**A red and white circular logo

Description automatically generated**

**รายงาน**

**เรื่อง : The Epic of Linkle**

**เสนอ**

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

**วิชาการเขียนโปรแกรมเชิงวัตถุ (OOP)**

ภาคเรียนที่ 1/2566

**โดย**

นาย วชิรสรณ์ เติมรัตนสุวรรณ รหัส 6504062630251

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**คำนำ**

**รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาการเขียนโปรแกรมเชิงวัตถุ (OOP)   
คณะวิทยาศาสตร์ประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์ ปีที่ 2 เพื่อให้ได้ศึกษาหาความรู้ในเรื่องการเขียนโปรแกรมเชิงวัตถุ และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการเรียน**

**ผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา   
ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อแนะนำหรือข้อผิดพลาดประการใด   
ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย**

**วชิรสรณ์ เติมรัตนสุวรรณ**

**7 พฤศจิกายน 2566**

**บทที่ 1**

**บทนำ**

**1.1 ที่มาและความสำคัญของโปรเจ็ค**

การพัฒนาเกมไม่เพียงแต่เป็นกระบวนการทางเทคนิค แต่ยังเป็นศิลปะที่ต้องการความคิดสร้างสรรค์และความใส่ใจในรายละเอียดทุกรายละเอียด โปรเจกต์ "The Epic of Linkle" นี้ไม่เพียงแค่เป็นเกม แต่เป็นสื่อที่เล่าเรื่องราวในโลกแห่งแฟนตาซี ผ่านการเขียนโปรแกรมที่ใส่ใจกับโครงสร้างและการออกแบบที่ดี ซึ่งเป็นผลลัพธ์ของความพยายามและความมุ่งมั่นของนักศึกษาในการเรียนรู้และปฏิบัติทักษะในวิชาการเขียนโปรแกรมเชิงวัตถุ (OOP)

**1.2 ประเภทของโครงการ**

**Java Applications**

**1.3 ประโยชน์ของโครงการ**

1. ช่วยฝึกการสร้างโครงสร้างของโปรแกรมที่มีความเป็นระเบียบ

2. ช่วยฝึกการลดโค้ดที่ต้องเขียนซ้ำซากโดยการใช้การ Inheritance และ Encapsulation

3. ช่วยฝึกการจัดโครงสร้างของเกมเป็นระเบียบและเข้าใจง่าย

**1.4 ขอบเขตของโครงการ**

**1.4.1 การวางแผน (Planning)**

กำหนดวัตถุประสงค์ของโครงการ โดยกำหนดว่าต้องการสร้างเกมประเภทใด   
และมีฟีเจอร์หรือระบบใดบ้างที่ต้องการให้มีในเกม ออกแบบคลาสและวัตถุเพื่อกำหนดคลาสต่างๆ   
และหน้าที่ของแต่ละคลาส เช่น ผู้เล่น (Player), ศัตรู (Enemy), ไอเท็ม (Item) เป็นต้น

**1.4.2 การเขียนโปรแกรม (Programming)**

สร้างคลาสและวัตถุตามที่วางแผนไว้ โดยการเขียนโค้ดสร้างคลาสของแต่ละวัตถุ   
และใส่เมธอด (methods) ที่ใช้ในการจัดการข้อมูลและกระทำต่างๆ

**1.4.3 การทดสอบและปรับปรุง (Testing and Refining)**

ทดสอบโปรแกรม โดยทดสอบการทำงานของโปรแกรมเพื่อตรวจสอบว่าทุกฟีเจอร์ทำงานตามที่ต้องการหรือไม่ และมีการปรับปรุงโค้ดหรือเพิ่มฟีเจอร์ตามความต้องการและข้อเสนอแนะ

**1.4.4 เขียนรายงาน (Documentation)**

การเขียนรายงานโครงการ โดยภายในเล่มรายงานรวบรวมข้อมูลเกี่ยวกับโครงการ เช่น วัตถุประสงค์, วิธีการทำ, ปัญหาที่พบและวิธีแก้ไข, และผลลัพธ์ที่ได้ รวมทั้งแสดงตัวอย่างโค้ดที่เกี่ยวข้องกับการใช้ OOP ในโปรเจกต์

**1.4.5 ตารางแผนการทํางานเดือนกันยายน - พฤศจิกายน**

1. ตารางแผนการทํางานเดือนกันยายน

|  |  |  |  |
| --- | --- | --- | --- |
| ลำดับ | รายการ | 16 – 22 ก.ย. | 23 – 30 ก.ย. |
| 1 | วางแผนโปรเจ็ค  • วางแผนโปรเจ็ควางแผนโปรเจ็ค  • ออกแบบโครงสร้างของเกมและคลาสต่างๆ |  |  |
| 2 | เขียนโค้ด  • เริ่มเขียนโค้ดในภาษา Java  • สร้างคลาสต่างๆ และเมธอดพื้นฐานที่จำเป็น |  |  |

