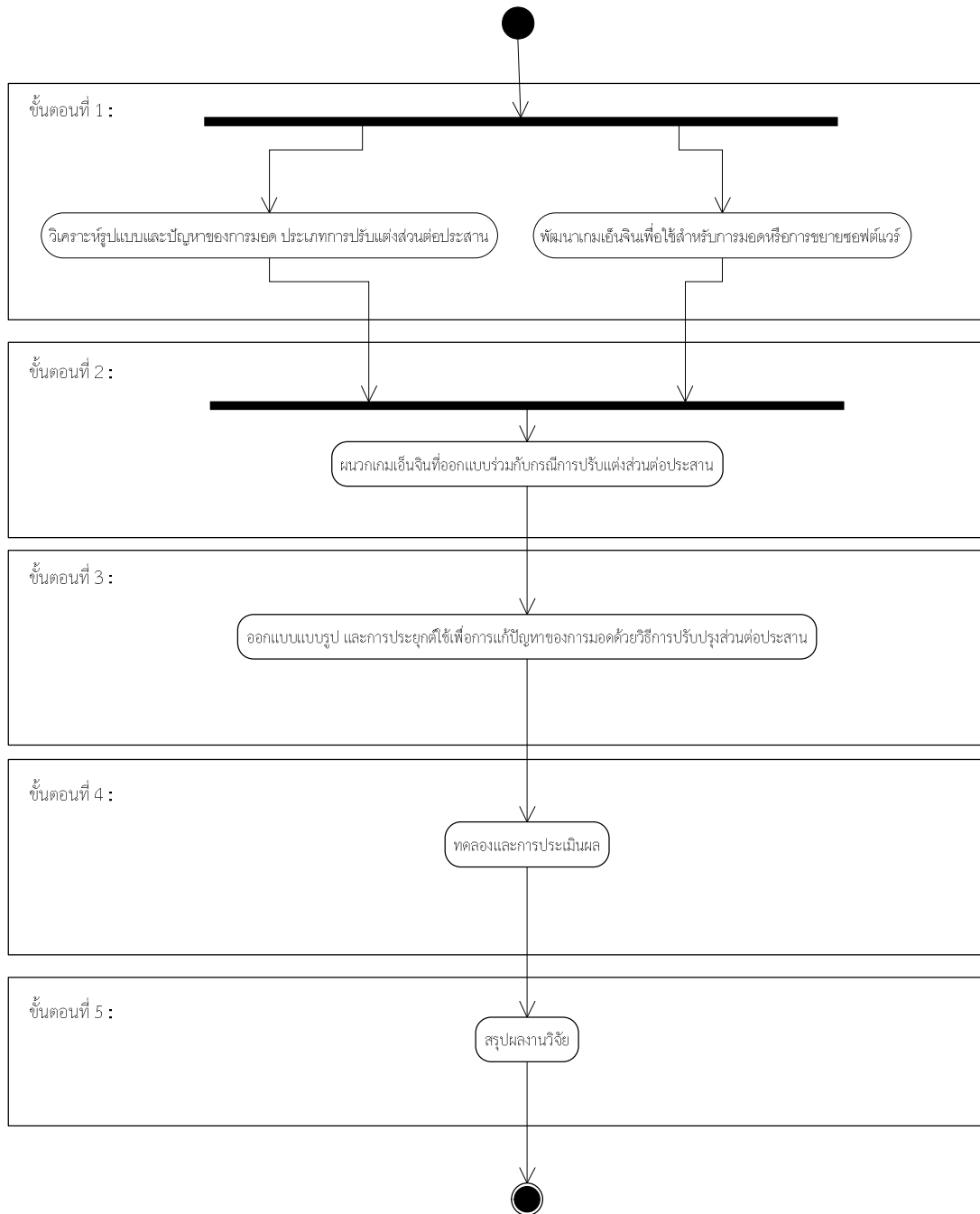


ภาพที่ 2.23 ส่วนต่อประสานแสดงภาพหน้าจอเมื่อโดนโจมตี [19]

### บทที่ 3

#### การศึกษา วิเคราะห์ปัญหาของการปรับแต่งและหาแนวทางแก้ไข

งานวิทยานิพนธ์นี้แบ่งขั้นตอนการดำเนินงานออกเป็น 5 ขั้นตอน โดยแสดงผังแผนภาพกิจกรรมดังภาพที่ 3.1



ภาพที่ 3.1 ผังแผนภาพกิจกรรมขั้นตอนการดำเนินงานวิจัย

ในบทนี้กล่าวถึงวิธีการจัดการข้อผิดพลาดที่เกิดจากการทำมอด นำเสนอผ่านการออกแบบ เกมเอ็นจิน และการนำแบบรูปมาประยุกต์ใช้ ตามหลักวิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลาย(End-User Software Engineer) เพื่อสนับสนุนการทำมอดเกม โดยการวิเคราะห์จากปัญหา และสาเหตุของ ข้อผิดพลาดจากการทำมอดครอบคลุมใน 2 กรณีได้แก่ ความผิดพลาดของผู้ใช้ชั้นปลาย (End-User) ในการทำมอด และ การออกแบบระบบที่ไม่รองรับการทำมอด จากปัญหาข้างต้นนำมาวิเคราะห์หา แนวทางแก้ปัญหาเพื่อควบคุมปัญหาการทำมอดที่นำเสนอแก่ผู้ใช้

### 3.1 วิเคราะห์รูปแบบและปัญหาของการมอด ประเภทการปรับแต่งส่วนต่อประสาน

จากงานวิจัยอ้างอิง นำไปสู่การค้นคว้า และนำผลมาวิเคราะห์ โดยมีจุดประสงค์เพื่อศึกษาวิธีการ มอดด้วยการปรับแต่งส่วนต่อประสาน และข้อผิดพลาดที่เกิดขึ้นจากการปรับแต่งส่วนต่อประสาน โดยมีหลักการดังนี้

#### 3.1.1 ศึกษาและวิเคราะห์กรณีในการปรับปรุงและจำแนกกรณีตามประเภทมอด

ศึกษาลักษณะการทำมอดที่เกิดจากผู้ใช้ โดยการหารูปแบบการมอด ด้วยวิธีปรับแต่งส่วนต่อ ประสาน ผลลัพธ์ที่ได้คือกรณีการปรับปรุงส่วนต่อประสาน จากจุดประสงค์ในการปรับแต่ง ได้ตั้ง ตารางที่ 3.1 และมีรายละเอียดแต่ละการปรับแต่งดังตารางที่ 3.2

ตารางที่ 3.1 รูปแบบ จุดประสงค์ และตัวอย่างการปรับแต่งส่วนต่อประสาน

| รูปแบบการปรับแต่ง ส่วนต่อประสาน              | จุดประสงค์ในการปรับแต่ง | ตัวอย่างการปรับแต่ง   |
|--|-------------------------|---|
| 1.สร้างส่วนต่อประสาน เป็นเอกลักษณ์ของ ตัวเอง | ความพึงพอใจของผู้ใช้    | การปรับแต่งเพื่อความสวยงาม เช่น เปลี่ยนรูปภาพ พื้นหลัง ขนาดภาพ เคอร์เซอร์ |
|  | การใช้งาน               | แก้การแสดงผลเพื่อให้รู้ว่าใช้งาน เพิ่ม เหตุการณ์ทำให้เกิดเสียง            |
| 2.การปรับแต่งการ แสดงผล                      | ฟังก์ชันงาน             | แก้การทำงานของการแสดงผล ใหม่ เปลี่ยนฟังก์ชันในการทำงาน                    |
|  | ประสิทธิภาพ             | การลดกิจกรรมของผู้ใช้ เช่น การรวมปุ่มให้มีจำนวนการกด น้อยลง               |

ตารางที่ 3.1 รูปแบบ จุดประสงค์ และตัวอย่างการปรับแต่งส่วนต่อประสาน (ต่อ)

| รูปแบบการปรับแต่งส่วนต่อประสาน                  | จุดประสงค์ในการปรับแต่ง | ตัวอย่างการปรับแต่ง  |
|---|-------------------------|--|
| 3.เพิ่มส่วนต่อประสานอันเนื่องมาจากการเพิ่มโมดูล | การปรับตัว              | เพิ่มส่วนต่อประสาน<br>เพิ่มการแสดงผลใหม่ๆ                              |
|   | การบำรุงรักษา           | เมื่อมีการเพิ่มฟังก์ชันมีผลต่อส่วนต่อประสาน ปรับแต่งให้ทำงานตามเหมาะสม |

ตารางที่ 3.2 รายละเอียดการปรับแต่ง

| จุดประสงค์ในการปรับแต่ง | คำอธิบาย  |
|-------------------------|---|
| ความพึงพอใจของผู้ใช้    | การปรับปรุงส่วนต่อประสานเพื่อให้มีลักษณะต่างไปจากส่วนต่อประสานเดิมเพื่อความพึงพอใจส่วนตัวของผู้ใช้งาน เพื่อมีลักษณะเด่น แตกต่าง หรือดึงดูดใจผู้ใช้งานอื่นๆ  |
| การใช้งาน               | การปรับปรุงส่วนต่อประสานเพื่อให้มีลักษณะเด่นในการนำเสนอบางสิ่งบางอย่าง เพื่อช่วยเหลือแก่ผู้ใช้เฉพาะกลุ่ม เพื่อให้เกิดความสะดวกคล่องตัว ในการใช้งานของผู้ใช้กลุ่มนั้น อนึ่งการปรับแต่งกรณีดังกล่าวอาจจะส่งผลกระทบต่อการใช้งานของผู้ใช้กลุ่มอื่นๆที่ได้   |
| ฟังก์ชันงาน             | การปรับปรุงส่วนต่อประสานในการเพิ่มส่วนต่อประสานใหม่ที่มีฟังก์ชันการทำงานอยู่ภายใน เพื่อมุ่งเน้นในการนำเสนอข้อมูลหรือปฏิสัมพันธ์กับผู้ใช้เป็นหลัก<br>เพื่อให้ผู้ใช้ทราบถึงการกระทำ กิจกรรม ระยะเวลา หรือสร้างความเข้าใจถึงการทำงานในขณะนั้นแก่ผู้ใช้   |
| ประสิทธิภาพ             | การปรับปรุงส่วนต่อประสานเพื่อลดปฏิสัมพันธ์ของผู้ใช้กับระบบ เพื่อให้เกิดการทำงานที่มีประสิทธิภาพ ผู้ใช้สามารถทำงานได้รวดเร็วยิ่งขึ้น การปรับปรุงลักษณะนี้อาจจะเหมาะกับผู้ใช้งานในที่มีความเชี่ยวชาญ เนื่องจากการนำเสนอข้อมูลระบบจะลดลงไป ผู้ใช้ทำให้พลาดข้อมูลบางอย่าง อาจจะเป็นผลแก่ผู้ใช้ระดับเริ่มต้นที่ขาดความเข้าใจ |

ตารางที่ 3.2 รายละเอียดการปรับแต่ง (ต่อ)

| จุดประสงค์ในการปรับแต่ง | คำอธิบาย  |
|-------------------------|---|
| การปรับตัว              | การปรับแต่งส่วนต่อประสานเพื่อสนับสนุนการปรับแต่งประเภทอื่นๆที่มี<br>การเพิ่มฟังก์ชันงานใหม่แก่ระบบ<br>ซึ่งการปรับแต่งประเภทนี้จะทำงานได้เมื่อมีการสร้างส่วนต่อประสาน<br>เพื่อให้ผู้ใช้สามารถเรียกใช้ฟังก์ชันงานประเภทเหล่านั้นได้ |
| การบำรุงรักษา           | บำรุงรักษา เป็นการปรับแต่งส่วนต่อประสานเนื่องจากการแก้ไข<br>ซอฟต์แวร์ในส่วนอื่นทำให้เกิดผลกระทบต่อส่วนต่อประสาน   |

จากกรณีการปรับแต่งข้างต้นนำไปสู่การสร้างกรอบสำหรับการแก้ไขข้อผิดพลาด พบว่าในแต่ละกรณีการปรับแต่ง สามารถเกิดข้อผิดพลาดได้ในหลากหลายรูปแบบ และมีวิธีแก้ไขข้อผิดพลาดที่หลากหลายเช่นกัน ทั้งแบบที่เหมือนและแตกต่างกันตามแต่กรณีในการนำแบบรูปจัดการสิ่งผิดปรกติมาใช้แก้ปัญหาข้อผิดพลาดนั้น พบว่ามีบางกรณีที่ขึ้นต่อกันจึงสามารถใช้วิธีแก้ไขเดียวกันได้ ดังตัวอย่างกรณีการปรับแต่งเพื่อความพึงพอใจของผู้ใช้ ประกอบด้วยตัวอย่างกรณีการเกิดข้อผิดพลาด การแก้ไขข้อผิดพลาดที่เกิดขึ้นและตัวอย่างใช้แบบรูปในการแก้ไขข้อผิดพลาดในแต่ละรายการการปรับแต่ง ดังตารางที่ 3.3 และ 3.4 ตามลำดับ

ตารางที่ 3.3 ตัวอย่างการทำงาน การมอดปรับแต่งส่วนต่อประสานเพื่อความพึงพอใจของผู้ใช้

| รายการปรับแต่ง   | กรณีผิดพลาด    | ตัวอย่าง                | ผลของข้อผิดพลาด | วิธีแก้ไข                        |
|------------------|----------------|-------------------------|-----------------|----------------------------------|
| แก้ไขภาพพื้นหลัง | นามสกุลไฟล์ผิด | ระบบใช้ jpeg มอดใช้ gif | แสดงผลผิดพลาด   | ตรวจสอบนามสกุลไฟล์อื่นๆที่รองรับ |
|                  |                |                         |                 | ใช้ค่าโดยปริยายแทน               |
|                  |                |                         |                 | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |

ตารางที่ 3.3 ตัวอย่างการทำงาน การมอดปรับแต่งส่วนต่อประสานเพื่อความพึงพอใจของผู้ใช้ (ต่อ)

| รายการปรับแต่ง    | กรณีผิดพลาด                    | ตัวอย่าง                                 | ผลของข้อผิดพลาด         | วิธีแก้ไข                        |                                  |
|-------------------|--------------------------------|--|-------------------------|----------------------------------|----------------------------------|
| แก้ไขขนาดภาพ      | ขนาดภาพไม่เท่ากับพื้นที่หน้าจอ | ขนาดภาพใหญ่เกินขนาดหน้าจอ                | แสดงผลผิดพลาด           | ตรวจสอบความใกล้เคียงของขนาด      |                                  |
|                   |                                |  |                         | ใช้ค่าโดยปริยายแทน               |                                  |
|                   |                                |  |                         | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |                                  |
|                   |                                | ขนาดภาพเล็กเกินขนาดหน้าจอ                | แสดงผลผิดพลาด           | ตรวจสอบขนาดของขนาด               |                                  |
|                   |                                |  |                         | ใช้ค่าโดยปริยายแทน               |                                  |
|                   |                                |  |                         | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |                                  |
| เปลี่ยนเคอร์เซอร์ | ขนาดภาพไม่เท่ากับพื้นที่       | ขนาดภาพใหญ่เกินขนาดหน้าจอ                | แสดงผลผิดพลาด           | ตรวจสอบความใกล้เคียงของขนาด      |                                  |
| เปลี่ยนเคอร์เซอร์ | รูปแบบไม่สามารถใช้งานได้       | ขนาดภาพเล็กเกินขนาดหน้าจอ                | แสดงผลผิดพลาด           | ใช้ค่าโดยปริยายแทน               |                                  |
|                   |                                |  |                         | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |                                  |
|                   |                                |  |                         | ตรวจสอบขนาดของขนาด               |                                  |
|                   |                                | ขนาดภาพเล็กเกินขนาดหน้าจอเคอร์เซอร์หายไป | แสดงผลผิดพลาดในการทำงาน | ใช้ค่าโดยปริยายแทน               |                                  |
|                   |                                |  |                         | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |                                  |
|                   |                                |  |                         | ใช้ค่าโดยปริยายแทน               |                                  |
|                   | รูปแบบไม่สามารถใช้งานได้       | เคอร์เซอร์หายไป                          | ผิดพลาดในการทำงาน       | ผิดพลาดในการทำงาน                | แสดงหน้าต่างแจ้งเตือนการทำงานผิด |
|                   |                                |  |                         |                                  | ใช้ค่าโดยปริยายแทน               |



### ตารางที่ 3.4 ตัวอย่างการกรณีแก้ไขข้อผิดพลาดด้วยแบบรูป

| วิธีแก้ไข                        | แบบรูปที่ใช้               |
|----------------------------------|----------------------------|
| การตรวจสอบ                       | การแก้ไขแบบทันที           |
| ใช้ค่าโดยปริยาย                  | ทางเลือกมีลำดับ/ไม่มีลำดับ |
| แสดงหน้าต่างแจ้งเตือนการทำงานผิด | การแก้ไขแบบเชื่อมโยง       |

#### 3.1.2 จำแนกประเภทของความล้มเหลวที่เกิดขึ้น

จากข้อผิดพลาดที่เกิดขึ้นในแต่ละประเภทสามารถจำแนกประเภทความล้มเหลวของแต่ละข้อผิดพลาด หรือรูปแบบการทำงานที่ไม่พึงประสงค์ จากการสำรวจในแต่ละประเภทส่งผลให้เกิดข้อผิดพลาดหรือสาเหตุ แตกต่างกัน ซึ่งแยกแยะถึงความรุนแรงของข้อผิดพลาดที่อาจเกิดขึ้นได้ในแต่ละการปรับแต่ง โดยจำแนกได้ดังต่อไปนี้

- 1) ข้อผิดพลาดในการนำเสนอ มีการนำเสนอที่ผิดพลาดแต่ระบบยังสามารถทำงานได้ปกติ
- 2) ข้อผิดพลาดในฟังก์ชันงาน การทำงานที่ผิดพลาดส่งผลกระทบต่อส่วนต่อประสานหรือการนำเสนอข้อมูลแก่ผู้ใช้
- 3) ข้อผิดพลาดในระบบ ทำให้ระบบทำงานผิดพลาด ผลเสียทำให้ระบบล้ม ช้า หรือไม่สามารถทำงานต่อไปได้

#### 3.1.3 วิเคราะห์ข้อผิดพลาดเพื่อแก้ไข

จากข้อผิดพลาดที่เกิดขึ้น ทางผู้วิจัยได้วิเคราะห์การแก้ไขเป็น 2 แนวทางอ้างอิงตามหลักวิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลายได้แก่ การแสดงข้อผิดพลาดแก่ผู้ใช้เพื่อให้ผู้ใช้ชั้นปลายสามารถเข้าใจถึงวิธีการแก้ไขสามารถพัฒนาออกมาได้สำเร็จ และการคัดกรองข้อผิดพลาดเป็นรูปแบบการแก้ไขเพื่อให้ระบบสามารถทำงานได้ตามปกติ ดังตารางที่ 3.5

ตารางที่ 3.5 ตารางแนวทางแก้ไขปัญหาและแบบรูปจัดการสิ่งผิดปกติที่นำมาประยุกต์ใช้

| รูปแบบการแก้ไข               | การแก้ไขปัญหา            | ประเภทแบบรูปจัดการสิ่งผิดปกติที่นำใช้ |
|------------------------------|--------------------------|---------------------------------------|
| การคัดกรองข้อผิดพลาด         | การนำเสนอการทำงานแบบอื่น | การนำเสนอ                             |
|                              | การเพิ่มกระบวนการ        | การแทรก                               |
| การแสดงผลข้อผิดพลาดแก่ผู้ใช้ | การยกเลิกการทำงาน        | การล้มเหลว                            |

การแสดงผลข้อผิดพลาดแก่ผู้ใช้ เป็นการยกเลิกการทำงานเมื่อผลของการมอดเกมอื่นๆทำให้ระบบไม่สามารถทำงานได้ตามปกติ หรือไม่สามารถแก้ไขได้ โดยแสดงข้อความแสดงข้อผิดพลาดแก่ผู้ใช้เพื่อให้ผู้ใช้แก้ไขต่อไป ก่อนจะจบการทำงาน

การคัดกรองข้อผิดพลาด เป็นกระบวนการแก้ไขเมื่อการมอดเกมอื่นๆ ระบบไม่สามารถทำงานได้ตามปกติ โดยมีแนวทางแก้ไข 2 รูปแบบได้แก่ การนำเสนอการทำงานแบบอื่นที่ใกล้เคียงทดแทน และการเพิ่มกระบวนการทำงานใดๆเข้าไปเพื่อแก้ไขข้อผิดพลาดจนสามารถทำงานได้ตามปกติ

### 3.2 พัฒนาเกมเอ็นจินเพื่อใช้สำหรับการมอดหรือการขยายซอฟต์แวร์

ในขั้นตอนนี้มีจุดประสงค์เพื่อศึกษาการวิเคราะห์และออกแบบเอ็นจินของเอ็นจินใดๆ ให้เหมาะสมแก่การมอดซอฟต์แวร์เพื่อให้ระบบมีประสิทธิภาพรองรับการปรับแต่งจากผู้ใช้ขั้นปลาย โดยผลลัพธ์ที่ได้ในขั้นตอนนี้คือแผนภาพแสดงถึงเอ็นจินเกม ที่แสดงถึงคุณลักษณะเด่นในด้านการปรับตัว และสามารถลดปัญหาในการแสดงผลต่อผู้ใช้งานจริง

#### 3.2.1. ขั้นตอนการกำหนดคุณลักษณะเด่นของเอ็นจิน

กำหนดคุณลักษณะเด่นโดยอิงคุณลักษณะจากเอ็นจินต้นแบบ โดยมุ่งเน้นในการเพิ่มคุณลักษณะด้านการทำส่วนขยาย เรียกว่าการจัดการการใช้ส่วนขยาย เป็นการแยกคุณลักษณะส่วนขยายออกจากคุณลักษณะทั่วไปเพื่อจัดการการเข้าถึง การใช้งาน ตรวจสอบข้อผิดพลาดของการทำงานและการนำเสนอได้ง่าย

#### 3.2.2 กำหนดเป้าหมายและกลยุทธ์ในการออกแบบ

กำหนดเป้าหมายของการพัฒนาเอ็นจินโดยเพิ่มเป้าหมายในการรองรับการขยาย การเพิ่ม (Add-on) ของซอฟต์แวร์ และต้องมีการเรียนรู้การใช้งานที่เข้าใจง่าย โดยมุ่งกลยุทธ์ในการออกแบบไปที่การปรับตัวของซอฟต์แวร์มากขึ้น ดังนี้

#### 1) การใช้งาน (Usability)

การออกแบบต้องมีการสร้างแผนภาพที่เข้าใจง่าย สามารถแยกมิติเพื่อให้ผู้ใช้งานจริงที่มีความรู้แตกต่างกันสามารถมุ่งประเด็นที่สนใจได้ง่าย ทำความเข้าใจได้รวดเร็ว โดยการแยกส่วนขยายของเกมออกจากส่วนอื่นๆ เพื่อให้ผู้ใช้ทำความเข้าใจเพียงส่วนเดียว ในการมอดทุกครั้งเมื่อมีการเรียกใช้การทำงานที่แตกต่างจากค่าปริยาย(Default) ของระบบ ส่วนขยายจะเป็นตัวตรวจสอบการทำงานก่อนทุกครั้ง โดยเป้าหมายคือผู้ใช้ 2 ระดับ ได้แก่ ผู้ใช้ระดับเริ่มต้นสามารถเลือกตัวช่วยในการใช้งานส่วนขยายที่ทีมพัฒนาจัดทำขึ้นเช่น เครื่องมือ เพื่อสามารถทำความเข้าใจได้ง่าย และผู้ใช้ในระดับสูงสามารถพัฒนาในแต่ละส่วนได้ตามต้องการ สุดท้ายแล้วการจัดการข้อผิดพลาดที่เกิดขึ้นจากการทำส่วนขยายจะรวมอยู่ที่จุดเดียว

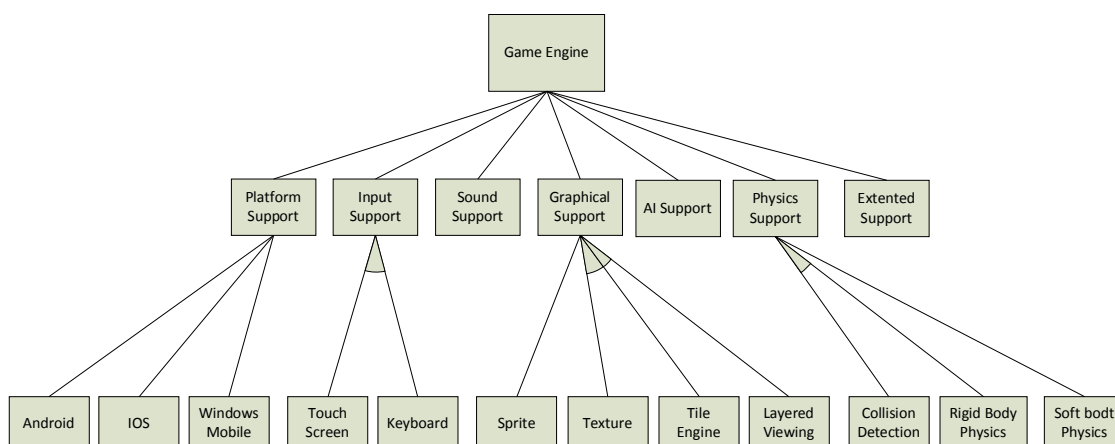
#### 2) การปรับตัว (Adaptability)

การออกแบบที่มุ่งเน้นไปที่การตรวจสอบการแสดงผลของระบบเมื่อมีการใช้งานมอด โดยการกำหนดเงื่อนไขการแสดงผล 2 ประเภทดังนี้

- การแสดงผลปกติ ยินยอมให้เกิดการแสดงผลแก่ผู้ใช้
- การแสดงผลผิดปกติ จะเลือกให้ระบบนำค่าโดยปริยายมาแสดงผลแทน

### 3.2.3 สกัดเป้าหมายจากคุณลักษณะ

การออกแบบเป้าหมายในแต่ละคุณลักษณะที่กำหนดไว้ ด้วยการสร้าง แผนภาพต้นไม้ แสดงเป้าหมายของแต่ละคุณลักษณะทั้งนี้ยังอธิบายถึงเป้าหมายหลักและเป้าหมายรอง ดังภาพที่ 3.2 เส้นเน้นสีแดงถึงระบบนั้นต้องการ โดยการเพิ่ม คุณลักษณะการช่วยเหลือการทำขยายซอฟต์แวร์ (Extended Support) เพื่อเป็นตัวกลางในการช่วยเหลือการปรับแต่ง ในขณะที่ส่วนอื่นๆยังคงลักษณะทั่วไปของคุณลักษณะของเอ็นจินเดิมที่ออกแบบไว้

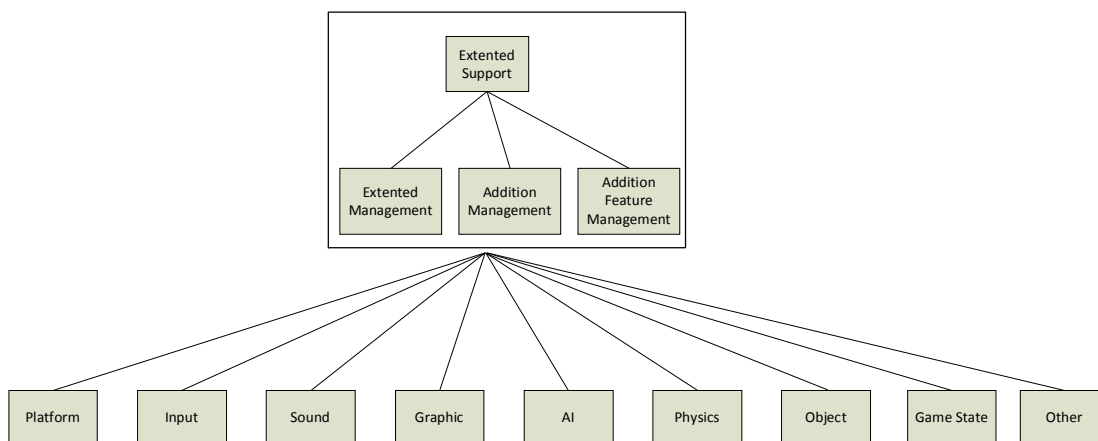


ภาพที่ 3.2 องค์ประกอบคุณลักษณะของเอ็นจิน

เป้าหมายของคุณสมบัติส่วนขยายมุ่งเน้นในการจัดการการทำส่วนขยายซอฟต์แวร์เพื่อตรวจสอบความถูกต้องของมอดและควบคุมหรือจัดการความผิดพลาดจากการทำงานมอด ดังภาพที่ 3.3 โดยมีส่วนประกอบดังนี้

1. ส่วนจัดการส่วนขยาย (Extension Management) ส่วนในการควบคุมการทำส่วนขยายของเกม การปรับปรุงจากส่วนประกอบหรือคุณลักษณะเดิมของเกม
2. ส่วนจัดการส่วนเพิ่ม (Addition Management) ส่วนในการควบคุมการทำส่วนเพิ่ม (Add-on) การเพิ่มคุณสมบัติใหม่ใดๆเข้าไปในซอฟต์แวร์เกม
3. ส่วนจัดการการเพิ่มคุณสมบัติ (Addition Feature Management) ส่วนจัดการการเรียกใช้หรือไม่เรียกใช้คุณสมบัติใดๆในเอ็นจิน เมื่อมีการปรับปรุงคุณลักษณะในระดับเดียวกับคุณลักษณะส่วนขยายจากการแก้ไขซอฟต์แวร์โดยผู้พัฒนาซอฟต์แวร์(patching) คุณลักษณะนี้จะทำหน้าที่การคัดกรอง และแทนที่ คุณลักษณะที่มีความขึ้นต่อกันในคุณลักษณะส่วนขยายให้สอดคล้องกับคุณสมบัติที่เปลี่ยนแปลงไปของคุณสมบัติหลัก

จากการออกแบบข้างต้นมีจุดประสงค์เพื่อให้เอ็นจินมีความสามารถรองรับการปรับแต่งได้ดียิ่งขึ้น มีความเข้าใจที่ง่ายขึ้นจากการแยกรูปแบบการปรับแต่งในลักษณะที่แตกต่างออกจากกันและจัดสรรให้แก่ผู้ใช้สามารถเรียนรู้การใช้งานของระบบจากการศึกษาจากส่วนสนับสนุนส่วนขยาย (Extend Support) เพียงอย่างเดียว โดยไม่ต้องทำความเข้าใจในส่วนสนับสนุนอื่นๆ เอ็นจิน ผู้ใช้ชั้นปลายสามารถมุ่งเน้นในการแก้ไขเฉพาะในส่วนสนับสนุนส่วนขยายเพียงจุดเดียว



ภาพที่ 3.3 องค์ประกอบภายในของคุณลักษณะส่วนขยายของเอ็นจิน

### 3.2.4 การเลือกใช้แบบรูป

เป็นขั้นตอนในการนำเสนอการออกแบบแบบรูป เพื่อจัดการตามเป้าหมายและกลยุทธ์ต่างๆ ที่ออกแบบไว้ในขั้นตอนก่อนหน้ามุ่งเน้นไปที่ การเลือกแบบรูปมาใช้ให้เกิดความสอดคล้องกับคุณลักษณะที่ออกแบบไว้และให้มีความสามารถในการจัดการส่วนขยายในการแสดงผลโดยใช้ แบบรูปของจัดการสิ่งผิดปกติในการจัดการด้านพฤติกรรม โดยมีการเลือกใช้แบบรูปดังนี้

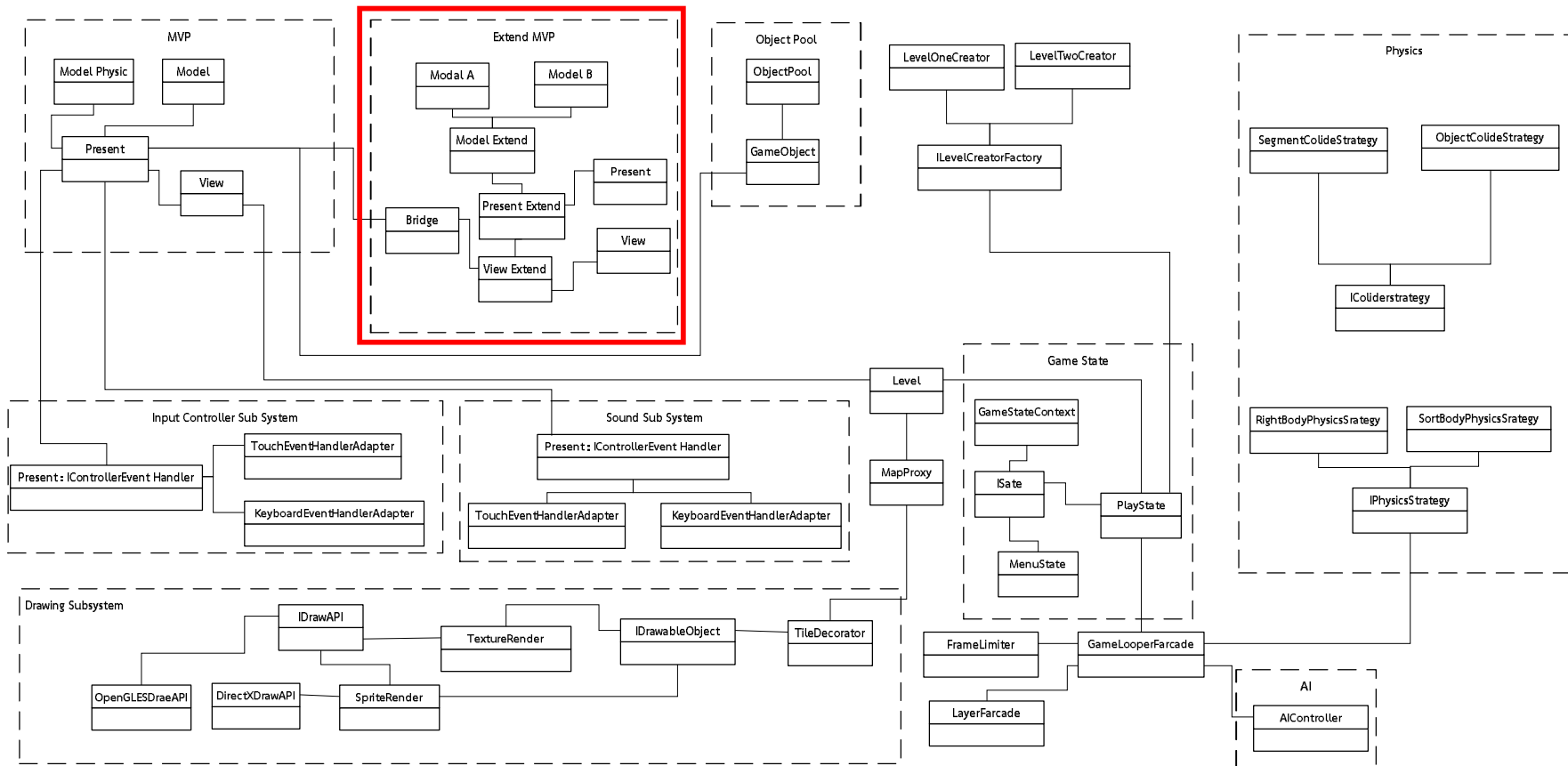
1. แบบรูปการนำเสนออ้างอิงจากเกมเอ็นจินที่นำมาใช้งาน โดยในตัวอย่างใช้แบบรูปเอ็มวีพีในการนำเสนอ จึงเลือกใช้แบบรูปเอ็มวีพีมาใช้อ้างอิงจากการโครงสร้างเดิมของเกมเอ็นจิน ในส่วนโปรแกรมทดสอบจะเลือกใช้แบบรูปการนำเสนอที่แตกต่างกันไปของแต่ละเกมเอ็นจินมาใช้

กรณีตัวอย่าง เกมเอ็นจินใช้แบบรูปเอ็มวีพี(MVP) เป็นแบบรูปในการนำเสนอ จึงเลือกที่จะสร้างแบบรูปเอ็มวีพีโดยอ้างอิงจากแบบรูปการนำเสนอของเกมเอ็นจินปกติ เพื่อให้เกิดความสอดคล้องกับการทำของเอ็มวีพีเดิม ตั้งชื่อว่า เอ็มวีพีส่วนขยาย ทำหน้าที่เสมือนการทำงานของเอ็มวีพีปกติ เน้นการนำเสนอการทำงานของมอดเท่านั้น กรณีที่ไม่มีการใช้งานมอดเอ็มวีพีส่วนขยายจะไม่ถูกเรียกใช้งาน

2. แบบรูปตัวเชื่อม (Bridge Pattern) ทำหน้าที่เป็นตัวเชื่อมการทำงานของแบบรูปเอ็มวีพีปกติกับแบบรูปเอ็มวีพีส่วนขยายเข้าด้วยกัน เพื่อให้การทำงานของเอ็มวีพีปกติสามารถทำงานได้โดยไม่ต้องทราบว่ามีการทำงานของเอ็มวีพีส่วนขยายทำงานซ่อนอยู่หรือไม่ ตัวอย่างเมื่อมีการปรับแต่งแบบรูปตัวเชื่อมจะเรียกใช้การทำงานของเอ็มวีพีส่วนขยายแทนการทำงานของเอ็มวีพีปกติ

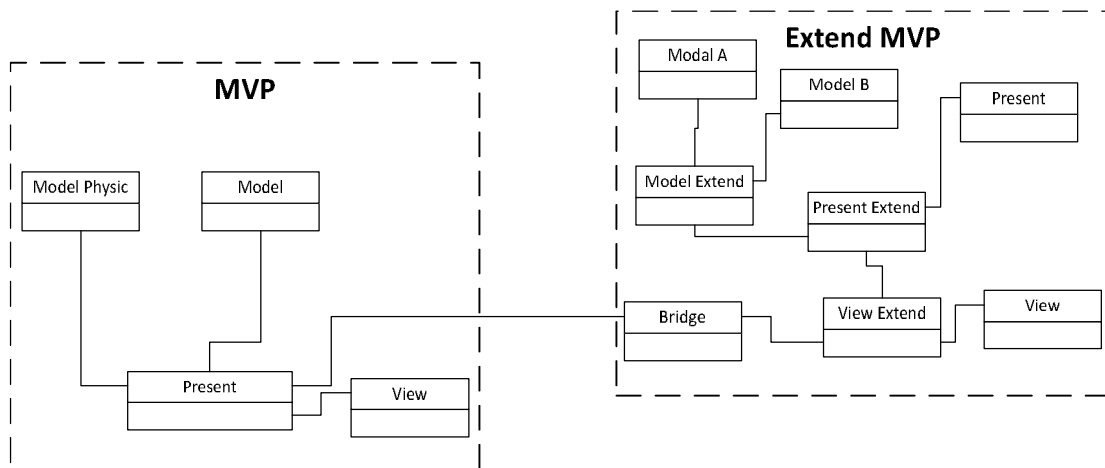
### 3.2.5 ออกแบบเกมเอ็นจิน

นำข้อมูลที่สังเคราะห์ได้จากขั้นตอนก่อนหน้า ได้แก่ คุณลักษณะ เป้าหมาย กลยุทธ์และแบบ  
รูปมาผนวก ออกแบบเกมเอ็นจิน มาสร้างเป็นโครงสร้างและนำเสนอผ่านแผนภาพ ดังภาพที่ 3.4



ภาพที่ 3.4 แผนภาพคลาสโครงสร้างของเอ็นจิน

มุ่งเน้นไปที่ลักษณะเฉพาะในการนำเสนอส่วนขยายของระบบ ซึ่งมีสัมพันธ์กันระหว่างการทำงานของเอ็มวีพีปกติและเอ็มวีพีส่วนขยายโดยทำงานผ่านแบบรูปตัวเชื่อม (Bridge) เอ็มวีพีขยายจะทำงานเมื่อมีการมอด แบบรูปส่วนเชื่อม ทำหน้าที่จัดการผลการการทำงานของเอ็มวีพีส่วนขยายเพื่อประเมินผล ตัวอย่างเช่น เมื่อมีการเรียกใช้เอ็มวีพีส่วนขยาย แบบรูปตัวเชื่อมจะคัดกรองผลการการทำงานตามกรณีการปรับแต่งว่ายอมรับได้หรือไม่ และถ้ายอมรับไม่ได้จะหาแนวทางแก้ไขอย่างไร อนึ่งกรณีปรับปรุงส่วนต่อประสาน เอ็มวีพีส่วนขยายนั้นอาจจะมีการเชื่อมต่อกับ พรีเซ็นเตอร์ โมเดล วิว ได้หลายตัวตามกรณีการมอด ยกตัวอย่างเช่น โมเดล A ใช้กับมอด X หรือ โมเดล B ใช้กับมอด Y เป็นต้น ดังภาพที่ 3.5

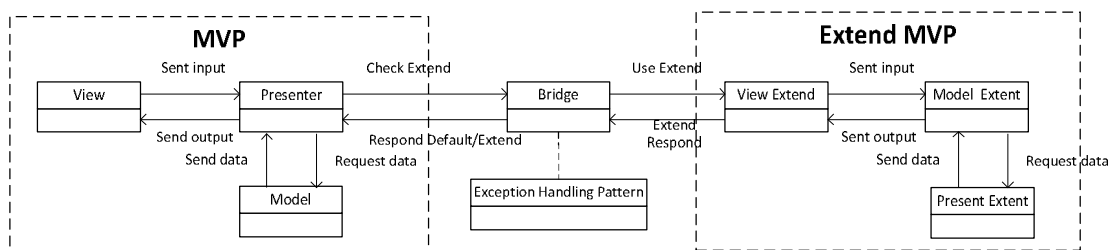


ภาพที่ 3.5 แผนภาพคลาสความสัมพันธ์ของ เอ็มวีพีส่วนขยายและ เอ็มวีพีของเอ็นจิน

### 3.3 ผนวกเกมเอ็นจินที่ออกแบบร่วมกับกรณีการปรับแต่งส่วนต่อประสาน

ขั้นตอนในการนำเสนอการทำงานของส่วนสนับสนุนส่วนขยาย (Extend Support) ของเอ็นจิน มีลักษณะการทำงานหลักเป็นการคัดกรองข้อผิดพลาดที่เกิดขึ้นจากการทำงานของเอ็มวีพีส่วนขยาย ซึ่งการทำงานปกติของระบบจะไม่ผ่านเข้าสู่แบบรูปตัวเชื่อมจะจบการทำงานในส่วนของเอ็มวีพีปกติ นั้น แต่กรณีที่มีการมอดจึงจะมีการเรียกใช้ให้เอ็มวีพีส่วนขยายทำงาน และมีแบบรูปตัวเชื่อมทำหน้าที่คัดกรองผลการการทำงานของมอดเพื่อยินยอมให้ระบบใช้ผลการการทำงานมอดหรือไม่ ตัวอย่างเช่น จะเลือกนำเสนอผลการการทำงานของเอ็มวีพีส่วนขยายหรือการนำเสนอค่าโดยปริยาย(Default) เป็นต้น โดยใช้แบบรูปจัดการสิ่งผิดปกติตรวจสอบการผลการการทำงาน อ้างอิงการตรวจสอบตามกรณีการปรับแต่งจาก 3.1 นำเสนอองค์ประกอบสำคัญ การทำงานส่วนขยายกับของเอ็นจินผ่านแผนภาพคลาส ดังภาพที่ 3.6 โดยมีรายละเอียดดังนี้



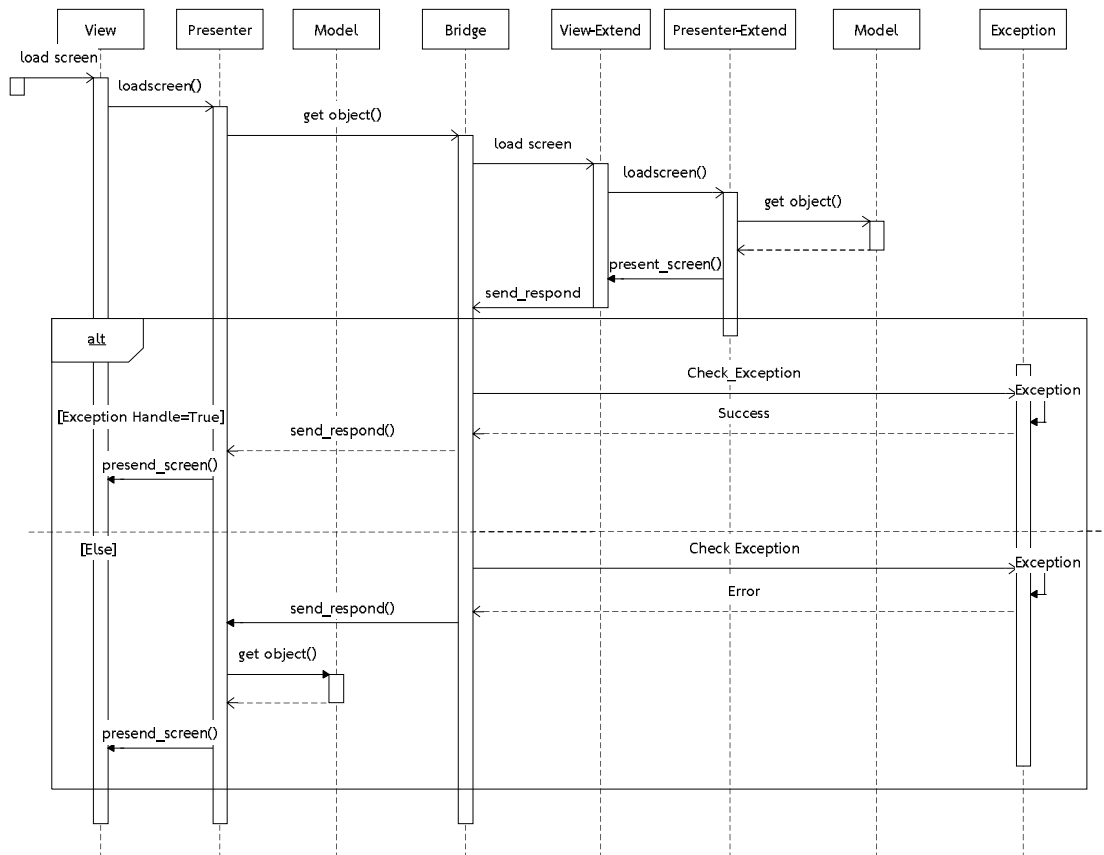


ภาพที่ 3.6 แผนภาพการไหลของข้อมูลของการทำงานส่วนขยายในเอ็นจิน

- 1) แบบรูปเอ็มวีพีของเอ็นจิน ส่วนการทำงานหลักในการควบคุมการรับส่ง ข้อมูลนำเข้า และ ข้อมูลนำออกของระบบ
- 2) แบบรูปตัวเชื่อม (bridge) ทำงานเป็นตัวกลางในการเชื่อมต่อระหว่าง เอ็มวีพีของเอ็นจิน และเอ็มวีพีของส่วนขยาย มีหน้าที่ส่งข้อมูลนำเข้าจาก เอ็มวีพีของเอ็นจินให้แก่เอ็มวีพีของส่วนขยายแทนและรับส่วนผลการทำงานของเอ็มวีพีส่วนขยายส่งกลับให้แก่เอ็มวีพีของเอ็นจิน ซึ่งจะผ่านการคัดกรองข้อผิดพลาดจากแบบรูปควบคุมเงื่อนไข
- 3) แบบรูปควบคุมเงื่อนไข (Exception Handling Pattern) ส่วนทำหน้าที่ในการกรอง ตรวจสอบ ผลการทำงานของเอ็มวีพีเอ็นจิน และเป็นส่วนคัดเลือก ตัดสินใจ เลือกรการทำงานของเอ็มวีพีส่วนขยายหรือไม่
- 4) เอ็มวีพีส่วนขยาย ทำหน้าที่ทำงานแทน เอ็มวีพีของเอ็นจินเมื่อมีการทำมอด

