

**โครงงาน**

**Mini project**

**จัดทำโดย**

6504062630171 นายพสิษฐ์ มัสสะอาด

**เสนอ**

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

**วิชา Object Oriented Programming**

ภาคเรียนที่ 1/2565

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**บทนำ**

**ที่มาและความสำคัญ**

**:** เพื่อฝึกฝนการเขียนโปรแกรมรูปแบบ Object-Orient ให้ได้อย่างมีประสิทธิภาพ โดยใช้สร้างในรูปแบบเกม เพื่อประยุกต์สิ่งที่ได้ศึกษามา เพื่อเพิ่มความแม่นยำ และความชำนาญในศาสตร์การเขียนโปรแกรมลักษณะนี้ยิ่งขึ้น

**ประเภทโครงงาน**

**:** เกม

**ประโยชน์**

**- ฝึกฝนความสามารถในการหาความรู้ และประยุกต์ใช้**

**- ฝึกฝนการใช้งานเครื่องมือเพื่อเพิ่มประสิทธิภาพการทำงานต่างๆ**

**ตารางแผนการทำงาน**

|  |  |  |  |
| --- | --- | --- | --- |
| ลำดับ | งานที่ทำ | 1-15 พย. | 16-30 พย. |
| 1 | ศึกษาและวางแผนการทำเกม | X |  |
| 2 | วางโครงสร้างและคลาสต่างๆ | X |  |
| 3 | อิมพลีเม้นท์แผนที่วางไว้ | X | X |
| 4 | จัดทำเอกสาร |  | X |
| 5 | ตรวจสอบจุดผิดพลาด |  | X |

**ขอบเขตการทำงาน**

**- รายละเอียดของเกม**

**:** ผู้เล่นคือนักผจญภัยที่เริ่มต่อสู้กับมอนสเตอร์ระหว่างทางไปเมืองต่างๆ ไปเรื่อยๆจนกว่าจะหมดอายุขัย

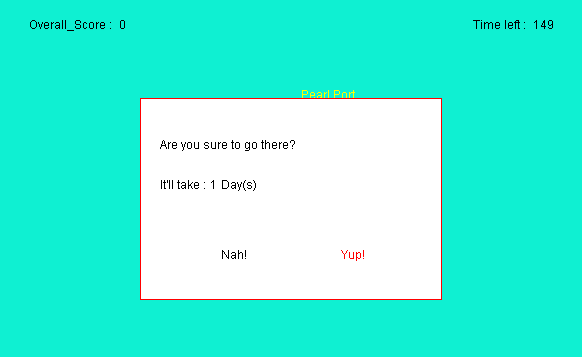
**- วิธีการเล่น**

**:** ปุ่มชี้ทิศทาง ซ้าย และ ขวา ในการเลื่อนไปมาระหว่างตัวเลือก ปุ่ม ขึ้น เพื่อเลือก เข้าแถบอัพสเตตัสด้วยการกด E ผู้เล่นต้องอัพสเตตัสของตัวเอง เพื่อที่จะมีค่าต่างๆมากพอที่จะสู้กับมอนสเตอร์ชุดต่อๆไป

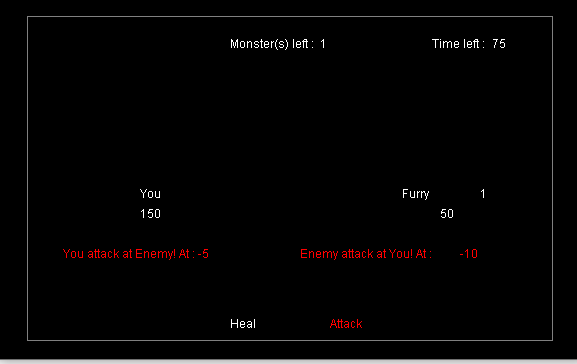
**- StoryBoard**

****

**:** หน้าตัวเลือกพื้นที่ที่จะไป ซึ่งแต่ละที่ก็จะมีจำนวนมอนสเตอร์ไม่เท่ากัน โดยที่ Player ไม่สามารถวนอยู่แค่เมืองเดียวได้