2. ตารางแผนการทํางานเดือนตุลาคม

|  |  |  |  |
| --- | --- | --- | --- |
| ลำดับ | รายการ | 01 – 25 ต.ค. | 26 – 31 ต.ค. |
| 3 | พัฒนาความสามารถของเกม  • เพิ่มฟีเจอร์และระบบต่างๆ เช่น การเคลื่อนที่  • ทำการทดสอบความถูกต้องของระบบที่เพิ่มเข้าไป |  |  |
| 4 | ทดสอบและปรับปรุง  • ทดสอบโปรแกรมโดยให้ผู้ทดสอบใช้งานเกม  • รับข้อเสนอแนะและแก้ไขปัญหาที่พบ |  |  |

3. ตารางแผนการทํางานเดือนพฤศจิกายน

|  |  |  |
| --- | --- | --- |
| ลำดับ | รายการ | 01 – 09 พ.ย. |
| 5 | เขียนรายงานและจัดรูปแบบ  • เขียนรายงานโครงการโดยรวบรวมข้อมูลจากการทดลองเล่นเกม  • จัดรูปแบบรายงานและตรวจสอบความถูกต้องของข้อมูลและการเขียน |  |

**บทที่ 2**

**ส่วนการพัฒนา**

**2.1 เนื้อเรื่องย่อและวิธิการเล่น**

**2.1.1 เนื้อเรื่อง**

ในโลกที่อยู่ในอนาคตที่ห่างไกลจากโลกปกติที่เรารู้จัก มีเกาะร้างหนึ่งซึ่งอยู่ตรงกลางของมหาสมุทรและเต็มไปด้วยสัตว์ประหลาดและธรรมชาติที่ร้ายแรง คุณตื่นขึ้นบนเกาะนี้โดยไม่มีความทรงจำว่าทำไมคุณถึงมาอยู่ที่นี่ แต่คุณสัมผัสได้ว่ามีบางสิ่งที่อยู่ใต้ดันเจี้ยนกำลังเรียกหาคุณอยู่

**2.1.2 วิธีการเล่น**

**ลักษณะการทำงาน ปุ่ม ลักษณะการทำงาน ปุ่ม**

เคลื่อนที่ไปข้างหน้า W เปิดหน้าเมนู ESC

เคลื่อนที่ไปข้างหลัง S เปิดหน้าต่างตัวละคร C

เคลื่อนที่ไปทางซ้าย A เปิดมินิแมพ X

เคลื่อนที่ไปทางขวา D เปิดแมพขนาดใหญ่ M

ทำการโจมตี ENTER ทำการใช้พลังเวท F

หยุดเกมชั่วคราว P ทำการเลือกสิ่งของ ENTER

**2.2 Class Diagram**

**A close-up of a text

Description automatically generated**

**A close-up of a list of text

Description automatically generated**

**A close-up of several papers

Description automatically generated**

**A diagram of a data flow

Description automatically generated**

**A diagram of a computer code

Description automatically generated with medium confidence**

**A diagram of a company

Description automatically generated**

**A close-up of a document

Description automatically generated**

**A diagram of a computer program

Description automatically generated with medium confidence**

**A close-up of a document

Description automatically generated**

**A close-up of several different types of text

Description automatically generated**

**A close-up of a paper

Description automatically generated**

**A close-up of a paper

Description automatically generated**

**คำอธิบายการอ่าน Class diagram**

**A diagram of a student

Description automatically generated**

1. ส่วนด้านบนใช้เพื่อตั้งชื่อคลาส

2. อันที่สองใช้เพื่อแสดง Constructer ของคลาส

3. อันที่สามใช้เพื่อแสดงคุณสมบัติของคลาส

4. อันที่สี่ใช้เพื่ออธิบายการดำเนินการที่ดำเนินการโดยคลาส

**2.3 อธิบายส่วนของโปรแกรมที่มีหลักการเขียนโปรแกรมเชิงวัตถุ**

**2.3.1 ตัวอย่างของ Constructor ที่มีภายในโปรแกรมA screenshot of a computer screen

Description automatically generated**

ตัวอย่างนี้อยู่ภายใน Class ที่มีชื่อว่า GamePanel โดย Constructors หรือตัวสร้าง   
คือเมธอดที่ถูกเรียกใช้เมื่อสร้าง Object ซึ่ง Constructors มีหน้าที่กำหนดค่าเริ่มต้นให้กับตัวแปร   
และประกาศและกำหนดค่าให้กับ Object ที่ถูกสร้างขึ้น เช่น

1. Public TileManager tileM = new TileManager(this); คือการสร้างวัตถุ TileManager   
และส่งตัวอ็อบเจ็กต์ปัจจุบันเป็นพารามิเตอร์ โดยที่มีระดับการเข้าถึงคือ public