ขั้นตอนในการทำงานของแบบรูปจัดการสิ่งผิดปกติจะเป็นแกนหลักในการทำงานเมื่อมีการมอดเกิดขึ้นผ่านการใช้แบบรูปตัวเชื่อม โดยการคัดกรองข้อผิดพลาด การจัดการข้อผิดพลาด ผู้วิจัยเลือกใช้แบบรูปดังกล่าวเพื่อให้เกิดความสอดคล้องกับหลักวิศวกรรมซอฟต์แวร์ผู้ใช้ชั้นปลายที่คำนึงถึงการจัดการ จัดการสิ่งผิดปกติอันเกิดจากการพัฒนาซอฟต์แวร์โดยผู้ใช้ชั้นปลาย

ผลที่ได้คือการทำงานของเอ็นจินนำเสนอในรูปแบบแผนภาพลำดับซึ่งแสดงการทำงานปกติ ดังภาพที่ 3.7 อนึ่งแม้การพัฒนาเกิดขึ้นแต่จากโครงสร้างเบื้องต้นยังไม่สามารถควบคุมการนำเสนอข้อผิดพลาดออกมาแก่ผู้ใช้ได้ ดังนั้นจากการสังเกต (Observe) โดยการเพิ่มคลาสตัวเชื่อมสำหรับตรวจสอบผลการทำงานของเอ็มวีพีส่วนขยายซึ่งมีแบบรูปจัดการสิ่งผิดปกติทำงานอยู่ภายใน



ภาพที่ 3.7 แผนภาพลำดับ (Sequence Diagram) การทำงานของส่วนขยายภายในเอ็นจิน

จากการวิเคราะห์ปัญหาข้างต้นนำไปสู่การจำแนกรูปแบบการมอดกรณีปรับปรุงส่วนต่อประสานเพื่อใช้เป็นกรณีศึกษา ทดสอบในขั้นตอนถัดไป และการออกแบบเกมเอ็นจินเพื่อให้เหมาะสมกับการทำส่วนขยาย ตลอดจนการผนวกรวมการออกแบบเพื่อให้เกิดความสามารถในการคัดกรองข้อผิดพลาดที่เกิดขึ้นตามรูปแบบการมอดได้ ทั้งนี้รูปแบบโครงสร้างดังกล่าวจะสามารถทำงานได้เมื่อมีการออกแบบแบบรูปเพื่อควบคุมข้อผิดพลาดตามประเภทการมอดซึ่งจะกล่าวถึงในบทถัดไป

## บทที่ 4

### ออกแบบแบบรูป

#### และการประยุกต์ใช้เพื่อการแก้ปัญหาของการมอดด้วยวิธีการปรับปรุงส่วนต่อประสาน

ขั้นตอนในการนำเสนอวิธีการออกแบบรูปและการประยุกต์ใช้แบบรูป โดยการนำเสนอในระดับโครงสร้างและระดับพฤติกรรมเพื่ออธิบายถึงลักษณะการทำงานของระบบในการปรับปรุงส่วนต่อประสาน พร้อมทั้งประยุกต์ใช้แบบรูปสำหรับการจัดการสิ่งผิดปกติในการจัดการปัญหาที่เกิดขึ้นอันมีสาเหตุจากการทำมอดด้วยวิธีการปรับปรุงส่วนต่อประสานเพื่อลดข้อผิดพลาดที่แสดงผลตามกรณีที่กำหนดไว้ โดยนำเสนอการวิเคราะห์ผ่านแผนภาพลิตเติลจิล ใช้แผนภาพคลาสในการนำเสนอโครงสร้างและตัวอย่าง จึงเสนอการใช้งานวิธีการนำแบบรูปไปใช้ด้วย ซึ่งมีองค์ประกอบดังตารางที่ 4.1 ซึ่งอ้างอิงจากเอกสารอ้างอิง [15, 16, 17, 18, 21]

ตารางที่ 4.1 ตัวอย่างส่วนประกอบของแบบรูป

| หัวข้อ                         | คำอธิบาย  | ตัวอย่าง   |
|--------------------------------|---|--|
| ชื่อ (name)                    | ชื่อการประยุกต์แบบรูปนั้นๆ<br>ชื่อของแบบรูป   | แบบรูปควบคุมเชิงเงื่อนไข   |
| จุดประสงค์ (intent)            | ความจำเป็นที่ต้องนำแบบรูปไปใช้<br>วัตถุประสงค์ในการออกแบบ<br>เพื่อใช้ในการแก้ปัญหาใด ทำหน้าที่<br>ได้                   | จัดการกรณีการทำงานที่ไม่พึงประสงค์   |
| ปัญหา (Problem)                | ปัญหาซึ่งนำไปสู่การออกแบบ<br>ปัญหาที่เกิดขึ้น ที่มาของปัญหา<br>สาเหตุ และเงื่อนไขของปัญหานั้นๆ                          | กระบวนการมีปัญหาจนไม่สามารถ<br>ทำงานต่อไปไม่ได้  |
| การประยุกต์ใช้ (applicability) | สถานการณ์ที่ควรนำแบบรูปนี้ไปใช้<br>สถานการณ์หรือเงื่อนไขที่<br>เหมาะแก่การใช้แบบรูป และหน้าที่<br>การทำงานหลักของแบบรูป | เมื่อกระบวนการมีปัญหาเกิดขึ้น ทำ<br>การสร้างแนวทางในการแก้ปัญหาว่ามี<br>กี่แนวทาง และแก้แต่ละแนวทาง<br>อย่างไร |

ตารางที่ 4.1 ตัวอย่างส่วนประกอบของแบบรูป (ต่อ)

| หัวข้อ                   | คำอธิบาย  | ตัวอย่าง   |
|--------------------------|---|--|
| โครงสร้าง<br>(structure) | โครงสร้างทั่วไปของ แบบรูป<br>องค์ประกอบในการทำงาน<br>ของแบบรูปและหน้าที่ของแต่ละ<br>องค์ประกอบ                      | จัดการสิ่งผิดปกติประกอบตัว<br>องค์ประกอบดังนี้<br><i>ตรวจสอบ</i> คัดกรองเงื่อนไขที่ไม่พึง<br>ประสงค์ในกระบวนการทำงานแก้ไข<br><i>แก้ไข</i> แก้ไขข้อผิดพลาดเพื่อให้<br>กระบวนการสามารถทำงานต่อไปได้<br><i>ต่อเนื่อง</i> ทำให้การทำงานเข้าสู่<br>กระบวนการทำงานปกติ |
| วิธีการใช้               | วิธีการนำไปใช้หรือพัฒนา<br>การพัฒนา การนำไปใช้  | การไปเที่ยวต่างจังหวัดกับครอบครัว<br>ด้วยรถยนต์ส่วนตัว แล้วรถยนต์ชำรุด   |
| ตัวอย่าง<br>(example)    | ตัวอย่าง<br>ตัวอย่างการทำงานของแบบ<br>รูป ได้แก่สถานการณ์ตัวอย่าง<br>ลำดับการทำงานและแผนภาพเพื่อ<br>สร้างความเข้าใจ | ปรากฏว่า<br>1. (ตรวจสอบ)รถยนต์ชำรุด<br>2. ทางครอบครัวจึงหาวิธีการอื่น<br>(แก้ไข) ด้วยการ<br>3. ยืมรถยนต์ของญาติ จึง<br>สามารถเดินทางได้ต่อไป<br>(ต่อเนื่อง)  |

ทั้งนี้จากงานวิจัยกำหนดองค์ประกอบเกี่ยวกับศัพท์ที่ใช้ในเกมดังต่อไปนี้

**อ็อบเจกต์เกม (Game Objects)** องค์ประกอบวัตถุที่อาจจะอยู่ในโลกของเกม เช่น รูปภาพที่ปรากฏ  
ในเกม สถานะพื้นดิน ท้องฟ้า เป็นต้น หรือ อยู่นอกโลกของเกม เช่น หน้าต่างเมนู ข้อความ เป็นต้น  
ซึ่งองค์ประกอบเหล่านี้สามารถปรับแต่งได้เป็นส่วนต่อประสานของเกม

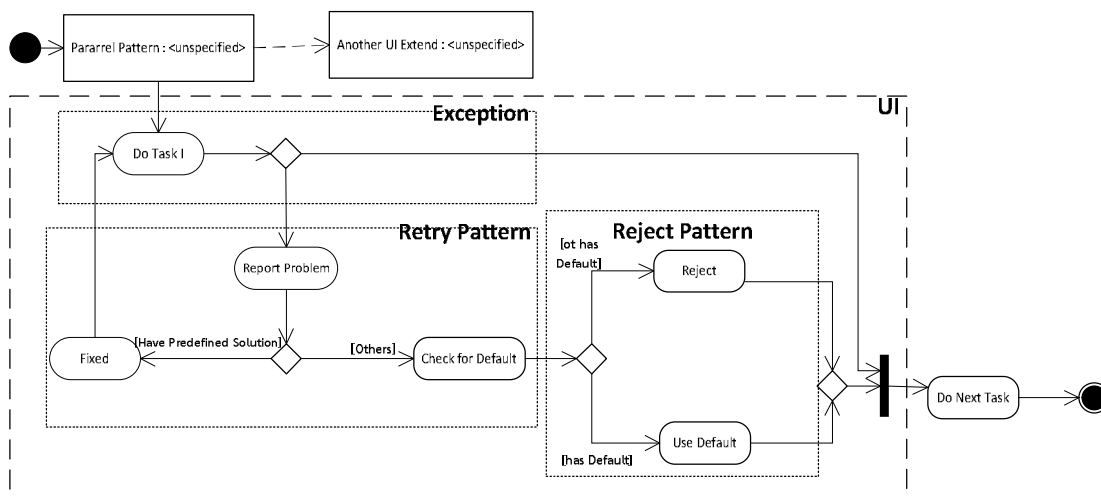
**คุณสมบัติของอ็อบเจกต์เกม (Game Objects Properties)** ค่าใดๆที่มีผลต่ออ็อบเจกต์เกม เช่น  
อ็อบเจกต์กล่องรูปภาพ มีคุณลักษณะขนาดภาพ ตำแหน่ง นามสกุล เป็นส่วนประกอบ

**เอฟเฟกต์เกม (Game Effect)** การเน้นจุดสำคัญในเกมเพื่อให้ผู้ใช้หรือผู้เล่นทราบถึงสิ่งที่สามารถมีปฏิสัมพันธ์ด้วยได้ สิ่งที่กำลังปฏิสัมพันธ์อยู่หรือผลของการกระทบใดๆ เช่นการแสดงผลเอฟเฟกต์เพื่อให้ผู้เล่นทราบว่ากำลังเลือกใช้ตัวละครตัวใดอยู่ เป็นต้น

**ผู้ปรับแต่ง (Modder)** คือผู้พัฒนามอดของเกมนั้น

#### 4.1. การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการนำเสนอข้อผิดพลาดจากการปรับปรุงส่วนต่อประสาน

การวิเคราะห์การจัดการกับเงื่อนไขไม่พึงประสงค์ของการจัดการส่วนต่อประสานในภาพรวม นำผลการศึกษาวิจัยในขั้นตอน 3.3 ออกแบบเป็นแผนภาพกิจกรรมอธิบายพฤติกรรมการทำงานของจัดการสิ่งผิดปกติเมื่อเกิดกรณีข้อผิดพลาดและการการทำงานภายหลังในจัดการสิ่งผิดปกติมีการทำงานอย่างไร เพื่อนำไปสู่การวิเคราะห์การเกิดข้อผิดพลาดตามกรณีปรับปรุงส่วนต่อประสานในขั้นตอนถัดไป มีรูปแบบการทำงานใช้แบบรูปทดลองใหม่ (retry) ในการตรวจสอบเงื่อนไข ใช้แบบรูปปฏิเสธและแบบรูปซ่อมแซมตามในการทวนสอบการทำงานดังภาพที่ 4.1



ภาพที่ 4.1 แผนภาพกิจกรรมการทำงานของส่วนขยาย เพื่อจัดการการทำงานของมอด

**จุดประสงค์** เพื่อจัดการข้อผิดพลาดในกรณีที่มีการมอดหรือการใช้งานส่วนขยายของซอฟต์แวร์เกม

**การประยุกต์ใช้** แบบรูปนี้จะทำงานเพื่อตรวจสอบการทำงานของมอด (mod) และแก้ไขโดยการเปลี่ยนองค์ประกอบเพื่อให้การมอดสามารถทำงานได้หรือยกเลิกการมอดเพื่อป้องกันการนำเสนอข้อผิดพลาดแก่ผู้ใช้นี้

1. ตรวจสอบคัดกรองข้อผิดพลาด

2. การแก้ไขข้อผิดพลาดซึ่งอาจเป็นการเพิ่มกระบวนการเพื่อแก้ไขหรือนำการแสดงผลอื่นๆ มาทดแทน
3. ยกเลิกการใช้งานมอด ซึ่งสามารถทำได้โดยการใช้การนำค่าปริยาย (Default) หรือการยกเลิกการทำงานพร้อมแจ้งข้อความถึงสิ่งที่ Error ให้ผู้ใช้ทราบ

**โครงสร้าง** มืองค์ประกอบสามส่วนใหญ่ๆซึ่งทำหน้าที่ต่างกันแต่มีลำดับการทำงาน โดยในทุกๆการปรับแต่งส่วนต่อประสานจะมีการทำงานในรูปแบบเดียวกันดังนี้

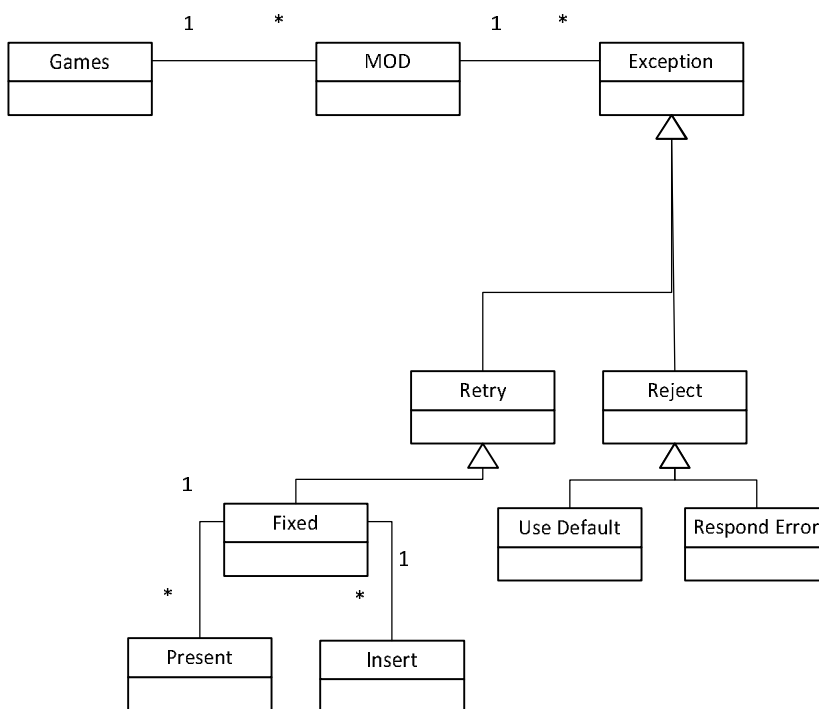
- **ส่วนตรวจสอบสิ่งผิดปกติ (Exception)** ทำหน้าที่ในตรวจสอบผลการทำงานของมอดเมื่อเกิดปัญหาจะทำ *แก้ไข* และส่งกลับไป *ทำงานใหม่* และผ่าน *ส่วนตรวจสอบ* ผลการทำงานอีกครั้ง จนถึงจำนวนครั้งที่กำหนดหรือผลการทำงานสำเร็จตามเงื่อนไข จะส่งไปสู่การทำงานถัดไป
- **แบบรูปการเริ่มการทำงานใหม่ (retry pattern)** ทำหน้าที่ตรวจสอบหาแนวทางแก้ไขผ่านส่วนแก้ไข (Fixed) โดยมีองค์ประกอบภายในรองรับใน 2 กรณีได้แก่ กรณีนำเสนองานอย่างอื่นทดแทนการทำงานของมอด และ ส่วนเพิ่มงานทำหน้าที่เพิ่มการทำงานใดๆโดยนิยามว่าผลการทำงานของงานนั้นๆจะทำให้ผลการมอดนั้นถูกต้อง
- **แบบรูปการปฏิเสธ (reject pattern)** ทำหน้าที่ *ปฏิเสธ* ผลของการมอดเมื่อผลการทำงานของแบบรูปก่อนหน้าไม่สามารถแก้ไขการทำงานใดๆได้ โดยการเลือกใช้ค่าโดยปริยาย (Default) ของระบบแทน หรือแจ้งข้อผิดพลาดให้แก่ผู้ใช้ทราบเพื่อดำเนินการแก้ไขต่อไป

**วิธีการใช้หรือตัวอย่าง** จากรูปแสดงขั้นตอนการทำงานของเกมเมื่อมีการมอดดังนี้

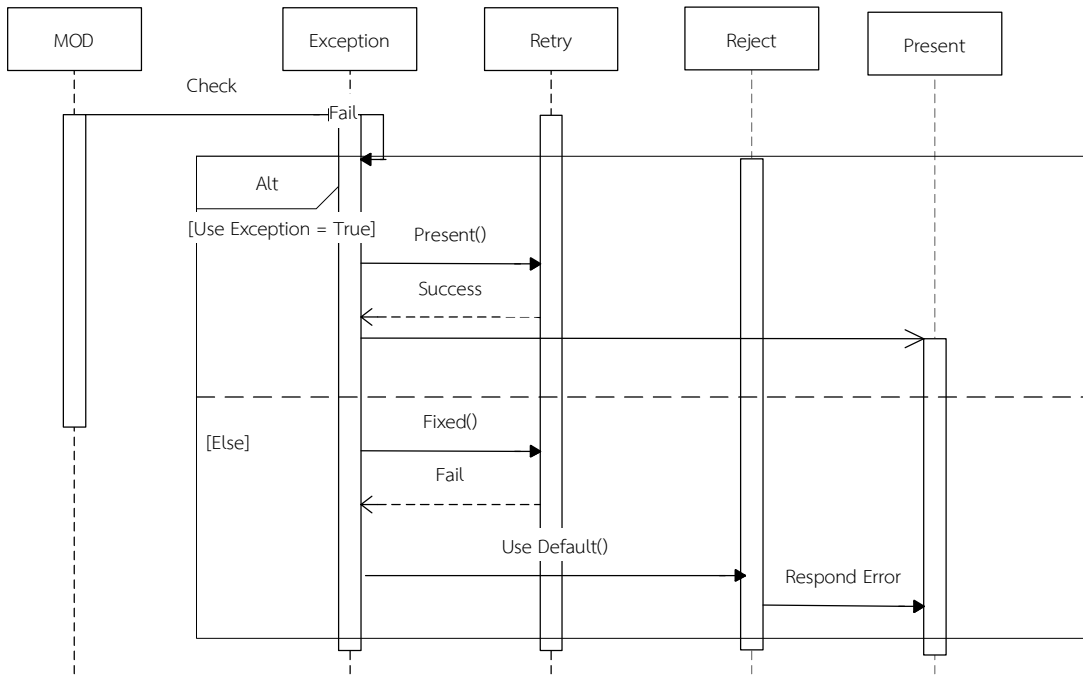
1. ตรวจสอบผลการทำงานของมอดในกรณีที่มีข้อผิดพลาด
2. เข้าสู่การแก้ไขข้อผิดพลาดผ่านแบบรูปการเริ่มการทำงานใหม่ ตามเงื่อนไขที่เกิดขึ้น โดยเลือกการแก้ไขจาก 2 กรณีคือ นำเสนอสิ่งอื่นทดแทน และเพิ่มงานเพื่อแก้ไข
3. กรณีที่ไม่สามารถใช้แบบรูปการเริ่มการทำงานใหม่ได้ จะทำการแบบรูปยกเลิกเพื่อยกเลิกการทำงานของมอด 2 กรณี ได้แก่ การใช้ค่าโดยปริยาย (Default) แทนผลมอดที่อาจจะผิดพลาด หรือก็คือการยกเลิกการเริ่มมอดพร้อมนำเสนอข้อความแจ้งข้อผิดพลาดแก่ผู้ใช้

ดังภาพที่ 4.2 แผนภาพคลาสแสดงโครงสร้างของแบบรูป เมื่อเกมเริ่มทำงานจึงจะเกิดการใช้งานมอดโดยมีส่วนตรวจสอบสิ่งผิดปกติทำหน้าที่ตรวจสอบถึงข้อผิดพลาดที่เกิดขึ้น เมื่อเกิดข้อผิดพลาดจะมีแนวทางจัดการปัญหา 2 กรณีคือ แบบรูปเริ่มการทำงานใหม่ทำหน้าที่แก้ไขการทำงานของการมอด

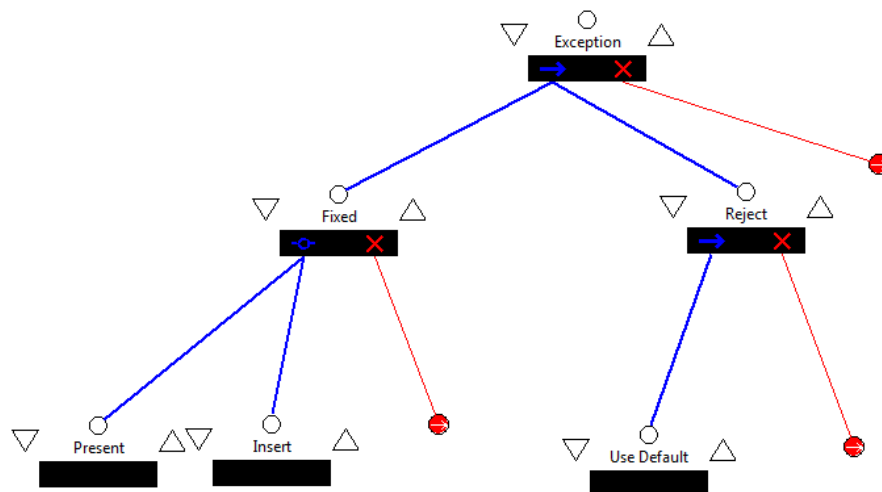
เพื่อให้สามารถทำงานต่อไปได้ โดยมีทั้งการเลือกส่วนการทำงานอื่นมานำเสนอแทนหรือเพิ่มงานเพื่อแก้ไขการทำงานให้สมบูรณ์ และแบบรูปการปฏิเสธทำหน้าที่ยกเลิกการทำงานของมอดโดยการนำค่าปริยายมาใช้แทนหรือแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้นเพื่อให้ผู้ปรับแต่งสามารถพัฒนาแนวทางแก้ไขข้อผิดพลาดนั้นๆต่อไปได้ นอกจากนี้ยังเสนอแผนภาพลำดับแสดงลำดับการทำงานของแบบในภาพที่ 4.3 และ ภาพที่ 4.4 แผนภาพลิตเติลจิลแสดงโครงสร้างการทำงานของแบบรูป



ภาพที่ 4.2 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติในการจัดการการมอดเกม



ภาพที่ 4.3 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการนำเสนอ ข้อผิดพลาดจากการปรับปรุงส่วนต่อประสาน



ภาพที่ 4.4 แผนภาพ ลิตเติ้ล-จิล ตัวอย่างของการจัดการมอดเกม จากการทำมอดเกม



จากกรณีการการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติเพื่อจัดการปัญหาปรับปรุงส่วนต่อประสานข้างต้นเป็นการแสดงการตรวจสอบการทำมอด และการจัดการภายหลังการตรวจสอบ โดยไม่ลงรายละเอียดเกี่ยวกับกรณีข้อผิดพลาดที่เกิดขึ้นแต่เป็นเพียงวิธีการจัดการปัญหาในภาพรวมว่าเมื่อนำแบบรูปจัดการสิ่งผิดปรกติมาใช้จะมีลักษณะการทำงานอย่างไร ผู้วิจัยได้ศึกษาและทำการออกแบบแบบรูปเพื่อควบคุมเงื่อนไขที่ไม่พึงประสงค์จากการทำมอดด้วยวิธีการปรับปรุงส่วนต่อประสาน ในแต่ละกรณีตามหัวข้อ 3.1 นำเสนอในหัวข้อถัดไป

#### 4.2 แบบรูปลำดับที่ 1: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีเพื่อความพึงพอใจ

การปรับปรุงส่วนต่อประสานกรณีเพื่อความพึงพอใจ เป็นการปรับปรุงเปลี่ยนแปลงส่วนต่อประสานเพื่อให้มีลักษณะต่างไปจากส่วนต่อประสานเดิมของเกมเพื่อความพึงพอใจส่วนตัวของผู้ใช้งาน เพื่อมีลักษณะเด่น แตกต่าง หรือสร้างแรงดึงดูดใจผู้ใช้งาน โดยการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ ตัวอย่างเช่น การแก้ไขฉากหรือรูปภาพของเมนูเกม การเปลี่ยนรูปแบบเคอร์เซอร์ของเกม เป็นต้น ดังตารางที่ 4.2

ตารางที่ 4.2 ตัวอย่างการปรับแต่งเพื่อความพึงพอใจ

| ลำดับ | การปรับแต่ง           | รูปแบบการแก้ไข                          | จุดประสงค์การปรับแต่ง             | หมายเหตุ                            |
|-------|-----------------------|---|-----------------------------------|-------------------------------------|
| 1     | เปลี่ยนภาพเมนูเกม     | เปลี่ยนเมนูเกม                          | เพื่อความสวยงามเฉพาะตัว           |                                     |
| 2     | เปลี่ยนเคอร์เซอร์     | เปลี่ยนเคอร์เซอร์เป็นรูปดาบ             | ให้ได้บรรยากาศเหมาะสมกับโลกของเกม | โลกของเกมเป็นยุคกลางสมัยสงครามศาสนา |
| 3     | เปลี่ยนรูปภาพตัวละคร  | เปลี่ยนรูปภาพตัวละครเป็นภาพผู้เล่นจริงๆ | เพื่อความโดดเด่นเป็นเอกลักษณ์     |                                     |
| 4     | เปลี่ยนไอคอนแสดงสถานะ | เปลี่ยนไอคอนให้บรรยากาศเหมือนก้อนหิน    | ให้ได้บรรยากาศเหมาะสมกับโลกของเกม | โลกของเกมเกี่ยวกับเหมืองหิน         |

ตารางที่ 4.2 ตัวอย่างการปรับแต่งเพื่อความพึงพอใจ (ต่อ)

| ลำดับ | การปรับแต่ง        | รูปแบบการแก้ไข                   | จุดประสงค์การปรับแต่ง           | หมายเหตุ                            |
|-------|--------------------|----------------------------------|---------------------------------|-------------------------------------|
| 5     | เปลี่ยนไอคอนปุ่มกด | เปลี่ยนไอคอนให้บรรยากาศไม้กระดาน | ให้ได้บรรยากาศเหมาะกับโลกของเกม | โลกของเกมเป็นยุคกลางสมัยสงครามศาสนา |

**จุดประสงค์** จัดการปัญหาที่เกิดกรณีการมอดเกมโดยการแก้ไขหรือปรับเปลี่ยนส่วนต่อประสานจากอ็อบเจกต์เกม เช่น รูปภาพของเมนูเกม ไอคอน เคอร์เซอร์ โดยแต่ละอ็อบเจกต์เกมมีการปรับเปลี่ยนคุณสมบัติที่ไม่เหมาะสมกับโครงสร้างเดิมของระบบจำเป็นต้องมีการปรับสถานะคุณสมบัติของอ็อบเจกต์เกมนั้นให้สามารถแสดงผลได้ กรณีที่ไม่สามารถปรับใช้ได้แบบรูปจะยกเลิกการปรับเปลี่ยนนั้นๆ โดยไปใช้ค่าเดิมของระบบแทนหรือพร้อมนำเสนอข้อความแจ้งข้อผิดพลาด

**ปัญหา** การมอดเกมโดยการแก้ไขหรือปรับเปลี่ยนรูปภาพของ ทำให้เกิดปัญหาไม่สามารถแสดงผลได้ เนื่องจากรูปภาพที่นำมาแก้ไข มีคุณสมบัติที่ระบบไม่รองรับเช่น ขนาดของรูปภาพไม่สามารถแสดงผลได้ มีขนาดที่ไม่พอดีกับกรอบของรูปภาพเดิมของเมนูเกม ทำให้ระบบแสดงผลผิดพลาด ปัญหาเกิดจากกรณีที่มีการปรับแต่งรูปภาพนั้นผู้ปรับแต่งไม่ได้ทราบถึงเงื่อนไขที่ต้องพิจารณา เช่น ขนาด เป็นต้น

วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดขอบเขตคุณสมบัติของอ็อบเจกต์เกมที่ต้องการจัดการสิ่งผิดปรกติ ตัวอย่างเช่น การแก้ไขรูปภาพเมนูเกม ต้องมีการกำหนดขนาด ที่ชัดเจน
2. กำหนดวิธีการแนวทางการแก้ปัญหาของแต่ละอ็อบเจกต์ของเกมที่ต้องการจัดการสิ่งผิดปรกติ เช่น เมื่อขนาดของรูปภาพเมนูเกมไม่สอดคล้องกับขอบเขตจะมีการปรับค่าเมนูใหม่ให้เหมาะสม เป็นต้น
3. กำหนดการแสดงผลและยกเลิกการทำงานเมื่อมีข้อผิดพลาดจากการปรับเปลี่ยนอ็อบเจกต์นอกเหนือจากที่กำหนดเพื่อให้ทราบถึง กรณีอ็อบเจกต์ใหม่ของเกมที่ต้องนำมาพิจารณานำไปกำหนดขอบเขตเพิ่มเติม

**การประยุกต์ใช้** แบบรูปทำหน้าที่ควบคุมเงื่อนไข เมื่อมีการปรับแต่งอ็อบเจกต์เกมเช่น รูปภาพของเมนูเกม ไอคอนเกม โดยจัดการกรณีที่เกิดปัญหาจากการปรับแต่งต้องมีการปรับค่าให้เข้ากับขอบเขตของอ็อบเจกต์ของเกมนั้นๆ

**โครงสร้าง** มีโครงสร้างการทำงานแบ่งออกเป็น 4 ส่วนโดยแบ่งเป็นส่วนที่เลือกทำงาน 2 ส่วนและส่วนบังคับใช้งาน 2 ส่วนดังนี้

ส่วนบังคับใช้งานงาน

- **การจัดการสิ่งผิดปกติ (Exception)** กำหนดรายละเอียดของการแก้ไขเกมนี้ๆ มีอ็อบเจกต์ใดของเกมที่ต้องการแก้ปัญหา
- **การยกเลิก (Reject)** นำเสนอแนวทางการยกเลิกการปรับแต่งเกมในแต่ละการปรับแต่ง และในแต่ละอ็อบเจกต์เกม โดยมีคลาส การชดเชย (Compensate) อยู่ภายในทำหน้าที่นำเสนอแนวทางการยกเลิกการปรับแต่งอ็อบเจกต์เกมที่มีความสัมพันธ์กัน

ส่วนเลือกใช้งาน

- **ทางเลือกแบบมีลำดับ (ordered Alternative)** นำเสนอการแก้ไขในแต่ละคุณสมบัติของอ็อบเจกต์ของเกม โดยจะประกอบด้วยคุณสมบัติที่คล้ายคลึงกันเพื่อแยกการนำไปใช้ซ้ำ
- **การทางเลือกแบบไม่มีลำดับ (Unordered Alternative)** นำเสนอการแก้ไขคุณสมบัติพิเศษของอ็อบเจกต์เกม ที่มีลักษณะเฉพาะที่ไม่สามารถนำไปใช้ซ้ำได้ในอ็อบเจกต์เกมประเภทอื่นๆ

การเลือกใช้องค์ประกอบแต่ละส่วนจะขึ้นกับเงื่อนไขการตรวจสอบการปรับแต่งนั้นๆ โดยจะต้องให้ความสำคัญในการจัดลำดับความสำคัญของหลักที่มี เพื่อให้องค์ประกอบการชดเชยทำงานได้อย่างมีประสิทธิภาพ

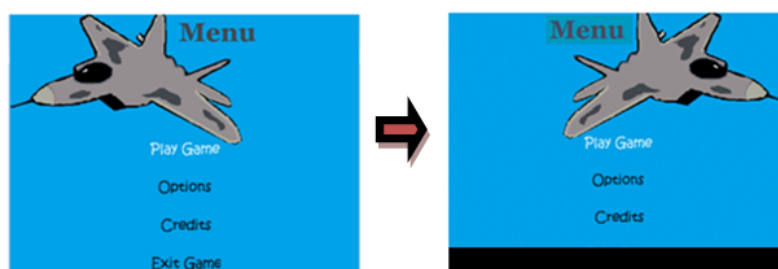
**วิธีการใช้** เป็นไปตามลำดับดังต่อไปนี้

1. ผู้ปรับแต่งกำหนดสิ่งที่ต้องการปรับแต่งในเกม ระบุว่าใช้อ็อบเจกต์ใดในการปรับแต่ง พร้อมกำหนดคุณสมบัติของอ็อบเจกต์และขอบเขตของอ็อบเจกต์เกมของผู้ปรับแต่งทราบลงไปในส่วนจัดการสิ่งผิดปกติ
2. กำหนดวิธีการแก้ปัญหาสิ่งผิดปกติตามขอบเขตของคุณสมบัติเกม โดยเลยเลือกใช้จากแบบรูป 2 กลุ่มได้แก่
  - 2.1 ทางเลือกแบบมีลำดับใช้แก้ปัญหาของสิ่งผิดปกติในกรณีที่คุณสมบัติอ็อบเจกต์เกมเป้าหมายเป็นคุณสมบัติมาตรฐานของคุณสมบัติอ็อบเจกต์เกมทั่วไปที่สามารถนำไปใช้งานซ้ำได้ เช่น ขนาดของรูปภาพ ตำแหน่ง สี เป็นต้น

- 2.2 ทางเลือกแบบไม่มีลำดับใช้แก้ปัญหาของสิ่งผิดปกติในกรณีที่คุณสมบัติอ็อบเจกต์เกมเป้าหมายมีคุณสมบัติที่แตกต่างจากคุณลักษณะปกติ เช่น ความเร็วในการเคลื่อนที่ของอ็อบเจกต์เกมในเกมที่มีการกำหนดขอบเขตให้เคลื่อนไหวได้ เป็นต้น
3. กำหนดแนวทางการยกเลิกการทำงาน โดยทั่วไปจะออกแบบให้ยกเลิกการทำงานได้ 2 กลุ่ม ขึ้นกับเงื่อนไขของการปรับแตงนั้นๆ ดังนี้
- 4.1 การยกเลิกการทำงานโดยแสดงข้อความแจ้งเตือนเพื่อให้ผู้ปรับแต่งทราบถึงข้อผิดพลาดใหม่ๆที่ไม่ได้ออกแบบไว้
- 4.2 การยกเลิกแบบขดเซย เพื่อขดเซยผลการทำงานกรณีที่ความผิดพลาดนั้นๆมีความต่อเนื่องจากการแก้ไขอื่นๆ เช่น การแก้ไขขนาดรูปภาพของเมนูเกมไปแล้ว แต่กลับพบปัญหาจากตำแหน่งรูปภาพที่ไม่สามารถแก้ไขได้ การที่ระบบจะยกเลิกการทำงานและเลือกใช้งานค่าปริยายจำเป็นต้องปรับขนาดรูปภาพของเมนูเกมให้เป็นค่าเริ่มต้นก่อน ไม่เช่นนั้นขนาดรูปภาพของเมนูเกมที่เป็นค่าปริยายจะผิดเพี้ยน

ทั้งนี้แบบรูปจะช่วยลดข้อผิดพลาดจากการปรับแตงอ็อบเจกต์เกม เมื่อมีการปรับแตงเกมโดยจะช่วยบรรเทาความผิดพลาดจากคุณสมบัติของอ็อบเจกต์เกมเป้าหมาย นอกจากนี้แบบรูปยังนำเสนอสาเหตุข้อผิดพลาดผิดพลาดใหม่ๆจากคุณสมบัติหรืออ็อบเจกต์ที่ไม่ได้กำหนดไว้ ณ ตอนต้นการออกแบบเพื่อให้ผู้ปรับแตงสามารถนำมาปรับใช้เพิ่มเติมในแบบรูปได้ต่อไป

**ตัวอย่าง** การมอดส่วนต่อประสานโดยการแก้ไขรูปภาพที่นำเสนอเป็นรูปภาพของเมนูเกมดังตัวอย่าง โดยใช้ขนาดของรูปภาพที่ไม่พอดีกับขนาดของพื้นหลังเดิม ผลคือทำให้พื้นที่ที่ขาดหายถูกแทนด้วยสีดำซึ่งทับกับเมนูในเกม ดังภาพที่ 4.5



ภาพที่ 4.5 ภาพตัวอย่างการเปลี่ยนขนาดภาพ

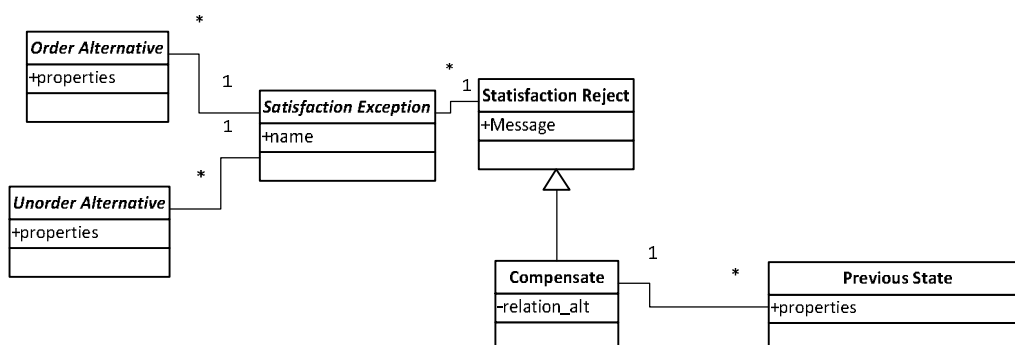
1. กำหนดอ็อบเจกต์ของเกมที่ทำให้ความสนใจลงในการจัดการสิ่งผิดปกติและทำการกำหนดคุณสมบัติของรูปภาพเท่าที่ผู้ใช้ทราบ ดังในกรณีตัวอย่าง ใช้ขนาดภาพของเอกสารรูปภาพใช้เดิมใช้ “640 x 480”

2. กำหนดแนวทางแก้ไข ตัวอย่างเช่น การกำหนดขนาดอื่นๆที่คุณสมบัติอ็อบเจกต์ของเกมสามารถเรียกใช้งานได้ใน *ส่วนทางเลือกแบบมีลำดับ*
3. ผลของการทำงานของแบบรูปจัดการสิ่งผิดปกติเป็นไปได้ 3 กรณีดังตัวอย่างต่อไปนี้
  - 3.1 *กรณีที่สามารถจัดการความผิดปกติได้* เมื่อเกมเริ่มทำงาน และเกิดข้อผิดพลาดขึ้น แบบรูปจะแก้ไขการมอดโดยการปรับค่าที่มีปัญหา ตามกรณีในส่วนทางเลือกมีลำดับ ในตัวอย่างนี้คือ การใช้ *ส่วนทางเลือกแบบมีลำดับ* โดยการหาค่าใกล้เคียงที่ระบบยอมรับได้ เช่น “800x600” หรือค่าใกล้เคียงจาก “640x480” ไม่เกิน 10 หน่วยเป็นต้น
  - 3.2 *กรณีที่ไม่สามารถจัดการความผิดปกติได้* แบบรูปจะสั่งให้ใช้ *ค่าโดยปริยาย* ของเกมที่ไม่ผ่านการมอดทำงานแทน จากตัวอย่างอย่างคือรูปภาพเดิมของเกมที่มี “640x480”
  - 3.3 *ในกรณีที่ไม่สามารถแก้ไขด้วยค่าปริยายได้* เช่น รูปภาพต้นฉบับของค่าปริยายถูกลบหรือถูกย้ายตำแหน่ง แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น ตัวอย่างเช่น ไม่พบเอกสารรูปภาพเดิมในตำแหน่งที่กำหนด

อย่างไรก็ตาม กรณีในการแก้ไขการมอดโดยการปรับค่าที่มีปัญหาสามารถปรับใช้ แบบรูปทดแทนแบบไม่มีลำดับได้ตามกรณีที่เหมาะสม เช่น มีการเรียกใช้คุณสมบัติเฉพาะของรูปภาพผ่านส่วนเสริม(Plug-in) ตัวอย่างเช่น ส่วนเสริมทำให้ภาพพื้นหลังเคลื่อนไหวได้ เป็นต้น ดังนั้นการใช้ส่วนเสริมทำให้มีคุณสมบัติใหม่เพิ่มเติมจากคุณสมบัติเดิมของอ็อบเจกต์รูปภาพ จึงต้องสร้างตัวกำหนดข้อผิดพลาดคุณสมบัติใหม่นี้แยกออกมา

นอกจากนี้กรณีการปรับแต่งที่มีความสัมพันธ์กันอาจทำให้เกิดผลกระทบได้ ตัวอย่างเช่น การปรับขนาดของอ็อบเจกต์รูปภาพที่ปรับ แต่เมื่อมีการปรับไปใช้ภาพปริยายของเกม ทำให้ขนาดกรอบรูปภาพในเกมผิดเพี้ยน ดังนั้นจึงเรียกใช้ การชดเชย เพื่อปรับขนาดรูปภาพให้เป็นไปตามขนาดรูปภาพเดิมทำให้สามารถแสดงผลค่าปริยายได้ถูกต้องตามเดิม ทั้งนี้การปรับแต่งบางกรณีต้องอาศัยเงื่อนไขตรวจสอบหลากหลายเพื่อให้ได้ผลการมอดที่ถูกต้อง

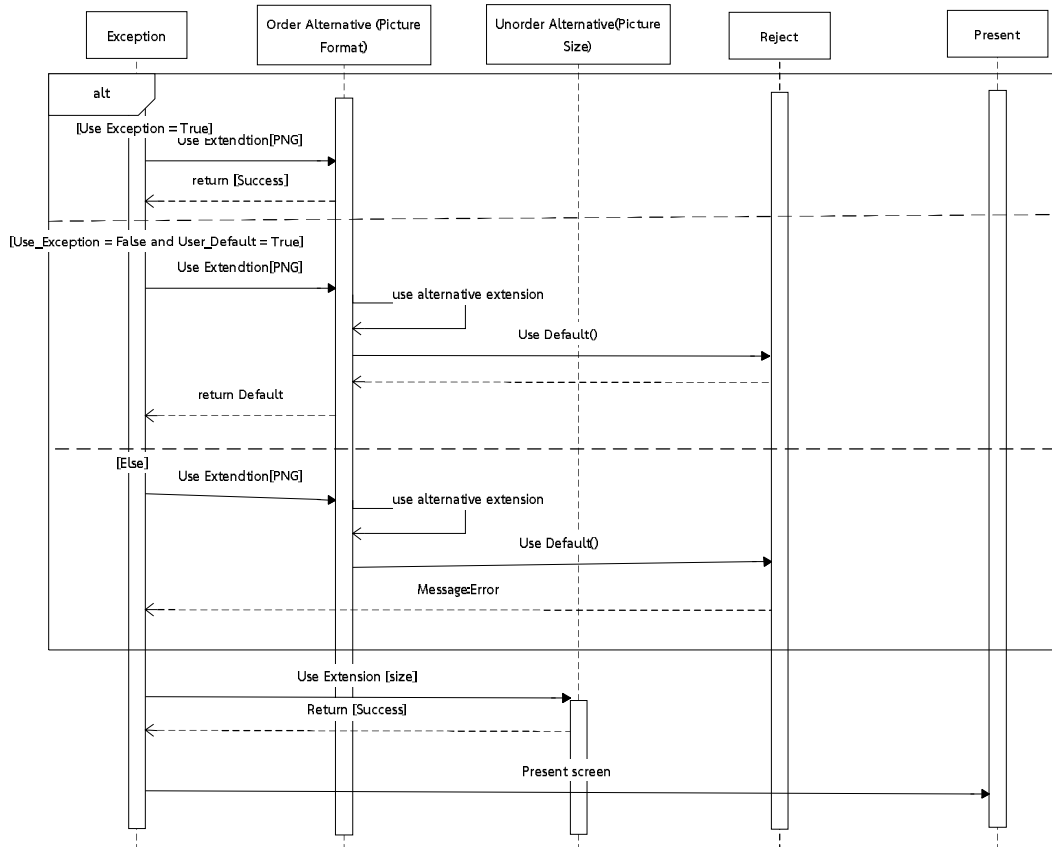
ดังภาพที่ 4.6 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อความพึงพอใจ



ภาพที่ 4.6 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติในการทำงานกรณีการปรับแต่งเพื่อความพึงพอใจของผู้ใช้

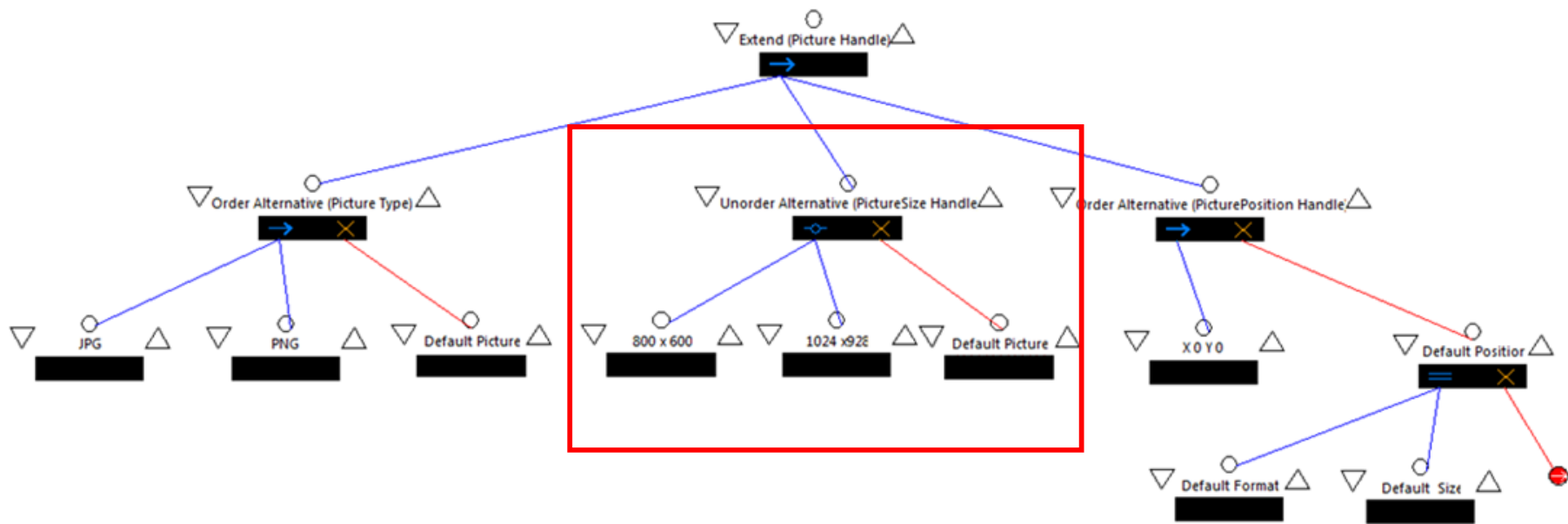
การทำงานของคลาสเมื่อกรณีการปรับแต่งนั้นๆสอดคล้องกับคลาสจัดการสิ่งผิดปกติเพื่อความพึงพอใจ (*Satisfaction Exception*) คลาสดังกล่าวจะกำหนดกรณีการปรับแต่งนั้นๆว่าเป็นอ็อบเจกต์ใดที่มีการปรับแต่งและตรวจสอบขอบเขตของคุณสมบัติอ็อบเจกต์เพื่อแก้ไขผ่านส่วนเลือกใช้งาน 2 กรณีได้แก่ทางเลือกแบบมีลำดับเพื่อความพึงพอใจ (*Satisfaction Ordered Alternative*) และทางเลือกแบบไม่มีลำดับเพื่อความพึงพอใจ (*Satisfaction Unordered Alternative*) โดยการใช้จะใช้งานทางเลือกแบบมีลำดับเมื่อคุณสมบัติของอ็อบเจกต์เกมที่ทำให้ความสนใจเป็นคุณสมบัติที่มีความสอดคล้องกับคุณสมบัติอื่นๆของอ็อบเจกต์อื่นๆที่สามารถประยุกต์ใช้ได้ ส่วนทางเลือกแบบไม่มีลำดับมุ่งให้ความสนใจเฉพาะในส่วนของคุณสมบัติอ็อบเจกต์พิเศษที่แตกต่างจากอ็อบเจกต์ปกติไม่สามารถนำกลับไปใช้ซ้ำได้ กรณีที่ส่วนจัดการสิ่งผิดปกติทั้ง 2 กรณีไม่สามารถจัดการกับสิ่งผิดปกติได้แบบรูปจะเข้าสู่ *ส่วนยกเลิกการทำงานเพื่อความพึงพอใจ (Satisfaction Reject)* เพื่อไม่ให้ระบบแสดงข้อผิดพลาดโดยการปรับใช้ค่าปริยายหรือแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้น โดยในกรณีที่ข้อผิดพลาดใดๆมีความสัมพันธ์กับคุณสมบัติอื่นๆที่ได้แก้ไขไปแล้วแบบรูปจะใช้ *ส่วนชดเชยเพื่อความพึงพอใจ (Satisfaction Compensate)* เพื่อทำการย้อนกลับการปรับแต่งก่อนหน้า (*Previous State*) ให้เหมาะสมตัวอย่างเช่น รูปภาพของเมนูเกมมีการแก้ไขขนาดและกรอบของรูปภาพ (*border*) กรณีที่แบบรูปสามารถจัดการปัญหาสิ่งผิดปกติของขนาดของรูปภาพได้โดยการปรับขนาดให้เหมาะสม แต่ไม่สามารถจัดการสิ่งผิดปกติในส่วนกรอบรูปภาพได้ เมื่อมีการยกเลิกย้อนกลับไปใช้รูปภาพของเมนูเกมเดิมจะต้องมีการปรับค่าขนาดรูปภาพให้เป็นค่าเริ่มต้นด้วย เนื่องจากการปรับขนาดจากแบบรูปจัดการสิ่งผิดปกติก่อนหน้านี้จะทำให้ขนาดรูปภาพเปลี่ยนแปลงไปซึ่งอาจจะไม่เหมาะสมกับขนาดของรูปในค่าปริยาย