**:** หน้ายืนยันการเดินทาง โดยจะมีจำนวนมอนสเตอร์ บอกอยู่ที่ Days



**:** หน้าต่อสู้ โดยที่ทางซ้าย คือผู้เล่น และทางขวา คือ มอนสเตอร์



**:** เมื่อแพ้ ก็จะถูกส่งมายังหน้านี้ เมื่อชนะจะถูกส่งกลับไปหน้าแมพเหมือนเดิม เพื่อเล่นต่อไป

**ส่วนการพัฒนา**

**- รูปแบบการพัฒนา**

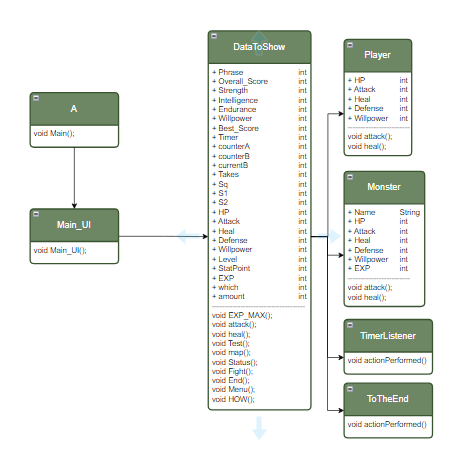
**1. สร้างระบบ Panel เพื่อการแสดงผล.**

**2. สร้างหน้าต่างๆรองรับฟังก์ชั่นต่างๆที่ต้องการให้มี**

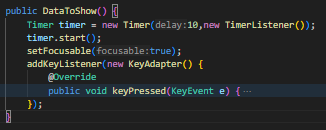
**3. สร้างสเตตัสต่างๆ ฟังก์ชั่นที่เกี่ยวข้องกับระบบเกม และไลบรารี่ศัตรู**

**4. ปรับระบบการต่อสู้ให้ผู้เล่นมีทางสู้ได้ แม้ในระดับที่สูงขึ้น**

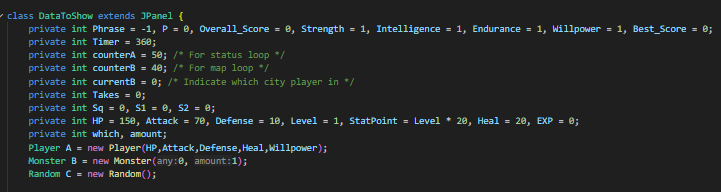
**- Class Diagram**

****

**- คำอธิบายส่วนโปรแกรม**

****

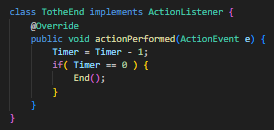
ตัวอย่าง Constructor ของ DataToShow ที่จะกำหนดสิ่งที่เกิดขึ้นภายในโปรแกรมระหว่างทำงาน และติดตั้งการรับคีย์จากผู้ใช้



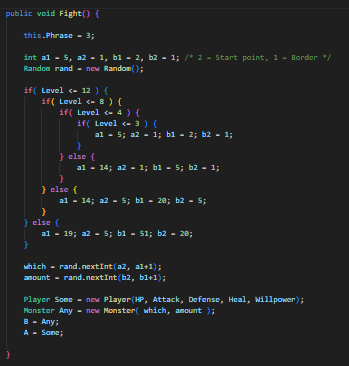
ตัวอย่างการใช้งาน Encapsulation กับตัวแปร คือ การกำหนดไม่ให้เกิดการเปลี่ยนตัวแปรในคลาสนี้จาก Class อื่นโดยตรง



ตัวอย่างการใช้งาน Composition ในการรับค่าจาก Constructor เพื่อแปลงค่าในเอทิตี้ตัวเองเพื่อใช้งานต่อๆไป



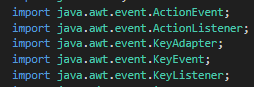
ตัวอย่างการใช้งาน Polymorphism ในการ Override ฟังก์ชั่นเปล่า จาก ActionListener ลงในชื่อ actionPerformed() ซึ่งเป็น Method จาก ActionListener



หนึ่งในอัลกอรึทึมสำตัญของเกมนี้ คือ การกำหนดระดับความยาก โดยอิงจาก Level ของผู้เล่น โดยตัวฟังก์ชั่นจะกำหนดศัตรูที่จะเจอ และจำนวน Magnifier เพื่อความยากที่มากขึ้น

**- ส่วน GUI ภายในเกม**

**มีการใช้งานดังนี้ :**

****

**EventHandling :**

ใช้งานปุ่มลูกศร ซ้าย-ขวา ในการเลือกระหว่างตัวเลือก

ใช้งานลูกศร บน เพื่อเลือกตัวเลือกนั้นๆ

ใช้งานปุ่ม E เพื่อเข้าหน้าอัพเดทสเตตัส

ใช้งาน G ในหน้าหน้าอัพเดทสเตตัสเพื่อจะโดนส่งไปที่ไร้วันกลับ

**สรุป**

**ปัญหาที่พบระหว่างพัฒนา**

**-** การเบิร์นเอ้าท์ระหว่างพัฒนา

- OverEngineering และ Over-expectation

- การเชื่อมไปมาระหว่าง EventHandler และ DrawComponent

- การควบคุมฟังก์ชั่นต่างๆให้ทำงานภายใต้ความสมดุล

**จุดเด่นที่สุดของโปรแกรม**

- มีจุดหลายจุดที่สามารถพัฒนาต่อยอด ที่สามารถทำให้เกมนี้มีมิติมากขึ้นกว่านี้ได้ หากมีเวลามากพอ

- Replay value ที่ไม่แย่ สามารถเล่นได้ 2 – 3 ครั้งก่อนที่จะเบื่อ

- โค้ดอ่านยาก จากการรวมทุกอย่างลงไฟล์เดียว ทำให้จะต้องใช้เวลาแกะนานขึ้น

**คำขอร้องแด่สำหรับผู้สอน**

: คงจะดีกว่านี้ ถ้างานนี้ สเกลคะแนนที่ได้เยอะกว่านี้ และได้เวลาในการทำมากกว่า 1 เดือนครับ