2. Sound music = new Sound(); - สร้างวัตถุ Sound โดยไม่มีการส่งพารามิเตอร์

**2.3.2 ตัวอย่างของ Encapsulation ที่มีภายในโปรแกรม**

A screen shot of a computer program

Description automatically generated

ตัวอย่างนี้อคือ Character Attributes บางส่วนของ Entity โดยมีการ Encapsulation ผ่านระดับการเข้าถึงของตัวแปรบางตัวแปร ซึ่งถูกประกาศเป็น private หมายความว่าสามารถเข้าถึงได้เฉพาะภายในคลาสที่ถูกประกาศ โดยต้องมีสร้างเมทอด getter และ setter เพื่อให้สามารถเข้าถึงและตั้งค่าข้อมูลที่ถูกป้องกันไว้ได้

A screenshot of a computer program

Description automatically generated**2.3.3 ตัวอย่างของ Inheritance ที่มีภายในโปรแกรม**

ตัวอย่างนี้อยู่ภายใน Class ที่มีชื่อว่า MON\_GreenSlime โดยมีการ Inheritance มาจาก   
Class ที่มีชื่อว่า Entity ทำให้ MON\_GreenSlime สามารถใช้งานฟังก์ชันและข้อมูลที่ถูกกำหนดใน Entity ได้โดยตรง โดยไม่ต้องนำเข้ามาใหม่

**2.4 ส่วนประกอบของ Graphics และ GUI ที่มีภายในโปรแกรม**

**2.4.1 กราฟิก (Graphics)**

java.awt.AlphaComposite: ใช้ในการกำหนดการผสมภาพแบบต่าง ๆ เช่น การผสมภาพด้วยค่าโปร่งใส

java.awt.Color: ใช้ในการจัดการสีและแสดงสีในรูปแบบ RGB.

java.awt.Graphics: เป็นอ็อบเจกต์ที่ใช้ในการวาดรูปภาพและตัวหนังสือบน component

java.awt.Graphics2D: คุณลักษณะเพิ่มเติมสำหรับการวาดภาพจาก java.awt.Graphics

java.awt.RenderingHints: ใช้กำหนดคุณลักษณะการวาดที่ต้องการ เช่น คุณภาพของการวาด

java.awt.BasicStroke: ใช้กำหนดลักษณะของเส้นทางที่วาด เช่น ความหนาของเส้น

java.awt.Font: ใช้ในการจัดการตัวหนังสือ เช่น การตั้งความหนา, ขนาด, และลักษณะอื่น ๆ

java.awt.FontFormatException: จัดการข้อผิดพลาดที่เกี่ยวข้องกับรูปแบบ Font ที่ไม่ถูกต้อง

java.awt.image.BufferedImage: เป็นคลาสที่ใช้สร้างและจัดการกับรูปภาพในรูปแบบของ buffer

java.awt.RadialGradientPaint: ใช้สร้างลวดลายแสงแบบรัศมี

java.awt.Rectangle: ใช้ในการจัดการข้อมูลของสี่เหลี่ยมผืนผ้า

java.awt.Dimension: ใช้ในการระบุขนาดของ component หรือขอบเขตของพื้นที่

java.awt.GraphicsDevice: ใช้ในการจัดการอุปกรณ์กราฟิก (เช่น หน้าจอ) ในเครื่องคอมพิวเตอร์

java.awt.GraphicsEnvironment: ใช้เพื่อให้เข้าถึงข้อมูลเกี่ยวกับกรณีพิเศษของกราฟิกที่ระบบใช้งาน

**2.4.2 GUI (Graphical User Interface)**

javax.swing.ImageIcon: ใช้ในการโหลดรูปภาพจากไฟล์

javax.swing.JFrame: เป็นคลาสที่ใช้สร้างหน้าต่าง GUI หลักของแอปพลิเคชัน

**2.5 Event Handling ที่มีภายในโปรแกรม**

Event Handling ของโปรแกรมนี้อยู่ใน Class ที่มีชื่อว่า KeyHandler โดยมีการ implements KeyListener โดยมีเมทอดที่ต้อง implement คือ keyTyped(), keyPressed(), และ keyReleased()

โดยเมทอดที่สำคัญคือ keyPressed และ keyReleased

keyPressed : เมทอดนี้ถูกเรียกเมื่อมีการกดปุ่มบนคีย์บอร์ด ใช้ในการตรวจสอบปุ่มที่ถูกกด   
และทำการปรับค่าตัวแปรต่าง ๆ ตามปุ่มที่ถูกกด

keyReleased : เมทอดนี้ถูกเรียกเมื่อปุ่มบนคีย์บอร์ดถูกปล่อยหลังจากที่ถูกกดไว้   
ใช้ในการตรวจสอบปุ่มที่ถูกปล่อย และทำการปรับค่าตัวแปรต่าง ๆ ตามปุ่มที่ถูกปล่อย