โดยนำเสนอลำดับการทำงานผ่านภาพที่ 4.7 แผนภาพลำดับแสดงลำดับการทำงานของแบบรูป



ภาพที่ 4.7 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณืเพื่อความพึงพอใจ

แผนภาพลิตเติลจิลแสดงโครงสร้างการทำงานของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วน ต่อประสานกรณืเพื่อความพึงพอใจ โดยนำเสนอตัวอย่างการนำเสนอแบบรูปในกรอบพร้อมทั้ง นำเสนอตัวอย่างรูปแบบอื่นๆ ได้แก่กรณืการแก้ไขนามสกุลภาพและตำแหน่งภาพดังภาพที่ 4



ภาพที่ 4.8 แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อความพึงพอใจของผู้ใช้



#### 4.3 แบบรูปลำดับที่ 2: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการปรับแต่งเพื่อการใช้งานเฉพาะ

การปรับปรุงส่วนต่อประสานกรณีเพื่อการใช้งานเฉพาะ เป็นการปรับปรุงส่วนหนึ่งส่วนใดของ ต่อประสานเกมเพื่อแสดงลักษณะเด่นบางสิ่งบางอย่างในช่วยเหลือแก่ผู้ใช้กลุ่มใดกลุ่มหนึ่งโดยเฉพาะ เพื่อให้ผู้เล่นเกมเกิดความเข้าใจ สะดวก คล่องตัว ในการใช้งานของผู้เล่นเกมเฉพาะกลุ่มนั้นๆ อนึ่งการ ปรับแต่งกรณีดังกล่าวอาจจะส่งผลกระทบต่อการใช้งานของผู้ใช้กลุ่มอื่นๆที่ได้ เช่นการเปลี่ยนสีของ อักษร การสีแสดงผลในเกมให้ต่างออกไปให้ผู้ใช้ที่ตาบอดสีสามารถอ่านหรือทำความเข้าใจได้ การนำ เลือดออกจากเกมทดแทนด้วยสีอื่นเพื่อให้เหมาะกับผู้เล่นเกมที่เป็นเด็กหรือการเพิ่มข้อความรูปภาพ แสดงยินดีเมื่อเล่นเกมชนะเพื่อให้ผู้ใช้ทราบถึงผลการเล่นเกม ซึ่งการเปลี่ยนแปลงที่กล่าวมาข้างต้นจะ ส่งผลต่อผู้ใช้ปกติทำให้ใช้งานยากกว่าปกติได้ เป็นต้น ดังตารางที่ 4.3

ตารางที่ 4.3 ตัวอย่างการปรับแต่งเพื่อการใช้งานเฉพาะ

| ลำดับ | การปรับแต่ง                     | รูปแบบการแก้ไข                       | จุดประสงค์การปรับแต่ง   | หมายเหตุ  |
|-------|---------------------------------|--------------------------------------|---|---|
| 1     | เปลี่ยนสีตัวอักษร               | เปลี่ยนเป็นสีเหลืองแทนน้ำเงิน        | เพื่อช่วยเหลือคนตาบอดสี                                       |   |
| 2     | เพื่อข้อความบอกเตรียมตัว        | เพิ่มข้อความ “เตรียมพร้อม”           | เพื่อให้ผู้เล่นที่ไม่เข้าใจสีเกมแข่งรถทราบว่า จะต้องทำอะไรต่อ | เกมแข่งรถบางประเภทใช้สีน้ำเงินแทนการเริ่มแข่งแทนสีเขียว |
| 3     | ข้อความแจ้งขณะกำลังโหลด         | เพิ่มข้อความ “กำลังโหลด”             | เพื่อแจ้งให้ทราบสถานะการทำงานของระบบ                          | บางครั้งเกมจะจอต่าไปเฉยๆไม่ทราบผลการทำงานเป็นอย่างไร    |
| 4     | เปลี่ยนสีเอฟเฟกต์ของเกม         | เปลี่ยนเป็นสีแดงแทนสีเขียว           | เพื่อให้มองเห็นง่ายขึ้นในพื้นที่ป่าสีเขียวในเกม               | พื้นที่สีเขียวจะกลืนกับเอฟเฟกต์มองเห็นยาก               |
| 5     | เพิ่มข้อความแจ้งเมื่อเล่นเกมชนะ | เพิ่มข้อความ “ยินดีดีด้วยคุณชนะแล้ว” | เพื่อแจ้งให้ทราบสถานะ ผลการเล่นของผู้เล่น                     |   |

**จุดประสงค์** เพื่อจัดการปัญหาจากการมอด โดยการเพิ่มส่วนต่อประสานใหม่เช่น การเพิ่มข้อความรูปภาพแสดงยินดีเมื่อเล่นเกมชนะ เป็นต้น ให้สามารถนำเสนอออกมาได้และไม่เกิดข้อผิดพลาด กรณีที่ไม่สามารถแสดงผลได้จะยกเลิกการปรับแต่งนั้นๆเพื่อให้เกมสามารถดำเนินต่อไปได้

**ปัญหา** การมอดเกมโดยการเพิ่มอ็อบเจกต์เกมเข้าไป ตัวอย่างเช่น การเพิ่มข้อความรูปภาพแสดงยินดี แต่ทำให้เกิดปัญหาไม่สามารถแสดงผลได้เนื่องจากอ็อบเจกต์ที่เพิ่มมีการกำหนดระดับชั้นของภาพ (layer) ไม่ถูกต้องจนผลที่นำเสนอถูกบดบังจากชั้นของภาพที่อยู่ชั้นบนกว่า ปัญหาเกิดจากกรณีที่มีการปรับแต่งรูปภาพนั้นผู้ปรับแต่งไม่ได้ทราบถึงเงื่อนไขที่ต้องพิจารณาดังกล่าว วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดอ็อบเจกต์เกมที่ต้องการสร้างเพิ่มเติมเพื่อจัดการสิ่งผิดปกติที่จะเกิดขึ้น ตัวอย่างคือการเพิ่มข้อความรูปภาพแสดงยินดี
2. กำหนดเงื่อนไขของการเพิ่มอ็อบเจกต์ในเกม เพื่อใช้เป็นรายการตรวจสอบปัญหา เช่น การเพิ่มรูปภาพใหม่ในเกมเงื่อนไขคือชั้นของอ็อบเจกต์ภาพ เป็นต้น
3. กำหนดวิธีการแนวทางการแก้ปัญหาโดยอ้างอิงจากเงื่อนไขที่สร้างไว้ เช่น กำหนดให้มีการย้ายชั้นของอ็อบเจกต์ภาพให้มาอยู่ด้านบน ทีละระดับชั้นจนสามารถอยู่เหนือชั้นภาพที่บังอ็อบเจกต์ภาพอยู่ เป็นต้น
4. กำหนดการแสดงผลและยกเลิกการทำงานเมื่อมีข้อผิดพลาดจากการปรับเปลี่ยนอ็อบเจกต์นอกเหนือจากที่กำหนดเพื่อให้ทราบถึง อ็อบเจกต์ใหม่ของเกมที่ต้องนำมาพิจารณากำหนดขอบเขตเพิ่มเติม

**การประยุกต์ใช้** แบบรูปทำหน้าที่จัดการสิ่งผิดปกติ เมื่อมีการแก้ไขค่ามีการเพิ่มอ็อบเจกต์ใหม่ โดยจัดการกรณีที่เกิดปัญหาจากการปรับแต่งให้เข้ากับเงื่อนไขขอบเขตของอ็อบเจกต์ของเกมนั้นๆ

**โครงสร้าง** มีโครงสร้างการทำงานแบ่งออกเป็น 5 ส่วนได้แก่

ส่วนบังคับใช้งานงาน

- **การจัดการสิ่งผิดปกติ (Exception)** กำหนดรายละเอียดของการแก้ไขเกมนั้นๆ มีอ็อบเจกต์ใดของเกมที่ต้องการแก้ปัญหา
- **การแก้ไขปัญหา (Solution)** กำหนดเงื่อนไขและขอบเขตของเงื่อนไขที่นำไปสู่ปัญหาที่ต้องแก้ไข เช่นคุณสมบัติของอ็อบเจกต์เกมที่สร้างขึ้นใหม่ คุณสมบัติใดที่สามารถทำให้เกิดปัญหาขึ้นได้

- **การยกเลิก (Reject)** นำเสนอแนวทางการยกเลิกการปรับแต่งเกมในแต่ละการปรับแต่งหรือในแต่ละอ็อบเจกต์เกม โดยมีคลาส การชดเชย (Compensate) อยู่ภายในทำหน้าที่นำเสนอแนวทางการยกเลิกการปรับแต่งอ็อบเจกต์เกมที่มีความสัมพันธ์กัน

ส่วนเลือกใช้งาน

- **แบบรูปแก้ไขแบบทันทีทันใด (Immediate fixed Pattern)** นำเสนอการแก้ไขแต่ละคุณสมบัติของ อ็อบเจกต์เกมที่สร้างขึ้นใหม่เพียงคุณสมบัติเดียวไม่มีความสัมพันธ์กับการแก้ไขอื่นๆ
- **แบบรูปแก้ไขแบบเชื่อมโยง (Defer fixed Pattern)** นำเสนอการแก้ไขในแต่ละคุณสมบัติของ อ็อบเจกต์ที่สร้างขึ้นใหม่ ซึ่งมีได้หลายคุณสมบัติอันเป็นเงื่อนไขความสัมพันธ์ซึ่งอาจนำไปสู่ความผิดปกติ

**วิธีการใช้** เป็นไปตามลำดับต่อไปนี้

1. ผู้ปรับแต่งกำหนดสิ่งที่ต้องการปรับแต่งในเกม ระบุความต้องการเพิ่มอ็อบเจกต์ใดในการปรับแต่ง พร้อมกำหนดคุณสมบัติ ขอบเขตคุณสมบัติที่ปรับแต่ง
2. กำหนดขอบเขตคุณสมบัติที่คาดว่าจะมีผลต่อการปรับแต่งลงในส่วนการแก้ไขปัญหา
3. กำหนดวิธีแนวทางเพื่อเป็นรายการแก้ไขคุณสมบัติลงในส่วนแก้ไขโดยเลือกใช้จากแบบรูป 2 กลุ่มได้แก่
  - 3.1 แบบรูปแก้ไขแบบทันทีทันใด ในกรณีที่มีการแก้ไขคุณสมบัติของอ็อบเจกต์เกมเป็นการแก้ไขคุณสมบัติแบบเดียวใช้เงื่อนไขเพียงเงื่อนไขเดียวในการพิจารณา
  - 3.2 แบบรูปแก้ไขแบบทันทีทันใด ในกรณีที่มีการแก้ไขคุณสมบัติของอ็อบเจกต์เกมเป็นการแก้ไขคุณสมบัติแบบหลายเงื่อนไขในการพิจารณา
4. กำหนดแนวทางการยกเลิกการทำงาน โดยทั่วไปจะออกแบบให้ยกเลิกการทำงานได้ 2 กลุ่มขึ้นกับเงื่อนไขของการปรับแต่งนั้นๆดังนี้
  - 4.1 การยกเลิกการทำงานโดยแสดงข้อความแจ้งเตือนให้ผู้ปรับแต่งทราบถึงข้อผิดพลาดใหม่ๆที่ไม่ได้ออกแบบไว้
  - 4.2 การยกเลิกแบบชดเชย เพื่อชดเชยผลการทำงานกรณีที่มีความผิดพลาดนั้นๆมีความต่อเนื่องจากการแก้ไขอื่นๆ

แบบรูปจะช่วยลดข้อผิดพลาดจากการปรับแต่งอ็อบเจกต์ของเกมหรือเพิ่มอ็อบเจกต์ใหม่ลงในเกม แบบรูปจะช่วยบรรเทาความผิดพลาดจากการเพิ่มส่วนต่อประสานใหม่ที่มีคุณสมบัติทำให้ไม่

สามารถแสดงผลได้ โดยการเพิ่มกระบวนการแก้ไขข้อผิดพลาดตามเงื่อนไขที่ผู้ปรับแต่งกำหนด นอกจากนี้แบบรูปยังนำเสนอสาเหตุข้อผิดพลาดผิดพลาดใหม่ๆจากคุณสมบัติหรืออ็อบเจกต์ที่ไม่ได้กำหนดไว้ ณ ตอนต้นการออกแบบเพื่อให้ผู้ปรับแต่งสามารถนำมาปรับใช้เพิ่มเติมในแบบรูปได้ต่อไป

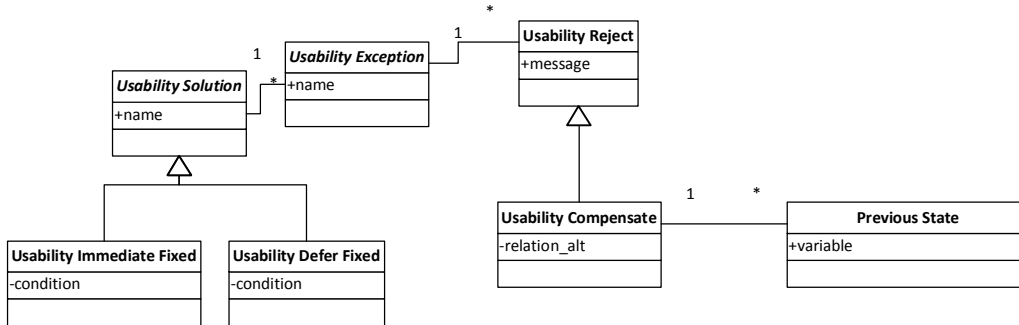
**ตัวอย่าง** การมอดส่วนต่อประสานโดยการเพิ่มข้อความแจ้งเตือนเพิ่มเติมกรณีที่สามารถกระทำใดๆ สำเร็จเช่นเล่นเกมชนะ จะต้องม็ข้อความแสดงความยินดี ดังภาพที่ 4.9



ภาพที่ 4.9 รูปภาพการเพิ่มข้อความแสดงความยินดี

1. กำหนดอ็อบเจกต์ของเกมที่จะเพิ่มเข้าไปใหม่ในส่วนจัดการสิ่งผิดปกติ ในกรณีตัวอย่างคือรูปภาพแสดงความยินดี
2. กำหนดคุณสมบัติและขอบเขตของคุณสมบัติของอ็อบเจกต์เกมที่ทำให้เกิดข้อผิดพลาดลงในส่วนการแก้ไขปัญหา กรณีตัวอย่างคือ ชั้นของรูปภาพ (layer) และระดับของชั้นของรูปภาพ (layer level) แสดงความยินดีในเกมอยู่ด้านหลังพื้นหลังทำให้ไม่สามารถแสดงผลได้
3. กำหนดการแก้ไขสิ่งผิดปกติเมื่อเกิดคุณสมบัติเป็นไปตามเงื่อนไขจากส่วนเงื่อนไข โดยการออกแบบกระบวนการใดๆเพื่อแก้ไขค่าคุณสมบัตินั้นลงในส่วนแก้ไขแบบทันทีทันใด กรณีตัวอย่าง แก้ไขข้อผิดพลาดโดยการปรับชั้นของรูปภาพแสดงความยินดี
4. กรณีที่ส่วนจัดการความผิดปกติสามารถทำงานได้ เมื่อเกมเริ่มทำงานและเกิดข้อผิดพลาดขึ้นแบบรูปจะแก้ไขการมอดโดยการปรับชั้นของรูปภาพให้ขยับมาด้านบนเรื่อยๆ จนกว่าจะสมบูรณ์ตามเงื่อนไขที่กำหนด
5. ในกรณีที่ไม่สามารถแก้ไขได้ หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อยกเลิกการปรับแต่งและแจ้งถึงข้อผิดพลาดที่เกิดขึ้น

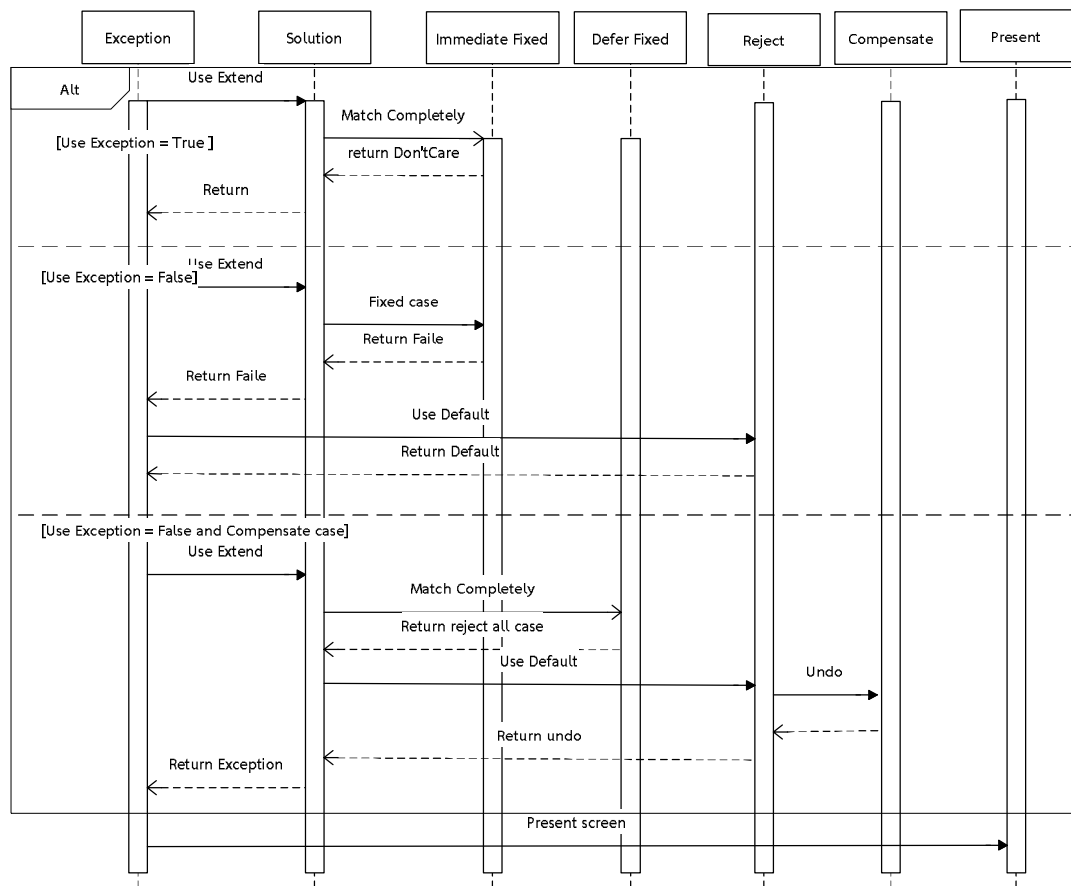
ดังภาพที่ 4.10 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสานกรณีเพื่อการใช้งานเฉพาะ



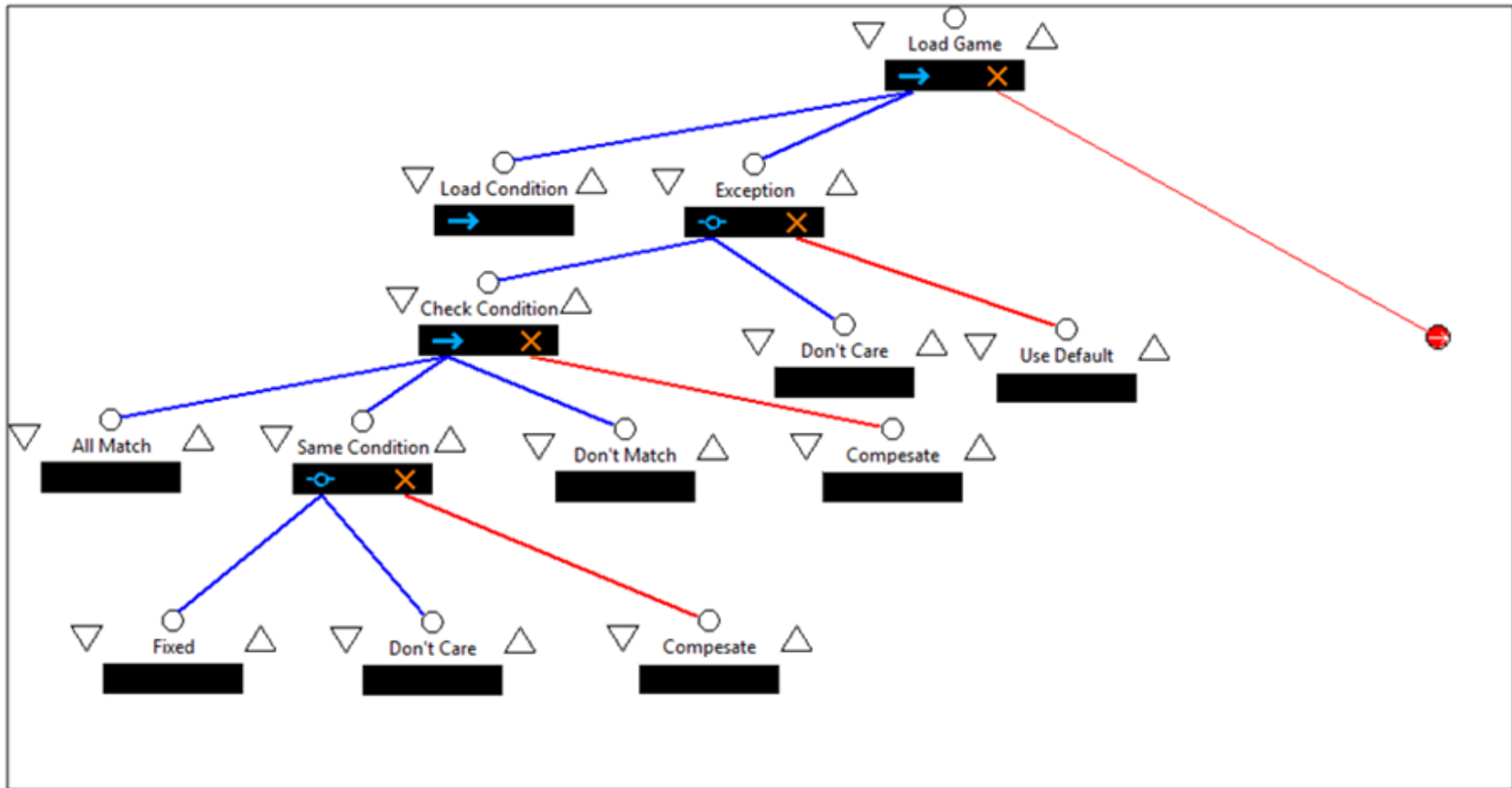
ภาพที่ 4.10 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่ง เพื่อการใช้งานเฉพาะ

การทำงานของคลาสเมื่อกรณีการปรับแต่งนั้นๆสอดคล้องกับคลาสจัดการสิ่งผิดปกติเพื่อการใช้งานเฉพาะ (*Usability Exception*) คลาสดังกล่าวจะกำหนดกรณีการปรับแต่งนั้นๆว่าเป็นข้อบกพร่องใดที่มีการปรับแต่งและตรวจสอบขอบเขตของคุณสมบัติข้อบกพร่องเพื่อกำหนดเงื่อนไขและขอบเขตของเงื่อนไขที่นำไปสู่ปัญหาที่ต้องแก้ไข เช่นคุณสมบัติของข้อบกพร่องเกมที่สร้างขึ้นใหม่ คุณสมบัติใดที่สามารถทำให้เกิดปัญหาขึ้นได้การแก้ไขปัญหาเพื่อการใช้งานเฉพาะ (*Usability Solution*) โดยมีรูปแบบการแก้ไข 2 รูปแบบได้แก่ แก้ไขแบบทันทีทันใดเพื่อการใช้งานเฉพาะ (*Usability Immediate fixed*) เลือกใช้ในการแก้ไขคุณสมบัติของข้อบกพร่องเกมที่สร้างขึ้นใหม่เพียงคุณสมบัติเดียวไม่มีความสัมพันธ์กับการคุณสมบัติอื่นๆ และ แบบรูปแก้ไขแบบเชื่อมโยงเพื่อการใช้งานเฉพาะ (*Usability Defer fixed*) เลือกใช้ในการแก้ไขในคุณสมบัติของข้อบกพร่องที่สร้างขึ้นใหม่ซึ่งมีความสัมพันธ์หลายคุณสมบัติซึ่งมีความสัมพันธ์เป็นเงื่อนไขซึ่งอาจนำไปสู่ความผิดปกติ

กรณีในส่วนจัดการสิ่งผิดปกติทั้ง 2 กรณีไม่สามารถจัดการกับสิ่งผิดปกติได้แบบรูปจะเข้าสู่ ส่วนยกเลิกการทำงานเพื่อการใช้งานเฉพาะ (*Usability Reject*) เพื่อไม่ให้ระบบแสดงข้อผิดพลาดโดยการปรับใช้ค่าปริยายหรือแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้น โดยในกรณีที่ข้อผิดพลาดใดๆมีความสัมพันธ์กับคุณสมบัติอื่นๆที่ได้แก้ไขไปแล้วแบบรูปจะใช้ส่วนชดเชยเพื่อการใช้งานเฉพาะ (*Usability Compensate*) เพื่อทำการย้อนกลับการปรับแต่งก่อนหน้า (*Previous State*) ให้เหมาะสม โดยนำเสนอลำดับการทำงานดังภาพที่ 4.11 และ ภาพที่ 4.12 แผนภาพลิตเติลจิสแสดงโครงสร้างการทำงานของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อการใช้งานเฉพาะ ซึ่งนำเสนอรูปแบบการทำงานอื่นๆควบคู่ไปด้วย



ภาพที่ 4.11 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการ  
ปรับปรุงส่วนต่อประสาน กรณีเพื่อการใช้งานเฉพาะ



ภาพที่ 4.12 แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการใช้งานเฉพาะ

#### 4.4 แบบรูปลำดับที่ 3: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน

การปรับแต่งเพื่อเพิ่มฟังก์ชันงาน เป็นการปรับปรุงส่วนต่อประสานโดยการเพิ่มส่วนต่อประสานใหม่ที่มีฟังก์ชันการทำงานอยู่ภายใน ฟังก์ชันงานที่เพิ่มมานั้นมุ่งเน้นในการนำเสนอข้อมูลหรือ การปฏิสัมพันธ์กับผู้ใช้ เพื่อให้ผู้ใช้ทราบถึงการทำงานของเกม การกระทำ กิจกรรม ระยะเวลา หรือ สร้างความเข้าใจถึงการทำงานในขณะนั้นแก่ผู้ใช้ เนื่องจากเกมบางประเภทนั้นออกแบบมาสำหรับผู้ใช้ที่มีความเชี่ยวชาญเกมเท่านั้น เช่น เกมยิงมุมมองบุคคลที่หนึ่งเมื่อยิงปืนจนกระสุนหมดจะต้องเติมกระสุนแมกกาซีนใหม่ โดยไม่มีการแจ้งใดๆ จึงมีแนวคิดในการเพิ่มเสียงแจ้งให้ทราบว่าต้องทำอะไร เป็นต้น โดยแบบรูปจะให้ความสำคัญกับฟังก์ชันงานที่ทำหน้าที่สนับสนุนการทำงานของส่วนต่อประสานเป็นหลักเพื่อให้ส่วนต่อประสานนั้นนำเสนอข้อมูลที่ถูกต้องแก่ผู้ใช้ ซึ่งจะไม่ครอบคลุมถึง ฟังก์ชันที่เกี่ยวข้องกับการทำงานของระบบ ดังตารางที่ 4.4

ตารางที่ 4.4 ตัวอย่างการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน

| ลำดับ | การปรับแต่ง                        | รูปแบบการแก้ไข                                  | จุดประสงค์การปรับแต่ง                               | หมายเหตุ                             |
|-------|------------------------------------|---|---|--------------------------------------|
| 1     | เพิ่มตัวเลขบอกแต้มเมื่อได้แต้ม     | เพิ่มแต้มบอกเมื่อเล่นเกมบาสเก็ตบอลแล้วได้แต้ม   | เพื่อให้ทราบถึงแต้มที่ได้ ณ ลูกยิงนั้นๆ             | ปกติต้องดูจากกระดานคะแนนเอาเอง       |
| 2     | เพื่อข้อความยืนยันขณะขายของ        | เพิ่มหน้าต่างยืนยันอีกครั้งหนึ่งขณะขายของ       | ป้องกันกรณีที่ผู้เล่นขายของผิดขึ้น                  | ในเกมสวมบทบาทจะมี การซื้อขายของในเกม |
| 3     | เพิ่มบาร์โหลดขณะกำลังโหลดเกม       | เพิ่มบาร์แสดงสถานะการทำงาน                      | เพื่อให้ผู้เล่นทราบถึงเวลาในการคอย                  |                                      |
| 4     | เพิ่มตัวเลขบอกความเสียหาย          | เมื่อโจมตีศัตรูจะมีตัวเลขบอกความเสียหายที่ทำได้ | เพื่อให้ผู้ใช้ทราบถึงความสามารถในการโจมตีของตัวละคร |                                      |
| 5     | เพิ่มตัวเลขบอกค่าพลังชีวิตของศัตรู | เพิ่มตัวเลขบอกค่าพลังชีวิตของศัตรู              | เพื่อให้ผู้เล่นวางแผนการทำงานได้                    | ต่อเนื่องจากตัวอย่างลำดับที่ 5       |



**จุดประสงค์** เพื่อจัดการปัญหาจากการมอด โดยการเพิ่มส่วนต่อประสานสำหรับการนำเสนอข้อมูลแก่ผู้ใช้โดยจะจัดการให้การปรับแต่งดังกล่าวไม่แสดงข้อมูลที่ผิดพลาด หรือสร้างความเสียหายแก่ระบบ

**ปัญหา** การมอดเกมโดยการเพิ่มส่วนต่อประสานใหม่ ทำให้เกิดปัญหาส่วนต่อประสานที่เพิ่มขึ้นใหม่ นำเสนอข้อมูลได้ไม่ถูกต้อง ตัวอย่างเช่น เกมบาสเก็ตบอลเวลาได้แต้มเราจะไม่ทราบถึงแต้มที่ได้ถ้าจดจำแต้มรวมก่อนหน้าไว้ จึงมีแนวโน้มในการทำให้เวลาลูกบาสเก็ตบอลลงห่วงมีเลขแต้มลอยขึ้นมา โดยเลขแต้มมีขอบเขตอยู่ที่ 1-3 แต้มเท่านั้น ผลการมอดปรากฏว่า เลขแต้มกลับเลื่อนลงได้แป้น และแสดงค่าติดลบ ซึ่งเป็นตัวเลขที่ผิดพลาด ซึ่งเป็นการนำเสนอที่ผกผันจากที่ผู้ปรับแต่งต้องการ หรือการนำเสนอค่าติดลบในตัวเลขค่าความแต้ม ทั้งนี้สาเหตุของปัญหามาจากฟังก์ชันควบคุมส่วนต่อประสานที่ผิดพลาด เช่นการให้ค่าตัวแปรผิด เป็นต้น

วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดอ็อบเจกต์ของเกมที่ต้องการแก้ไขหรือสร้างเพิ่มเติมเพื่อจัดการสิ่งผิดปกติที่จะเกิดขึ้น
2. กำหนดฟังก์ชันเป้าหมายซึ่งควบคุมการทำงานของอ็อบเจกต์เกมที่สร้างขึ้น เช่น ฟังก์ชันการเคลื่อนที่ของตัวอักษร
3. กำหนดเงื่อนไขการทำงานของฟังก์ชันที่ทำให้เกิดข้อผิดพลาด เช่น การกำหนดค่าทิศทางการเคลื่อนที่ของตัวอักษรที่ไม่พึงประสงค์ เป็นต้น หรือขอบเขตค่าของข้อมูลที่ระบบควรนำเสนอตามจุดประสงค์การปรับแต่งนั้นๆ เช่น เกมบาสเก็ตบอลมีแต้มได้ 3 ประเภท คือ 1 2 และ 3 แต้มเท่านั้น
4. กำหนดแนวทางการแก้ปัญหาของฟังก์ชันอ้างอิงตามเงื่อนไขการทำงานที่ทำให้เกิดข้อผิดพลาด
5. กำหนดการแสดงผลและยกเลิกการทำงานเมื่อมีข้อผิดพลาดจากทำงานของฟังก์ชัน นอกเหนือจากที่กำหนดเพื่อให้ทราบถึงกรณีการทำงานของฟังก์ชันใหม่ของเกมที่ต้องนำมาพิจารณาในการกำหนดขอบเขตเพิ่มเติม

**การประยุกต์ใช้** แบบรูปทำหน้าที่จัดการสิ่งผิดปกติ จากการเพิ่มส่วนต่อประสานใหม่ซึ่งมีฟังก์ชันงานเป็นองค์ประกอบ โดยมุ่งเน้นในการควบคุมความถูกต้องในการนำเสนอตามจุดประสงค์ของผู้ปรับแต่งและความถูกต้องของข้อมูลที่นำเสนอ ไม่ให้นำเสนอข้อมูลผิดพลาดและส่งกระทบต่อระบบ โดยมีองค์ประกอบการนำเสนอต่อยอดมาจาก แบบรูปเพื่อการใช้งานเฉพาะหรือแบบรูปลำดับที่ 2 สร้างเงื่อนไขของฟังก์ชันที่ทำให้เกิดข้อผิดพลาดเพิ่ม โดยอ้างอิงจากแบบรูปเพื่อการใช้งานเฉพาะโดยมุ่งในการควบคุมการทำงานของฟังก์ชันแทนอ็อบเจกต์เกม

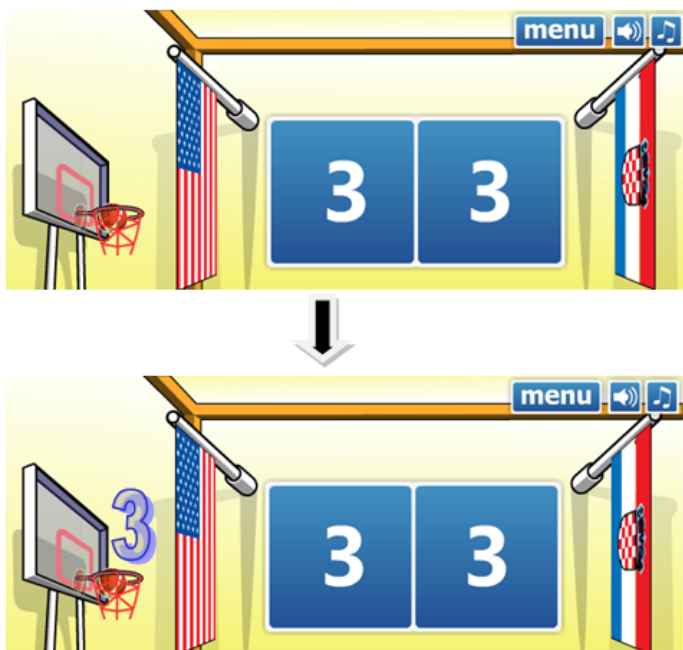
**โครงสร้าง** มีองค์ประกอบการทำงานที่ต่อยอดจากแบบรูปเพื่อการใช้งานเฉพาะ โดยเพิ่มส่วนจัดการสิ่งผิดปกติย่อย(Sub Exception) เพื่อคัดกรองผลการทำงานของฟังก์ชันงานภายในส่วนต่อประสาน โดยนำโครงสร้างภายในของจัดการสิ่งผิดปกติเพื่อการใช้งานเฉพาะทั้งหมด เพียงแต่มุ่งเน้นในส่วนการทำงานของฟังก์ชันและทำงานต่อยอดจากส่วนจัดการสิ่งผิดปกติของแบบรูปเพื่อการใช้งานเฉพาะ

**วิธีการใช้** เป็นไปตามลำดับต่อไปนี้

1. ผู้ปรับแต่งกำหนดสิ่งที่ต้องการปรับแต่งในเกม ระบุความต้องการเพิ่มอ็อบเจกต์ใดในการปรับแต่ง
2. ผู้ปรับแต่งระบุฟังก์ชันที่ถูกเพิ่มมาพร้อมอ็อบเจกต์เกมใหม่
3. พร้อมกำหนดฟังก์ชัน ขอบเขตของการทำงานฟังก์ชันที่คาดว่าจะมีผลการทำงานที่ผิดปกติลงในส่วนการแก้ไข
4. กำหนดวิธีแนวทางเพื่อเป็นรายการแก้ไขฟังก์ชันลงในส่วนแก้ไขโดยเลือกใช้จากแบบรูป 2 กลุ่มได้แก่
  - 3.1 แบบรูปแก้ไขแบบทันทีทันใด ในกรณีที่มีการแก้ไขฟังก์ชันของอ็อบเจกต์เกมเป็นการแก้ไขคุณสมบัติแบบเดี่ยวใช้เงื่อนไขเพียงเงื่อนไขเดียวในการพิจารณา
  - 3.2 แบบรูปแก้ไขแบบทันทีทันใด ในกรณีที่มีการแก้ไขฟังก์ชันของอ็อบเจกต์เกมเป็นการแก้ไขฟังก์ชันแบบใช้หลายเงื่อนไขในการพิจารณา
5. กำหนดแนวทางการยกเลิกการทำงาน โดยทั่วไปจะออกแบบให้ยกเลิกการทำงานได้ 2 กลุ่มขึ้นกับเงื่อนไขของการปรับแต่งนั้นๆดังนี้
  - 4.3 การยกเลิกการทำงานโดยแสดงข้อความแจ้งเตือนเพื่อให้ผู้ปรับแต่งทราบถึงข้อผิดพลาดใหม่ๆที่ไม่ได้ออกแบบไว้
  - 4.4 การยกเลิกแบบขดเชย เพื่อขดเชยผลการทำงานกรณีที่มีความผิดพลาดนั้นๆมีความต่อเนื่องจากการแก้ไขอื่นๆ

แบบรูปจะช่วยลดข้อผิดพลาดจากการเพิ่มอ็อบเจกต์ใหม่ลงในเกม แบบรูปจะช่วยบรรเทาความผิดพลาดจากกรณีการเพิ่มส่วนต่อประสานใหม่อันมีฟังก์ชันการทำงานที่ทำให้ไม่สามารถแสดงผลได้ โดยการเพิ่มกระบวนการแก้ไขข้อผิดพลาดตามเงื่อนไขที่ผู้ปรับแต่งกำหนดเพื่อควบคุมความผิดพลาดของฟังก์ชันนั้น นอกจากนี้แบบรูปยังนำเสนอสาเหตุข้อผิดพลาดผิดพลาดใหม่ๆจากฟังก์ชันที่ไม่ได้กำหนดไว้ ณ ตอนต้นการออกแบบเพื่อให้ผู้ปรับแต่งสามารถนำมาปรับใช้เพิ่มเติมในแบบรูปได้ต่อไป

**ตัวอย่าง** การมอดส่วนต่อประสานโดยการเพิ่มหน้าต่างแจ้งให้ทราบถึงสิ่งที่เกิดขึ้น เช่น เล่นเกมบาสเก็ตบอลได้รับแต้มเท่าไร ในระบบเดิมไม่มีการแสดงผลให้ชัดเจน ค่าตัวเลขที่ได้ถูกบวกรวมคะแนนรวมกับแต้มเดิมไป ดังนั้นจึงเกิดแนวความคิดการแจ้งตัวเลขในรูปแบบของกราฟิกข้อความแจ้งให้ทราบ ดังภาพที่ 4.13



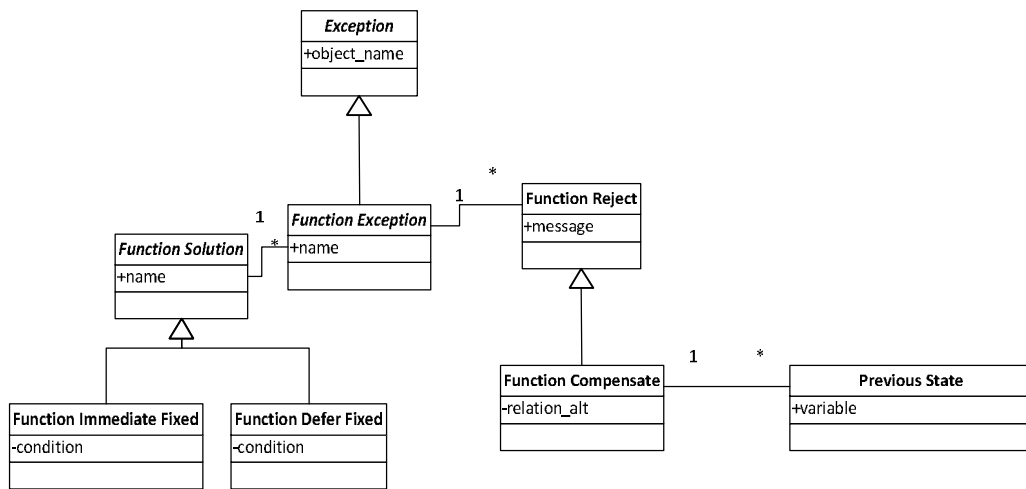
ภาพที่ 4.13 รูปภาพการเพิ่มข้อความแจ้งตัวเลขเพิ่มเติม เพื่อแสดงแต้มที่รับ

1. กำหนดอ็อบเจกต์ของเกมที่จะเพิ่มเข้าไปใหม่ ในกรณีตัวอย่างคือ รูปภาพเลขแต้ม
2. กำหนดฟังก์ชันการทำงานของอ็อบเจกต์ของเกมที่จะเพิ่มเข้าไปใหม่ลงในส่วนจัดการสิ่งผิดปกติย่อย ในกรณีตัวอย่างคือ ค่าที่แสดงในรูปแบบเลขแต้ม
3. กำหนดเงื่อนไขฟังก์ชันการทำงานและขอบเขตการทำงานที่ทำให้เกิดข้อผิดพลาดลงในส่วนแก้ไขของ ส่วนจัดการสิ่งผิดปกติย่อย ในกรณีตัวอย่างคือขอบเขตตัวเลขแต้มของเกมบาสเก็ตบอลคือ 1 ถึง 3 แต้ม
4. กำหนดการแก้ไขสิ่งผิดปกติเมื่อฟังก์ชันทำงานเป็นไปตามเงื่อนไข โดยการออกแบบกระบวนการใดๆเพื่อแก้ไขการทำงาน กรณีตัวอย่าง แก้ไขการตั้งค่าแต้มมาแสดงผิด โดยการคำนวณแต้มจากกระดานคะแนนเดิมของระบบ

5. เมื่อเกมเริ่มทำงาน เกิดข้อผิดพลาด แบบรูปจะแก้ไขข้อผิดพลาดที่เกิดขึ้นได้ตามเงื่อนไขและวิธีแก้ไขที่ออกแบบไว้ โดยกรณีตัวอย่างใช้ส่วนแก้ไขแบบทันทีทันใด แก้ไขข้อผิดพลาดโดยการคำนวณแต้มที่ได้จากค่าตัวเลขที่เพิ่มขึ้นในกระดานคะแนนรวม
6. ในกรณีที่ไม่สามารถแก้ไขได้หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น และยกเลิกการนำเสนอส่วนต่อประสานนั้นๆ

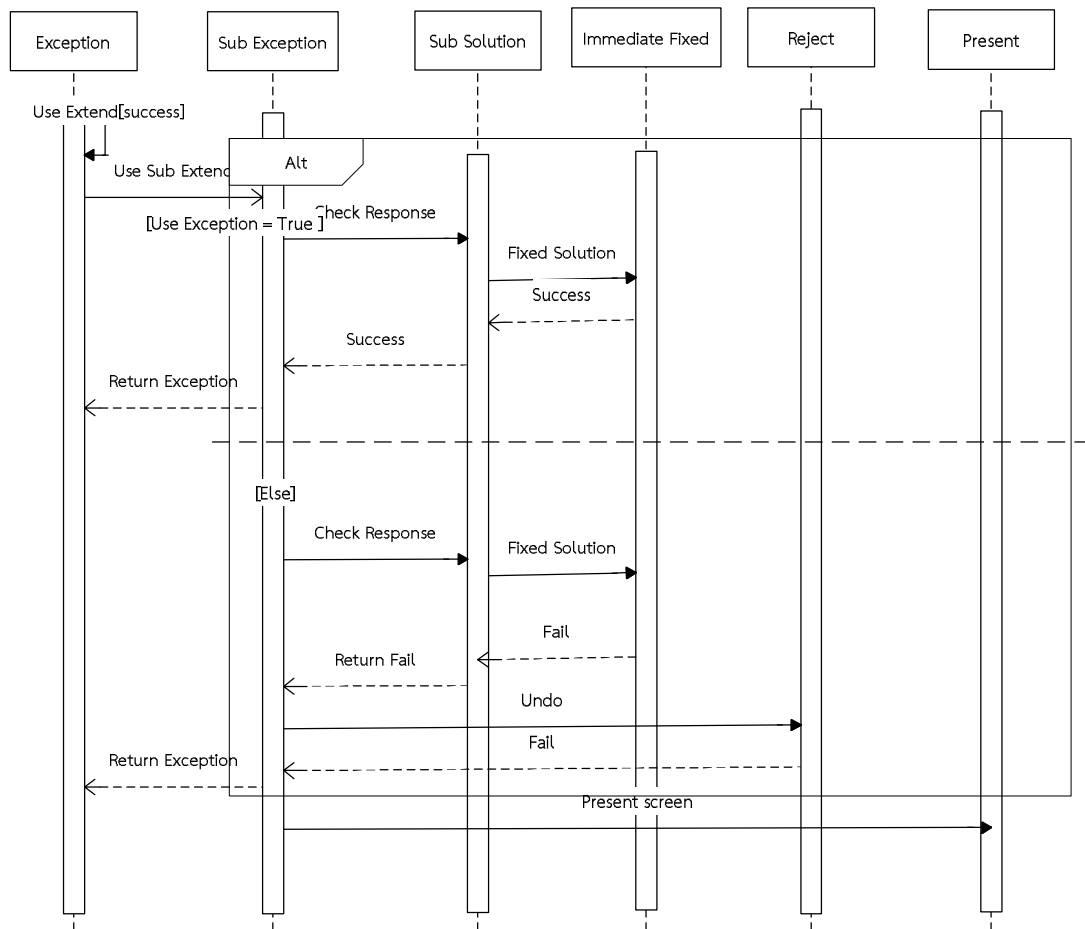
ดังภาพที่ 4.14 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีจัดการสิ่งผิดปกติ (Exception) คลาสดังกล่าวจะกำหนดกรณีการปรับแตงนั้นๆ ว่าเป็นอ็อบเจกต์ใดที่มีการปรับแตงเพื่อส่งผลต่อสู่คลาสจัดการสิ่งผิดปกติฟังก์ชันงาน (Function Exception) เพื่อกำหนดฟังก์ชันการทำงานที่ใช้งานร่วมกับอ็อบเจกต์เกมที่เพิ่มขึ้นมากำหนดเงื่อนไขและขอบเขตของเงื่อนไขของฟังก์ชันที่นำไปสู่ปัญหาที่ต้องแก้ไขลงใน การแก้ไขปัญหาเพื่อฟังก์ชันงาน (Function Solution) โดยมีรูปแบบการแก้ไข 2 รูปแบบได้แก่ แก้ไขแบบทันทีทันใดเพื่อฟังก์ชันงาน (Function Immediate fixed) เลือกใช้ในการแก้ไขการทำงานของฟังก์ชันในอ็อบเจกต์เกมที่สร้างขึ้นใหม่เพียงโดยมีเงื่อนไขการตรวจสอบที่เป็นเอกเทศไม่มีความสัมพันธ์กับการเงื่อนไขอื่นๆ และ แบบรูปแก้ไขแบบเชื่อมโยงฟังก์ชัน (Function Defer fixed) เลือกใช้ในการแก้ไขการทำงานของฟังก์ชันของอ็อบเจกต์ที่สร้างขึ้นใหม่โดยมีเงื่อนไขการตรวจสอบความสัมพันธ์หลากหลายเงื่อนไขที่นำไปสู่ความผิดปกติ

กรณีที่ส่วนจัดการสิ่งผิดปกติทั้ง 2 กรณีไม่สามารถจัดการกับสิ่งผิดปกติได้แบบรูปจะเข้าสู่ ส่วนยกเลิกการทำงานฟังก์ชัน (Function Reject) เพื่อไม่ให้ระบบแสดงข้อผิดพลาดโดยการปรับใช้ค่าปริยายหรือแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้น โดยในกรณีที่ข้อผิดพลาดใดๆมีความสัมพันธ์กับคุณสมบัติอื่นๆที่ได้แก้ไขไปแล้วแบบรูปจะใช้ส่วนชดเชยฟังก์ชัน (Function Compensate) เพื่อทำการย้อนกลับการปรับแตงก่อนหน้า (Previous State) ให้เหมาะสม

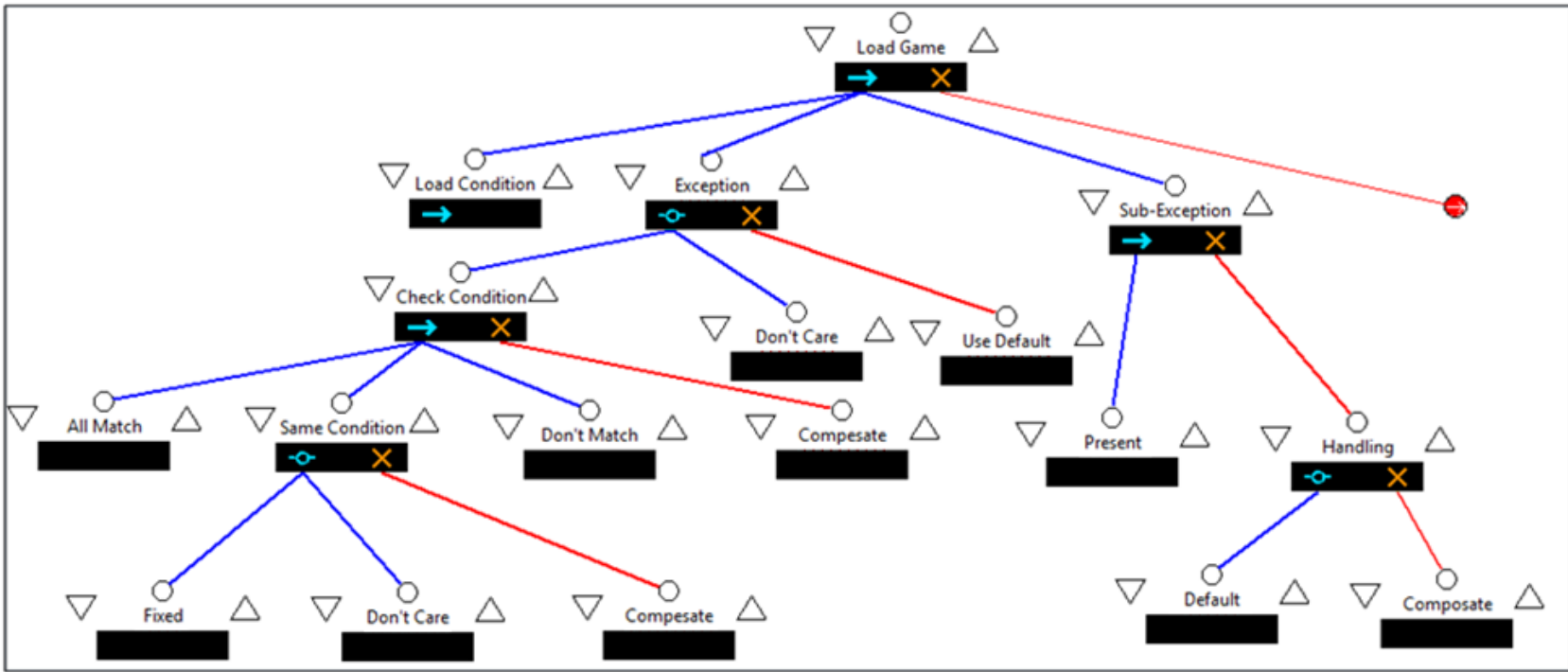


ภาพที่ 4.14 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่ง  
เพื่อการเพิ่มฟังก์ชันงาน

ภาพที่ 4.15 แผนภาพลำดับแสดงการทำงานของแบบรูป และ ภาพที่ 4.16 แผนภาพลิตเติล  
จิลแสดงโครงสร้างการทำงานของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อ  
การเพิ่มฟังก์ชันงาน



ภาพที่ 4.15 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการ  
ปรับปรุงส่วนต่อประสาน กรณีเพิ่มฟังก์ชันงาน



ภาพที่ 4.16 แผนภาพ ลิตเติล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน

#### 4.5 แบบรูปลำดับที่ 4: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการปรับแต่งเพื่อเพิ่มประสิทธิภาพ

กรณีการปรับปรุงส่วนต่อประสานกรณีเพื่อเพิ่มประสิทธิภาพ เป็นการปรับแต่งส่วนต่อประสานเพื่อลดปฏิสัมพันธ์ของผู้เล่นเกม เพื่อให้เกิดการทำงานที่รวดเร็วมีประสิทธิภาพ ผู้ใช้สามารถเล่นเกมได้อย่างรวดเร็วยิ่งขึ้น อนึ่งการปรับปรุงลักษณะอาจจะเหมาะกับผู้เล่นที่มีความเชี่ยวชาญ เนื่องจากการนำเสนอข้อมูลของระบบจะลดลงไป อาจจะทำให้พลาดข้อมูลบางอย่าง ทำให้ผู้ใช้ระดับเริ่มต้นที่ขาดความเข้าใจจะไม่สามารถทำความเข้าใจการทำงานได้ ตัวอย่างเช่น การซื้อขายของจากร้านค้าในเกม จะต้องเลือกสินค้าก่อนกดปุ่มซื้อ ในกรณีที่ซื้อสินค้าปริมาณมาก ทำให้เสียเวลาในการเล่นเกม แต่เมื่อปรับลดกิจกรรมในขั้นตอนการซื้อขายอาจจะทำให้ผู้เล่นเริ่มต้นสับสนได้ เป็นต้น ดังตารางที่ 4.5

ตารางที่ 4.5 ตัวอย่างการปรับแต่งเพื่อประสิทธิภาพ

| ลำดับ | การปรับแต่ง                           | รูปแบบการแก้ไข  | จุดประสงค์การปรับแต่ง                                | หมายเหตุ                                |
|-------|---------------------------------------|---|--|---|
| 1     | รวมปุ่มการซื้อขายให้เหลือปุ่มเดียว    | ปกติต้องกดเลือกและซื้อสินค้าทำการรวมปุ่มให้กดเลือกสินค้าเพียงปุ่มเดียวได้ | เพื่อให้ผู้เล่นดำเนินการซื้อขายสินค้าได้อย่างรวดเร็ว |   |
| 2     | ลดการกดปุ่มทำไม้ตาย                   | รวมการกดปุ่มเพื่อให้สามารถใช้ไม้ตายได้ด้วยปุ่มเดียว                       | เพื่อให้ผู้เล่นบางคนกดใช้ทำไม้ตายได้ง่ายขึ้น         | เกมต่อสู้ต้องกดทำไม้ตายด้วยปุ่มหลายปุ่ม |
| 3     | การรวมปุ่มใช้ทักษะหลายปุ่มเข้าด้วยกัน | การรวมการใช้ทักษะต่างๆเข้าด้วยกัน   | เพื่อความสะดวกแก่ผู้ใช้                              | ตัวละครมีทักษะหลากหลาย การจะกดครบยาก    |



**จุดประสงค์** เพื่อจัดการมอดในการลดกิจกรรม ขั้นตอน หรือกระบวนการที่ผู้เล่นกระทำกับระบบเกมในการทำงาน โดยการสร้างคำสั่งลัด เมื่อเกิดปัญหาขึ้นระบบสามารถตรวจจับถึงขั้นตอนที่มีปัญหา และแก้ไขให้สามารถประสานงานต่อไปได้

**ปัญหา** การมอดเกมโดยการรวมส่วนต่อประสาน เช่นปุ่มเข้าด้วยกัน เพื่อให้เกิดการง่ายในการเล่น เช่นการซื้อสินค้าในเกมต้องกดเลือกสินค้าและซื้อหลายขั้นตอน จึงนำไปสู่แนวคิดในการรวมปุ่มการทำงานให้สามารถซื้อสินค้าได้โดยการ ดับเบิ้ลคลิกที่สินค้าแทนการซื้อได้ทันที แต่กลับเกิดข้อผิดพลาดขึ้นเมื่อมีคำสั่งในขั้นตอนใดๆขาดหายไปหรือไม่ทำงานโดยที่ผู้ใช้ไม่ทราบถึงขั้นตอนนั้น ทำให้ไม่สามารถสานต่อการซื้อสินค้าจากเดิมได้ ทั้งนี้สาเหตุของปัญหามาจากการควบคุมคำสั่งการทำงานที่ผิดพลาด

วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดอ็อบเจกต์เกมเป้าหมายที่ต้องการใช้เป็นตัวรวมคำสั่ง โดยทั่วไปมักเป็นคำสั่งแรกของกระบวนการทำงานที่ต้องการรวม ในกรณีนี้คือ ตัวรายการสินค้า
2. กำหนดคำสั่งและลำดับคำสั่งที่ต้องการรวม
3. กำหนดการตรวจสอบความสมบูรณ์ในการทำงานของแต่ละคำสั่งที่นำมารวม
4. กำหนดเงื่อนไขกรณีที่เกิดข้อผิดพลาดขึ้นและการแก้ไข วิธีแนวทางในการแก้ปัญหาตามเงื่อนไขที่ออกแบบไว้
5. กำหนดการแสดงผลและยกเลิกการทำงานเมื่อมีข้อผิดพลาดจากทำงานของฟังก์ชัน นอกเหนือจากที่กำหนดโดยการยกเลิกการทำงานของคำสั่งการทำงานลัดทั้งหมด เพื่อให้ผู้ใช้สามารถดำเนินการเล่นเกมได้ตามปกติ

**การประยุกต์ใช้** แบบรูปทำหน้าที่จัดการสิ่งผิดปกติ จากกรณีการรวมส่วนต่อประสานเข้าด้วยกัน โดยมุ่งเน้นในการตรวจสอบกระบวนการทำงานของชุดคำสั่งในเกมที่นำมารวมเข้าด้วยกันเพื่อสร้างชุดคำสั่งทางลัดในการเล่น และตรวจสอบจำนวนความครบถ้วนของแต่ละคำสั่ง ทั้งยังรวมถึงการยกเลิกคำสั่งเมื่อเกิดปัญหาเพื่อให้ผู้เล่นสามารถชดเชยไปสานต่อคำสั่งที่เกิดปัญหาและเล่นเกมต่อไปได้ตามปกติ

**โครงสร้าง** ประกอบด้วยองค์ประกอบ 4 ส่วนมีรายละเอียดดังต่อไปนี้

ส่วนบังคับใช้งานงาน

- **เส้นทางการทำงานของปุ่ม (Button Flow)** กำหนดรายละเอียดการรวมคำสั่ง ลำดับคำสั่งที่รวม

- **การทำงานปุ่ม (Button Function)** กำหนดเงื่อนไขเพื่อตรวจสอบความสมบูรณ์ในการทำงานของแต่ละคำสั่งที่รวม
- **การจัดการสิ่งผิดปกติ (Exception)** กำหนดรายละเอียดเงื่อนไขกรณีที่เกิดปัญหาในการทำงานในแต่ละคำสั่ง

#### ส่วนเลือกใช้งาน

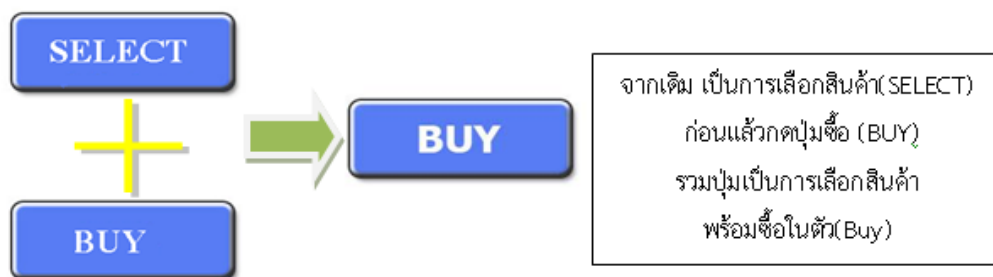
- **เงื่อนไขในการทำงานใหม่ (Exception-Driven Rework)** กำหนดการแก้ปัญหาข้อผิดพลาดที่ทำให้การรวมปุ่มไม่สามารถทำงานได้สมบูรณ์ โดยขดเซยการทำงานคำสั่งล่าสุดอีกครั้ง
- **ลองใหม่ (Retry)** นำเสนอแนวทางการยกเลิกการรวมคำสั่ง โดยการยกเลิกคำสั่งการทำงานของการรวมคำสั่งทั้งหมด เพื่อให้ผู้เล่นเริ่มทำงานใหม่อีกครั้ง

#### วิธีการใช้ เป็นไปตามลำดับต่อไปนี้

1. ผู้ปรับแต่งกำหนดคำสั่งที่จะรวมเป็นคำสั่งลัด ระบุความต้องการชุดคำสั่งที่จะรวมและลำดับการทำงาน
2. กำหนดเงื่อนไขการตรวจสอบความสมบูรณ์ของฟังก์ชัน
3. กำหนดการแก้ปัญหาการทำงานที่ผิดปกติโดยการทำงานคำสั่งเดิมซ้ำ หรือยกเลิกคำสั่งลัดเพื่อเริ่มต้นทำงานใหม่ทั้งหมดตามกระบวนการปกติ ตามความเหมาะสม

แบบรูปจะช่วยลดข้อผิดพลาดจากการสร้างคำสั่งลัดลงในเกม แบบรูปจะช่วยบรรเทาความผิดพลาดจากการการทำงานที่ไม่สมบูรณ์อันส่งผลต่อการแสดงผล โดยทำงานคำสั่งที่ไม่สมบูรณ์อีกครั้งหรือยกเลิกคำสั่งการทำงานลัดทั้งหมดเพื่อให้ผู้เล่นสามารถเล่นเกมได้ตามปกติ

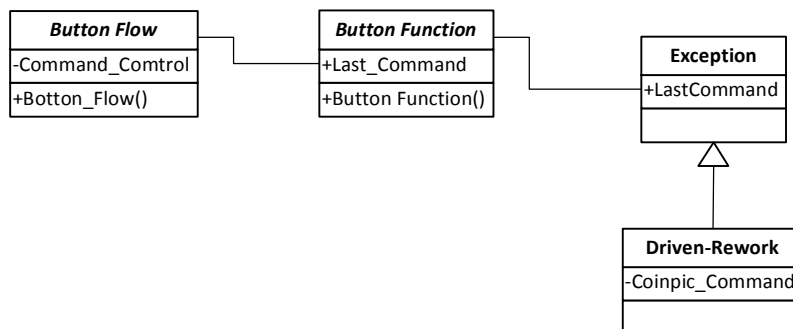
**ตัวอย่าง** การมอดส่วนต่อประสานโดยการสร้างปุ่มลัดสำหรับใช้งานหลายฟังก์ชันงานเพื่อลดกิจกรรมของผู้ใช้มีตัวอย่างดังภาพที่ 4.17 และมีลำดับการทำงานดังนี้



ภาพที่ 4.17 รูปภาพการรวมปุ่มการทำงาน

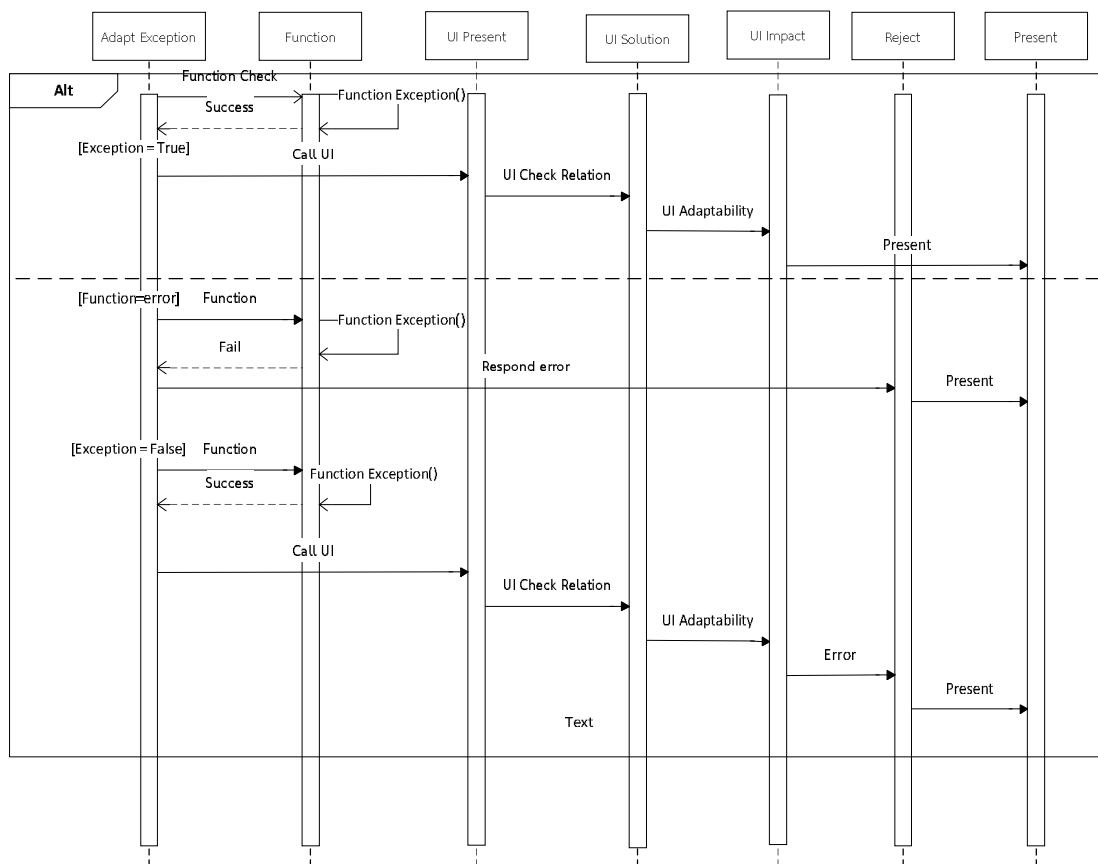
1. กำหนดชุดคำสั่งที่ต้องการทำคำสั่งลัด พร้อมกำหนดลำดับลงในส่วน เส้นทางการทำงานปุ่ม กรณีตัวอย่างคือ การเลือกสินค้า และ ซื้อสินค้า ตามลำดับ
2. กำหนดเงื่อนไขตรวจสอบความสมบูรณ์ของแต่ละคำสั่ง เช่นคำสั่งซื้อจะไม่ทำงานเมื่อเลือกสินค้าในเกมไม่ได้
3. กำหนดเงื่อนไขความผิดปกติเมื่อแต่ละคำสั่งไม่ทำงาน เช่น เลือกสินค้าได้แต่ซื้อไม่ได้ระบบจะค้าง เป็นต้นและสร้างกรณีแก้ไขโดยกำหนดการแก้ไขสิ่งผิดปกติโดยเลือกอย่างใดอย่างหนึ่งจาก 2 รูปแบบ
  - 3.1 เริ่มต้นคำสั่งลัดใหม่ โดยให้ทำงาน ณ คำสั่งที่เกิดปัญหาทันที
  - 3.2 ชดเชยการทำงานโดยยกเลิกคำสั่งลัดทั้งหมดเพื่อให้เกมสามารถดำเนินไปได้ปกติ

ดังภาพที่ 4.18 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสานกรณีเพื่อประสิทธิภาพ โดยเริ่มต้นการทำงานจากการตรวจสอบการทำงานของปุ่ม คำสั่งการทำงานที่รวมเป็นคำสั่งลัด ลำดับการทำงานผ่าน*เส้นทางการทำงานปุ่ม (Button Flow)* หลังจากนั้นสร้าง*ส่วนการทำงานปุ่ม (Button Function)* เพื่อตรวจสอบการทำงานของคำสั่งแต่ละ คำสั่ง และกำหนดเงื่อนไขการจัดการสิ่งผิดปกติที่เกิดขึ้นผ่าน *ส่วนจัดการสิ่งผิดปกติ (Efficiency Exception)* โดยกำหนดการแก้ไขและยกเลิกการทำงานเมื่อเกิดปัญหาเลือกจาก 2 รูปแบบได้แก่ *ส่วนเงื่อนไขการทำงานใหม่ (Exception-Driven Rework)* ทำหน้าที่จัดการสิ่งผิดปกติโดยชดเชย การทำงานคำสั่งล่าสุดอีกครั้งและ*ส่วนลองใหม่ (Retry)* ทำหน้าที่จัดการสิ่งผิดปกติโดยการยกเลิก คำสั่งการทำงานของการรวมคำสั่งทั้งหมดเพื่อให้ผู้เล่นเริ่มทำงานใหม่รูปแบบปกติใหม่อีกครั้ง

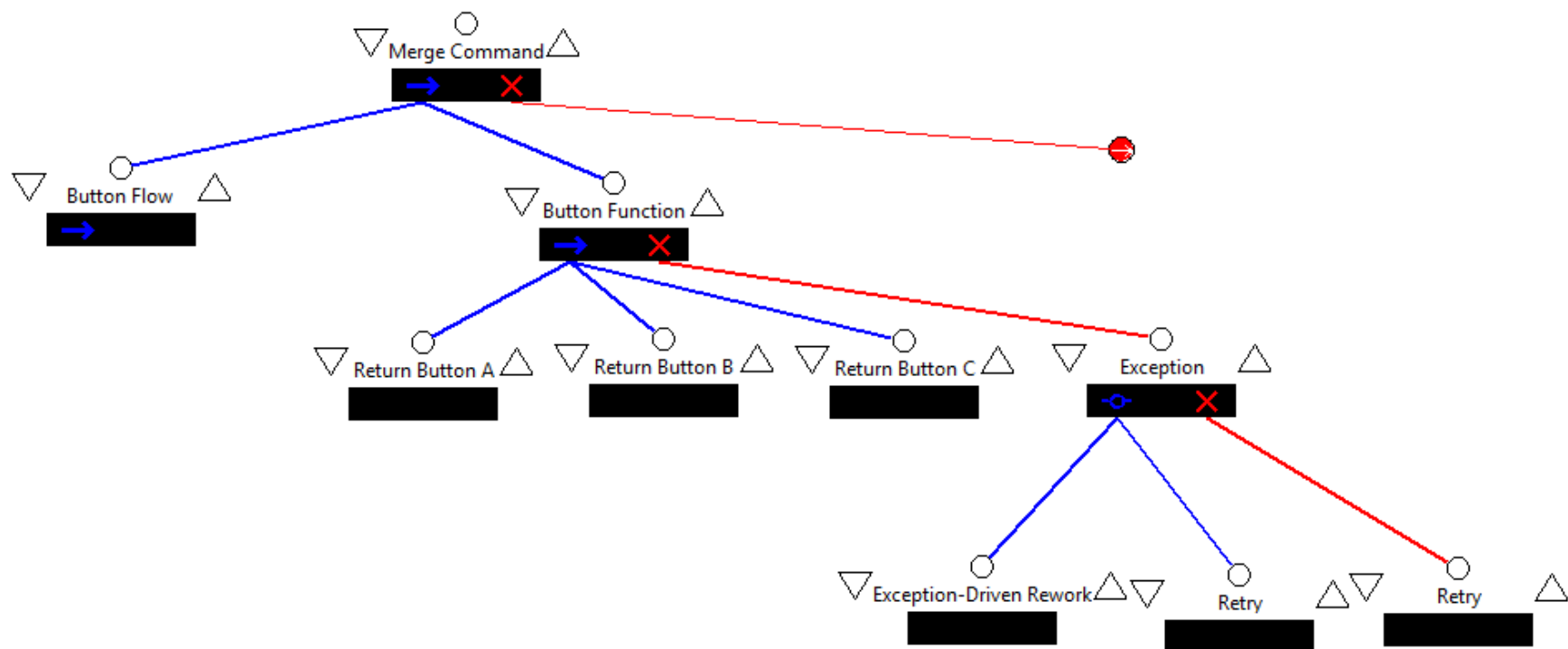


ภาพที่ 4.18 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่ง เพื่อประสิทธิภาพ

ภาพที่ 4.19 แผนภาพลำดับแสดงการทำงานของแบบรูปกรณีที่ฟังก์ชันงานปุมทำงาน สมบูรณ์แต่ไม่ครบตามลำดับ และภาพที่ 4.20 แผนภาพลิตเติลจิลแสดงโครงสร้างการทำงานของแบบ รูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อประสิทธิภาพ



ภาพที่ 4.19 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีเพื่อประสิทธิภาพ



ภาพที่ 4.20 แผนภาพ ลิตเติ้ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อเพิ่มประสิทธิภาพ

#### 4.6 แบบรูปลำดับที่ 5: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการปรับแต่งเพื่อการปรับตัว

การปรับแต่งส่วนต่อประสานกรณีเพื่อการปรับตัว เป็นการปรับแต่งส่วนต่อประสานเพื่อสนับสนุนการปรับแต่งประเภทในการเพิ่มฟังก์ชันงานในเกมพร้อมส่วนต่อประสาน ซึ่งแบบรูปช่วยในการสร้างส่วนต่อประสานในรูปแบบของอ็อบเจกต์เกม โดยการปรับอ็อบเจกต์เกมใหม่ที่มีลักษณะใกล้เคียงกับอ็อบเจกต์เกมเดิม สามารถปรับใช้คุณสมบัติร่วมกันได้ แบบรูปดังกล่าวออกแบบมาเพื่อรองรับการเพิ่มฟังก์ชันพร้อมกับส่วนต่อประสานเพื่อให้ระบบสามารถทำให้ระบบทำงานต่อไปได้ กรณีที่ฟังก์ชันดังกล่าวเกิดข้อผิดพลาดที่ส่งผลกระทบต่อระบบเกม แบบรูปจะทำหน้าที่จัดการปัญหาดังกล่าวตามเงื่อนไขที่กำหนดตัวอย่างเช่น เกมสวมบทบาทได้เพิ่มค่าสถานะชุ่มซำเข้าไปให้ตัวละครพร้อมหน้าต่างแสดงสถานะก็ต้องเพิ่มอ็อบเจกต์เกมสำหรับแสดงสถานะชุ่มซำด้วย ทั้งนี้หน้าต่างสถานะที่เพิ่มต้องอาศัยการตั้งค่าคุณสมบัติอ็อบเจกต์เกมหลากหลายส่วนจึงจะทำให้อ็อบเจกต์สถานะชุ่มซำออกมาสวยงามเข้ากับอ็อบเจกต์ของสถานะตัวละครอื่นๆ ดังตารางที่ 4.6

ตารางที่ 4.6 ตัวอย่างการปรับแต่งเพื่อการปรับตัว

| ลำดับ | การปรับแต่ง                        | รูปแบบการแก้ไข   | จุดประสงค์การปรับแต่ง                      | หมายเหตุ  |
|-------|------------------------------------|--|--|---|
| 1     | การเพิ่มสถานะใหม่ตัวละคร           | การเพิ่มสถานะใหม่ตัวละคร ต้องเพิ่มสถานะในหน้าต่างสถานะด้วย             | เพื่อให้ผู้ใช้ทราบถึงการมีอยู่ของสถานะนั้น |   |
| 2     | การเพิ่มช่องสล็อตในเกม สล็อตแมชชีน | การเพิ่มช่องสล็อตในเกมสล็อตแมชชีนต้องปรับให้สล็อตที่เพิ่มใช้งานได้ด้วย | เพื่อสร้างความท้าทายในการเล่น              |   |
| 3     | การเพิ่มทักษะใหม่ๆให้ตัวละคร       | การเพิ่มทักษะใหม่ๆให้ตัวละคร ต้องเพิ่มปุ่มการใช้ทักษะใหม่ให้ด้วย       | เพื่อสร้างความบันเทิงแก่ผู้เล่น            | เช่น เพิ่มทักษะ ดี่มายาซู กำลังให้ชาวนา เป็นต้น |

**จุดประสงค์** เพื่อจัดการการมอด ในการปรับใช้งานอ็อบเจกต์เกมร่วมกับอ็อบเจกต์เกมเดิมของระบบ เมื่อมีปัญหาที่เกิดขึ้นจะแก้ไขตามกรณีที่ออกแบบไว้ หรือแสดงข้อความแจ้งเตือนผู้ใช้ เพื่อให้สามารถปรับแก้ไขการปรับแต่งทำให้ระบบสามารถดำเนินการต่อไปได้

**ปัญหา** การมอดเกมโดยการเพิ่มฟังก์ชันใหม่ลงในเกมพร้อมส่วนต่อประสานที่มีลักษณะเดียวกับที่เกมมีอยู่ แต่กลับมีคุณสมบัติของอ็อบเจกต์เกมใหม่ที่ไม่สอดคล้องกับอ็อบเจกต์เกมเดิมทำให้ขาดความสวยงามและมีการดึงข้อมูลมานำเสนอผิดพลาดได้ ตัวอย่างเช่น การเพิ่มสถานะชุ่มซำมเข้าไปให้ตัวละครในเกมสวมบทบาท แต่หน้าต่างค่าสถานะกลับนำเสนอรูปแบบตัวอักษร ขนาดกรอบสถานะที่ไม่สอดคล้องกับค่าสถานะอื่นๆของเกม พร้อมทั้งนำเสนอข้อมูลที่ผิดพลาดโดยการนำข้อมูลจากค่าสถานะความโหดดีในเกมมานำเสนอ สาเหตุของปัญหาเกิดจากผู้ใช้ไม่ทราบคุณสมบัติของอ็อบเจกต์เกมประเภทเดียวกัน หรือการใช้งานที่มีฟังก์ชันการทำงานที่ผิดปรกติ

วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดสิ่งที่ต้องการปรับแต่ง ในกรณีนี้คือ การเพิ่มสถานะชุ่มซำมของตัวละคร
2. กำหนดเงื่อนไขการทำงานของฟังก์ชันที่ปรับแต่ง ในกรณีตัวอย่างคือ การจัดการค่าสถานะชุ่มซำม
3. กำหนดส่วนต่อประสานหรืออ็อบเจกต์เกมที่สร้างขึ้นมาควบคู่กับฟังก์ชันใหม่ ในกรณีตัวอย่างคือชุดอ็อบเจกต์เกม หน้าต่างสถานะ
4. กำหนดความสัมพันธ์ของอ็อบเจกต์เกมกับอ็อบเจกต์เกมที่เพิ่มขึ้น เมื่อมีความสัมพันธ์จะเกิดการปรับใช้คุณสมบัติอ็อบเจกต์เกมใหม่ไปใช้ร่วมคุณสมบัติของอ็อบเจกต์เกมเดิม
5. กำหนดเงื่อนไขการตรวจสอบการแสดงผลของอ็อบเจกต์เกมใหม่ว่ามีความผิดปรกติหรือไม่ พร้อมทั้งกำหนดขอบเขตของคุณลักษณะ
6. กำหนดการแก้ไขสิ่งผิดปรกติเมื่อฟังก์ชันทำงานเป็นไปตามเงื่อนไข โดยการออกแบบกระบวนการใดๆเพื่อแก้ไขการทำงาน กรณีตัวอย่าง แก้ไขการดึงค่าสถานะมาแสดงผล โดยอาจจะดึงข้อมูลค่าอื่นๆมาแสดงทำให้ผู้เล่นสับสน แก้ไขด้วยการดึงค่าสถานะจากตัวแปรที่ประกาศ
7. เมื่อเกมเริ่มทำงาน เกิดข้อผิดพลาด แบบรูปจะแก้ไขข้อผิดพลาดที่เกิดขึ้นตามเงื่อนไขและวิธีแก้ไขที่ออกแบบไว้
8. ในกรณีที่ไม่สามารถแก้ไขได้หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น และยกเลิกการนำเสนอส่วนต่อประสานนั้นๆ



**การประยุกต์ใช้** แบบรูปทำหน้าที่จัดการสิ่งผิดปกติ จากการเพิ่มฟังก์ชันใหม่พร้อมส่วนต่อประสาน โดยส่วนต่อประสานที่เพิ่มขึ้นเป็นอ็อบเจกต์เกมที่มีความสัมพันธ์กับอ็อบเจกต์เกมเดิม แบบรูปทำหน้าที่มุ่งเน้นในการปรับคุณสมบัติของอ็อบเจกต์เกมใหม่เข้ากับคุณสมบัติของอ็อบเจกต์เกมเดิมเพื่อ ความเกิดความสอดคล้อง พร้อมทั้งคัดกรองและควบคุมผลการทำงานที่ผิดพลาดของอ็อบเจกต์เกม หรือฟังก์ชันที่มีผลกระทบต่อระบบทำให้ระบบไม่สามารถทำงานได้อย่างสมบูรณ์ได้ เพื่อให้ระบบเกม สามารถดำเนินการทำงานต่อไปได้

**โครงสร้าง** ประกอบด้วยองค์ประกอบ 7 ส่วนมีรายละเอียดดังต่อไปนี้

- **การจัดการสิ่งผิดปกติ (Exception)** กำหนดรายละเอียดของปรับแต่งเกมนั้นฟังก์ชันของ เกมที่เพิ่มขึ้นจากการปรับแต่ง
- **ฟังก์ชัน (Function)** กำหนดรายละเอียดเงื่อนไขและการแก้ปัญหาฟังก์ชันที่เพิ่มมาใหม่
- **ส่วนนำเสนอส่วนต่อประสาน (UI Interface)** กำหนดรายละเอียดส่วนต่อประสาน อ็อบเจกต์เกมที่เพิ่มขึ้นมาใหม่
- **แก้ปัญหาส่วนต่อประสาน (UI Solution)** กำหนดความสัมพันธ์กับอ็อบเจกต์เกมเดิม เงื่อนไขการตรวจสอบ คุณสมบัติของอ็อบเจกต์เกมเดิมที่สามารถนำไปปรับใช้กับอ็อบเจกต์ เกมใหม่
- **ส่วนตรวจสอบผลกระทบส่วนต่อประสาน (UI Impact)** กำหนดเงื่อนไขตรวจสอบการ แสดงผลของ อ็อบเจกต์ที่ผิดปกติ จากคุณสมบัติของอ็อบเจกต์และค่าที่แสดงผล
- **การแก้ไข (fixed)** นำเสนอการแก้ไขในแต่ละคุณสมบัติของอ็อบเจกต์เกมที่สร้างขึ้นใหม่ รวมถึงค่าที่แสดงผลในอ็อบเจกต์เกม
- **การยกเลิก (Reject)** นำเสนอแนวทางการยกเลิกการปรับแต่งเกมในแต่ละการปรับแต่ง และ ในแต่ละอ็อบเจกต์เกม

ทั้งนี้กรณีออกแบบดังกล่าวใช้โครงสร้างการจัดการสิ่งผิดปกติจากระบบเดิมหรือกรณีการ ออกแบบก่อนหน้าในส่วนการจัดการสิ่งผิดปกติ โดยโครงสร้างสามารถประยุกต์ใช้รูปแบบเดิมที่อยู่ให้ สามารถทำงานร่วมกับการออกแบบใหม่ได้ กรณีที่ผลการทำงานเป็นข้อผิดพลาดใหม่ แบบรูปจะทำ หน้าที่แจ้งรูปแบบสิ่งผิดปกตินั้นๆ เพื่อให้ผู้พัฒนานำไปออกแบบเป็นกรณีทดสอบต่อไป

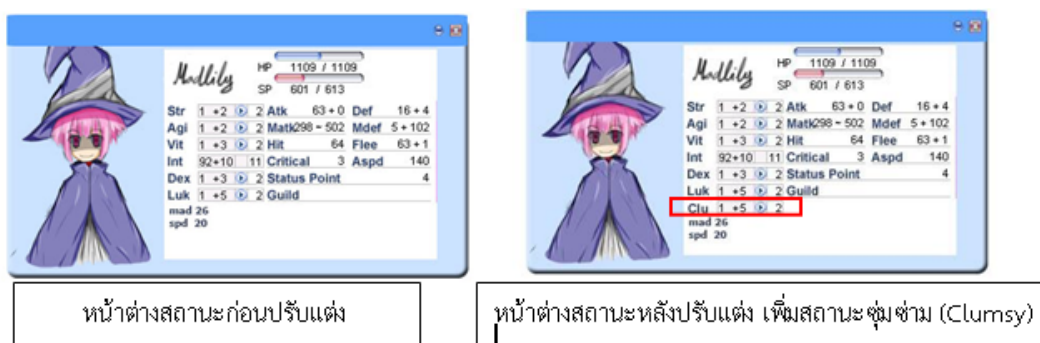
**วิธีการใช้** เป็นไปตามลำดับต่อไปนี้

1. ผู้ปรับแต่งกำหนดฟังก์ชันงานที่เพิ่มขึ้นใหม่
2. ระบุเงื่อนไขตรวจสอบการทำงานของฟังก์ชันใหม่

- กำหนดรายละเอียดส่วนต่อประสานที่เพิ่มขึ้นพร้อมกับฟังก์ชันใหม่
- ระบุเงื่อนไขการปรับใช้คุณสมบัติของอ็อบเจกต์เกมเดิมให้กับคุณสมบัติอ็อบเจกต์เกมใหม่ของส่วนต่อประสาน
- กำหนดขอบเขตคุณสมบัติของอ็อบเจกต์เกมหรือค่าในการแสดงผลที่คาดว่าจะมีผลการทำงานที่ผิดปกติลงในส่วนเงื่อนไข
- กำหนดรายการแก้ไขลงในส่วนแก้ไข
- กำหนดแนวทางการยกเลิกการทำงาน โดยแสดงข้อความแจ้งเตือนเพื่อให้ผู้ปรับแต่งทราบถึงข้อผิดพลาดใหม่ๆที่ไม่ได้ออกแบบไว้

แบบรูปจะช่วยลดข้อผิดพลาดจากการปรับใช้คุณสมบัติของเดิมของส่วนต่อประสานใหม่ อันมีขอบเขตของคุณสมบัติหรือค่าการทำงานที่ทำให้ไม่สามารถแสดงผลได้ โดยการเพิ่มกระบวนการแก้ไขข้อผิดพลาดตามเงื่อนไขที่ผู้ปรับแต่งกำหนด นอกจากนี้แบบรูปยังนำเสนอสาเหตุข้อผิดพลาดใหม่ๆจากฟังก์ชันที่ไม่ได้กำหนดไว้ ณ ตอนต้นการออกแบบเพื่อให้ผู้ปรับแต่งสามารถนำมาปรับใช้เพิ่มเติมในแบบรูปได้ต่อไป

ตัวอย่าง การปรับแต่งส่วนต่อประสานโดยการเพิ่มฟังก์ชันงานใหม่ๆเข้าให้แก่ระบบ ตัวอย่างเช่นการเพิ่มความสามารถใหม่ของตัวละคร หรือช่องแสดงผลเพิ่ม เมื่อระบบทำงาน แบบรูปจะทำการตรวจสอบการทำงานของฟังก์ชันงานว่าสามารถทำงานได้สมบูรณ์หรือไม่ ภายหลังตรวจสอบการทำงานของฟังก์ชันแล้ว จะเข้าสู่การทำงานของแบบรูปมีตัวอย่างดังภาพที่ 4.21 และมีลำดับการทำงานดังนี้



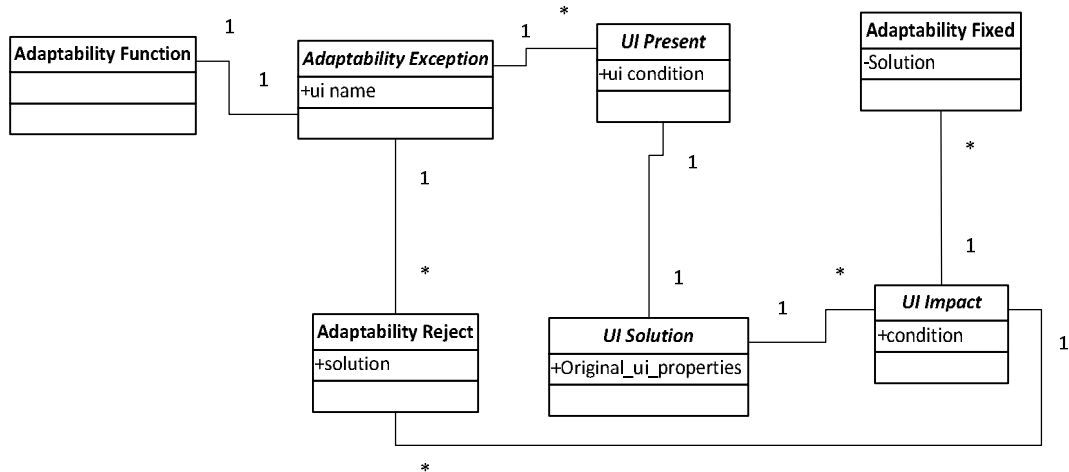
ภาพที่ 4.21 รูปภาพการปรับตัวจากฟังก์ชันการทำงานใหม่

- กำหนดฟังก์ชันที่เพิ่มจากการปรับแต่งลงในส่วนจัดการสิ่งผิดปกติ ในกรณีตัวอย่างคือฟังก์ชันสถานะชุ่มซำมของตัวละครในเกม

2. กำหนดเงื่อนไขในการจัดการสิ่งผิดปกติจากฟังก์ชันสถานะชุ่มซำมของตัวละครที่เพิ่มขึ้น ลงในส่วนฟังก์ชัน
3. กำหนดรายละเอียดส่วนต่อประสานที่เพิ่มขึ้นจากการเพิ่มฟังก์ชันสถานะชุ่มซำมของตัวละคร ลงในส่วนนำเสนอส่วนต่อประสาน ในกรณีตัวอย่างคือชุดอ็อบเจกต์เกมสถานะชุ่มซำม
4. กำหนดส่วนต่อประสานซึ่งมีความสัมพันธ์กับส่วนต่อประสานใหม่ที่เพิ่มขึ้นและปรับคุณสมบัติของอ็อบเจกต์เกมใหม่ให้ตรงกับอ็อบเจกต์เกมเดิมตามเงื่อนไข โดยทำงานในส่วนแก้ปัญหาส่วนต่อประสาน จากกรณีตัวอย่าง คือ การปรับคุณสมบัติของชุดอ็อบเจกต์สถานะชุ่มซำมเข้ากับคุณสมบัติของชุดอ็อบเจกต์ตัวละครในเกม ตามความสัมพันธ์ที่กำหนดไว้
5. กำหนดเงื่อนไขการตรวจสอบสิ่งผิดปกติของชุดอ็อบเจกต์เกมชุ่มซำม และค่าที่นำเสนอ ในกรณีตัวอย่างคือ การกำหนดขอบเขตของคุณสมบัติชุดอ็อบเจกต์สถานะชุ่มซำมให้มีขนาดเท่ากับคุณสมบัติชุดอ็อบเจกต์สถานะอื่นๆของตัวละคร โดยต้องไม่ให้ตำแหน่งซ้อนทับกัน และตรวจสอบค่าที่แสดงผลไม่ซ้ำซ้อนกับค่าสถานะอื่นๆของเกม
6. กำหนดการแก้ไขสิ่งผิดปกติเมื่อชุดอ็อบเจกต์เกมชุ่มซำมเป็นไปตามเงื่อนไข โดยการออกแบบกระบวนการใดๆเพื่อแก้ไขการทำงาน กรณีตัวอย่าง ตำแหน่งของชุดอ็อบเจกต์สถานะชุ่มซำมซ้อนทับกับชุดอ็อบเจกต์สถานะโศคติซึ่งคือสถานะที่อยู่ใกล้เคียง แบบรูปจะปรับแก้ตำแหน่งให้ออกห่างจากชุดอ็อบเจกต์สถานะโศคติเพื่อให้เกิดความเหมาะสม
7. เมื่อเกมเริ่มทำงาน เกิดข้อผิดพลาด แบบรูปสามารถแก้ไขข้อผิดพลาดที่เกิดขึ้นตามเงื่อนไข และวิธีแก้ไขที่ออกแบบไว้
8. ในกรณีที่แบบรูปไม่สามารถแก้ไขได้หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น และยกเลิกการนำเสนอส่วนต่อประสานนั้นๆ