ตัวอย่างโค้ดในการใช้งาน Event Handling ในสถานการณ์ที่แตกต่างกัน เช่น   
ในโหมด titleState() เมื่อกดปุ่ม W หรือ S จะทำให้ตัวแปร gp.ui.commandNum ถูกเพิ่มหรือลดลง และเมื่อกดปุ่ม Enter แล้วตัวแปร gp.ui.commandNum ถูกตรวจสอบเพื่อทำการเปลี่ยนสถานะเกม (เช่น เริ่มเกม, โหลดเกม, หรือออกจากเกม)

A video game screen with text and cartoon character

Description automatically generated

**2.6 อัลกอริทึมที่สำคัญในโปรแกรม**

หนึ่งในอัลกอริทึมที่สำคัญในโปรแกรมมีชื่อว่า A\* (A-Star) เป็นการใช้วิธีการค้นหาเส้นทาง   
เพื่อหาเส้นทางที่สั้นที่สุดจากตำแหน่งเริ่มต้นไปยังตำแหน่งปลายทางในตารางขนาดใหญ่ (grid)   
ซึ่งแต่ละช่องในตารางมีค่าความยาก (cost) ที่ต่างกัน ซึ่งอยู่ในส่วน Package ที่มีชื่อว่า ai

Node : คลาสนี้เก็บข้อมูลของแต่ละช่องในตารางที่ใช้ในการคำนวณ A\* Pathfinding   
โดยมีข้อมูลเช่น col และ row แทนตำแหน่งของช่อง, gCost, hCost, และ fCost ที่ใช้ในการคำนวณ  
ค่าความยาก, และตัวแปร solid, open, และ checked ที่บอกถึงสถานะของช่องนั้น ๆ

PathFinder : คลาสนี้คือเมนต์ของการค้นหาเส้นทาง A\* โดยมีเมทอดที่สำคัญได้แก่

• instantiateNodes(): สร้างตารางขนาดใหญ่ที่ใช้ในการคำนวณ A\* Pathfinding

• resetNodes(): รีเซ็ตสถานะของช่องทั้งหมดให้เป็นสถานะเริ่มต้น

• setNodes(): ตั้งค่าตำแหน่งเริ่มต้นและปลายทางของการค้นหาเส้นทาง A\*   
พร้อมกับกำหนดของ entity ที่มีผลต่อการคำนวณค่าความยากของแต่ละช่อง

• getCost(Node node): คำนวณค่า gCost, hCost, และ fCost ของช่องที่กำหนด

• search(): ทำการค้นหาเส้นทางโดยใช้อัลกอริทึม A\*

• openNode(Node node): เปิดช่องที่ยังไม่ได้เปิดให้ตามค่าความยากและเพิ่มลงใน openList

• trackThePath(): ติดตามเส้นทางที่ได้จากผลลัพธ์ของการค้นหาเส้นทาง A\*

**บทที่ 3**

**สรุป**

**3.1 ปัญหาที่พบระหว่างการพัฒนา**

• การควบคุมการประสานงานระหว่างเหตุการณ์ เช่น การเดินพร้อมกับการโจมตี

• ปัญหาทางด้านการเช็ค collision ทำให้ตัวละครอัดมุมไม่สามารถเดินออกมาได้

• ปัญหาด้านการเรียงลำดับการวาดภาพ เช่น วาด NPC ทับตัวละครผู้เล่น

• ปัญหาทางด้านแหล่งเสียงที่จัดการกับการทำงานของเอฟเฟกต์แต่ละอย่างในเวลาเดียวกัน

**3.2 จุดเด่นของโปรแกรม**

โปรแกรมนั้นมีการแบ่งส่วนการทำงานของแต่ละ Class อย่างชัดเจนทำให้การเพิ่ม Map, Monster, NPC, OBJ และ Item เข้ามาใหม่นั้นสามารถทำได้อย่างง่ายดาย

**3.3 คำแนะนำสำหรับน้องรุ่นต่อไป**

• ช่วงต้นเทอมอยากทำอะไรให้รีบทำ และให้ศึกษาไว้ก่อนล่วงหน้าและทำโปรเจคก่อนได้เลย เพราะต้องคุณจะรอจนสอนจบยันบทสุดท้ายแล้วคุณค่อยทำโปรเจค ช่วงนั้นจะติดช่วงสอบพอดี   
แล้วคุณจะต้องเลือกระหว่าง เวลาทำโปรเจค, เวลาอ่านหนังสือ และเวลานอน ว่าสิ่งไหนสำคัญที่

**GitHub:** https://github.com/Eshwaldz/The-Epic-of-Linkle