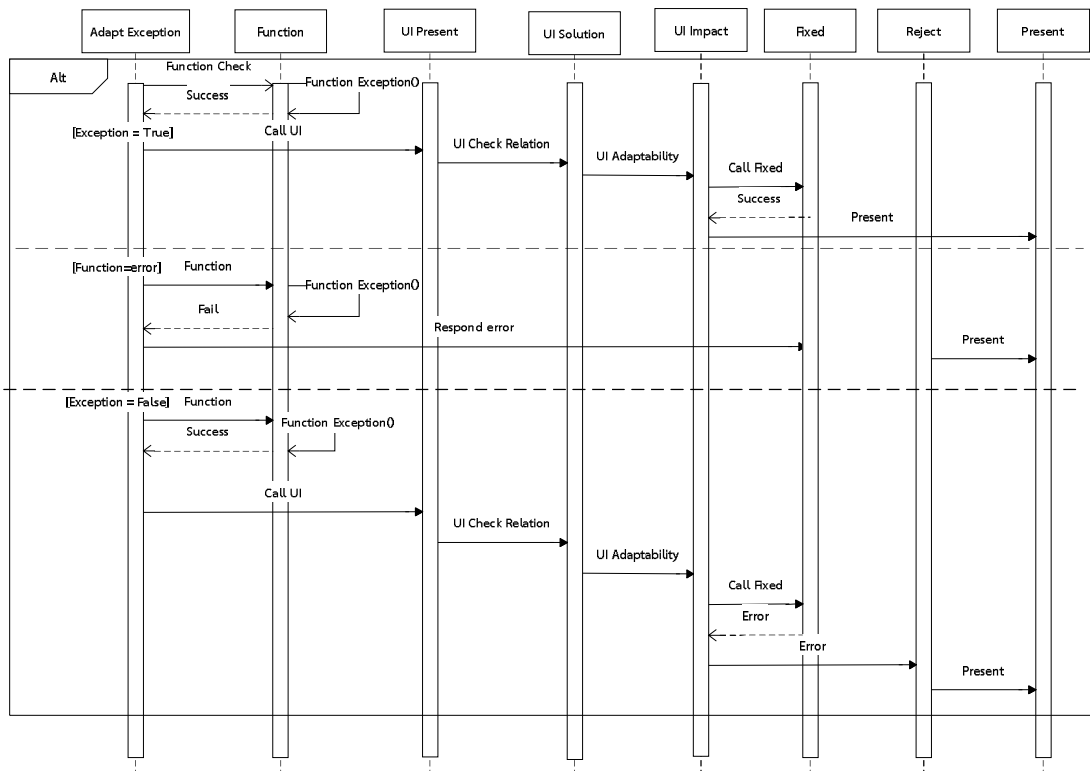
ดังภาพที่ 4.22 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อการปรับตัว ทำงานผ่านส่วนการจัดการสิ่งผิดปกติการปรับตัว (*Adaptability Exception*) เพื่อกำหนดฟังก์ชันงานที่เข้ามาใหม่และตรวจสอบฟังก์ชันผ่านส่วน ฟังก์ชันการปรับตัว (*Adaptability Function*) หลังจากผ่านกระบวนการฟังก์ชันแล้วจะเข้าสู่ ส่วนนำเสนอส่วนต่อประสาน (*UI Present*) เพื่อตรวจสอบส่วนต่อประสาน อ็อบเจกต์เกมที่ฟังก์ชันเรียกใช้ และตรวจสอบความสัมพันธ์ความสอดคล้องเพื่อปรับใช้อ็อบเจกต์เกมร่วมกับอ็อบเจกต์เกมเดิมของระบบผ่านส่วน แก้ปัญหาส่วนต่อประสาน (*UI Solution*) ภายหลังปรับใช้ส่วนต่อประสานของอ็อบเจกต์เกมแล้วจะใช้ ส่วนตรวจสอบผลกระทบส่วนต่อประสาน (*UI Impact*) ทำหน้าที่กำหนดเงื่อนไขตรวจสอบสิ่ง

ผิดพลาดจากการปรับใช้ส่วนต่อประสาน เมื่อเกิดปัญหาจะแก้ไขข้อผิดพลาดผ่านส่วนแก้ไขการปรับตัว (*Adaptability Reject*) ในกรณีที่ไม่สามารถแก้ไขข้อผิดพลาดจะใช้ เพื่อไม่ให้ระบบแสดงข้อผิดพลาด โดยการแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้น

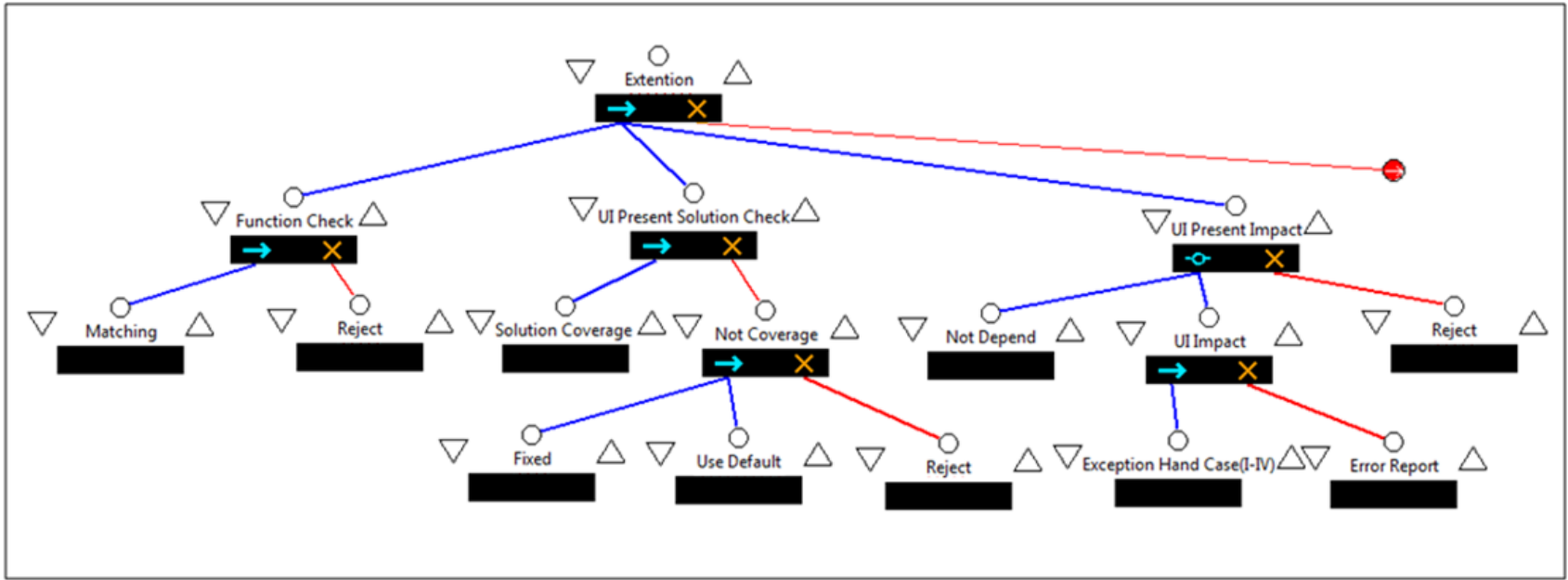


ภาพที่ 4.22 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดพลาดกรณีการปรับแต่ง  
เพื่อการปรับตัว

ภาพที่ 4.23 แผนภาพลำดับแสดงการทำงานของแบบรูปและ ภาพที่ 4.24 แผนภาพลิตเติล  
จิลแสดงโครงสร้างการทำงานของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อการ  
ปรับตัว



ภาพที่ 4.23 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีการปรับตัว



ภาพที่ 4.24 แผนภาพผลิตเต็ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการปรับตัว

#### 4.7 แบบรูปลำดับที่ 6: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการปรับแต่งเพื่อการบำรุงรักษา

การปรับปรุงส่วนต่อประสานกรณีการปรับแต่งเพื่อการบำรุงรักษา เป็นการปรับแต่งส่วนต่อประสานเนื่องจากการแก้ไขฟังก์ชันของเกมแล้วทำให้เกิดผลกระทบต่อส่วนต่อประสาน แบบรูป ออกแบบมาเพื่อปรับแต่งส่วนต่อประสานที่ได้รับผลกระทบให้สามารถทำงานได้ตามปกติ ตัวอย่างเช่น การเพิ่มตู้เกมอีก 1 ตู้ส่งผลให้ส่วนต่อประสานหยุดเหรียญของเกมไม่สามารถใช้งานร่วมกับตู้เกมใหม่ที่เพิ่มเข้ามาได้ เป็นต้น ดังตารางที่ 4.7

ตารางที่ 4.7 ตัวอย่างการปรับแต่งเพื่อการปรับตัว

| ลำดับ | การปรับแต่ง                               | รูปแบบการแก้ไข   | จุดประสงค์การปรับแต่ง                   | หมายเหตุ |
|-------|---|--|---|----------|
| 1     | การเพิ่มตู้เกมใหม่ ในเกม สล็อตแมชชีน      | การเพิ่มตู้เกมใหม่ มีผลต่อส่วนต่อประสานหยุดเหรียญ                                    | เพื่อให้เกิดความสุนทรีย์ภาพในการเล่น    |          |
| 2     | การเพิ่มตัวละครอีกหนึ่งตัวในเกมเล่นเดี่ยว | การเพิ่มตัวละครอีกหนึ่งตัวในเกมเล่นเดี่ยว ต้องมีการแยกส่วนต่อประสานเพื่อให้บังคับได้ | เพื่อให้สามารถเล่นร่วมกันหลายตัวละครได้ |          |

**จุดประสงค์** จัดการกรณีการปรับแต่ง เพื่อแก้ไขผลกระทบจากการแก้ไขปรับแต่งฟังก์ชันงานในเกม แต่ส่งผลต่อส่วนต่อประสาน จนทำให้ระบบทำงานบกพร่องหรือไม่สามารถทำงานต่อไปได้ โดยมุ่งเน้นปรับให้ระบบสามารถทำงานได้เป็นปกติ

**ปัญหา** การมอดเกมโดยการเพิ่มหรือแก้ไขฟังก์ชันซึ่งส่งผลกระทบต่อส่วนต่อประสานเดิมระบบทำให้ไม่สามารถทำงานได้ ตัวอย่างเช่น การเพิ่มตู้เกมใหม่ลงไปในเกมสล็อตแมชชีนเพื่อสร้างสีสันหรือต้องการเล่น 2 ตู้พร้อมกันแต่เนื่องจากเหรียญและคั่นโยกใช้ส่วนต่อประสานตัวเดียวกัน ทำให้ส่วนต่อ

ประสานไม่สามารถแยกแยะคู่แข่งเป้าหมายที่ต้องการ จากกรณีตัวอย่างสาเหตุมาจากความซ้ำซ้อน  
 ของคู่แข่งเป้าหมายที่ส่วนต่อประสานทำงานด้วย

วิธีการแก้ปัญหาต้องเป็นตามหลักดังต่อไปนี้

1. กำหนดสิ่งที่ต้องการปรับแต่ง ในกรณีนี้คือ การเพิ่มคู่แข่ง
2. กำหนดเงื่อนไขความสัมพันธ์กับส่วนต่อประสานเดิมของเกม
3. กำหนดเงื่อนไขการทำงานของฟังก์ชันที่ปรับแต่ง ในกรณีตัวอย่างคือ การทำงานของ  
 คู่แข่งที่เพิ่มเข้ามา
4. กำหนดความขึ้นต่อกันของอ็อบเจกต์เกม อันได้แก่ชื่อของอ็อบเจกต์เกมเพื่อป้องกัน  
 ปัญหาการเรียกใช้งานซ้ำซ้อน
5. กำหนดเงื่อนไขความผิดพลาดของส่วนต่อประสานอื่นๆนอกเหนือจากกรณีการส่งผล  
 กระทบต่อส่วนต่อประสาน
6. กำหนดผลการทำงานของฟังก์ชัน ดูผลการทำงานของส่วนต่อประสานที่เกิด  
 ผลกระทบจากการทำงาน และสร้างแนวทางแก้ไขเมื่อได้รับผลกระทบโดยสร้างส่วน  
 ต่อประสานใหม่เพื่อช่วยการทำงาน
7. เมื่อเกมเริ่มทำงาน เกิดข้อผิดพลาด แบบรูปจะแก้ไขข้อผิดพลาดที่เกิดขึ้นตาม  
 เงื่อนไขและวิธีแก้ไขที่ออกแบบไว้
8. ในกรณีที่ไม่สามารถแก้ไขได้หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด  
 แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น และ  
 ยกเลิกการนำเสนอส่วนต่อประสานนั้นๆ

**การประยุกต์ใช้** แบบรูปเพื่อจัดการสิ่งผิดปกติ เพื่อจัดการผลกระทบของการปรับแต่งฟังก์ชันงาน  
 หรือองค์ประกอบส่วนใดๆของระบบอันส่งผลกระทบต่อส่วนต่อประสานอ็อบเจกต์เดิมของเกม โดย  
 การปรับแก้ไขการทำงานหรือแยกการทำงานเพื่อไม่ให้เกิดความขึ้นต่อกันของส่วนต่อประสานอ็อบ  
 เจกต์เกม เพื่อให้ระบบเกมสามารถดำเนินการทำงานต่อไปได้

**โครงสร้าง** จากแบบรูปมีโครงการทำงานแบ่งเป็น 3 องค์ประกอบ มีรายละเอียดดังนี้

- **การจัดการสิ่งผิดปกติ (Exception)** กำหนดรายละเอียดของปรับแต่งเกมและฟังก์ชันของ  
 เกมต่างๆที่ทำให้เกิดความขึ้นต่อกันของส่วนต่อประสาน
- **การยกเลิก (Reject)** นำเสนอแนวทางการยกเลิกการปรับแต่งเกมในแต่ละการปรับแต่ง และ  
 ในแต่ละอ็อบเจกต์เกม



- **ส่วนความสัมพันธ์ส่วนต่อประสาน (UI Relate)** กำหนดส่วนต่อประสานอ็อบเจกต์เกมที่ฟังก์ชันมีการเรียกใช้โดยมีส่วนการทำงานย่อย 3 ส่วนได้แก่
  - บำรุงรักษา (Maintain) สร้างเงื่อนไขตรวจสอบการทำงานของฟังก์ชัน
  - ส่วนตรวจสอบความขึ้นต่อกัน (Depend Case) หาแนวทางแก้ไขปัญหาการทำงานของส่วนต่อประสานที่อาจขึ้นต่อกัน
  - ส่วนแก้ไขปัญหาที่เกี่ยวข้อง (Relate Solution) โดยนำส่วนการจัดการสิ่งผิดปกติจากการอาศัยแบบรูปควบคุมความผิดปกติอื่นๆ
- **ส่วนแยกการทำงาน (Separate)** กำหนดเงื่อนไขตรวจสอบการทำงานในแต่ละส่วนต่อประสานหรือส่วนการทำงานย่อยเพื่อหาความผิดปกติจากการทำงาน โดยมีเลือกใช้งานจาก 2 ส่วนย่อยในการแก้ปัญหาได้แก่
  - ส่วนที่ให้ความสนใจ (Attentive) การทำงานมีประเด็นต้องกำหนดแนวทางแก้ไขโดยการสร้างส่วนต่อประสานอ็อบเจกต์ของเกมใหม่เพื่อให้ระบบทำงานได้เป็นปกติหรือแจ้งเตือนผู้ปรับแต่งถึงข้อผิดพลาดที่เกิดขึ้น
  - ส่วนไม่ให้ความสนใจ (Disregard) เนื่องจากอาจจะมีการยกเลิกการทำงานบางส่วนออกในขณะที่บำรุงรักษาทำให้ยังคงมีการทำงานบางประเภทที่เราไม่ต้องให้ความสนใจแก้ปัญหาส่วนต่อประสานกลุ่มนั้นๆ

วิธีการใช้ เป็นไปตามลำดับต่อไปนี้

1. ผู้ปรับแต่งกำหนดการแก้ไขฟังก์ชันงานของเกม
2. ระบุอ็อบเจกต์เกมที่เกี่ยวข้องกับฟังก์ชันงานที่แก้ไขและกำหนดขอบเขตของอ็อบเจกต์เกมหรือค่าในการแสดงผลที่คาดว่าจะมีผลการทำงานที่ผิดปกติลงในส่วนเงื่อนไข และกำหนดรายการแก้ไขลงในส่วนแก้ไขโดยอ้างอิงจากแบบรูปจัดการสิ่งผิดปกติกรณีอื่น
3. กรณีที่การแก้ไขปัญหาในข้อ 2 ไม่สามารถจัดการปัญหาได้สร้างแนวทางแก้ไขเพิ่มโดยการสร้างอ็อบเจกต์เกมใหม่เพื่อช่วยควบคุมให้การทำงานสามารถดำเนินไปได้ตามปกติ
4. การสร้างแนวทางการยกเลิกการทำงานเพื่อแจ้งข้อผิดพลาดที่เกิดขึ้นเมื่อมีข้อผิดพลาดนอกเหนือจากเงื่อนไขที่กำหนดไว้

แบบรูปจะช่วยลดข้อผิดพลาดจากการแก้ไขฟังก์ชันอันทำให้เกิดผลกระทบต่ออ็อบเจกต์เกมที่นำเสนอแก่ผู้เล่น ทำให้มีการนำเสนอที่ผิดพลาด ซ้ำซ้อน หรือไม่สามารถแสดงผลได้ โดยการเพิ่มอ็อบเจกต์เกมเพื่อให้ระบบสามารถดำเนินไปได้ตามปกติ นอกจากนี้แบบรูปยังนำเสนอสาเหตุข้อผิดพลาด

ผิดพลาดใหม่ๆจากฟังก์ชันที่ไม่ได้กำหนดไว้ ณ ตอนต้นการออกแบบเพื่อให้ผู้ปรับแต่งสามารถนำมาปรับใช้เพิ่มเติมในแบบรูปได้ต่อไป

ตัวอย่าง การปรับแต่งส่วนต่อประสาน โดยการปรับแต่งฟังก์ชันงานซึ่งส่งผลกระทบต่อระบบ เช่น เกมที่ต้องมีการหยุดเหรียญในการเล่น การทำงานกรณีที่มีการเพิ่มจำนวนตู้ใหม่แก่ระบบ ระบบจะไม่ทราบถึงการมีอยู่ของตู้ใหม่ เป็นต้น มีตัวอย่างดังภาพที่ 4.25 และมีลำดับการทำงานดังนี้

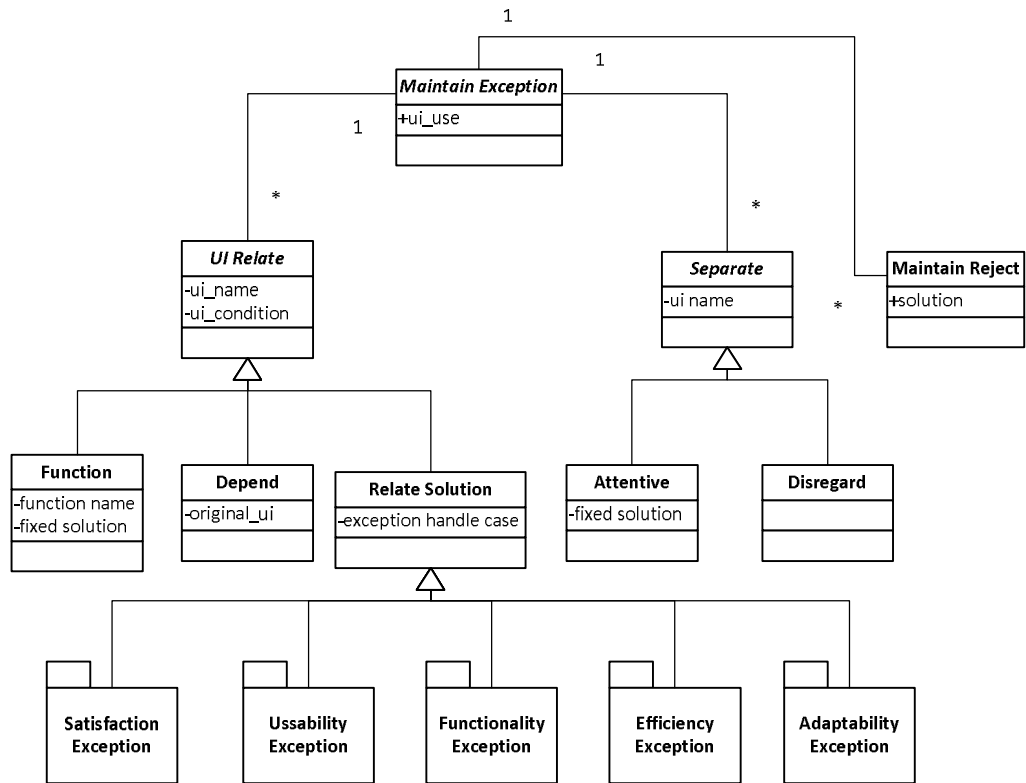


ภาพที่ 4.25 รูปภาพการเพิ่มตู้เกม

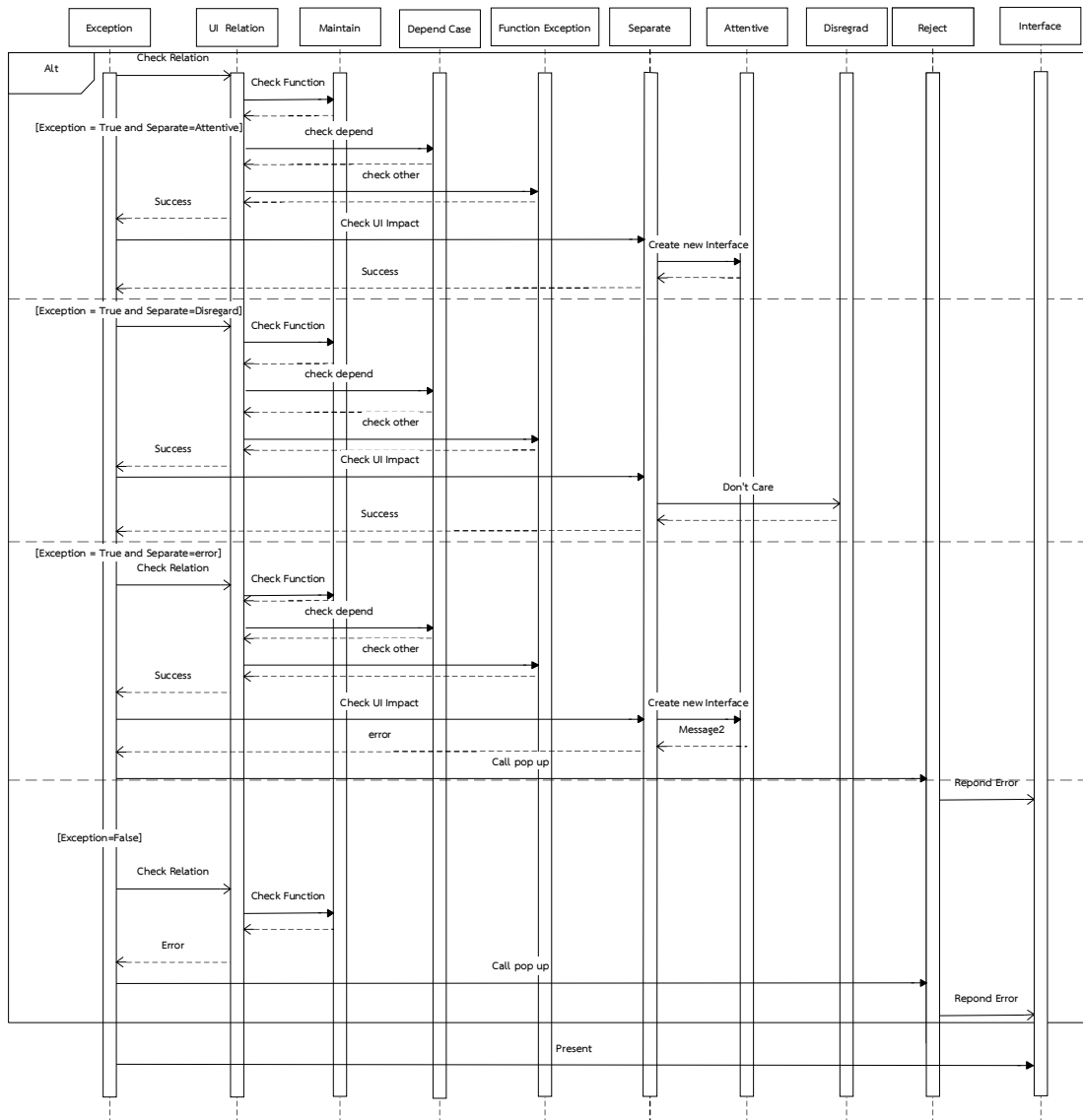
1. กำหนดฟังก์ชันที่แก้ไขจากการปรับแต่งลงในส่วนจัดการสิ่งผิดปกติ ในกรณีตัวอย่างคือ การเพิ่มตู้สล็อตแมชชีนใหม่อีกหนึ่งตู้
2. กำหนดเงื่อนไขการตรวจสอบความสัมพันธ์ของส่วนต่อประสานเกมที่ฟังก์ชันเรียกใช้ พร้อมตรวจสอบการทำงานของฟังก์ชัน จัดการความขึ้นต่อกันของอ็อบเจกต์เกมที่ฟังก์ชันเรียกใช้ โดยการแก้ไขชื่อเรียกใช้งาน และกำหนดแนวทางแก้ปัญหาอ้างอิงจากแบบรูปจัดการสิ่งผิดปกติอื่นๆ
3. กำหนดเงื่อนไขตรวจสอบการทำงานของส่วนต่อประสานอ็อบเจกต์เกม และสร้างข้อกำหนดการแก้ไข โดยสร้างอ็อบเจกต์เกมเพื่อให้เกมสามารถดำเนินต่อไปได้

4. ในกรณีที่แบบรูปสามารถแก้ไขข้อผิดพลาดที่เกิดขึ้นได้จะแก้ไขการทำงานตามเงื่อนไขและวิธีแก้ไขที่ออกแบบไว้
5. ในกรณีที่ไม่สามารถแก้ไขได้หรือข้อผิดพลาดนั้นๆอยู่นอกเหนือเงื่อนไขที่กำหนด แบบรูปจะเรียกใช้ใน การยกเลิก (reject) เพื่อแจ้งถึงข้อผิดพลาดที่เกิดขึ้น และยกเลิกการนำเสนอส่วนต่อประสานนั้นๆ

อนึ่งแบบรูป ส่วนแยกการทำงานการบำรุงรักษา สามารถเรียกใช้งานเพียงแบบเดียวได้ถ้าการปรับแต่งเกมครั้งนั้นๆมีการสร้างความสัมพันธ์รวมมากับการปรับแต่ง ดังภาพที่ 4.26 แผนภาพคลาสแสดงโครงสร้างของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อการบำรุงรักษา เริ่มต้นการทำงานส่วนจัดการสิ่งผิดปกติการบำรุงรักษา (*Maintain Exception*) จะตรวจสอบฟังก์ชันงานที่ทำงานแก้ไข และส่งส่วนต่อประสานอ็อบเจกต์เกมและฟังก์ชันงานที่เกี่ยวข้องแก่ส่วนความสัมพันธ์ส่วนต่อประสาน (*UI Relation*) เพื่อตรวจสอบใน 3 กรณีตามความเหมาะสมของการปรับแต่งนั้นๆได้แก่ การตรวจสอบความถูกต้องของฟังก์ชันใน ส่วนบำรุงรักษา (*Maintain*) ตรวจสอบความขึ้นต่อกันของส่วนต่อประสานใน ส่วนตรวจสอบความขึ้นต่อกัน (*Depend Case*) และการจัดการสิ่งผิดปกติในกรณีการปรับแต่งประเภทอื่นๆผ่านส่วนแก้ไขปัญหาที่เกี่ยวข้อง (*Relate Solution*) เมื่อไม่มีผลกรณีที่กล่าวมาจะเข้าใช้ส่วนแยกการทำงาน (*Separate*) เพื่อตรวจสอบการทำงานของส่วนต่อประสานว่ายังคงมีข้อผิดพลาดอยู่หรือไม่ และถ้ามีอยู่จะจัดการกับข้อผิดพลาดนั้นๆอย่างไร ขึ้นกับกรณีการปรับแต่งนั้นๆเช่นกรณีที่มีส่วนต่อประสานผิดพลาดและต้องให้ความสนใจในการแก้ไขใช้ส่วนที่ให้ความสนใจ (*Attentive*) ช่วยเหลือในการทำงาน ในขณะที่บางกรณีการปรับแต่งอาจจะมีการยกเลิกส่วนต่อประสานบางตัวไปถ้าเกิดกรณีข้อผิดพลาดที่ส่วนต่อประสานดังกล่าวเราจะไม่ให้ความสนใจโดยทำงานผ่านส่วนไม่ให้ความสนใจ (*Disregard*) เมื่อเกิดปัญหาจะแก้ไขข้อผิดพลาดผ่านส่วนแก้ไขการบำรุงรักษา (*Maintain Reject*) ในกรณีที่ไม่สามารถแก้ไขข้อผิดพลาดจะใช้ เพื่อไม่ให้ระบบแสดงข้อผิดพลาดโดยการแสดงข้อความแจ้งถึงข้อผิดพลาดที่เกิดขึ้น

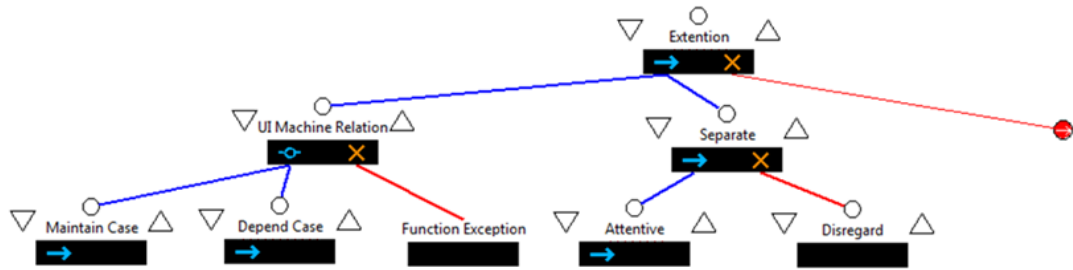


ภาพที่ 4.26 แผนภาพคลาส โครงสร้างของการประยุกต์แบบรูปจัดการสิ่งผิดปกติกรณีการ  
ปรับแต่งเพื่อการบำรุงรักษา



ภาพที่ 4.27 แผนภาพลำดับ การทำงานของแบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุง ส่วนต่อประสาน กรณีการบำรุงรักษา

ภาพที่ 4.27 แผนภาพลำดับแสดงการทำงานของแบบรูปและ ภาพที่ 4.28 แผนภาพลิตเติล จิลแสดงโครงสร้างการทำงานของแบบรูปเพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสานกรณีเพื่อการ บำรุงรักษา



ภาพที่ 4.28 แผนภาพผลิตเต็ล-จิล การทำงานตัวอย่างการทำงานกรณีการปรับแต่งเพื่อการบำรุงรักษา

#### 4.8 สรุปการประยุกต์ใช้

ผลที่ได้เห็นได้ว่าอ้างอิงจากการนำแบบรูปจัดการสิ่งผิดปกติมาประยุกต์ใช้โดยออกแบบเป็นแนวทางในการใช้งานกรณีที่มีการปรับแต่งส่วนต่อประสานของซอฟต์แวร์เกม แต่ระบบจะมีการใช้งานแบบรูปจัดการสิ่งผิดปกติที่แตกต่างกันโดยในงานวิจัยได้ประยุกต์แบบรูปจัดการสิ่งผิดปกติทั้งสิ้น 7 แบบรูปจาก 8 แบบรูปที่ได้นิยามไว้ ดังตารางที่ 4.8 โดยมีรายละเอียดปรากฏอยู่ในภาคผนวก ค

ตารางที่ 4.8 ตารางรายการสรุปการประยุกต์ใช้

| ลำดับ | กรณีการปรับแต่ง     | แบบรูปจัดการสิ่งผิดปกติที่นำมาประยุกต์ใช้ |
|-------|---------------------|---|
| 1     | เพื่อความพึงพอใจ    | ทางเลือกแบบมีลำดับ                        |
|       |                     | ทางเลือกแบบไม่มีลำดับ                     |
|       |                     | การแก้ไขแบบทันทีทันใด                     |
|       |                     | การยกเลิก                                 |
| 2     | เพื่อการใช้งานเฉพาะ | ทางเลือกแบบมีลำดับ                        |
|       |                     | ทางเลือกแบบไม่มีลำดับ                     |
|       |                     | การแก้ไขแบบทันทีทันใด                     |
|       |                     | การยกเลิก                                 |
|       |                     | การแก้ไขแบบเชื่อมโยง                      |
|       |                     | การชดเชย                                  |

ตารางที่ 4.8 ตารางรายการสรุปการประยุกต์ใช้ (ต่อ)

| ลำดับ | กรณีการปรับแต่ง    | แบบรูปจัดการสิ่งผิดปกติที่นำมาประยุกต์ใช้ |
|-------|--------------------|---|
| 3     | เพื่อฟังก์ชันงาน   | ทางเลือกแบบมีลำดับ                        |
|       |                    | ทางเลือกแบบไม่มีลำดับ                     |
|       |                    | การแก้ไขแบบทันทีทันใด                     |
|       |                    | การแก้ไขแบบเชื่อมโยง                      |
|       |                    | การยกเลิก                                 |
| 4     | เพื่อประสิทธิภาพ   | เงื่อนไขการทำงานใหม่                      |
| 5     | เพื่อการปรับตัว    | ทางเลือกแบบมีลำดับ                        |
|       |                    | ทางเลือกแบบไม่มีลำดับ                     |
|       |                    | การแก้ไขแบบทันทีทันใด                     |
|       |                    | การยกเลิก                                 |
|       |                    | การแก้ไขแบบเชื่อมโยง                      |
| 6     | เพื่อการบำรุงรักษา | ทางเลือกแบบมีลำดับ                        |
|       |                    | ทางเลือกแบบไม่มีลำดับ                     |
|       |                    | การแก้ไขแบบทันทีทันใด                     |
|       |                    | การยกเลิก                                 |
|       |                    | การแก้ไขแบบเชื่อมโยง                      |
|       |                    | การชดเชย                                  |

จากตารางเห็นได้ว่าแนวทางการแก้ปัญหาของแบบรูปจัดการสิ่งผิดปกติสำหรับการปรับแต่ง ส่วนต่อประสานมีความสอดคล้องกับแบบรูปจัดการสิ่งผิดปกติซึ่งนำมาประยุกต์ใช้งานสอดคล้อง และเป็นแนวทางการแก้ปัญหาที่ออกแบบจากปัญหาที่เกิดขึ้นตามแต่ละกรณี ทั้งนี้แบบรูปซึ่งถูกนิยามลงในแต่ละกรณีการปรับแต่งอาจจะไม่ถูกเลือกใช้ทั้งหมดในการทำงานแต่ละครั้งเพื่อให้เกิดความเหมาะสมกับกรณีความผิดปกติที่เกิดขึ้นจริงในการมอด

#### 4.9 วิเคราะห์ความสัมพันธ์ของแบบรูปจัดการสิ่งผิดปกติและหลักการการออกแบบส่วนต่อประสานเกมไคเจสซิส

ผู้วิจัยได้วิเคราะห์หาความสัมพันธ์ร่วมกับกับหลักการการออกแบบส่วนต่อประสานเกมไคเจสซิส เพื่อหาความสอดคล้องว่ารูปแบบการทำงานของแบบรูปครอบคลุมตามหลักการการออกแบบส่วนต่อประสานเกมในส่วนใด ได้ผลดังตารางที่ 4.9

ตารางที่ 4.9 ตารางความสัมพันธ์แบบรูปและหลักการการออกแบบส่วนต่อประสานเกมไคเจสซิส

| แบบรูป              | การปรับแต่ง  | การปรับแต่งส่วนต่อ<br>ประสานเกมประเภท |
|---------------------|--|---------------------------------------|
| เพื่อความพึงพอใจ    | เพื่อปรับเปลี่ยนองค์ประกอบของ<br>คุณสมบัติอ็อบเจกต์เกม   | ทุกกรณี                               |
| เพื่อการใช้งานเฉพาะ | เพื่อปรับเปลี่ยนหรือเพิ่มองค์ประกอบ<br>ของคุณสมบัติอ็อบเจกต์เกม  | เดียเจติก ,นอน-เดียเจติก              |
| เพื่อฟังก์ชันงาน    | การเพิ่มส่วนต่อประสานอ็อบเจกต์<br>เกมพร้อมฟังก์ชันงานที่ทำงาน<br>เกี่ยวข้องกับส่วนต่อประสาน              | เดียเจติก ,นอน-เดียเจติก              |
| ประสิทธิภาพ         | การรวมคำสั่งการทำงานของเกมผ่าน<br>ส่วนต่อประสานหรืออ็อบเจกต์เกม<br>จากการใช้ อ็อบเจกต์เกมเดิมของ<br>ระบบ | เดียเจติก ,นอน-เดียเจติก              |
| การปรับตัว          | การเพิ่มฟังก์ชันงานไปพร้อม<br>คุณสมบัติอ็อบเจกต์เกม  | ทุกกรณี                               |
| การบำรุงรักษา       | การแก้ไขฟังก์ชันซึ่งส่งผลกระทบต่อ<br>ส่วนต่อประสาน   | ทุกกรณี                               |

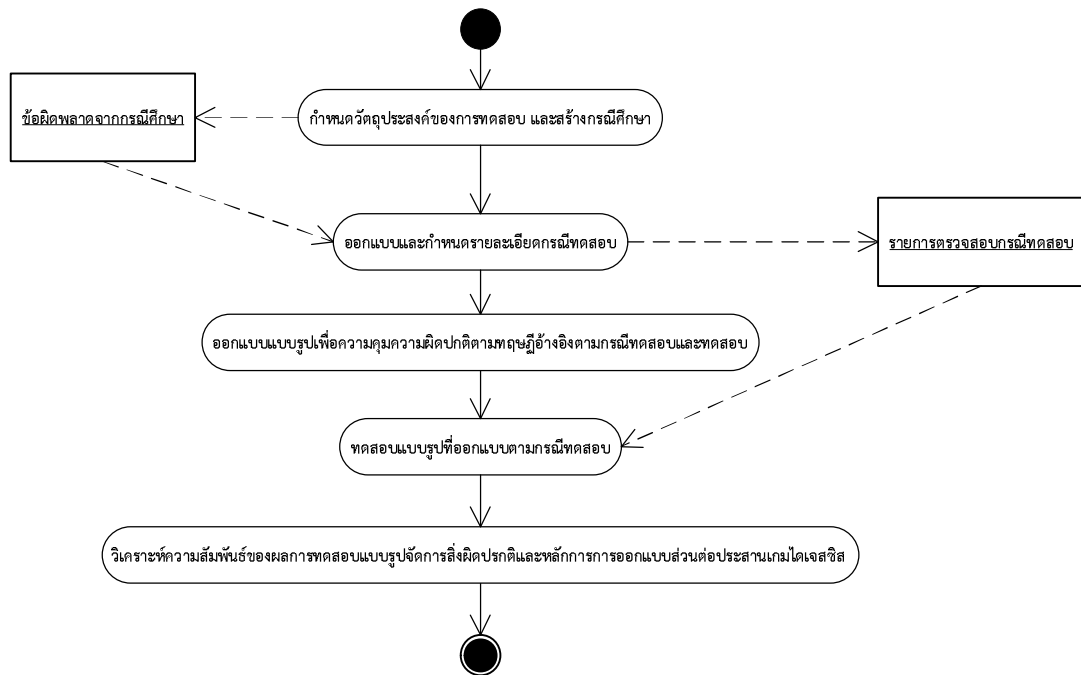
จากตารางเห็นได้ว่าแนวทางการปรับแต่งสามารถปรับให้เข้ากับหลักการออกแบบไคเจสซิสได้ในทุกกรณีขึ้นกับส่วนต่อประสานเป้าหมายที่เราต้องการควบคุมความผิดปกติและขอบเขตของซอฟต์แวร์ที่อนุญาตให้แก้ไขปรับแต่งในคุณสมบัติ ทั้งนี้แบบรูปซึ่งถูกนิยามลงในแต่ละกรณีการปรับแต่งอาจจะไม่ถูกเลือกใช้ทั้งหมดในการทำงานแต่ละครั้งเพื่อให้เกิดความเหมาะสมกับกรณีการปรับแต่ง



## บทที่ 5

### การทดสอบและการประเมินผลการออกแบบ

พัฒนาเครื่องมือทดสอบจากโอเพนซอร์สอ้างอิงตามแบบรูปที่ออกแบบไว้ เพื่อเป็นการทดสอบแบบรูปและเครื่องมือ เพื่อประเมินว่าแบบรูปแม่แบบที่พัฒนาขึ้นนั้นสามารถทำงานถูกต้องตามข้อกำหนดความต้องการ โดยการนำแบบรูปที่ออกแบบมาทดสอบพร้อมประเมินผล ซึ่งมีขั้นตอนดังภาพที่ 5.1 และมีรายละเอียดดังนี้



ภาพที่ 5.1 แผนภาพกิจกรรมขั้นตอนการทดสอบและการประเมินผลการออกแบบ

- 1) กำหนดวัตถุประสงค์ของการทดสอบ เลือกระบบพื้นฐาน และสร้างกรณีศึกษา จากซอฟต์แวร์โอเพนซอร์ส
- 2) ออกแบบและกำหนดรายละเอียดกรณีทดสอบ จากกรณีศึกษา
- 3) ทดสอบแบบรูปที่ออกแบบตามกรณีทดสอบที่กำหนด โดยการเพิ่มส่วนการปรับปรุงซอฟต์แวร์เกมตามทฤษฎีให้กับโปรแกรม เก็บข้อมูลผลการทดสอบ
- 4) สรุปผลการทดสอบ โดยวิเคราะห์จากผลการทดสอบ

โดยเกณฑ์การประเมินออกแบบในรูปแบบของรายการตรวจสอบความต้องการของการทำมอดอ้างอิงผลการทดสอบและประเมินผลการควบคุมข้อผิดพลาดที่เกิดขึ้นว่าสามารถควบคุมตามรายการที่ออกแบบไว้ในกรณีทดสอบได้หรือไม่

### 5.1 การสร้างกรณีศึกษาเพื่อกำหนดกรณีทดสอบ จากการการปรับแต่งส่วนต่อประสานตามกรณี

เป็นการกำหนดกรณีทดสอบจากการพัฒนาการปรับแต่งซอฟต์แวร์เกม เกมที่นำมาใช้เป็นพื้นฐานสำหรับการออกแบบคือเกม สล็อตแมชชีน ซึ่งมีลักษณะการเล่นเป็นการหมุนสล็อตเพื่อสุ่มผลที่ออกในหน้าสล็อตทั้ง 3 ถ้าผลของทั้ง 3 หน้าได้ตรงกันจึงจะเป็นการเล่นชนะในรอบนั้นๆ โดยการเล่นแต่ละครั้งต้องใช้เหรียญในการเล่น เมื่อเล่นชนะจะได้เหรียญตอบแทน เกมประเภทดังกล่าวจัดเป็นเกมประเภทเกมกระดาน มีรายละเอียดซอฟต์แวร์ต้นฉบับตามภาคผนวก ค [22]

พัฒนาการปรับแต่งเพื่อสร้างกรณีศึกษาเพื่อนำผลข้อผิดพลาดจากการปรับแต่งเป็นไปตามกรณีที่ออกแบบทั้ง 6 กรณีดังตารางที่ 5.1 ไปกำหนดเป็นกรณีทดสอบในการประยุกต์ใช้โหมด พร้อมเก็บข้อมูลผลข้อผิดพลาดที่คาดว่าจะเกิดขึ้นหรือได้ผลของการปรับแต่งเป็นไปตามที่คาดหวังหรือไม่อย่างไร โดยบันทึกผลที่เกิดขึ้นจริงเพื่อนำไปเป็นข้อมูลในการออกแบบกรณีทดสอบ ผลที่เกิดขึ้นเป็นไปตามแบบฟอร์มตัวอย่างเพื่อเก็บเป็นตัวอย่างกรณีศึกษา อนึ่งข้อผิดพลาดที่คาดว่าจะเกิดขึ้นนั้นไม่ได้เกิดขึ้นจริงผู้วิจัยได้ทำการวิเคราะห์เบื้องต้นถึงสาเหตุทำข้อผิดพลาดนั้นๆควบคู่ไปในแบบฟอร์มดังตัวอย่างในตารางที่ 5.2 โดยอ้างอิงตัวอย่าง 1 กรณีดังภาพที่ 5.2-5.3 โดยนำข้อมูลที่เก็บได้ในแต่ละกรณีศึกษามาวิเคราะห์เพื่อพัฒนาเป็นกรณีทดสอบต่อไป

ตารางที่ 5.1 การปรับแต่งเกมสล็อตแมชชีนให้สอดคล้องกับกรณีการปรับแต่งส่วนต่อประสานที่ ออกแบบ

| กรณีการปรับแต่ง         | การปรับแต่ง                                 | คำอธิบาย  |
|-------------------------|---|---|
| 1. ความพึงพอใจของผู้ใช้ | เปลี่ยนภาพในช่องสล็อต                       | เปลี่ยนภาพ “เลข 7” ใหม่เพื่อสร้างความโดดเด่น และเปลี่ยนภาพ เซอร์รี่เป็นแอปเปิ้ล |
| 2. การใช้งาน            | การเพิ่มข้อความแสดงความยินดีเมื่อชนะรางวัล  | สร้างข้อความแสดงที่หน้าจอความยินดีเมื่อชนะ                                      |
| 3. ฟังก์ชันงาน          | เพิ่ม กราฟิกแสดงเมื่อได้เงินรางวัล          | เพิ่มภาพแสดงเหรียญวิ่ง เมื่อชนะรางวัล เพื่อให้สอดคล้องกับตอนหยอดเหรียญ          |
| 4. ประสิทธิภาพ          | การรวมปุ่มหยอดเหรียญและสับคันโยกเข้าด้วยกัน | จากเดิมที่ต้อง กด หยอดเหรียญ และ กด สับคันโยก โดยรวมไว้ในปุ่มเดียว              |
| 5.การปรับตัว            | เพิ่มช่องสล็อตอีก 2 ช่องรวมเป็น 5 ช่อง      | เดิมสล็อตมีอยู่ 3 ช่อง เพิ่มอีก 2 ช่องสล็อตเป็น 5 ช่อง                          |
| 6 การบำรุงรักษา         | เพิ่มตู้สล็อตอีก 1 ตู้                      | เพิ่มตู้สล็อตอีก 1 ตู้ให้เลือกเล่นได้   |



ภาพที่ 5.2 การทำงานปกติก่อนทดสอบ



ภาพที่ 5.3 ผลการทดสอบกรณีเพื่อความพึงพอใจของผู้ใช้ โดยการเปลี่ยนภาพในช่องสล็อต

ตารางที่ 5.2 แบบฟอร์มผลการกรณีศึกษา

| ลำดับ | ชื่อกรณีทดสอบ                                | การแสดงผลที่ผิดปกติ   |
|-------|--|---|
| 1     | เปลี่ยนภาพในช่องสล็อต                        | ภาพมีขนาดใหญ่กว่าช่อง   |
| 2     | เพิ่มข้อความแสดงความยินดีเมื่อชนะ            | ไม่มีข้อความแจ้งใดๆเด่นชัด                                    |
| 3     | เพิ่มกราฟิกเหรียญวิ่งเข้ากระเป๋า             | เหรียญวิ่งเข้าไม่เข้ากระเป๋า และเลยหลุด<br>ขอบหน้าต่างระบบค้ำ |
| 4     | รวมปุ่มหยุดเหรียญและสับคันโยกเข้า<br>ด้วยกัน | กดปุ่ม 1 ครั้งทำงานได้เพียง 1 อย่างแม้<br>เป็นปุ่มที่รวม      |
| 5     | ช่องสล็อตอีก 2 ช่องรวมเป็น 5 ช่อง            | สล็อตยังทำงานเพียง 3 ช่องตามเกม<br>ต้นฉบับ                    |
| 6     | เพิ่มตู้สล็อตอีก 1 ตู้                       | เล่นเกมได้เพียงตู้เดียว                                       |

## 5.2 ออกแบบและกำหนดรายละเอียดกรณีทดสอบ จากกรณีศึกษา

จากผลข้างต้นทางผู้วิจัยคัดเลือกเฉพาะผลที่ผิดปกติเพื่อนำไปสร้างเป็นกรณีทดสอบ โดยกรณีทดสอบดังกล่าวจะเป็นกรณีความผิดปกติที่เกิดจากการปรับแต่งโดยวิธีการมอด ตามทฤษฎีที่กล่าวไว้ กรณีทดสอบทั้งหมดที่สร้างขึ้นจะเป็นเกณฑ์ในการประเมินการออกแบบแบบรูปเพื่อจัดการ

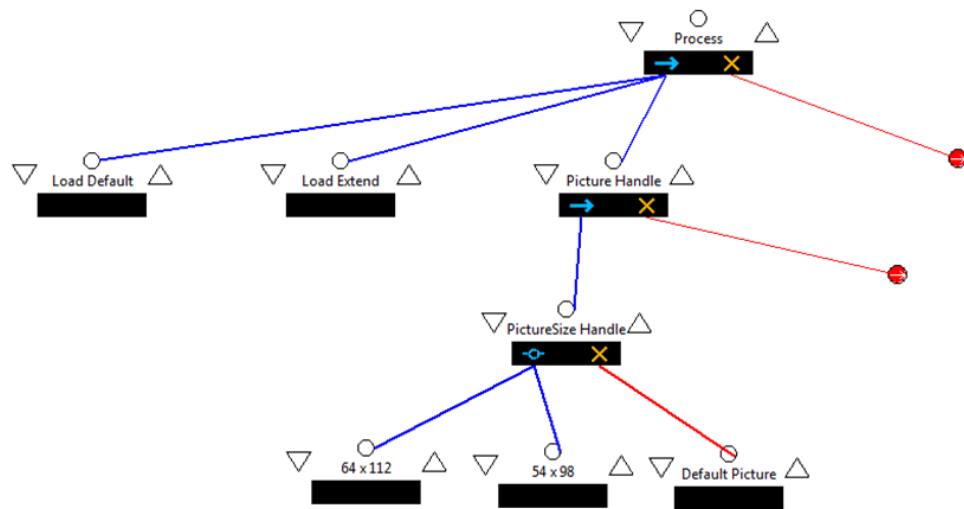
ข้อผิดพลาดตามแต่ละกรณีโดยมีรายละเอียดดังตารางที่ 5.3 และสร้างเป็นรายละเอียดกรณีทดสอบแต่ละกรณีลงในภาคผนวก ง ซึ่งจะกล่าวถึงในขั้นตอนต่อไป

ตารางที่ 5.3 กรณีทดสอบ

| ลำดับ | ชื่อกรณีทดสอบ                                | รายละเอียดกรณีทดสอบ             | แบบรูปที่ประยุกต์ใช้<br>กรณีการปรับแต่ง |
|-------|--|---------------------------------|---|
| 1     | เปลี่ยนภาพในช่องสล็อต                        | ภาพขนาดใหญ่เกินกรอบของ<br>สล็อต | ความพึงพอใจของผู้ใช้                    |
| 2     | เพิ่มข้อความแสดงความยินดีเมื่อ<br>ชนะ        | ข้อความไม่ปรากฏ                 | การใช้งาน                               |
| 3     | เพิ่มกราฟิกเหรียญวิ่งเข้ากระเป๋า             | เหรียญวิ่งผิดทิศทาง             | ฟังก์ชันงาน                             |
| 4     | รวมปุ่มหยุดเหรียญและสับคัน<br>โยกเข้าด้วยกัน | การรวมปุ่มทำงานไม่สมบูรณ์       | ประสิทธิภาพ                             |
| 5     | ช่องสล็อตอีก 2 ช่องรวมเป็น 5<br>ช่อง         | สล็อตช่องที่เพิ่มไม่ทำงาน       | การปรับตัว                              |
| 6     | เพิ่มตู้สล็อตอีก 1 ตู้                       | ตู้สล็อตทำงานทั้ง 2 ตู้         | การบำรุงรักษา                           |

### 5.3 ออกแบบแบบรูปเพื่อจัดการสิ่งผิดปรกติตามทฤษฎีอ้างอิงตามกรณีทดสอบและการทดสอบ

การนำทฤษฎีการออกแบบจากบทที่ 4 มาวิเคราะห์ตามกรณีทดสอบ เพื่อออกแบบแบบรูปตามทฤษฎีการประยุกต์ในแต่ละกรณีทดสอบ ทำการทดสอบ พร้อมนำเสนอโครงสร้างการออกแบบและซอร์ซโค้ดที่พัฒนาพร้อมบันทึกผลการทดสอบ กรณีที่แบบรูปไม่สามารถทำตามที่ได้คาดหวังได้ ผู้วิจัยจะออกแบบแบบรูปใหม่หรือแก้ไขทฤษฎีการออกแบบเพื่อให้ได้ผลตามที่คาดหวัง การบันทึกผลการประเมินเป็นตามแบบฟอร์มรายการตรวจสอบที่ออกแบบไว้ โดยการนำเสนอแบบรูปที่ออกแบบผ่าน แผนภาพลิตเติ้ลจิล ดังภาพที่ 5.4 ซึ่งเป็นตัวอย่างกรณีทดสอบที่ 1 การปรับแต่งเพื่อความพึงพอใจ พร้อมนำเสนอซอร์ซโค้ดดังภาพที่ 5.5 - ภาพที่ 5.7



ภาพที่ 5.4 แผนภาพ ลิตเติ้ล-จิล โครงสร้างของแบบรูปการปรับแต่งเพื่อความพึงพอใจของผู้ใช้  
สำหรับจัดการขนาดภาพช่องสล็อต

```

Public Sub Exception
  Dim ImageException As Image

  Try
    ImageException = Image.FromFile("Images\Apple.jpg") 'BMP,GIF,JPEG,PNG,TIFF
    OrderAlternative.Picture_Handling(ImageException)
  Catch ex As Exception
    OrderAlternative.Picture_Handling(ImageException)
  End Try
End Sub

```

ภาพที่ 5.5 ซอร์ซโค้ดในการเพิ่มรูปภาพใหม่แก่ระบบ

```

Private Sub Picture_Handling()
    PictureSize_Handle()
End Sub
Private Sub PictureSize_Handle()

    c = c + 1
    If imgseven.Image.Size.Height.ToString <> "47" Then
        Dim bm_source As New Bitmap(imgseven.Image)
        Dim bm_dest As New Bitmap(bm_source, 48, 47)
        ' Make a Graphics object for the result Bitmap.
        Dim gr_dest As Graphics = Graphics.FromImage(bm_dest)
        ' Copy the source image into the destination bitmap.
        gr_dest.DrawImage(bm_source, 0, 0, _
            bm_dest.Width + 1, _
            bm_dest.Height + 1)
        imgseven.Image = bm_dest
    End If
    If imgcherry.Image.Size.Height.ToString <> "47" Then
        Dim bm_source As New Bitmap(imgcherry.Image)
        ' Make a bitmap for the result.
        Dim bm_dest As New Bitmap(bm_source, 48, 47)
        ' Make a Graphics object for the result Bitmap.
        Dim gr_dest As Graphics = Graphics.FromImage(bm_dest)
        ' Copy the source image into the destination bitmap.
        gr_dest.DrawImage(bm_source, 0, 0, _
            bm_dest.Width + 1, _
            bm_dest.Height + 1).
        imgcherry.Image = bm_dest
    End If
End Sub

```

ภาพที่ 5.6 ซอร์ซโค้ดการคัดกรองกรณีการปรับแต่เกี่ยวกับรูปภาพ

```
Public Sub Reject(ByVal ex as String)
```

```
    MessageBox.Show(ex)
```

```
End Sub
```

ภาพที่ 5.7 ซอร์สโค้ด Reject แจ้งผลการทำงานกรณีที่ไม่สามารถปรับแต่งได้ ส่วนประกอบของแบบรูปประกอบด้วย 3 ส่วนได้แก่

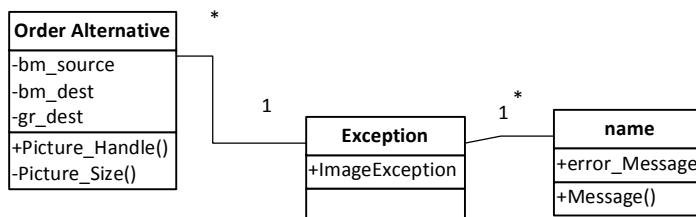
1. Exception เป็นส่วนขยายของเกม
2. Picture\_Handling เป็นส่วนหลักในการจัดการข้อผิดพลาดซึ่งภายในจะบรรจุ ส่วนย่อยในการจัดการข้อผิดพลาดในกรณีต่างๆ ตามแต่กรณี ซึ่งในกรณีทดสอบได้แก่ ส่วนจัดการขนาดภาพ (PictureSize\_Handle) เพียงกรณีเดียว
3. Reject แจ้งผลการทำงานกรณีที่ไม่สามารถปรับแต่งได้

โดยการทำงานของแบบรูปจะได้ผลตามภาพที่ 5.8 และโครงสร้างการทำงานดังรูปที่ 5.9



ภาพที่ 5.8 ภาพตัวอย่างส่วนต่อประสานที่ถูกแก้ไข ภาพเซอร์รีในช่องสล็อตถูกแทนที่ด้วย สับปะรด





ภาพที่ 5.9 แผนภาพคลาส แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อการบำรุงรักษาแก้ไขภาพเซอร์รี

ทำการบันทึกผลการทดสอบลงรายละเอียดกรณีทดสอบดังตารางที่ 5.4 ในแต่ละกรณีลงในแบบฟอร์มการประเมินรายการตรวจสอบเพื่อนำไปสรุปในขั้นตอนถัดไปดังตารางที่ 5.5 ตารางที่ 5.4 รายละเอียดกรณีทดสอบ การเปลี่ยนภาพ

|                                 |  |                       |
|---------------------------------|--|-----------------------|
| ชื่อกรณีทดสอบ                   | T011   | เปลี่ยนภาพในช่องสล็อต |
| ฟังก์ชันที่ทดสอบ                | R011   | PictureSize_Handle    |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีเพื่อความพึงพอใจ                |                       |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ  |                       |
| ข้อมูลนำเข้า                    | การเปลี่ยนภาพในช่องสล็อต   |                       |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | นำเสนอภาพที่ปรับแต่งอย่างถูกต้อง มีขนาดภาพพอดีกับช่อง                                |                       |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | นำเสนอภาพที่ผิดปกติ หรือ นำเสนอภาพเดิมของระบบ  |                       |
| ข้อมูลทดสอบ                     | ภาพดังภาพที่ 5.10 ซึ่งมีขนาด 96 x 172 pixel  |                       |
| ผลการทดสอบ (กรณีปกติ)           | ดังแสดงในภาพที่ 5.8  |                       |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน   หมายเหตุ |                       |



ภาพที่ 5.10 ภาพที่ใช้สำหรับการปรับแต่ง

ตารางที่ 5.5 แบบฟอร์มการประเมินรายการตรวจสอบ

| ลำดับ | ชื่อกรณีทดสอบ                            | ผลการทดสอบ   | สถานะ |
|-------|--|--|-------|
| 1     | เปลี่ยนภาพในช่องสล็อต                    | ภาพมีขนาดพอดีกับช่อง   | ผ่าน  |
| 2     | เพิ่มข้อความแสดงความยินดีเมื่อชนะ        | มีข้อความยินดีปรากฏเมื่อชนะ<br>เกม   | ผ่าน  |
| 3     | เพิ่มกราฟิกเหรียญวิ่งเข้ากระเป๋า         | มีเหรียญวิ่งเข้ากระเป๋าเงิน  | ผ่าน  |
| 4     | รวมปุ่มหยุดเหรียญและสับคันโยกเข้าด้วยกัน | สามารถใช้ปุ่มสับคันโยกเพียง<br>ปุ่มเดียวในการเล่นเกม เหรียญ<br>จะถูกหยุดโดยอัตโนมัติ | ผ่าน  |
| 5     | ช่องสล็อตอีก 2 ช่องรวมเป็น 5 ช่อง        | สามารถเล่นเกมปกติ เมื่อมีการ<br>เพิ่มช่อง ผลในแต่ละช่องเป็นไป<br>ตามปกติ             | ผ่าน  |
| 6     | เพิ่มตู้สล็อตอีก 1 ตู้                   | มีหน้าต่างเพิ่มให้เลือกว่าจะ<br>หยุด<br>เหรียญเพื่อเล่นตู้ใด                         | ผ่าน  |

#### 5.4 วิเคราะห์ความสัมพันธ์ของผลการทดสอบแบบรูปจัดการสิ่งผิดปกติและหลักการการออกแบบส่วนต่อประสานเกมไคเจสซิส

จากแบบรูปที่นำมาทดสอบผู้วิจัยได้วิเคราะห์หาความสัมพันธ์ร่วมกับกับหลักการออกแบบส่วนต่อประสานเกมไคเจสซิส เพื่อหาความสอดคล้องว่าผลการทำงานของแบบรูปครอบคลุมตามหลักการการออกแบบส่วนต่อประสานเกมในส่วนใด ได้ผลดังตารางที่ 5.6

ตารางที่ 5.6 ตารางความสัมพันธ์แบบรูปและหลักการการออกแบบส่วนต่อประสานเกมไดเจสซิส

| แบบรูป              | การปรับแต่งตัวอย่าง  | การปรับแต่งส่วนต่อ<br>ประสานเกมประเภท |
|---------------------|--|---------------------------------------|
| เพื่อความพึงพอใจ    | เปลี่ยนภาพในช่องสล็อต เป็นการปรับแต่งส่วนต่อประสานในส่วนโลกของเกม                                    | เดียเจติก                             |
| เพื่อการใช้งานเฉพาะ | เพิ่มข้อความแสดงความยินดีเมื่อชนะ เป็นการปรับแต่งส่วนต่อประสานในมิติของเกมไม่ใช่โลกของเกม            | นอน-เดียเจติก                         |
| เพื่อฟังก์ชันงาน    | เพิ่มกราฟิกเหรียญวิ่งเข้ากระเป๋า เป็นการปรับแต่งส่วนต่อประสานในส่วนโลกของเกม                         | เดียเจติก                             |
| ประสิทธิภาพ         | รวมปุ่มหยุดเหรียญและสับคันโยกเข้าด้วยกัน เป็นการปรับแต่งในโลกของเกมแต่มีความไม่สมเหตุสมผลในโลกของเกม | สเปเชียล                              |
| การปรับตัว          | ช่องสล็อตอีก 2 ช่องรวมเป็น 5 ช่อง เป็นการปรับแต่งส่วนต่อประสานในส่วนโลกของเกม                        | เดียเจติก                             |
| การบำรุงรักษา       | เพิ่มตู้สล็อตอีก 1 ตู้ เป็นการปรับแต่งส่วนต่อประสานในส่วนโลกของเกม                                   | เดียเจติก                             |

ทั้งนี้จะเห็นได้ว่าการทดสอบประเมินผลสามารถทดสอบประกอบกับหลักการการออกแบบส่วนต่อประสานไดเจสซิสครอบคลุมในส่วนของหลักการผนังสี่ด้านเท่านั้น เนื่องจากข้อจำกัดของแนวคิดการมอดซึ่งไม่สอดคล้องกับทุกประเภทของหลักการและขอบเขตของซอฟต์แวร์เกมทดสอบ

นอกจากนี้แบบรูปยังอาจจะสามารถใช้กับการปรับแต่งได้หลายหลักการซึ่งไม่สามารถประเมินได้ด้วยซอฟต์แวร์เกมทดสอบเพียงกลุ่มเดียว

### 5.5 สรุปผลการทดสอบและวิเคราะห์จากผลการทดสอบ

วัตถุประสงค์ของวิทยานิพนธ์ฉบับนี้คือ การพัฒนาแบบรูปเพื่อจัดการสิ่งผิดปกติจากการทำส่วนขยายซอฟต์แวร์ ดังนั้นในขั้นตอนการวิเคราะห์ผลการออกแบบ อ้างอิงจากผลของกรณีทดสอบข้อผิดพลาดที่เกิดขึ้นควรจะถูกจัดการแก้ปัญหาทั้งหมดให้ผลการทำงานสามารถทำงานได้ตามปกติพิจารณาจากจุดประสงค์การปรับแต่งและวิธีการแก้ปัญหาของแบบรูป จากการวิเคราะห์พบว่าแบบรูปสามารถจัดการข้อผิดพลาดที่เกิดขึ้นได้เป็นไปตามจุดประสงค์การออกแบบแบบรูปและกรณีการปรับแต่งในแต่ละกรณี และพบว่าแบบรูปที่ออกแบบในนั้นเป็นโครงสร้างในการจัดการข้อผิดพลาดเฉพาะเจาะจงกับเหตุผิดปกติที่เกิดขึ้น ซึ่งในรายละเอียดความผิดปกตินั้นสามารถปรับเปลี่ยนได้ตามความต้องการพัฒนาทำให้รายละเอียดภายในของแบบรูปจำเป็นต้องปรับเปลี่ยนตามกรณีนั้นๆตามไปด้วย โครงสร้างสามารถยึดหลักตามเดิมได้เฉพาะกรณีที่การปรับแต่งยังอยู่ในขอบเขตเดิม ซึ่งรายละเอียดผลการทดสอบอยู่ในภาคผนวก ข โดยครอบคลุมในทั้ง 6 กรณีการปรับแต่ง แต่เมื่อเทียบกับหลักการการออกแบบส่วนต่อประสานโต้ตอบผู้ใช้สละสลวยเฉพาะส่วนของผนังสี่ด้านเท่านั้น ทั้งนี้เป็นการเทียบเพียงหนึ่งแบบรูปต่อหนึ่งหลักการการออกแบบส่วนต่อประสาน มีความเป็นไปได้ว่าหนึ่งแบบรูปสามารถครอบคลุมหลักการการออกแบบส่วนต่อประสาน

ทั้งนี้แบบรูปที่ออกแบบสามารถใช้เป็นแม่แบบการประยุกต์ใช้งานในการมอดเกมอื่นในอนาคตได้ต่อไปเนื่องจากซอฟต์แวร์สามารถถูกปรับเปลี่ยนได้ตลอดเวลาตามความต้องการ ทรัพยากรสภาพแวดล้อมของระบบหรือสาเหตุอื่นๆนอกเหนือจากที่กล่าวมา ฉะนั้นการปรับแต่งที่เป็นนวัตกรรมใหม่ๆ สามารถเกิดขึ้นได้เสมอ แบบรูปที่ออกแบบเป็นการนำเสนอแนวทางในการแก้ปัญหาจากตัวอย่างกรณีปรับแต่งที่เกิดขึ้นในงานวิจัยเพื่อใช้อ้างอิงในการปรับแต่งต่อไปในอนาคต

### 5.6 การทดลองด้วยซอฟต์แวร์เกมตัวอื่นๆ

จากการทดลองที่กล่าวมาทางผู้วิจัยได้ทดลองนำวิธีการดังกล่าวไปทดสอบกับซอฟต์แวร์เกมประเภทอื่นๆอีก 2 ประเภท ได้แก่ เกมพัชเชิล และเกมแข่งรถ เพื่อเป็นการประเมินความถูกต้องของแบบรูปอีกระดับหนึ่ง

เกมพัชเชิลที่นำมาใช้ทดสอบคือ เกม ู ซึ่งมีลักษณะการเล่นเป็นบังคับงูให้กินอาหาร เมื่องูกินอาหารทุกครั้งความยาวของงูจะเพิ่มขึ้น เกมจะจบเมื่องูเลื้อยชนกำแพงหรือกินตัวเอง มีรายละเอียดซอฟต์แวร์ต้นฉบับตามภาคผนวก ฉ [23]

เกมแข่งรถ คือเกมแข่งรถที่มีลักษณะการเล่นให้บังคับรถวิ่งในสนามและจับเวลาเมื่อหาเวลาวิ่งที่ดีที่สุดในแต่ละรอบ โดนมึรายละเอียดซอฟต์แวร์ต้นฉบับตามภาคผนวก ซ [24]

โดยซึ่งมีรายละเอียดการทดสอบอ้างอิงจากหัวข้อที่ 5.1-5.4 รายละเอียดปรากฏในภาคผนวก ซ และ ฅ ตามลำดับเกมที่น่าเสนอ ช้างต้น

## บทที่ 6

### สรุปผลการวิจัย

#### 6.1 บทสรุปของผลงานวิจัย

งานวิทยานิพนธ์ฉบับนี้ได้นำเสนอแนวทางการออกแบบและประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติเพื่อจัดการข้อผิดพลาดที่เกิดขึ้นในการปรับแต่งซอฟต์แวร์เกม โดยการศึกษา วิเคราะห์ กรณีการปรับแต่งและการออกแบบ พัฒนาเกมเอ็นจิน เพื่อกำหนดเป็นกรณี วัตถุประสงค์ ขั้นตอนในการปรับแต่งซอฟต์แวร์เกมโดยผู้ใช้ชั้นปลาย มุ่งเน้นในการปรับแต่งกรณีส่วนต่อประสาน โดยศึกษาค้นคว้ากรณีการปรับแต่งส่วนต่อประสานของเกมเพื่อให้ทราบถึงจุดประสงค์ในการปรับแต่งแต่ละกรณี ผลลัพธ์ที่ได้คือรายการการปรับแต่ง 6 รายการ หลังจากนั้นศึกษาถึงขั้นตอนในการออกแบบพัฒนาเอ็นจินเกมซึ่งเป็นโครงสร้างของระบบเกมเพื่อนำมาออกแบบโครงสร้างของเกมให้เหมาะสมแก่การปรับแต่งซอฟต์แวร์เกมสำหรับผู้ใช้ชั้นปลาย ซึ่งช่วยในการคัดกรองผลการทำงานของการปรับแต่งเกม ทำความเข้าใจได้ง่าย และสามารถรองรับการคัดกรองข้อผิดพลาด และการนำเสนอข้อผิดพลาดแก่ผู้ใช้ได้ดียิ่งขึ้น

ในขั้นตอนต่อมา นำรายการการปรับแต่งทั้ง 6 กรณีมาออกแบบเพื่อคัดกรองความผิดพลาดที่สามารถเกิดขึ้นได้แก่กรณีและผนวกการออกแบบร่วมกับส่วนรองรับการปรับแต่งของเกมเอ็นจินหรือส่วนช่วยเหลือส่วนขยายซอฟต์แวร์ที่ออกแบบไว้ โดยทำการออกแบบแบบรูปสำหรับกรณีการปรับแต่งจากการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ โดยการออกแบบดังกล่าวนี้มีการตรวจสอบประเมินความถูกต้องของการออกแบบ เพื่อทำการปรับปรุงให้มีความสมบูรณ์และเป็นแม่แบบสำหรับการประยุกต์ใช้ต่อไป ผ่านโอเพ่นซอร์ส

ผลลัพธ์ของงานวิทยานิพนธ์นี้คือ แบบรูปเพื่อควบคุมข้อผิดพลาดจากการปรับแต่งซอฟต์แวร์ตามกรณีการปรับแต่งส่วนต่อประสานของซอฟต์แวร์เกมทั้ง 6 กรณี โดยนำเสนอโครงสร้าง การประยุกต์ใช้การออกแบบรูปในการจัดการข้อผิดพลาดที่เกิดขึ้นในแต่ละกรณีการปรับแต่งพร้อมตัวอย่าง การทำงานของแบบรูปผ่านแผนภาพ ยูเอ็มแอล และ ลิตเติ้ลจิล เพื่อให้ง่ายแก่การทำความเข้าใจและเป็นอ้างอิงในการใช้งาน ทั้งนี้การออกแบบแบบรูปเพื่อจัดการข้อผิดพลาดของการปรับแต่ง จะช่วยในการลดข้อผิดพลาดจากการปรับแต่งเกมที่นำเสนอต่อผู้ใช้ อีกทั้งยังช่วยลดปัญหาข้อผิดพลาดที่เกิดขึ้นจากการปรับแต่งเกมโดยผู้ใช้ชั้นปลายและสนับสนุนการปรับแต่งเกม สร้างความเข้าใจในข้อผิดพลาดที่เกิดขึ้นแก่ผู้ใช้ชั้นปลาย

## 6.2 ข้อจำกัดของงานวิจัย

1. กรณีการออกแบบส่วนต่อประสานซึ่งในส่วนของรายละเอียดแบบรูปนั้นไม่สามารถครอบคลุมในเชิงรายละเอียดข้อผิดพลาดทั้งหมดที่เกิดขึ้นได้เนื่องจาก กรณีการปรับแต่งในเชิงรายละเอียดนั้นสามารถเปลี่ยนแปลงหรือเกิดรูปแบบใหม่ได้ตลอดเวลา เนื่องจากผู้ใช้มีแนวคิดนวัตกรรมใหม่ออกมาอยู่เสมอทำให้การออกแบบเชิงรายละเอียดนั้นมักจะสามารถควบคุมได้เมื่อข้อผิดพลาดเกิดขึ้นแล้วมากกว่า นอกจากนี้ระบบ สภาพแวดล้อมของระบบเกมยังสามารถปรับเปลี่ยนตลอดเวลาตามเทคโนโลยีที่พัฒนาขึ้นซึ่งไม่สามารถคาดเดาล่วงหน้าได้ ดังรูปแบบรูปที่นำเสนอจึงเป็นเพียงโครงสร้างเพื่อรองรับการนำไปประยุกต์ใช้ต่อไปในกรณีความผิดพลาดที่เกิดขึ้นในอนาคต
2. แบบรูปที่ออกแบบสามารถครอบคลุมในส่วนหลักการการออกแบบส่วนต่อประสานเกมได้เจตคติเพียง 4 หลักการจาก 5 หลักการเท่านั้นนอกจากนี้ยังเป็นการประเมินความครอบคลุมแบบหนึ่งต่อหนึ่ง ดังนั้น 1 แบบรูปอาจจะสามารถครอบคลุมหลักการการออกแบบได้มากกว่าหนึ่งหลักการ

## 6.3 งานวิจัยในอนาคต

1. การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติจากการปรับแต่งส่วนต่อประสานสำหรับเกมที่มีผู้เล่นหลายคนเนื่องจากเกมประเภทดังกล่าวจะมีตรรกะพิเศษในการเลือกนำเสนอข้อมูลแก่ผู้ใช้หลายคนพร้อมกันแตกต่างกันไป ทั้งนี้กรณีปรับแต่งที่เกิดขึ้นอาจมีผลถึงตรรกะพิเศษดังกล่าว
2. การออกแบบแบบรูป โดยการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติในการปรับแต่งรูปแบบอื่นๆ ได้ การแปลงเกม การเปลี่ยนรูปแบบ การเจาะระบบเกมปิด เพื่อให้เกิดความครอบคลุมในส่วนการมอดเกมทุกกรณี

## รายการอ้างอิง

- [1] Burnett M., Cook C. and Rothermel G., End-User Software Engineering. Communications of the ACM 47(9) 2004: 53–58
- [2] Stephen Tang and Martin Hanneghan, Game Content Model: An Ontology for Documenting Serious Game Design, Developments in E-systems Engineering 2011: p. 431-431
- [3] Eike Falk Anderson, Steffen E., Peter C. and Leigh M., The Case for Research in Game Engine Architecture. Conference on Future Play: Research, Play Proceeding Future Play '08 Proceedings of the 2008: Pages 228-231
- [4] Eva Hudlicka, Affective Game Engines: Motivation and Requirement, International Conference on Foundations of Digital Games FDG '09 Proceedings of the 4<sup>th</sup>: Pages 299-306
- [5] Seung Hun Lee , Gum Hee Lee , Hyun Hoon , Doo Heon Song and Sung Yul Rhew, An Empirical Model of the Game Software Development Processes, Proceedings of the 4th International Conference on Software Engineering Research Management and Applications (SERA'06)
- [6] Magy Seif, El-Nasr and Brian K Smith, Learning Through Game Modding, ACM Computers in Entertainment Vol. 4, No. 1, January 2006: p.1-20
- [7] Walt Scacchi, Modding as a Basis for Developing Game Systems. International Workshop on Games and Software Engineering GAS 2011: p. 5-8
- [8] Blizzard Entertainment, World of Warcraft [Online]. Available from: <http://us.battle.net/wow/en/> [2012, September 07]
- [9] Spiel2001, nUI [Online]. Available from: <http://www.curse.com/addons/wow/nui> [2012, September 07]



- [10] M. Burnett, What is end-user software engineering and why does it matter? , In 2nd International Symposium on End-User Development 2009: p 15–28
- [11] Andrew J.Ko., The State of the art in end-user software engineering. ACM Computer Survey 2011(21)
- [12] Model View Presenter (MVP) VS Model View Controller (MVC). [Online], Available from:<http://blog.vuscode.com/malovicn/archive/2007/12/18/model-view-presenter-mvp-vs-model-view-controller-mvc.aspx> [Last Accessed: September 03, 2012].
- [13] Martin Hunter, The MVPC Software Design Pattern, Tidying the House 2006: p 1-5
- [14] Ali Gokalp Peker and Tolga Can, A Design Goal and Design Pattern Based Approach for Development of Game Engines for Mobile Platforms. The 16th International Conference on Computer Games 2011
- [15] Eric Gamma, Recharad Helm, Ralph Johnson and John Vlisside, Design Pattern Elements of Reusable Object-oriented Software. 1 edition: Addison-Wesley Professional, 1998
- [16] Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra and Elisabeth Robson, Head First Design Patterns. 1 edition. United States of America: O'Reilly Media, 2011
- [17] Barbara Staudt Lerner and Stefan Christov, Exception Handling Patterns for Process Modeling, IEEE Transactions on Software Engineering Vol. 36, No. 2, March/April 2010: p.162-183
- [18] Barbara Staudt Lerner, Stefan Christov, Alexander Wise and Leon J. Osterweil, Exception Handling Patterns for Processes, WEH '08, Proceedings of the 4th international workshop on Exception handling: p 55-61
- [19] Serious Games and the Games Based Learning conference 2010, User interface design in video games [Online]. Available from: <http://www.thewanderlust.net/blog/2010/03/29/user-interface-design-in-video-games/> [2013 May 03].

- [20] A game development magazine, Video game user interface design: Diegesis theory [Online]. Available from:  
<http://devmag.org.za/2011/02/02/video-game-user-interface-design-diegesis-theory/> [2013, May 03]
- [21] Markus Schumacher , Eduardo Fernandez-Buglioni , Duane Hybertson , Frank Buschmann and Peter Sommerlad, Security Patterns: Integrating Security and Systems Engineering. England: Wiley Software Patterns Series, 2006
- [22] BncApps, Slot Machine (888) [Online]. Available from:  
<http://bncapps.com/vbnet-slot-machine/> [2013, March 03].
- [23] Joost Rijneveld, Snake [Online]. Available from:  
<http://active.tutsplus.com/tutorials/games/build-a-classic-snake-game-in-actionscript-2-0/> [2013, April 08]
- [24] Witchhut LLC All Rights Reserved, Racing Games [Online]. Available from:  
<http://www.gamesheep.com/free/flash-tutorials/1-flash-racing-game-1/step1.php> [2013, April 08]

ภาคผนวก

## ภาคผนวก ก

### อภิธานศัพท์

|                                 |   |
|---------------------------------|---|
| Game Engine                     | โครงสร้างพื้นฐานของระบบเกม เป็นสถาปัตยกรรมพื้นฐานของระบบเกม   |
| MOD (mod)                       | มีรากศัพท์มาจากคำว่า modification หรือ Modify ซึ่งคือการปรับแต่งเกมโดยผู้ใช้ชั้นปลาย  |
| End User                        | ผู้ใช้ชั้นปลาย คือผู้ใช้งานระบบ   |
| End User                        | การพัฒนาซอฟต์แวร์โดยผู้ใช้ชั้นปลาย  |
| Programming/End User Developing |   |
| End User Software Engineer      | กระบวนการ โครงสร้างที่ออกแบบมาเพื่อช่วยเหลือช่วยเหลือการพัฒนาซอฟต์แวร์โดยผู้ใช้ชั้นปลาย อาจจะอยู่ในคู่มือหรือเครื่องมือเพื่อช่วยในการจัดการโครงการพัฒนาซอฟต์แวร์ให้มีคุณภาพ |

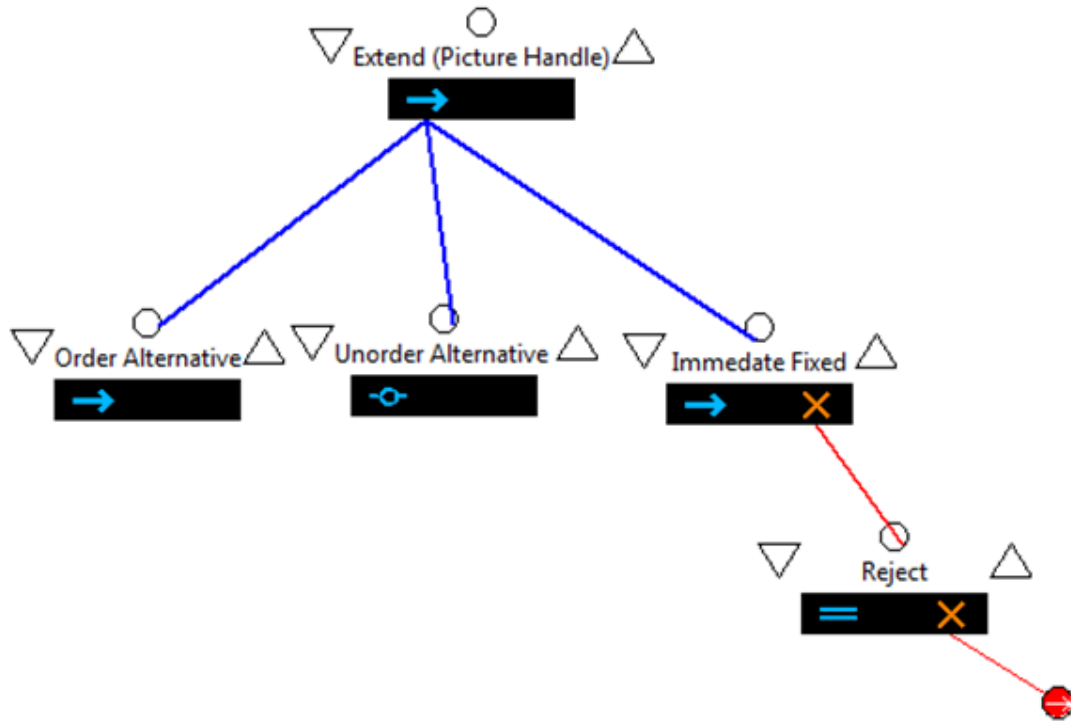
ภาคผนวก ข

โครงสร้างแบบรูปการประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ

เพื่อจัดการการนำเสนอข้อผิดพลาดจากการปรับปรุงส่วนต่อประสาน

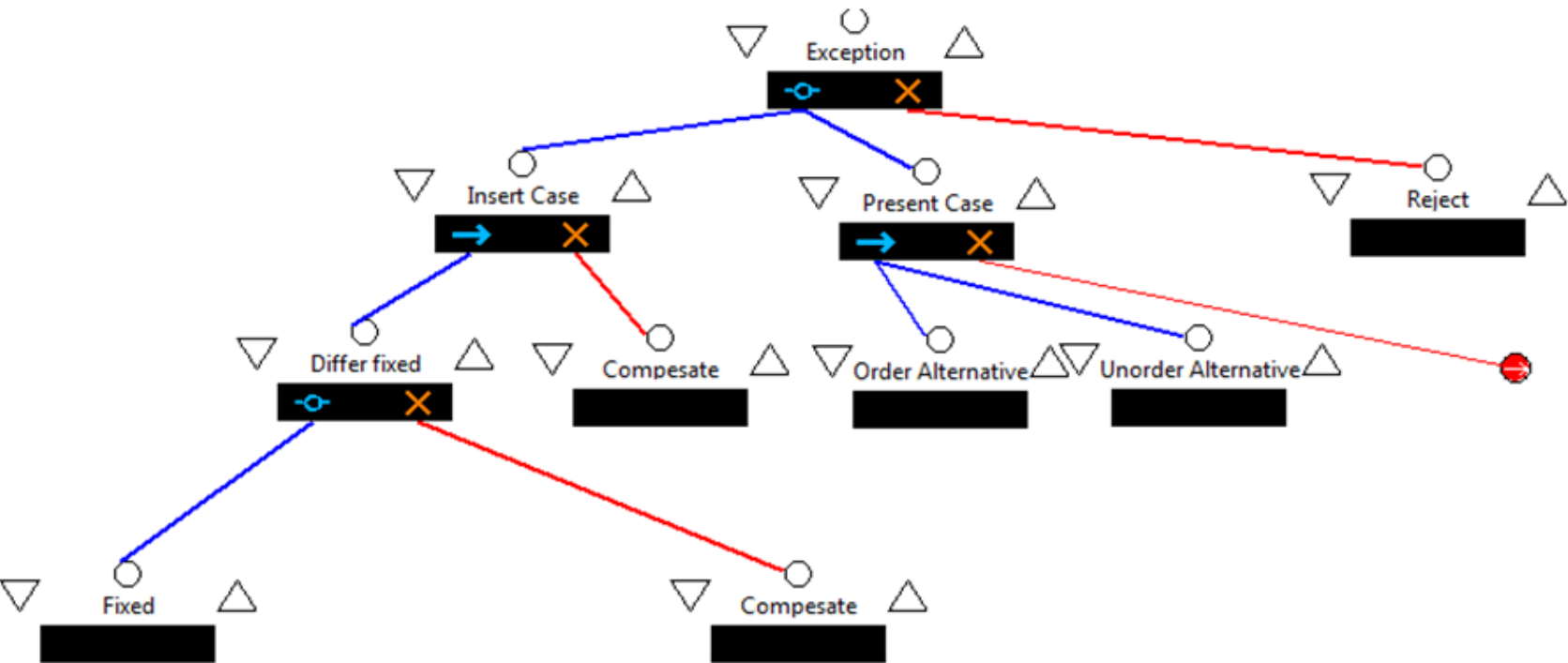
การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ นำเสนอโครงสร้างของแต่ละแบบรูปผ่านแผนภาพ  
ลิตเติ้ลจิล โดยอธิบายถึงองค์ประกอบต่างๆจากแบบรูปจัดการสิ่งผิดปรกติที่นำมาใช้ในทั้ง 6 กรณีการ  
ปรับแต่ง

ข.1. แบบรูปลำดับที่ 1: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน กรณีสื่อความพึงพอใจ



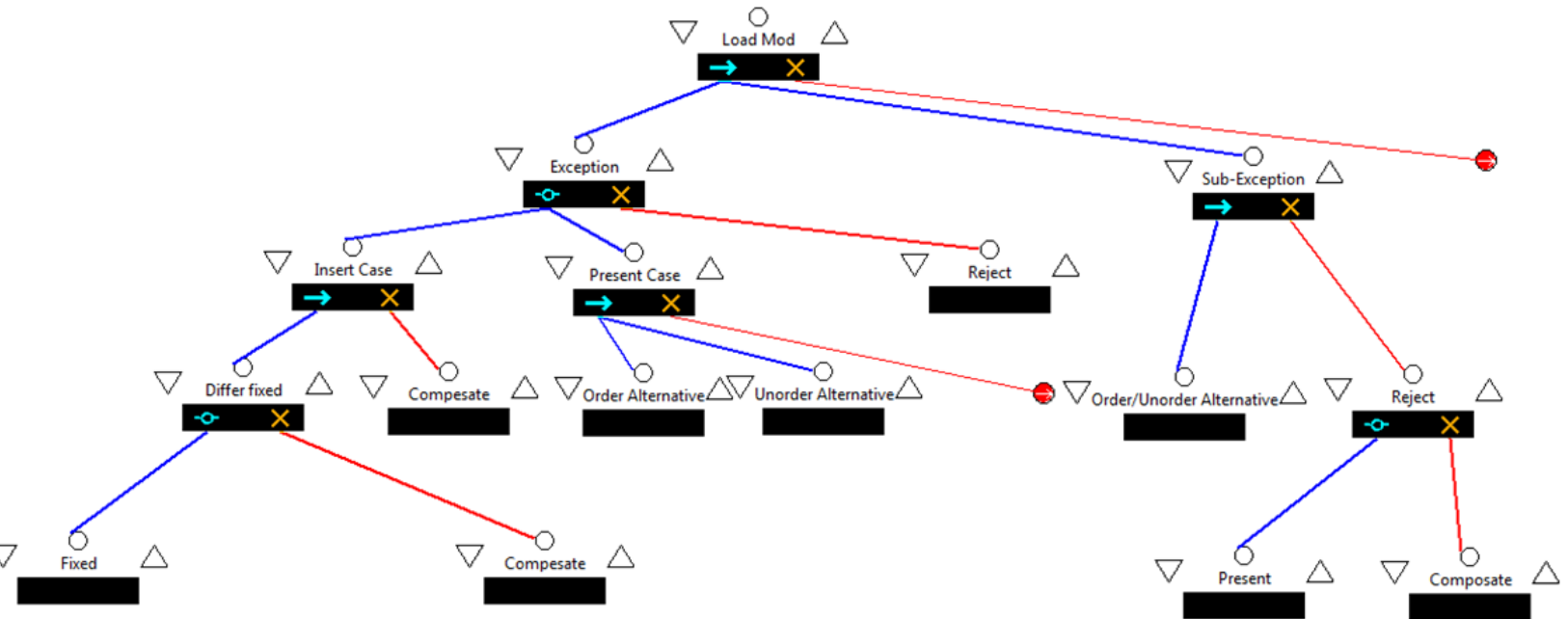
ภาพที่ ข.1 แผนภาพผลิตเต็ม-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อความพึงพอใจ

ข.2. แบบรูปลำดับที่ 2: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน  
 กรณีการปรับแต่งเพื่อการใช้งานเฉพาะ



รูปที่ ข.2 แผนภาพผลิตเต็ม-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปรกติกรณีการปรับแต่งเพื่อความพึงพอใจ  
 การใช้งานเฉพาะ

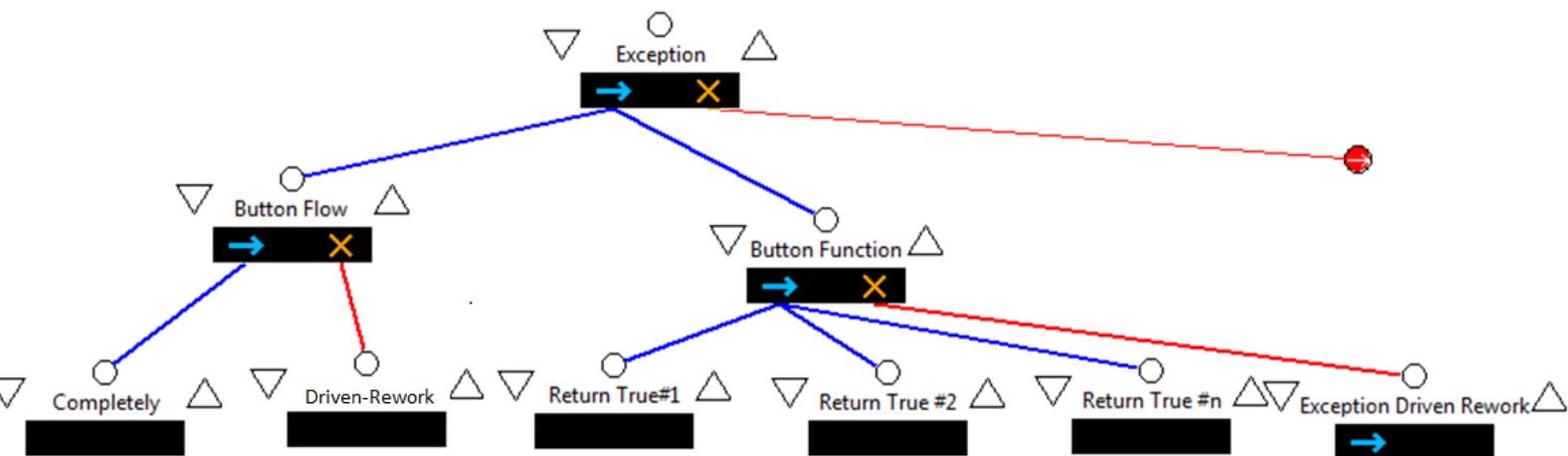
ข.3 แบบรูปลำดับที่ 3: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน  
 กรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน



รูปที่ ข.3 แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อเพิ่มฟังก์ชันงาน

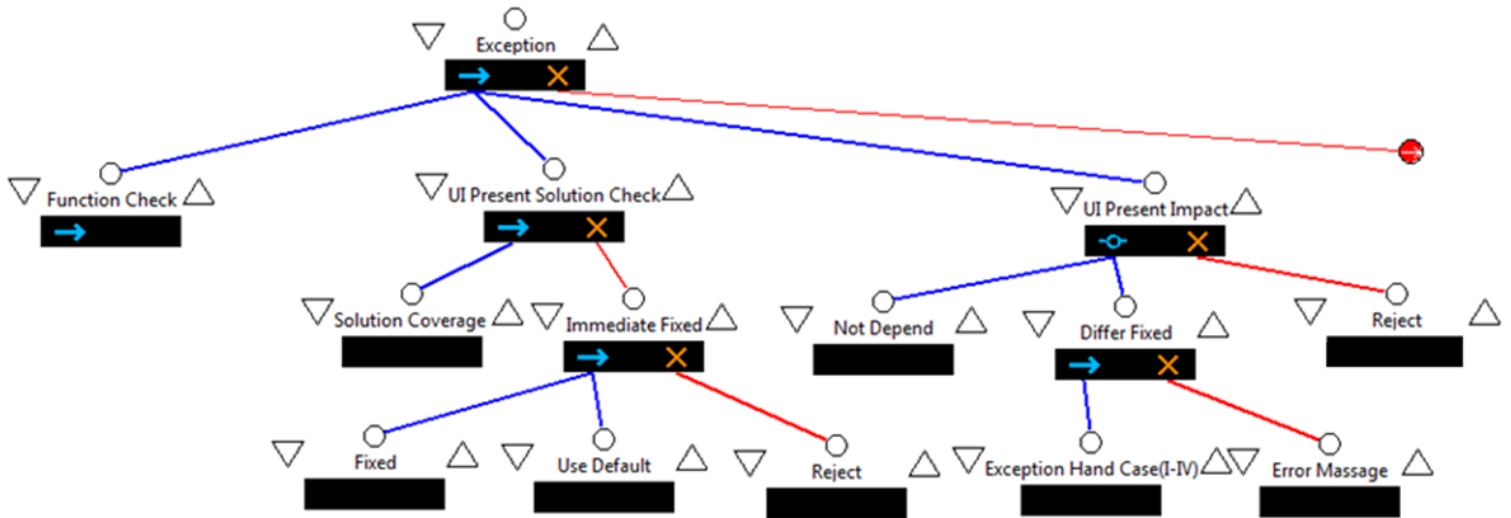


ข.4 แบบรูปลำดับที่ 4: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน  
กรณีการปรับแต่งเพื่อเพิ่มประสิทธิภาพ



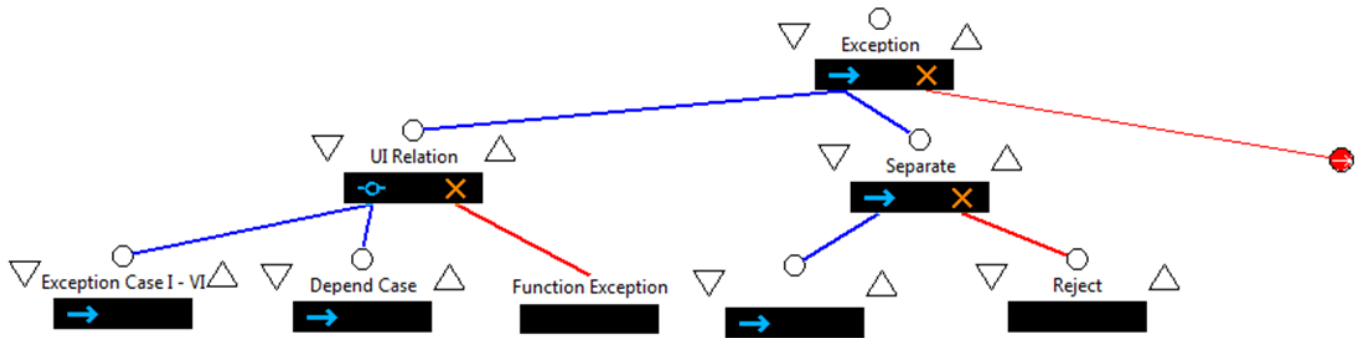
รูปที่ ข.4 แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อเพิ่มประสิทธิภาพ

ข.5 แบบรูปลำดับที่ 5: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน  
กรณีการปรับแต่งเพื่อการปรับตัว



ที่ ข.5 แผนภาพลิตเติ้ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อการปรับตัว

ข.6 แบบรูปลำดับที่ 6: การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติ เพื่อจัดการปัญหาการปรับปรุงส่วนต่อประสาน  
กรณีการปรับแต่งเพื่อการบำรุงรักษา



ที่ ข.6 แผนภาพผลิตเต็ล-จิล การประยุกต์ใช้แบบรูปจัดการสิ่งผิดปกติกรณีการปรับแต่งเพื่อการบำรุงรักษา

**ภาค ผนวก ค**  
**เกม สล็อตแมชชีน**

**ค.1 ข้อมูลจำเพาะ**

พัฒนาโดย <http://bncapps.com>

ภาษาที่ใช้พัฒนา : ภาษา วิชวลเบสิกคอตเน็ต (VB.NET)

ประเภทเกม : เกมกระดาน

เกมประเภทสล็อตแมชชีน เป็นเกมที่มีส่วนต่อประสานเป็นองค์ประกอบหลัก 3 ส่วนได้แก่

1. ช่องสล็อต เป็นตัวบอกแต้มของเกม โดยจำนวนช่องสล็อตเป็นเลขคู่ มีจำนวนตั้งแต่ 3 ช่อง สล็อตขึ้นไปตามที่ระบบกำหนด โดยจำนวนภาพเป็นมีจำนวนเป็นไปตามแต่ละเกมกำหนด โดยทั่วไปจะมี 3 ภาพเป็นอย่างต่ำได้แก่ ภาพเลข 7 ภาพผลเซอรัรี และภาพแถบบาร์ปกติซึ่งจะมีรูปแบบแตกต่างกันไป
2. คั่นโยก เป็นตัวทำหน้าที่เริ่มเกมในแต่ละรอบ
3. เหรียญ เป็นทำหน้าที่แทนจำนวนรอบเกมในแต่ละครั้ง จ่าย 1 เหรียญแทนการเล่นเกม 1 รอบ

กฎของเกมเป็นไปตามกฎของการเล่นเกมสล็อตแมชชีน ดังนี้

1. ในการเล่นแต่ละครั้งต้องจ่าย 1 เหรียญในการเล่น
2. เมื่อโยกคั่นโยก ช่องสล็อตจะทำการสุ่มค่าทุกช่อง
3. ผลการเล่นชนะเมื่อภาพบนช่องทั้งหมดเหมือนกัน
4. เมื่อชนะทุกรอบจะได้เหรียญเป็นการตอบแทน
5. เกมจบเมื่อเหรียญหมด

โหมดการเล่น ประกอบด้วย 2 โหมดได้แก่ โหมดปกติ และโหมดพลาแมวซึ่งโหมดดังกล่าวมีความแตกต่างจากโหมดปกติที่จะมีผลช่องเหลี่ยมกันระหว่างภาพได้ทำให้เกมยากขึ้นอีกระดับ

**ค.2 วิธีการเล่น**

1. เมื่อเริ่มเกมจะให้เลือกโหมด
2. เริ่มเกมระบบจะแสดงหน้าต่างถามชื่อผู้เล่นให้กรอก
3. ภายหลังกรอกแล้วจะขึ้นสู่เกม ดังภาพ
4. เริ่มเล่นเกมโดยการคลิกที่เหรียญ เหรียญจะวิ่งไปที่ช่องหยอดเหรียญของผู้เล่น
5. เมื่อเหรียญเข้าไปที่ตู้ ตู้เกมจะแสดงจำนวนเงินที่ตู้มี
6. โยกคั่นโยกเพื่อเริ่มเกม

7. ภาพในช่องสล็อตจะหมุนแบบสุ่ม
8. เมื่อภาพหยุดเกมจะคำนวณผลของภาพและคะแนนที่ได้
9. ถ้าผลของภาพตรงตามเงื่อนไขจะได้รับเหรียญตอบแทน
10. เกมจบเมื่อเหรียญหมดหรือปิดโปรแกรม

### ค.3 องค์ประกอบหน้าจอส่วนต่อประสานเกม

องค์ประกอบของส่วนต่อประสานอ้างอิงตามภาพ เป็นดังภาพที่ ค.1 มีรายละเอียดดังต่อไปนี้

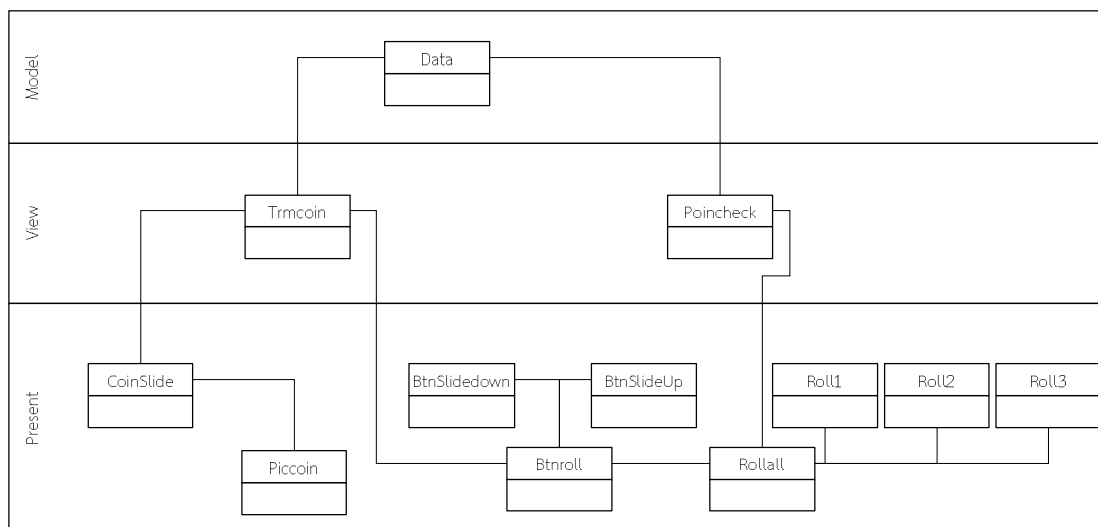


ภาพที่ ค.1 ส่วนต่อประสานของเกม

1. ชื่อผู้เล่น
2. กระเป๋าใส่เงิน แสดงข้อมูลเหรียญเงินที่มีสำหรับใช้จ่ายในการเล่น
3. คันโยกใช้สำหรับเริ่มเล่นเกม
4. สล็อตสำหรับแสดงผล
5. จำนวนเงินที่หยอดใส่ตู้
6. จำนวนเงินที่ได้รับเมื่อเล่นเกมชนะ
7. ช่องใส่เหรียญ

### ค.4 โครงสร้างระบบ

เดิมโครงสร้างของระบบไม่ได้ออกแบบให้อยู่ในลักษณะแบบรูป ทางผู้วิจัยได้ศึกษา วิเคราะห์ โครงสร้างและสร้างเป็นแผนภาพในลักษณะของแบบรูปได้ดังภาพที่ ค.2



ภาพที่ ค.2 แผนภาพคลาสโครงสร้างของเกมสล็อตแมชชีน

จากภาพข้างต้นแบ่งออกเป็นระดับใช้ได้ 3 ระดับดังนี้

- 1.โมเดล ระดับชั้นที่แสดงถึงข้อมูลในการทำงานของระบบเกม
- 2.วิว ชั้นที่เป็นศูนย์รวมตรรกะของระบบประกอบด้วย 2 องค์ประกอบได้แก่
  - 2.1 Trmcoin ทำหน้าที่ในการจัดการกับข้อมูลเหรียญที่ใช้ในเกม
  - 2.2 Poincheck ทำหน้าที่จัดการข้อมูลผลคะแนนของสล็อตโดยตีคะแนนที่ได้เป็นจำนวนเหรียญและส่งแกโมเดลต่อไป
- 3.พรีเซ้น ส่วนที่ทำหน้าที่ปฏิสัมพันธ์กับผู้ใช้ ซึ่งแบ่งออกได้เป็น 2 ชั้นย่อยได้แก่
  - 3.1 ส่วนปฏิสัมพันธ์กับผู้ใช้ เป็นชั้นที่รวมส่วนต่อประสานที่ผู้ใช้สามารถมีปฏิสัมพันธ์ได้โดยตรงได้แก่
    - 3.1.1 Piccoin ส่วนต่อประสานที่ให้ผู้เลือกในการหยอดเหรียญ
    - 3.1.2 Btnroll ส่วนต่อประสานเพื่อให้ผู้ใช้เลือกใช้งานเพื่อเลือกเล่นเกม
  - 3.2 ส่วนต่อประสานเพื่อนำเสนอข้อมูล ทำหน้าที่ในการนำเสนอข้อมูลแก่ผู้ใช้เพียงอย่างเดียวประกอบด้วย
    - 3.2.1 CoinSlide ทำหน้าที่นำเสนอภาพเหรียญวิ่ง
    - 3.2.2 Btnslidedown แสดงคั่นโยกเมื่อมีการใช้
    - 3.2.3 Btnslideup แสดงคั่นโยกเมื่อผ่านการโยกแล้วกลับที่เดิม
    - 3.2.4 Rollall แสดงการหมุนของสล็อต
    - 3.2.5 Roll1 นำเสนอการหมุนและหยุดของสล็อตแรก
    - 3.2.6 Roll2 นำเสนอการหมุนและหยุดของสล็อตที่สอง

### 3.2.7 Roll3 นำเสนอการหมุนและหยุดของสล๊อตที่สาม

ภาคผนวก ง

กรณีทดสอบทดสอบระบบต้นฉบับ

กรณีทดสอบที่นำมาแสดงในเอกสารประกอบวิทยานิพนธ์นี้ เป็นเพียงส่วนหนึ่งของกรณีทดสอบเพื่อหาข้อจำกัดของเอ็นจินเกม เนื่องจากในการทดสอบที่มีวัตถุประสงค์เดียวกันอาจมีกรณีทดสอบหลายชุด โดยชุดต้นแบบกรณีทดสอบซึ่งเป็นแบบร่างกรณีทดสอบหลังจากการทดสอบผลการทำงานของโปรแกรมเกมเพื่อหาข้อผิดพลาดครอบคลุมในตารางที่ ง.1-ง.6 ซึ่งในทุกๆกรณีจะมีการเขียนโปรแกรมเพิ่มเติมเพื่อให้ระบบสามารถทำงานตามการปรับแต่งได้ ยกเว้นกรณีที่ต้องเขียนโปรแกรมในการเข้าไปแก้ไขโครงสร้างของระบบ ทางผู้วิจัยจะนิยมว่าการเขียนในลักษณะดังกล่าวเป็นการปรับแต่งซึ่งไม่สอดคล้องกับหลักการเขียนโปรแกรมโดยผู้ใช้นั้น เนื่องจากกลุ่มผู้พัฒนา ลักษณะดังกล่าวจะแก้ไขซอฟต์แวร์โดยมีความรู้ในหลักการเขียนโปรแกรมระดับพื้นฐานเท่านั้น

ตารางที่ ง.1 กรณีทดสอบเปลี่ยนภาพในช่องสล็อต (T01)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T01 เปลี่ยนภาพในช่องสล็อต   |
| ฟังก์ชันที่ทดสอบ                | R01 ตรวจสอบการทำงานของระบบในภาพรวม  |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการปรับแต่งเพื่อความพึงพอใจ (แบบรูปลำดับที่ 1)  |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | การเปลี่ยนภาพในช่องสล็อต  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | นำเสนอภาพที่ปรับแต่งอย่างถูกต้อง มีขนาดภาพพอดีกับช่อง   |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | นำเสนอภาพที่ผิดปกติ หรือ นำเสนอภาพเดิมของระบบ   |
| ข้อมูลทดสอบ                     | ภาพดังภาพที่ ง.1 ซึ่งมีขนาด 96 x 172 pixel  |
| ผลการทดสอบ                      | ผลดังภาพที่ ง.2-ง.3   |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน<br><input checked="" type="checkbox"/> ไม่ผ่าน หมายเหตุ จากการศึกษพบว่า <u>นาสกุลของรูปภาพโปรแกรมรองรับปัญหาได้เนื่องจาก ภาษาที่ใช้พัฒนา มีรูปแบบที่รองรับได้</u> |





ภาพที่ ง.1 ภาพที่ใช้สำหรับการปรับแต่ง



ภาพที่ ง.2 ผลการทำงานก่อนทำการทดสอบ T01



ง.3 ผลกรณืทดสอบ T01

ตารางที่ ง.2 กรณืทดสอบเพิ่มข้อความนำเสนอข้อมูลเมื่อชนะเกม (T02)

|                                 |  |                                     |
|---------------------------------|--|-------------------------------------|
| ชื่อกรณืทดสอบ                   | T02  | เพิ่มข้อความนำเสนอข้อมูลเมื่อชนะเกม |
| ฟังก์ชันที่ทดสอบ                | R01  | ตรวจสอบการทำงานของระบบในภาพรวม      |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณืการปรับแต่งเพื่อการใช้งานเฉพาะ(แบบรูปลำดับที่ 2) |                                     |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ  |                                     |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดข้อความแสดงความยินดีเมื่อชนะเกม   |                                     |
| ผลลัพธ์ที่คาดหวัง (กรณืปกติ)    | ข้อความปรากฏขึ้นตามที่แจ้ง มีขนาดและตำแหน่งพอดีกับที่กำหนด   |                                     |
| ผลลัพธ์ที่คาดหวัง (กรณืผิดพลาด) | นำเสนอข้อความที่ผิดปกติ แสดงข้อความยินดีแต่ไม่หายไปเมื่อเริ่มเล่นเกมใหม่                                 |                                     |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดและข้อความภาพแสดงความยินดี ดังภาพที่ ง.4-ง.5  |                                     |
| ผลการทดสอบ                      | ผลดังภาพที่ ง.6-ง.7  |                                     |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน หมายเหตุ _____                 |                                     |

```
Private Sub ExtendSpecific_Congratz()  
    Try  
        Dim newPictureBox As New PictureBox  
        Dim nfile As String = Application.StartupPath & "\Images\images (4).jpg"  
        newPictureBox.Image = Image.FromFile(nfile)  
        newPictureBox.Visible = True  
        newPictureBox.Top = 60  
        newPictureBox.Width = 238  
        newPictureBox.Height = 60  
        newPictureBox.Left = 10  
        newPictureBox.Location = New Point(500, 24)  
        'add control to form.  
        Controls.Add(newPictureBox)  
        Catch ex As Exception  
        End Try  
End Sub
```

ภาพที่ ง.4 ซอร์ซโค้ดข้อความยินดี



ภาพที่ ง.5 ข้อความแสดงความยินดี



ภาพที่ ง.6 ผลการทำงานก่อนทำการทดสอบ T02

เพิ่มข้อความภาพแสดงความยินดีเมื่อได้รางวัล  
แต่ข้อความไม่หายไปเมื่อเริ่มเล่นรอบใหม่



ภาพที่ ง.7 ผลกรณืทดสอบ T02

ตารางที่ ง.3 กรณีทดสอบเพิ่มกราฟิกแสดงข้อมูลขณะใช้งาน (T03)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T03      เพิ่มกราฟิกแสดงข้อมูลขณะใช้งาน   |
| ฟังก์ชันที่ทดสอบ                | R01      ตรวจสอบการทำงานของระบบในภาพรวม   |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการปรับแต่งฟังก์ชันงาน (แบบรูปลำดับที่ 3) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดเพื่อนำเสนอกราฟิกแสดงเหรียญวิ่งออกจากตู้เกมเข้ากระเป๋าเมื่อชนะ                           |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | นำเสนอกราฟิกเหรียญวิ่งจากเครื่องเข้าสู่กระเป๋าเมื่อชนะ  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | นำเสนอผลการทำงานที่ปกติเช่นเหรียญวิ่งผิดทิศทางและเร็วจนมองตามไม่ทัน                               |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ ง.8  |
| ผลการทดสอบ                      | ระบบเกมค้าง ผลดังภาพที่ ง.9-ง.10  |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน    หมายเหตุ _____       |

```
Private Sub tmrcoinslidnewDown()  
    Dim tmrcoinsnewlide As New System.Windows.Forms.Timer(Me.components)  
    Dim x, y As Integer  
    Dim i As Integer  
    Me.piccoin.Location = New Point(601, Y_pincoin)  
    tmrcoinsnewlide.Enabled = True  
    For i = X_pincoin To 600  
        'tmrcoinsnewlide.Enabled = True  
        Me.piccoin.Left = Me.piccoin.Left + 45 '3  
        Me.piccoin.Top = Me.piccoin.Top + 45 '2  
        x = Me.piccoin.Left  
        y = Me.piccoin.Top  
        i = Me.piccoin.Left  
        System.Threading.Thread.Sleep(20)  
        Me.piccoin.Location = New Point(x, y)  
        ' tmrcoinslide.Enabled = False  
    Next  
    Me.piccoin.Location = New Point(600, 552)  
    tmrcoinslide.Enabled = False  
End Sub
```

ภาพที่ ง.8 ซอร์ซโค้ดการเพิ่มกราฟิกเหรียญวิ่ง



ภาพที่ ง.9 ผลการทำงานก่อนทำการทดสอบ T03

เพิ่มเหรียญวิ่งเข้ากระเป๋าเมื่อได้รางวัล



ภาพที่ ง.10 ผลกรณีทดสอบ T03

ตารางที่ ง.4 กรณีทดสอบการรวมปุ่มการทำงาน (T04)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T04 การรวมปุ่มการทำงาน  |
| ฟังก์ชันที่ทดสอบ                | R01 ตรวจสอบการทำงานของระบบในภาพรวม  |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการปรับแต่งประสิทธิภาพ (แบบรูปลำดับที่ 4) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดสำหรับรวมปุ่มการทำงาน หยอดเหรียญและโยกคันโยกเข้าด้วยกัน                                  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | ใช้ปุ่มคันโยกในการเล่นเกมนั่งทั้งที่หยอดเหรียญและโยกคันโยก  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | ทำให้ปุ่มโยกค้าง และเกิดข้อผิดพลาดในการเล่นต่อไม่ได้  |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ ง.11   |
| ผลการทดสอบ                      | ผลการทำงานเหมือนการทำงานปกติ แต่สามารถใช้ปุ่มโยกคู่แทนการหยอดเหรียญได้ ดังภาพที่ ง.12-ง.13        |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน หมายเหตุ _____          |

```
Private Sub Merge_Command()
```

```
    piccoin_Click(Me, EventArgs.Empty)
```

```
    tmrslidewdown.Enabled = True
```

```
End Sub
```

ภาพที่ ง.11 ซอร์ซโค้ดการรวมปุ่ม



ใช้การทำงานหลายจังหวะ ขั้นตอน



ภาพที่ ง.12 ผลการทำงานก่อนทำการทดสอบ T04

รวมปุ่มกด หยอดเหรียญและโยก  
คันโยกเข้าด้วยกัน  
แต่ทำให้ปุ่มโยกค้าง



ภาพที่ ง.13 ผลกรณีทดสอบ T04

ตารางที่ ง.5 กรณีทดสอบการเพิ่มฟังก์ชันใหม่ในระบบ (T05)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T05 การเพิ่มฟังก์ชันใหม่ในระบบ  |
| ฟังก์ชันที่ทดสอบ                | R01 ตรวจสอบการทำงานของระบบในภาพรวม  |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการปรับแต่งเพื่อการปรับตัว (แบบรูปลำดับที่ 5) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดสำหรับเพิ่มช่องในสล็อตในเกม  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | การเล่นเกมได้นำช่องสล็อตที่เพิ่มเข้าไป ร่วมในการเล่นด้วย  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | สล็อตที่เพิ่มไม่ถูกเรียกใช้   |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ ง.14   |
| ผลการทดสอบ                      | ผลดังภาพที่ ง.15-ง.16   |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน หมายเหตุ _____              |

```
Private Sub tmrollall_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrollall.Tick
    Randomize()
    If slot1rolling = True Then
        slot1 = Int(Rnd() * 10) + 1
        Select Case slot1
            Case 1 To 4
                lblslot1.Image = imgcherry.Image
                lblslot1.Tag = 1
            Case 5 To 7
                lblslot1.Image = imgbar.Image
                lblslot1.Tag = 2
            Case 8 To 9
                lblslot1.Image = imgdoublebar.Image
                lblslot1.Tag = 3
            Case 10
                lblslot1.Image = imgseven.Image
                lblslot1.Tag = 4
        End Select
    End If
    If slot2rolling = True Then
        slot2 = Int(Rnd() * 10) + 1
        Select Case slot2
            Case 1 To 4
                lblslot2.Image = imgcherry.Image
                lblslot2.Tag = 1
            Case 5 To 7
                lblslot2.Image = imgbar.Image
                lblslot2.Tag = 2
```

รูป ที่ ง.14 ซอร์ซโค้ดการเพิ่มฟังก์ชันใหม่ในระบบ

```
Case 8 To 9
    lblslot2.Image = imgdoublebar.Image
    lblslot2.Tag = 3
Case 10
    lblslot2.Image = imgseven.Image
    lblslot2.Tag = 4
End Select
End If
If slot3rolling = True Then
    slot3 = Int(Rnd() * 10) + 1
    Select Case slot3
        Case 1 To 4
            lblslot3.Image = imgcherry.Image
            lblslot3.Tag = 1
        Case 5 To 7
            lblslot3.Image = imgbar.Image
            lblslot3.Tag = 2
        Case 8 To 9
            lblslot3.Image = imgdoublebar.Image
            lblslot3.Tag = 3
        Case 10
            lblslot3.Image = imgseven.Image
            lblslot3.Tag = 4
    End Select
End If
If slotirolling = True Then
    slot4 = Int(Rnd() * 10) + 1
    Select Case slot4
```

รูป ที่ ง.14 ซอร์ซโค้ดการเพิ่มฟังก์ชันใหม่ในระบบ (ต่อ 1)

```
Case 1 To 4
    lblslot4.Image = imgcherry.Image
    lblslot4.Tag = 1
Case 5 To 7
    lblslot4.Image = imgbar.Image
    lblslot4.Tag = 2
Case 8 To 9
    lblslot4.Image = imgdoublebar.Image
    lblslot4.Tag = 3
Case 10
    lblslot4.Image = imgseven.Image
    lblslot4.Tag = 4
End Select
slot5 = Int(Rnd() * 10) + 1
Select Case slot3
    Case 1 To 5
        lblslot5.Image = imgcherry.Image
        lblslot5.Tag = 1
    Case 5 To 7
        lblslot5.Image = imgbar.Image
        lblslot5.Tag = 2
    Case 8 To 9
        lblslot5.Image = imgdoublebar.Image
        lblslot5.Tag = 3
    Case 10
        lblslot5.Image = imgseven.Image
        lblslot5.Tag = 4
End Select
End If
End Sub
```

รูป ที่ ง.14 ซอร์ซโค้ดการเพิ่มฟังก์ชันใหม่ในระบบ (ต่อ 2)



ภาพที่ ง.15 ผลการทำงานก่อนทำการทดสอบ T05

เพิ่มช่องสล๊อตเป็น 5 ช่อง 2 ช่องที่เพิ่มทำงานผิดปกติ



รูป ที่ ง.16 ผลกรณีทดสอบ T05

ตารางที่ ง.6 กรณีทดสอบการบำรุงรักษาระบบเมื่อมีการปรับแต่ง (T06)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T06 การบำรุงรักษาระบบเมื่อมีการปรับแต่ง   |
| ฟังก์ชันที่ทดสอบ                | R01 ตรวจสอบการทำงานของระบบในภาพรวม  |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการปรับแต่งเพื่อการบำรุงรักษา (แบบรูปลำดับที่ 6)  |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มตู้เกมสล็อตอีก 1 ตู้   |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | การเล่นเกมได้นำตู้เกมสล็อตที่เพิ่มเข้าไป ร่วมในการเล่นด้วย  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | ตู้เกมที่ถูกเพิ่มไม่ถูกเรียกใช้   |
| ข้อมูลทดสอบ                     | กรณีทดสอบดังกล่าว จำต้องมีการเขียนซอร์ซโค้ดเข้าไปเพิ่มเติมเพื่อให้ระบบรู้จักและทราบถึงมีการใช้งานอยู่ ซึ่งขึ้นกับรูปแบบในการเขียน (ซอร์ซโค้ดที่เขียนเข้าไปมีเพียงซอร์ซโค้ดสำหรับสร้างภาพเท่านั้น) |
| ผลการทดสอบ                      | เกมไม่สามารถทำงานได้เนื่องจากไม่ทราบถึงตู้ที่ต้องการเล่น ภาพที่ ง.17-ง.18   |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน หมายเหตุ ไม่สามารถทำได้ถ้าไม่ศึกษา โค้ดของซอฟต์แวร์   |



ภาพที่ ง.17 ผลการทำงานก่อนทำการทดสอบ T06

เพิ่มตู้สล็อตอีก 1 ตู้  
ตัวระบบไม่ทราบถึง  
การมีอยู่ของตู้ที่เพิ่ม



รูป ที่ ง.18 ผลกรณีทดสอบ T06



ภาคผนวก จ

กรณีทดสอบของแบบรูป

กรณีทดสอบที่นำมาแสดงในเอกสารประกอบวิทยานิพนธ์นี้ เป็นส่วนหนึ่งของกรณีทดสอบ เพื่อตรวจสอบความถูกต้องของแบบรูป โดยนำเสนอผ่านตารางที่ จ.1-จ.6

ตาราง จ.1 รายละเอียดกรณีทดสอบ แบบรูปกรณีการใช้งานเฉพาะโดยการเพิ่มข้อความนำเสนอข้อมูลเมื่อชนะเกม (T012)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T012 แบบรูปกรณีการใช้งานเฉพาะโดยการเพิ่มข้อความนำเสนอข้อมูลเมื่อชนะเกม  |
| ฟังก์ชันที่ทดสอบ                | R012 ExtendSpecific_Congratz  |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีเพื่อการใช้งานเฉพาะ (แบบรูปลำดับที่ 2)                       |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดข้อความแสดงความยินดีเมื่อชนะเกม โดยมีโครงสร้างเป็นไปตามแบบรูปการจัดการการปรับแต่งกรณีเพื่อการใช้งานเฉพาะ |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | ข้อความปรากฏขึ้นตามที่แจ้ง มีขนาดและตำแหน่งพอดีกับที่กำหนด  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | แจ้งข้อความแสดงถึงสิ่งที่ผิดปกติ หรือนำเสนอภาพเดิมของระบบ   |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ จ.1-จ.2  |
| ผลการทดสอบ                      | ดังแสดงในภาพที่ จ. 2  |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน    หมายเหตุ                             |

```
Public Sub Exception()  
    Dim nfile As String = Application.StartupPath & "\Images\images (4).jpg  
    ExtendSpecific_Congratz()  
End Sub  
  
Private Sub ExtendSpecific_Congratz()  
    Try  
        newPictureBox.Image = Image.FromFile(nfile)  
        newPictureBox.Visible = True  
        newPictureBox.Top = 60  
        newPictureBox.Width = 238  
        newPictureBox.Height = 60  
        newPictureBox.Left = 10  
        newPictureBox.Location = New Point(500, 24)  
    Catch ex As Exception  
        MessageBox.Show(ex.ToString)  
    End Try  
    ExtendSpecific_Match()  
End Sub
```

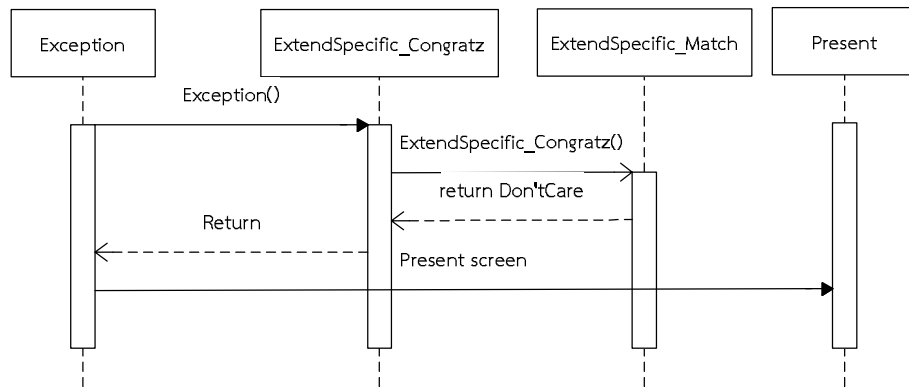
ภาพที่ จ.1 ซอร์ซโค้ดการเพิ่มข้อความภาพแสดงความยินดี

```
Private Sub ExtendSpecific_Match()  
    Try  
        If win = 1 Then  
            'add control to form.  
            Me.Controls.Add(newPictureBox)  
        Else  
            For Each c As Control In Me.Controls  
                If TypeOf c Is Button Or TypeOf c Is PictureBox Then  
                    If c.Name.StartsWith("newPictureBox") Then  
                        Me.Controls.Remove(c)  
                    End If  
                End If  
            Next  
        End If  
        Catch ex As Exception  
            MessageBox.Show(ex.ToString)  
        End Try  
    End Sub
```

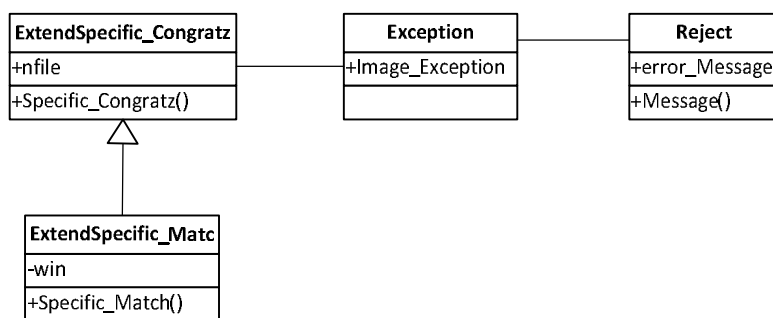
ภาพที่ จ.2 ซอร์ซโค้ดการแก้ไขการปรับแต่ง



ภาพที่ จ.3 ผลกรณืทดสอบ T012



ภาพที่ จ.4 แผนภาพลำดับ การทำงานของแบบรูปกรณืเพื่อการใช้งานเฉพาะ โดยการเพิ่มข้อความ  
แสดงความยินดีเมื่อชนะเกม



ภาพที่ จ.5 แผนภาพคลาส การทำงานของแบบรูปกรณีเพื่อการใช้งานเฉพาะ โดยการเพิ่มความ  
แสดงคามยินดีเมื่อชนะเกม

ตาราง จ.2 รายละเอียดกรณีทดสอบ แบบรูปกรณีการเพิ่มฟังก์ชัน โดยทดสอบเพิ่มกราฟิกแสดงข้อมูล  
ขณะใช้งาน (T013)

|                                 |  |
|---------------------------------|--|
| ชื่อกรณีทดสอบ                   | T013 แบบรูปกรณีการเพิ่มฟังก์ชัน ทดสอบโดยเพิ่มกราฟิกแสดงข้อมูลขณะใช้งาน   |
| ฟังก์ชันที่ทดสอบ                | R013 Exception_tmrcoinslidnewDown  |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีการปรับแต่งฟังก์ชันงาน (แบบรูปลำดับที่ 3)   |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ  |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดเพื่อนำเสนอกราฟิกแสดงเหรียญวิ่งออกจากตู้เกมเข้ากระเป๋าเมื่อชนะ โดยมีโครงสร้างเป็นไปตามแบบรูปการจัดการการปรับแต่งกรณีเพิ่มฟังก์ชัน |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | นำเสนอกราฟิกเหรียญวิ่งจากเครื่องเข้าสู่กระเป๋าเมื่อชนะ   |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | แจ้งข้อความแสดงถึงสิ่งที่ผิดปกติ หรือ นำเสนอภาพเดิมของระบบ   |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ จ.6-จ.8   |
| ผลการทดสอบ                      | ดังแสดงในภาพที่ จ.9  |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน      หมายเหตุ  |

```
Private Sub Exception()  
    Try  
        Dim tmrcoinsnewlide As New System.Windows.Forms.Timer(Me.components)  
        Dim x, y As Integer  
        Dim i As Integer  
        Me.piccoin.Location = New Point(601, Y_pincoin)  
        Exception_tmrcoinslidnewDown()  
        SubException_tmrcoinslidnewDown()  
        tmrcoinsnewlide.Enabled = True  
        For i = X_pincoin To 600  
            Me.piccoin.Left = Me.piccoin.Left + 45 '3  
            Me.piccoin.Top = Me.piccoin.Top + 45 '2  
            x = Me.piccoin.Left  
            y = Me.piccoin.Top  
            i = Me.piccoin.Left  
            System.Threading.Thread.Sleep(20)  
            Me.piccoin.Location = New Point(x, y)  
        Next  
        Me.piccoin.Location = New Point(600, 552)  
        tmrcoinslide.Enabled = False  
    Catch ex As Exception  
        MessageBox.Show(ex.ToString)  
    End Try  
End Sub
```

ภาพที่ จ.6 ซอร์ซโค้ดการเพิ่มกราฟิกเหรียญวิ่ง แกะไขให้เรียกใช้ แบบรูปได้

```

Private Sub SubException _tmrcoinslidnewDown()
    Try
        If Me.piccoin.Left > 600 Then
            X_pincoin = 60
            Y_pincoin = 182
            Me.piccoin.Location = New Point(X_pincoin, Y_pincoin)
        End If
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub

```

ภาพที่ จ.7 ซอร์ซโค้ดสำหรับตรวจสอบการแสดงผลของกราฟิก ให้แสดงผลได้ปกติ

```

Private Sub Exception _tmrcoinslidnewDown()
    Try
        If win = 1 Then
        Else
            MessageBox.Show("Error Scroll")
        End If
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
End Sub

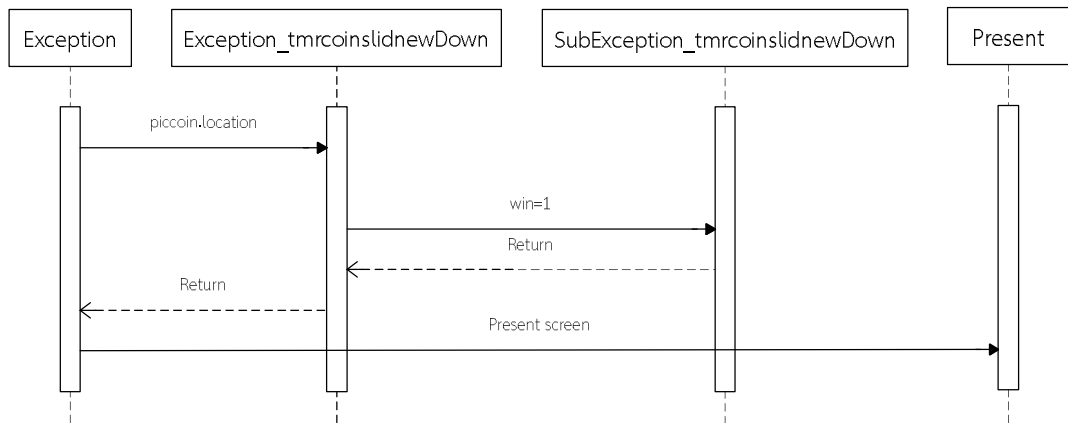
```

ภาพที่ จ.8 ซอร์ซโค้ดสำหรับตรวจสอบการทำงานของฟังก์ชัน ซึ่งทำให้กับการแสดงผลให้ถูกต้องตามเงื่อนไข

เพิ่มเหรียญวิ่งเข้ากระเ้าเมื่อได้รางวัล

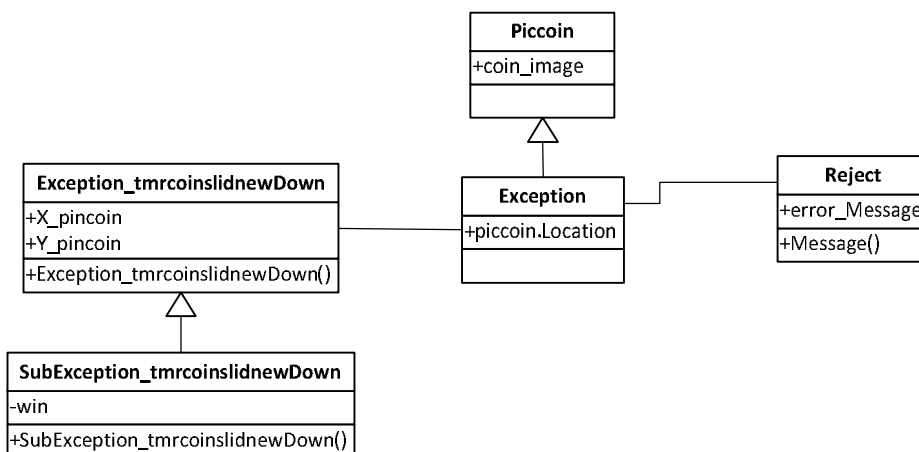


ภาพที่ จ.9 ผลกรณืทดสอบ T013



ภาพที่ จ.10 แผนภาพลำดับ การทำงานของแบบรูปกรณืการเพิ่มฟังก์ชัน โดยการเพิ่มกราฟิกแสดง ข้อมูลขณะใช้งาน





ภาพที่ จ.11 แผนภาพคลาส การทำงานของแบบรูปกรณีการเพิ่มฟังก์ชัน โดยการเพิ่มกราฟิกแสดงข้อมูลขณะใช้งาน

ตาราง จ.3 รายละเอียดกรณีทดสอบแบบรูปกรณีการเพิ่มฟังก์ชัน โดยทดสอบเพิ่มกราฟิกแสดงข้อมูลขณะใช้งาน (T014)

|                                 |  |
|---------------------------------|--|
| ชื่อกรณีทดสอบ                   | T014 แบบรูปกรณีเพื่อประสิทธิภาพ ทดสอบโดยการรวมปุ่มการทำงาน                                     |
| ฟังก์ชันที่ทดสอบ                | R014 Botton_Flow(),Botton_Function   |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีการปรับแต่งประสิทธิภาพ (แบบรูปลำดับที่ 4) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ  |
| ข้อมูลนำเข้า                    | เพิ่มโค้ดสำหรับรวมปุ่มการทำงาน หยอดเหรียญและโยกคันโยกเข้าด้วยกัน                               |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | ใช้ปุ่มคันโยกในการเล่นเกมนั่งทั้งทั้งหยอดเหรียญและโยกคันโยก                                    |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | โยกคันโยกครั้งแรกทำได้เพียง 1 กิจกรรมหรือฟังก์ชันเท่านั้น                                      |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดตั้งภาพที่ จ.12-จ.14  |
| ผลการทดสอบ                      | ดังแสดงในภาพที่ จ.15   |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน      หมายเหตุ        |

```

Private Sub ExceptionMerage_Comand()
Dim Merage_Command As Button
If Merage_Command.Visible = True Then
Button_Flow.Botton_Flow()
End If
End Sub

```

ภาพที่ จ.12 ซอร์ซโค้ดการรวมปุ่มการทำงานโดยมีโครงสร้างเป็นไปตามแบบรูปการจัดการการ  
ปรับแต่งกรณีเพื่อประสิทธิภาพ

```

Private Sub Botton_Flow()
If credits >= 1 Then
Form1.CoinPin_Comand()
Command_Control = Command_Control + 1
Else
Form1.Merage_Comand()
End If
End Sub

```

ภาพที่ จ.13 ซอร์ซโค้ดการตรวจสอบลำดับการทำงาน

```

Private Sub Botton_Function()
If Command_Control = 0 Then
Lastcommand = Merage_Comand
ElseIf Command_Control = 1 Then
Lastcommand = CoinPin_Comand
End If
credits = credits - 1
If Lastcommand <> “” Then
Call Lastcommand
End IF
End Sub

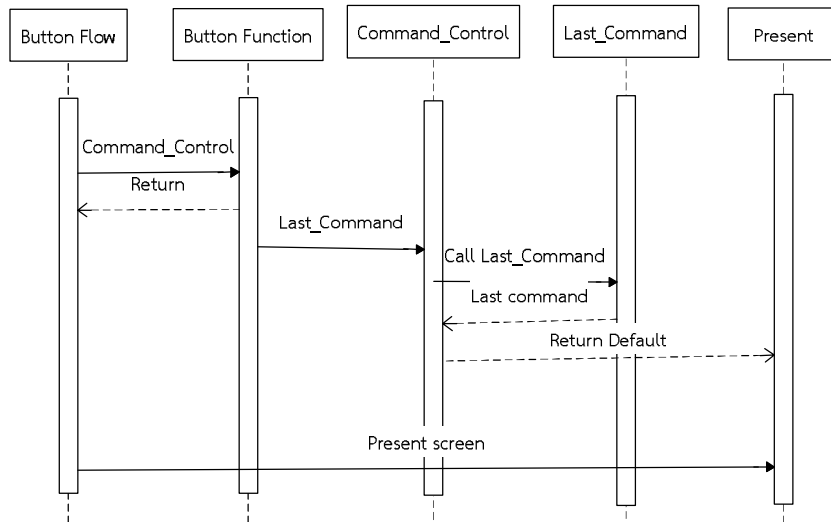
```

ภาพที่ จ.14 ซอร์ซโค้ดตรวจสอบความสมบูรณ์ของการทำงาน

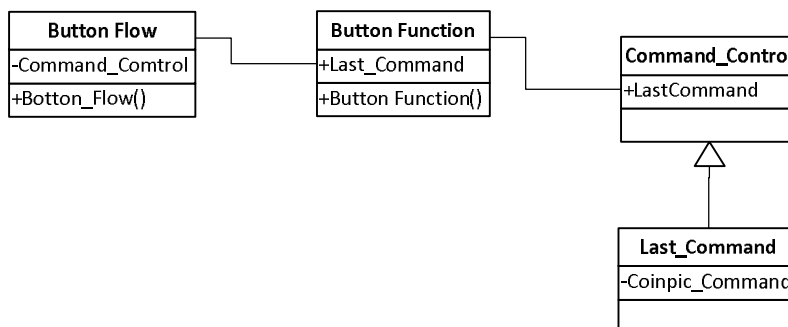
รวมปุ่มกด หยอดเหรียญและโยก  
คันโยกเข้าด้วยกัน



ภาพที่ จ.15 ผลกรณืทดสอบ T014



ภาพที่ จ.16 แผนภาพลำดับ การทำงานของแบบรูปกรณืเพื่อประสิทธิภาพ โดยการรวมปุ่มการทำงาน



ภาพที่ จ.17 แผนภาพคลาส การทำงานของแบบรูปกรณิเพื่อประสิทธิภาพ โดยการรวมปุมการทำงาน

ตาราง จ.4 รายละเอียดกรณิทดสอบ แบบรูปกรณิการปรับตัว โดยการเพิ่มฟังก์ชันใหม่ในระบบ (T015)

|                                 |  |  |
|---------------------------------|--|--|
| ชื่อกรณิทดสอบ                   | T015                                     | แบบรูปกรณิการปรับตัว ทดสอบโดยเพิ่มฟังก์ชันใหม่ในระบบ   |
| ฟังก์ชันที่ทดสอบ                | R015                                     | Function_Check(),Control_Check(), Adap_Exception   |
| วัตถุประสงค์การทดสอบ            |  | เพื่อตรวจสอบการทำงานของแบบรูปกรณิจัดการการปรับแต่กรณิการปรับแต่งเพื่อการปรับตัว (แบบรูปลำดับที่ 5) |
| ผู้ใช้งาน                       |  | ผู้ใช้งานระบบ  |
| ข้อมูลนำเข้า                    |  | เพิ่มโค้ดสำหรับเพิ่มช่องในสล็อตในเกม   |
| ผลลัพธ์ที่คาดหวัง (กรณิปกติ)    |  | การเล่นเกมนำช่องสล็อตที่เพิ่มเข้าไป ร่วมในการเล่นด้วย  |
| ผลลัพธ์ที่คาดหวัง (กรณิผิดพลาด) |  | สล็อตที่เพิ่มไม่ถูกเรียกใช้  |
| ข้อมูลทดสอบ                     |  | ซอร์ซโค้ดดังภาพที่ จ.17-จ.19   |
| ผลการทดสอบ (กรณิปกติ)           |  | ดังแสดงในภาพที่ จ.20   |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน | <input type="checkbox"/> ไม่ผ่าน      หมายเหตุ   |

```
Private Sub Control_Check()  
    Dim lbl_count As Integer  
    Dim ctrl As Control  
    Dim lblist As New ArrayList  
    For Each ctrl In Me.Controls  
        If (TypeOf ctrl Is Label) Then  
            lbl_count = lbl_count + 1  
            'do whatever you want with the label here  
            lblist.Add(ctrl)  
        End If  
    Next  
    Dim i As Integer  
    For i = 0 To lblist.Count - 1  
        ' Cast to a string.  
        Dim str As String = CObj(lblist.Item(i)).Name.ToString  
        If CObj(lblist.Item(i)).Location.Y = 232 Then  
            lblist_Exception.Add(lblist.Item(i))  
        End If  
    Next i  
    If lbl_count > 8 Then ' Over the Standard Lable  
        Function_Check()  
        Adap_Exception = 1  
    End If  
End Sub
```

ภาพที่ จ.18 ซอร์ซโค้ดส่วนการนำเสนอส่วนต่อประสาน

```
Private Sub Function_Check()  
    Dim i As Integer  
    For i = 0 To lblast_Exception.Count - 1  
        If CObj(lblast_Exception.Item(i)).Name.ToString = "lblslot1" Or  
Obj(lblast_Exception.Item(i)).Name.ToString = "lblslot2" Or  
CObj(lblast_Exception.Item(i)).Name.ToString = "lblslot3" Then  
            Else  
                lblast_Handle.Add(lblast_Exception.Item(i))  
                Adap_Exception = 1  
            End If  
        Next i  
    End Sub
```

ภาพที่ จ.19 ซอร์ซโค้ดการจัดการฟังก์ชันงาน

```
Private Sub tmrroll3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrroll3.Tick

    Dim slot3num As Int16
    Randomize()
    slot3num = Int(Rnd() * 10) + 1
    Select Case slot3num
        Case 1 To 4
            lblslot3.Image = imgcherry.Image
            lblslot3.Tag = 1
        Case 5 To 7
            lblslot3.Image = imgbar.Image
            lblslot3.Tag = 2
        Case 8 To 9
            lblslot3.Image = imgdoublebar.Image
            lblslot3.Tag = 3
        Case 10
            lblslot3.Image = imgseven.Image
            lblslot3.Tag = 4
    End Select
    slot3rolling = False
    tmrroll3.Enabled = False
    If Adap_Exception = 1 Then
        Dim i As Integer
        For i = 0 To lblist_Handle.Count
            tmrrolli_Tick()
        Next
    End If
```

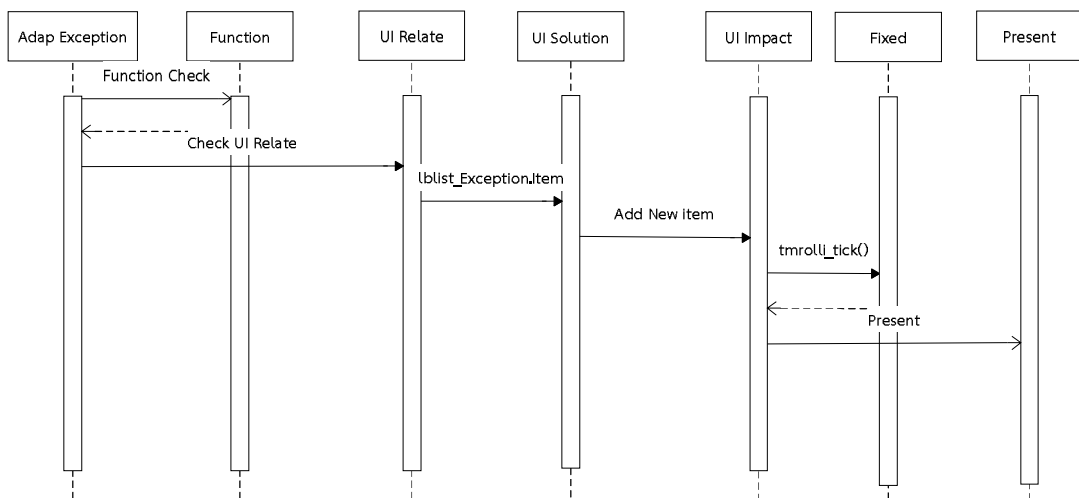
```

tmrrollall.Enabled = False
    pointcheck()
    btnroll.Enabled = True
End Sub
    
```

ภาพที่ จ.20 ซอร์ซโค้ดการจัดการผลกระทบส่วนต่อประสาน (ต่อ)

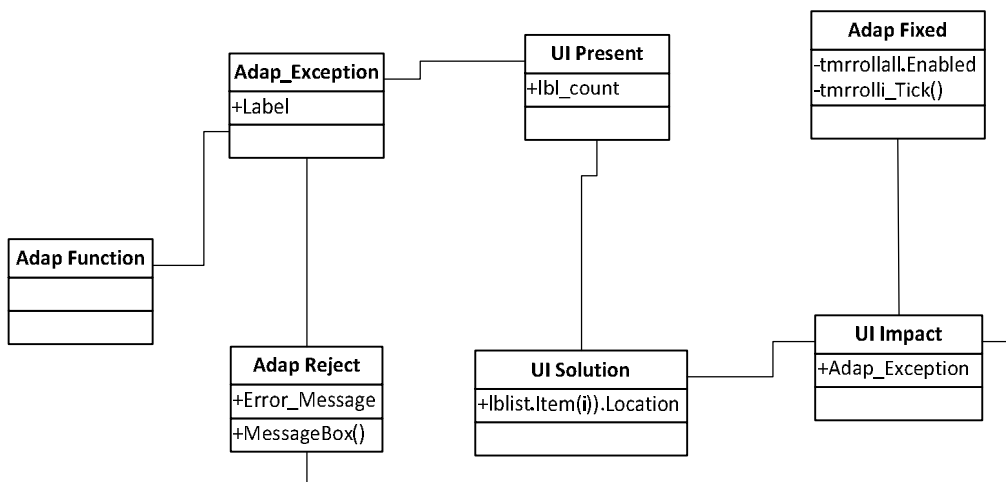


ภาพที่ จ.21 ผลกรณืทดสอบ T015



ภาพที่ จ.22 แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการปรับตัว โดยการเพิ่มฟังก์ชันใหม่





ภาพที่ จ.23 แผนภาพคลาส การทำงานของแบบรูปกรณีเพื่อการปรับตัว โดยการเพิ่มฟังก์ชันใหม่

ตาราง จ.5 รายละเอียดกรณีทดสอบ แบบรูปกรณีการบำรุงรักษา โดยการบำรุงรักษาระบบเมื่อมีการปรับแต่ง (T016)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T016 แบบรูปกรณีการบำรุงรักษา ทดสอบโดยการบำรุงรักษาระบบเมื่อมีการปรับแต่ง                              |
| ฟังก์ชันที่ทดสอบ                | R016 ตรวจสอบการทำงานของระบบในภาพรวม   |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีการปรับแต่งเพื่อการบำรุงรักษา (แบบรูปลำดับที่ 6) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | เพิ่มตู้เกมสล็อตอีก 1 ตู้   |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | การเล่นเกมนำตู้เกมสล็อตที่เพิ่มเข้าไป ร่วมในการเล่นด้วย   |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | ตู้เกมที่ถูกเพิ่มไม่ถูกเรียกใช้   |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดตั้งภาพที่ จ.22-จ.24   |
| ผลการทดสอบ (กรณีปกติ)           | ดังแสดงในภาพที่ จ.25  |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน      หมายเหตุ               |

```
Private Sub Control_Check()  
    Dim lbl_count As Integer  
    Dim lblist As New ArrayList  
    Dim ctrl As Control  
    Dim ptb_count As Integer  
    Dim ptblist As New ArrayList  
    For Each ctrl In Me.Controls  
        If (TypeOf ctrl Is Label) Then  
            lbl_count = lbl_count + 1  
            lblist.Add(ctrl)  
        ElseIf (TypeOf ctrl Is PictureBox) Then  
            ptb_count = ptb_count + 1  
            ptblist.Add(ctrl)  
        End If  
    Next  
    Dim i As Integer  
    For i = 0 To lblist.Count - 1  
        Dim str As String = CObj(lblist.Item(i)).Name.ToString  
        If CObj(lblist.Item(i)).Location.Y = 232 Then  
            lblist_Exception.Add(lblist.Item(i))  
        End If  
    Next i  
    For i = 0 To ptblist.Count - 1  
        Dim str As String = CObj(ptblist.Item(i)).Name.ToString  
        If CObj(ptblist.Item(i)).Location.Y = -8 Then  
            ptblist_Exception.Add(ptblist.Item(i))  
        End If  
    Next i
```

ภาพที่ จ.24 ซอร์ซโค้ดการบำรุงรักษาระบบโดยการตรวจสอบความสอดคล้อง  
ของส่วนต่อประสาน

```
If lbl_count > 8 Then ' Over the Standard Lable
    End If
If ptb_count > 14 Then
    If ptblist_Exception.Count > 1 Then
        Maintain_Check()
    End If
End If
End Sub
Private Function Maintain_Check() As Integer
    Dim i As Integer
    For i = 0 To ptblist_Exception.Count - 1
        If CObj(ptblist_Exception.Item(i)).Name.ToString = "picslotmachine" Then
        Else
            ptblist_Handle.Add(ptblist_Exception.Item(i))
            MainTain_Exception = 1
        End If
    Next i
End Function
```

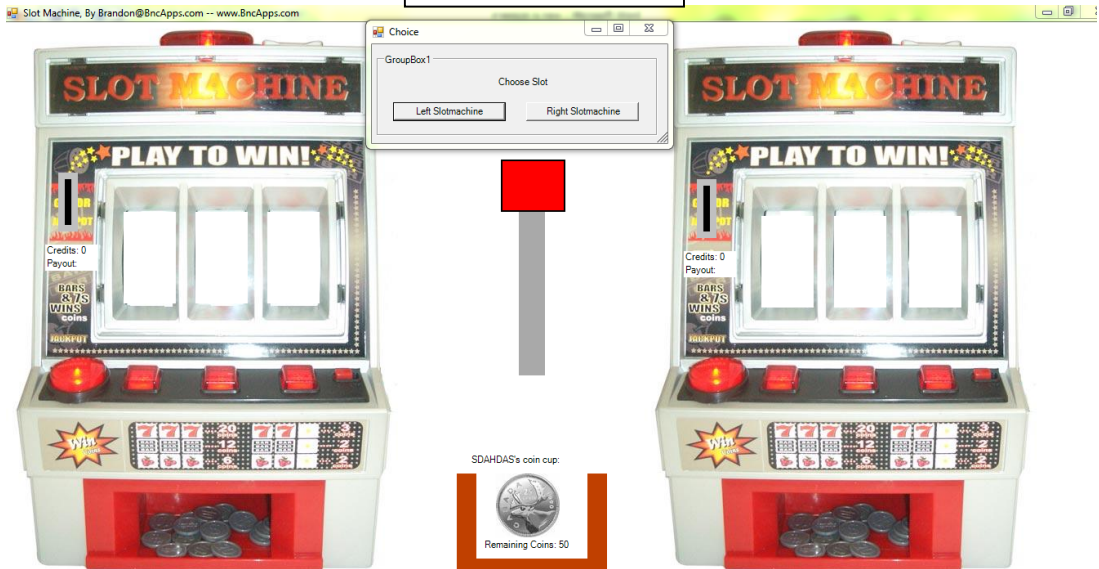
ภาพที่ จ.24 ซอร์ซโค้ดการบำรุงรักษาระบบโดยการตรวจสอบความสอดคล้อง  
ของส่วนต่อประสาน(ต่อ)

```
If MainTain_Exception = 1 Then
    Choice.ShowDialog()
    If SlotChoose = "RSlot" Then
        MessageBox.Show(SlotChoose)
        piccoin_Click(Me, EventArgs.Empty)
        Me.lblpayout.Text = "Payout: "
        credits2x = credits2x - 1
        tmrslidedown.Enabled = True
        btnroll.Enabled = False
    Else
        piccoin_Click(Me, EventArgs.Empty)
        Me.lblpayout.Text = "Payout: "
        credits = credits - 1
        tmrslidedown.Enabled = True
        btnroll.Enabled = False
    End If
End If

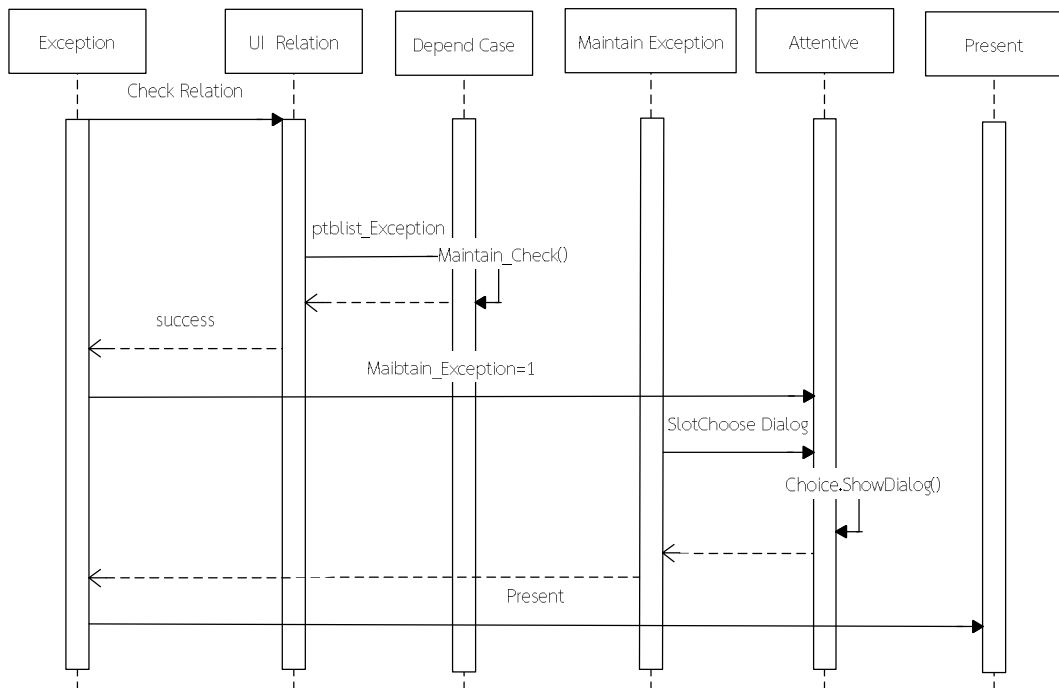
If Choice.Visible = False Then
    If credits >= 1 Then
        If SlotChoose = "RSlot" Then
            Else
            End If
        Else
        End If
    End If
End If
```

ภาพที่ จ.25 ซอร์ซโค้ดการบำรุงรักษาระบบโดยการแยกการทำงานของส่วนต่อประสาน

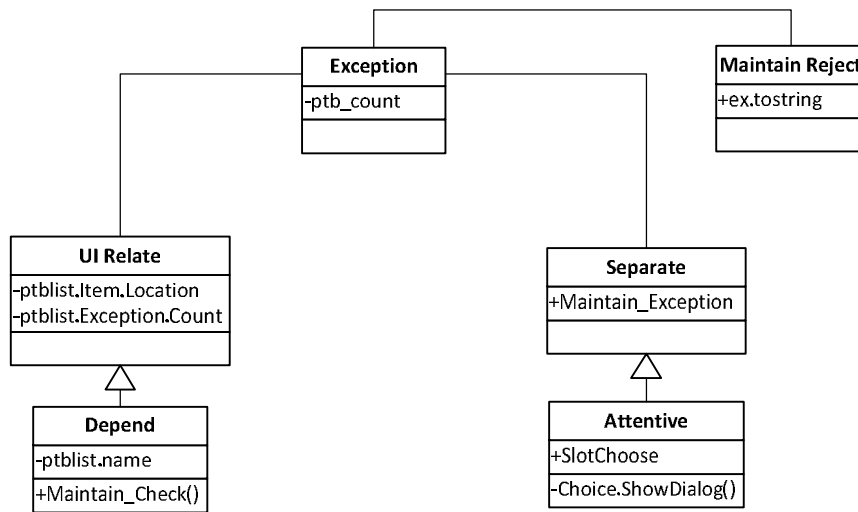
เพิ่มตู้สล็อตอีก 1 ตู้  
 ตัวระบบไม่ทราบถึง  
 การมีอยู่ของตู้ที่เพิ่ม



ภาพที่ จ.26 ผลกรณืทดสอบ T016



ภาพที่ จ.27 แผนภาพลำดับ การทำงานของแบบรูปกรณืเพื่อการบำรุงรักษา โดยการบำรุงรักษา  
 ระบบเมื่อมีการปรับแต่ง



ภาพที่ จ.28 แผนภาพลำดับ การทำงานของแบบรูปกรณีเพื่อการบำรุงรักษา โดยการบำรุงรักษา ระบบเมื่อมีการปรับแต่ง

## ภาค ผนวก ฉ

### เกม งู

#### ฉ.1 ข้อมูลจำเพาะ

พัฒนาโดย: <http://active.tutsplus.com> โดย คุณ *Joost Rijnveld*

ภาษาที่ใช้พัฒนา : ภาษา แฟรช แอ็คชั่น สคริป (Flash Actionscript)

ประเภทเกม : เกมพัชเชิ้ล

เกมประเภทงู เป็นเกมที่มีส่วนต่อประสานเป็นองค์ประกอบหลัก 3 ส่วนได้แก่

- 1 ตัวงู แทนตัวละครของผู้ให้ผู้เล่นสามารถบังคับได้จะมีขนาดยาวขึ้นเมื่อกินอาหาร งูจะตายเมื่องูกินตัวเองหรือเคลื่อนที่ชนฉาก
- 2 อาหาร แทนอาหารของงูเมื่อกินแล้วจะทำให้งูมีขนาดยาวขึ้น ปรากฏขึ้นแบบสุ่มในฉาก
- 3 ฉาก คือขอบเขตของเกมที่สามารถเคลื่อนที่ได้ เมื่องูเคลื่อนที่โดนฉากจะทำให้เกมจบทันที

กฎของเกมเป็นไปตามกฎของการเล่นเกมสล็อตแมชชีน ดังนี้

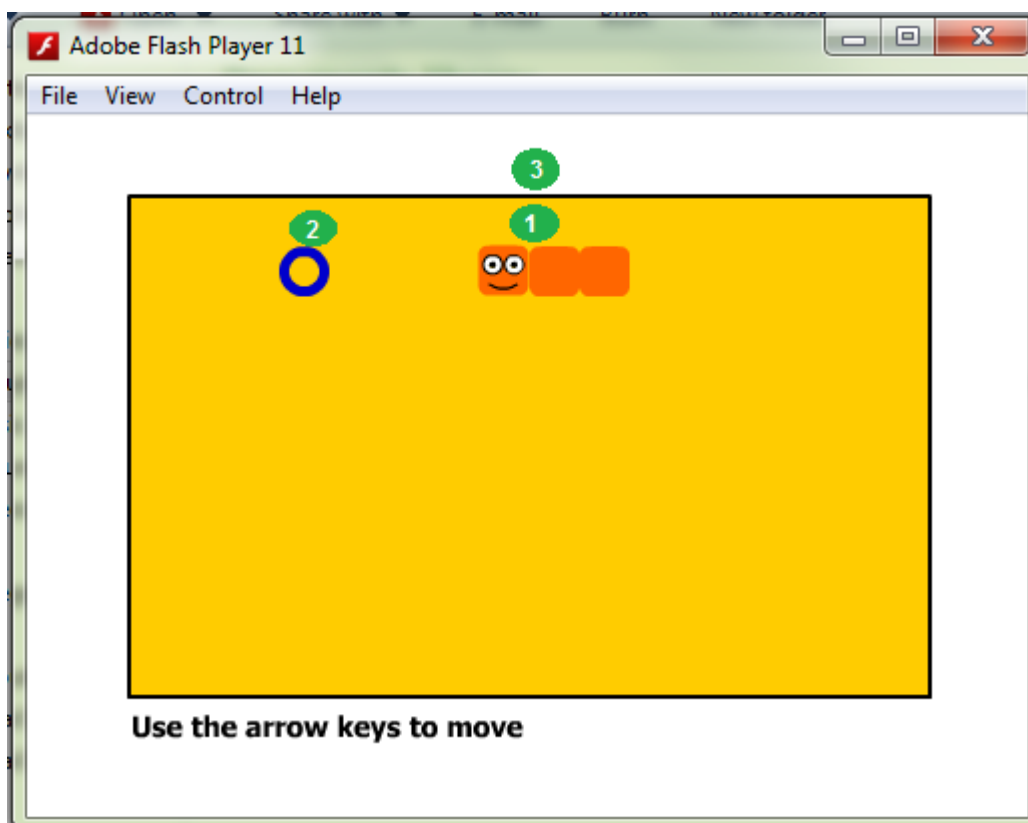
1. ผู้เล่นบังคับให้งูกินอาหาร
2. เมื่องูกินอาหารจะทำให้ตัวงูยาวขึ้น
3. เกมจบเมื่องูตาย
4. งูกินตัวเองทำให้งูตาย
5. งูเดินชนฉากทำให้งูตาย

#### ฉ.2 วิธีการเล่น

1. เมื่อเริ่มเกมอาหารและงูจะปรากฏ ผู้เล่นสามารถบังคับงูเข้าหาอาหารได้ทันที
2. เมื่อผู้เล่นบังคับงูแล้ว งูจะเคลื่อนที่ไปในทิศทางที่ผู้เล่นบังคับไปเรื่อยๆจนกว่าจะถึงเปลี่ยน
3. เมื่องูกินอาหารจะทำให้งูยาวขึ้น
4. เมื่องูตายจะถือว่าเกมจบ โดยงูตายมีได้ 2 กรณี
  - 4.1 งูกินตัวเอง
  - 4.2 งูเคลื่อนที่ชนฉาก
5. เมื่องูตายจะปรากฏหน้าต่างเล่นเกมใหม่

#### ฉ.3 องค์ประกอบหน้าจอส่วนต่อประสานเกม

องค์ประกอบของส่วนต่อประสานอ้างอิงตามภาพ เป็นดังภาพที่ ฉ.1 มีรายละเอียดดังต่อไปนี้



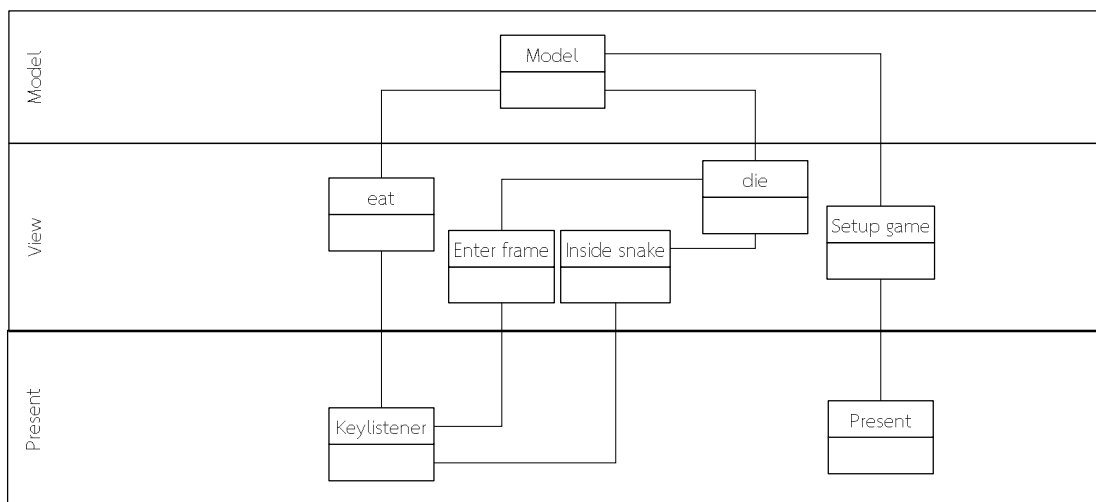
ภาพที่ ฉ.1 ส่วนต่อประสานของเกมงู

1. งู
2. อาหาร
3. ฉาก

#### ฉ.4 โครงสร้างระบบ

เดิมโครงสร้างของระบบไม่ได้ออกแบบให้อยู่ในลักษณะแบบรูป ทางผู้วิจัยได้ศึกษา วิเคราะห์ โครงสร้างและสร้างเป็นแผนภาพในลักษณะของแบบรูปได้ดังภาพที่ ฉ.2





ภาพที่ ๑.2 แผนภาพคลาสโครงสร้างของเกมงู

จากภาพข้างต้นแบ่งออกเป็นระดับใช้ได้ 3 ระดับดังนี้

- 1.โมเดล ระดับชั้นที่แสดงถึงข้อมูลในการทำงานของระบบเกม
- 2.วิว ชั้นที่เป็นศูนย์รวมตรรกะของระบบประกอบด้วย 2 องค์ประกอบได้แก่
  - 2.1 eat ทำหน้าที่ทำงานเมื่องูกินอาหาร
  - 2.2 inside snake ทำหน้าที่จัดการกรณีที่ถูกงูกินตัวเอง
  - 2.3 die ทำหน้าที่จัดการเมื่องูตาย
  - 2.4 Enter frame ทำหน้าที่จัดการเมื่องูชนฉาก
- 3.พรีเซ็น ส่วนที่ทำหน้าที่ปฏิสัมพันธ์กับผู้ใช้ ซึ่งแบ่งออกได้เป็น 2 ชั้นย่อยได้แก่
  - 3.1 ส่วนปฏิสัมพันธ์กับผู้ใช้ เป็นชั้นที่รวมส่วนต่อประสานที่ผู้ใช้สามารถมีปฏิสัมพันธ์ได้โดยตรงได้แก่ Keylistener เพียงส่วนเดียวทำหน้าที่รับข้อมูลจากผู้ใช้
  - 3.2 ส่วนต่อประสานเพื่อนำเสนอข้อมูล ทำหน้าที่ในการนำเสนอข้อมูลแก่ผู้ใช้เพียงอย่างเดียวประกอบด้วยได้แก่ Present ซึ่งก็คือหน้าต่างเฟรมการนำเสนอของแพรช

## ภาค ผนวก ช

### เกม แข่งรถ

#### ช.1 ข้อมูลจำเพาะ

พัฒนาโดย: <http://www.gamesheep.com/free/flash-tutorials/1-flash-racing-game-1/step1.php>

ภาษาที่ใช้พัฒนา : ภาษา แฟรช แอ็คชั่น สคริป (Flash Actionscript)

ประเภทเกม : เกมแข่งรถ

เกมประเภทแข่งรถ เป็นเกมที่มีส่วนต่อประสานเป็นองค์ประกอบหลัก 3 ส่วนได้แก่

- 1 รถ แทนตัวละครของผู้ ให้ผู้เล่นสามารถบังคับ
- 2 สนามแข่ง ขอบเขตที่รถวิ่งได้
- 3 กระดาน สำหรับบอกเวลาและจำนวนรอบ ข้อกำหนดในการเล่น ใช้แทนคะแนน
- 4 หน้าต่างแจ้งสถานะ หน้าต่างสำหรับแจ้งสถานะก่อนเริ่มเกมเพื่อเตรียมพร้อมและจบเกม เพื่อให้เลือกเล่นซ้ำ

กฎของเกมเป็นไปตามกฎของการเล่นเกมแข่งรถ ดังนี้

1. ผู้เล่นบังคับรถวิ่งในสนาม
2. เมื่อรถวิ่งนอกเขตสนามรถจะวิ่งช้าลง
3. รถวิ่งในสนามตามจำนวนรอบที่กำหนด
4. เวลาจะสิ้นสุดอยู่ตลอดเวลา และมีเก็บค่าทุกรอบที่วิ่ง
5. เกมจบเมื่อวิ่งครบตามรอบที่กำหนด

#### ช.2 วิธีการเล่น

1. เมื่อเริ่มจะมีข้อความแจ้งเพื่อให้ผู้เล่นบังคับรถ
2. เมื่อผู้เล่นบังคับรถไปตามเส้นทางในเกม
3. เมื่อนาฬิกาจะจับเวลาตลอดเวลา
4. นาฬิกาจะเก็บค่าทุกครั้งที่ครบรอบ และจะมีเก็บเวลาที่ดีที่สุดในทุกรอบ
5. เกมจบเมื่อรถวิ่งครบจำนวนรอบแล้ว

#### ช.3 องค์ประกอบหน้าจอส่วนต่อประสานเกม

องค์ประกอบของส่วนต่อประสานอ้างอิงตามภาพ เป็นดังภาพที่ ช.1 มีรายละเอียดดังต่อไปนี้

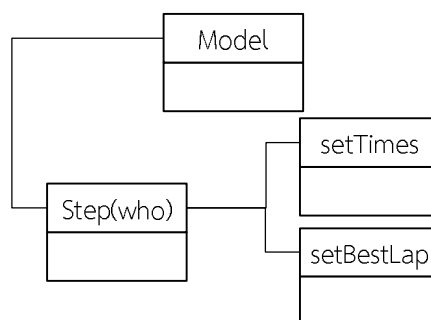


ภาพที่ ช.1 ส่วนต่อประสานของเกมแข่งรถ

1. รถ
2. สนาม
3. กระดาน
4. หน้าต่างสถานะ

#### ช.4 โครงสร้างระบบ

เดิมโครงสร้างของระบบไม่ได้ออกแบบให้อยู่ในลักษณะแบบรูป ทางผู้วิจัยได้ศึกษา วิเคราะห์โครงสร้างและสร้างเป็นแผนภาพในลักษณะของแบบรูปได้ดังภาพที่ ช.2



ภาพที่ ช.2 โครงสร้างของระบบเกมแข่งรถ

จากภาพข้างต้นมีองค์ประกอบ 4 ส่วนได้แก่


1. โมเดล (Model) ทำหน้าที่สร้างและเก็บส่วนตัวแปลงตัวแปร
2. Step(who) ทำหน้าที่สร้างแกนของสนาม คำนวณตำแหน่งรถกับสนาม นำเสนอข้อมูลตำแหน่งรถ รับข้อมูลการบังคับจากผู้ใช้
3. setTimes ทำหน้าที่จับเวลา และนำเสนอข้อมูลเวลาแก่ผู้ใช้
4. setBestLap ทำหน้าที่คำนวณหาเวลาที่ดีที่สุด และนำเสนอแก่ผู้ใช้

ภาค ผนวก ซ

การทดสอบแบบรูปการปรับแต่งส่วนต่อประสานกรณีเพื่อความพึงพอใจ กับเกมงู

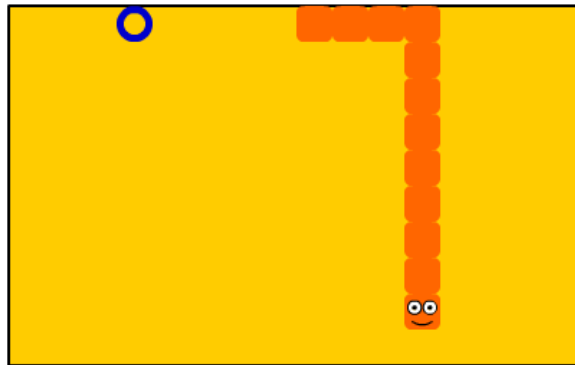
พัฒนาการปรับแต่งเพื่อสร้างกรณีศึกษาเพื่อนำผลข้อผิดพลาดจากการปรับแต่งเป็นไปตามกรณีการปรับแต่งส่วนต่อประสานเพื่อความพึงพอใจ ไปกำหนดเป็นกรณีทดสอบในการประยุกต์ใช้มอด พร้อมเก็บข้อมูลผลข้อผิดพลาดที่คาดว่าจะเกิดขึ้นหรือได้ผลของการปรับแต่งเป็นไปตามที่คาดหวังหรือไม่อย่างไร และบันทึกผลที่เกิดขึ้นจริงเพื่อนำไปเป็นข้อมูลในการออกแบบกรณีทดสอบผลที่เกิดขึ้นเป็นไปตามแบบฟอร์มตัวอย่างเพื่อเก็บเป็นตัวอย่างกรณีศึกษา อนึ่งข้อผิดพลาดที่คาดว่าจะเกิดขึ้นนั้นไม่ได้เกิดขึ้นจริงผู้วิจัยได้ทำการวิเคราะห์เบื้องต้นถึงสาเหตุทำข้อผิดพลาดนั้นๆควบคู่ไปในแบบฟอร์ม ดังตัวอย่างในตารางที่ ซ.1 โดยนำข้อมูลที่เก็บได้ในแต่ละกรณีศึกษามาวิเคราะห์เพื่อพัฒนาเป็นกรณีทดสอบเพื่อใช้ในการทดสอบแบบรูปในตารางที่ ซ.2

ตาราง ซ.1 รายละเอียดกรณีทดสอบ (T020)

|                                 |  |  |
|---------------------------------|--|--|
| ชื่อกรณีทดสอบ                   | T020   | เปลี่ยนภาพเมื่อกินอาหารมากกว่า 5 หน่วย |
| ฟังก์ชันที่ทดสอบ                | R020   | eatface                                |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีเพื่อความพึงพอใจ (แบบรูปลำดับที่ 1)   |  |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ  |  |
| ข้อมูลนำเข้า                    | ภาพงูกินจุ ตัวอย่างดังภาพ ซ.1 <br>ภาพที่ ซ.1 ภาพงูกินจุ |  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | นำเสนอภาพที่ปรับแต่งอย่างถูกต้อง มีขนาดภาพพอดีขนาดภาพอื่นๆที่เกี่ยวข้อง  |  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | นำเสนอภาพที่ผิดปกติ หรือ นำเสนอภาพเดิมของระบบ  |  |
| ข้อมูลทดสอบ                     | ภาพดังภาพ ซ.1 ซึ่งมีขนาด 11x25 และซอร์ซโค้ด ภาพที่ ซ.2   |  |
| ผลการทดสอบ (กรณีผิดพลาด)        | ดังแสดงในภาพ ซ.4   |  |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน      หมายเหตุ  |  |

```
function eatface(){
    bodypart0.gotoAndStop(3);
    delay = 30000;
    status_face="normal";
    exception_eatface();
}
```

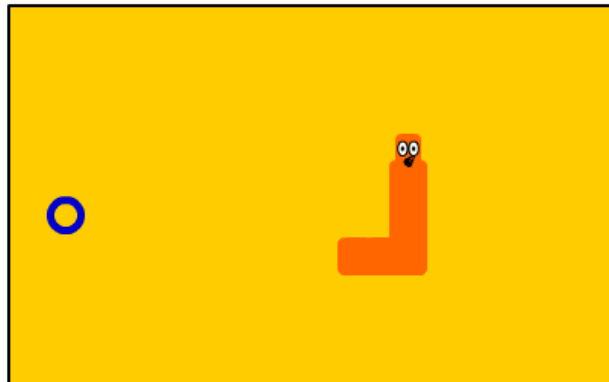
ภาพที่ ซ.2 ซอร์ซโค้ดในการเปลี่ยนรูปภาพใหม่แก่ระบบ



Use the arrow keys to move

ภาพที่ ซ.3 ผลการทำงานปกติก่อนทดสอบ T020

เปลี่ยนหน้าจอเมื่อตัวยาว



Use the arrow keys to move

ภาพที่ ซ.4 ผลการทดสอบกรณีเพื่อความพึงพอใจของผู้ใช้ โดยการภาพงู T020

ตาราง ซ.2 รายละเอียดกรณีทดสอบ การเปลี่ยนภาพ เมื่อประยุกต์ใช้แบบรูป (T021)

|                                 |  |   |
|---------------------------------|--|---|
| ชื่อกรณีทดสอบ                   | T021                                     | เปลี่ยนภาพเมื่อถูกกินอาหารมากกว่า 5 หน่วย   |
| ฟังก์ชันที่ทดสอบ                | R021                                     | exception_eatface,<br>exception_eatfacehandle   |
| วัตถุประสงค์การทดสอบ            |  | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่งกรณีเพื่อความพึงพอใจ (แบบรูปลำดับที่ 1)        |
| ผู้ใช้งาน                       |  | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    |  | เพิ่มโค้ดการเปลี่ยนภาพ โดยมีโครงสร้างเป็นไปตามแบบรูปการจัดการการปรับแต่งกรณีเพื่อการใช้งานเฉพาะ |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    |  | นำเสนอภาพที่ปรับแต่งอย่างถูกต้อง มีขนาดภาพพอดีขนาดภาพอื่นๆที่เกี่ยวข้อง                         |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) |  | นำเสนอภาพที่ผิดปกติ หรือ นำเสนอภาพเดิมของระบบ   |
| ข้อมูลทดสอบ                     |  | ซอร์ซโค้ดดังภาพที่ ซ.5-ซ.6  |
| ผลการทดสอบ                      |  | ดังแสดงในภาพที่ ซ.7   |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน | <input type="checkbox"/> ไม่ผ่าน      หมายเหตุ  |

```

function exception_eatface(){
    ///bodypart0._width = 10;
        trace(bodypart0._width)
    if(bodypart0._width == 25){
        if(bodypart0._height == 25){
            testtest.text = "a";
            trace(bodypart0.testtest.text)
        }else{
            trace("aaa")
        }
    }else{
        if(bodypart0._height == 25){
            trace("bbb")
            hcase= 25;
            exception_eatface1()
        }else{
            trace("cc")
        }
    }
    ///bodypart0.gotoAndStop(4);
    ///bodypart0.gotoAndStop(4);
    }
}

```

ภาพที่ ซ.5 ซอร์ซโค้ดในการตรวจสอบขนาดภาพ

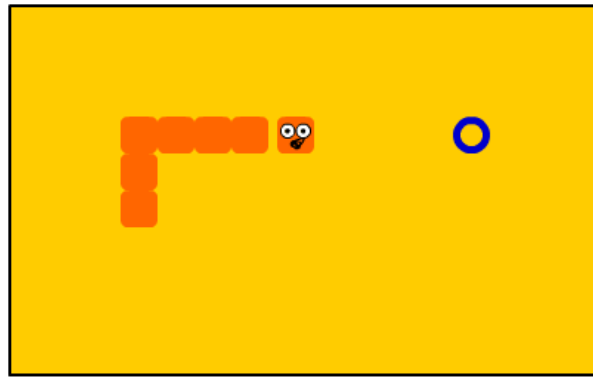
```

function exception_eatfacehandle(){
    bodypart0._width =hcase
}

```

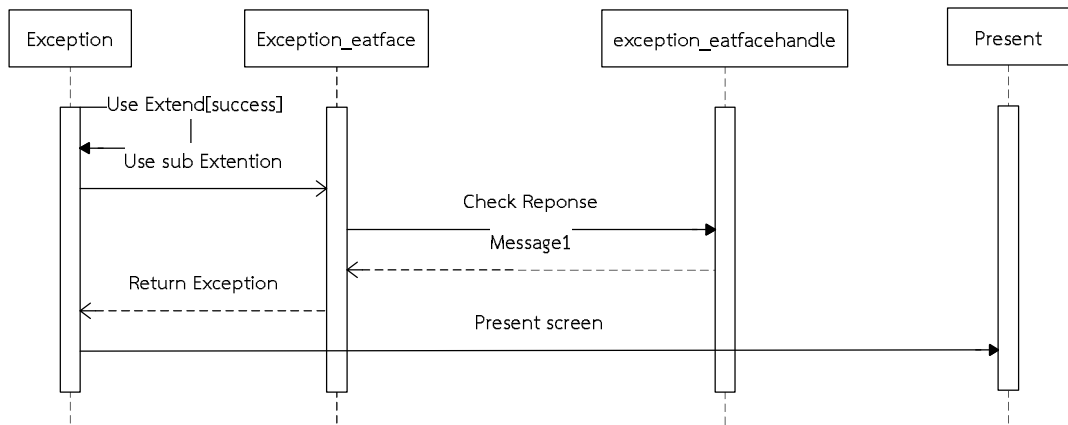
ภาพที่ ซ.6 ซอร์ซโค้ดในการปรับขนาดภาพ





Use the arrow keys to move

ภาพที่ ซ.7 ผลกรณืทดสอบ T021



ภาพที่ ซ.8 แผนภาพลำดับ การทำงานของแบบรูปกรณืเพื่อความพึงพอใจ โดยการเปลี่ยนภาพ



ภาพที่ ซ.9 แผนภาพคลาส การทำงานของแบบรูปกรณืเพื่อความพึงพอใจ โดยการเปลี่ยนภาพ

## ภาค ผนวก ฅ

### การทดสอบแบบรูปการปรับแต่งส่วนต่อประสานกรณีเพื่อการเพิ่มฟังก์ชันกับเกมแข่งรถ

พัฒนาการปรับแต่งเพื่อสร้างกรณีศึกษาเพื่อนำผลข้อผิดพลาดจากการปรับแต่งเป็นไปตามกรณีการปรับแต่งส่วนต่อประสานเพื่อการปรับตัว ไปกำหนดเป็นกรณีทดสอบในการประยุกต์ใช้โหมดพร้อมเก็บข้อมูลผลข้อผิดพลาดที่คาดว่าจะเกิดขึ้นหรือได้ผลของการปรับแต่งเป็นไปตามที่คาดหวังหรือไม่อย่างไร และบันทึกผลที่เกิดขึ้นจริงเพื่อนำไปเป็นข้อมูลในการออกแบบกรณีทดสอบ ผลที่เกิดขึ้นเป็นไปตามแบบฟอร์มตัวอย่างเพื่อเก็บเป็นตัวอย่างกรณีศึกษา อนึ่งข้อผิดพลาดที่คาดว่าจะเกิดขึ้นนั้นไม่ได้เกิดขึ้นจริงผู้วิจัยได้ทำการวิเคราะห์เบื้องต้นถึงสาเหตุทำข้อผิดพลาดนั้นๆควบคู่ไปในแบบฟอร์ม ดังตัวอย่างในตารางที่ ฅ.1 โดยนำข้อมูลที่เก็บได้ในแต่ละกรณีศึกษามาวิเคราะห์เพื่อพัฒนาเป็นกรณีทดสอบเพื่อใช้ในการทดสอบแบบรูปในตารางที่ ฅ.2

ตารางที่ ฅ.1 กรณีทดสอบการเพิ่มฟังก์ชันใหม่ในระบบ (T030)

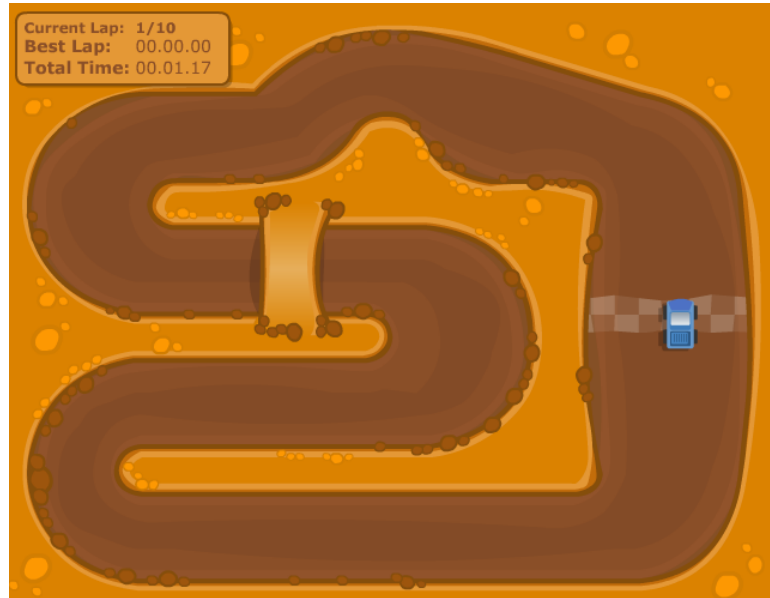
|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T030 การเพิ่มฟังก์ชันใหม่ในระบบ   |
| ฟังก์ชันที่ทดสอบ                | R030 ตรวจสอบการทำงานของระบบในภาพรวม   |
| วัตถุประสงค์การทดสอบ            | เพื่อทดสอบความสามารถในการทำงานเพื่อหาความครอบคลุมของกรณีการเพิ่มฟังก์ชัน (แบบรูปลำดับที่ 3) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | ซอร์ซโค้ดสำหรับเพิ่มปุ่มเร่งรถ  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | สามารถใช้งานปุ่มเพิ่มความเร็วได้ มีหน้าต่างแสดง จำนวนความเร็วที่ใช้ได้                      |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | รถไม่สามารถเร่งได้ ไม่มีหน้าต่างแสดงผล  |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ ฅ.1  |
| ผลการทดสอบ                      | ผลดังภาพที่ ฅ.2   |
| สรุปผลการทดสอบ :                | <input type="checkbox"/> ผ่าน <input checked="" type="checkbox"/> ไม่ผ่าน เหตุเหตุดู_____   |

```

if (Key.isToggled(65){
    _root.maxSpeed += maxSpeed +1;
    Gear.Text = "zxxxx";
}

```

ภาพที่ ฌ.1 ซอร์ซโค้ดใ้การเพิ่มฟังก์ชันเร่งความเร็ว



ภาพที่ ฌ.2 ผลการทำงานปกติก่อนทดสอบ (T030)

เพิ่มปั้ม A เพื่อให้รถเร็วขึ้น



ภาพที่ ฌ.3 ผลการทดสอบกรณี (T030)

ตาราง ฅ.2 รายละเอียดกรณีทดสอบ แบบรูปกรณีการปรับตัว โดยการเพิ่มฟังก์ชันใหม่ในระบบ (T031)

|                                 |   |
|---------------------------------|---|
| ชื่อกรณีทดสอบ                   | T031    แบบรูปกรณีการปรับตัว ทดสอบโดยเพิ่มฟังก์ชันใหม่ในระบบ  |
| ฟังก์ชันที่ทดสอบ                | R031    Exception_function, Exception_Screen, nitro   |
| วัตถุประสงค์การทดสอบ            | เพื่อตรวจสอบการทำงานของแบบรูปการจัดการการปรับแต่กรณีการปรับแต่งเพื่อการเพิ่มฟังก์ชัน (แบบรูปลำดับที่ 3) |
| ผู้ใช้งาน                       | ผู้ใช้งานระบบ   |
| ข้อมูลนำเข้า                    | ซอร์ซโค้ดสำหรับเพิ่มปุ้มเร่งรถ  |
| ผลลัพธ์ที่คาดหวัง (กรณีปกติ)    | สามารถใช้งานปุ้มเพิ่มความเร็วได้ มีหน้าต่างแสดง จำนวนความเร็วที่ใช้ได้                                  |
| ผลลัพธ์ที่คาดหวัง (กรณีผิดพลาด) | เล่นเกมได้ตามปกติด้วยความเร็วเท่าเดิม   |
| ข้อมูลทดสอบ                     | ซอร์ซโค้ดดังภาพที่ ฅ.15-ฅ.17  |
| ผลการทดสอบ (กรณีปกติ)           | ดังแสดงในภาพที่ ฅ.18  |
| สรุปผลการทดสอบ :                | <input checked="" type="checkbox"/> ผ่าน <input type="checkbox"/> ไม่ผ่าน    หมายเหตุ                   |

```
function nitro(){
    nitus=nitus-1;
    Exception_function();
}
```

ภาพที่ ฅ.4 ซอร์ซโค้ดในการเพิ่มฟังก์ชันเร่งความเร็ว (ใหม่)

```

function Exception_function(){
if ((nitus<100) && (nitus>0)) {
    _root.maxSpeed += 1;
}else if(nitus<0){
    this["speed"+who] =speedDecay;
}else{
    this["speed"+who] =speedDecay;
}
Exception_Screen ()
}

```

ภาพที่ ฌ.5 ซอร์ซโค้ดจัดการฟังก์ชันเร่งความเร็ว

```

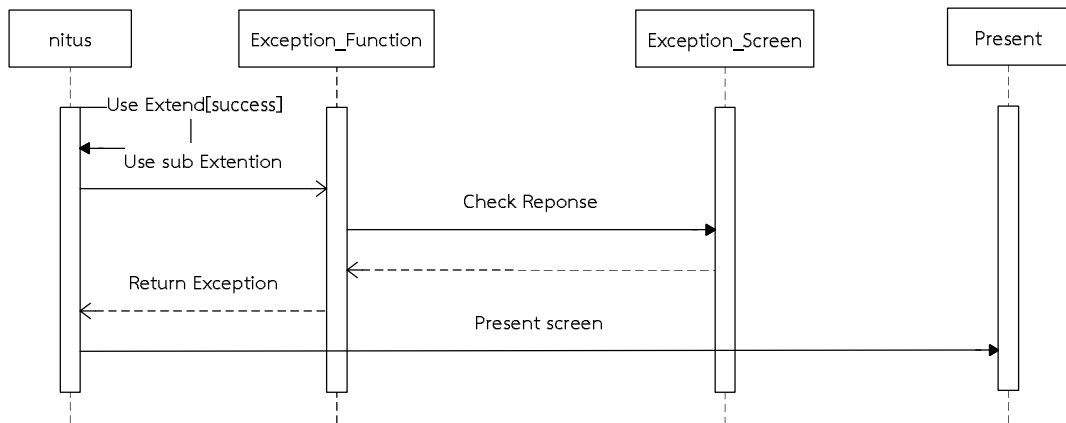
function Exception_Screen(){
if ((nitus<100) && (nitus>0)) {
    Gear_xx.text = nitus;
}else if(nitus<0){
    Gear_xx.text = "0";
}else{
    Gear_xx.text ="";
}
}

```

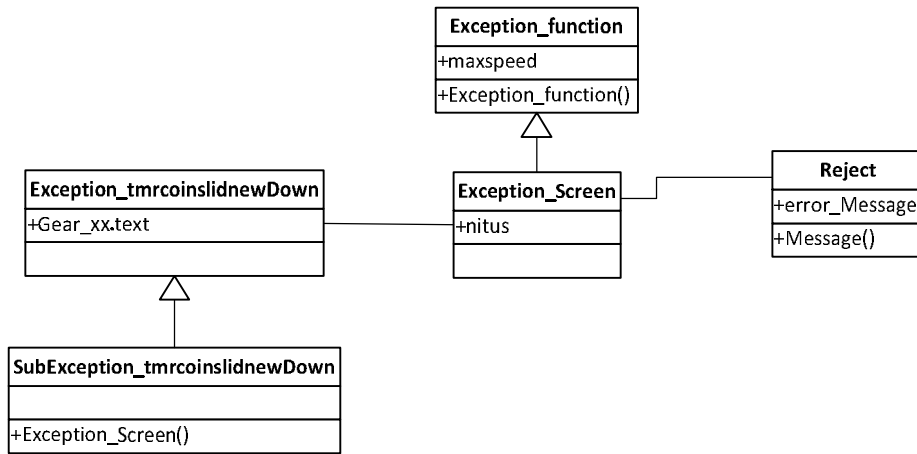
ภาพที่ ฌ.6 ซอร์ซโค้ดจัดการการนำเสนอข้อมูล



ภาพที่ ฅ.7 ผลกรณืทดสอบ T031



ภาพที่ ฅ.8 แผนภาพลำดับ การทำงานของแบบรูปกรณืเพิ่มฟังก์ชัน ปุ่มเร่งความเร็ว



ภาพที่ ๘.๙ แผนภาพคลาส การทำงานของแบบรูปกรณีเพิ่มฟังก์ชัน ปุ่มเร่งความเร็ว

**ประวัติผู้เขียนวิทยานิพนธ์**

นาย ฐิติพงศ์ เต็งไตรรัตน์ เกิดวันที่ 13 พฤศจิกายน 2528 สำเร็จการศึกษาระดับปริญญา  
วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยรังสิต  
ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรม  
ซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ในปีการศึกษา 2554 ที่อยู่ปัจจุบัน  
ที่สามารถติดต่อ 218 หมู่ 8 ตำบลปัว อำเภอปัว จังหวัดน่าน หมายเลขโทรศัพท์ 0802777833 อีเมลล์  
seedmonoeye@gmail.com