A red and black logo

Description automatically generated

**โครงงาน**

Project Myra

**จัดทำโดย**

6704062612197 นายพาทิศ เจริญแพทย์

**เสนอ**

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา 040613204 Object-Oriented Programming

ภาคเรียนที่ 1 ปีการศึกษา 2568

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**เกี่ยวกับโครงงาน**

ชื่อโปรเจค: Project Myra

นำเสนอโดย: นายพาทิศ เจริญแพทย์

อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

Source Code: <https://github.com/PT22315/Myra-Project>

**บทที่ 1**

**บทนำ**

**ที่มาและความสำคัญของโครงการ**

โครงงานนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object-Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเกม โดยใช้แนวคิดการเขียนโปรแกรมแบบเชิงวัตถุ และยังช่วยให้ผู้จัดทำเรียนรู้อุปกรณ์และเครื่องมือ ผู้จัดทำได้สร้างเกมขึ้นมา

**ประเภทของโครงงาน**

เกม 2D จากภาษา Java

**ประโยชน์**

1.เพื่อนำแนวคิดการเขียนโปรแกรมแบบเชิงวัตถุมาประยุกต์ใช้

2.สะสมประสบการณ์การเขียนโปรแกรมแบบ OOP

**ขอบเขตของโครงงาน**

1.ใช้ภาษา Java ในการเขียน

2.ใช้หลักการเขียนโปรแกรมแบบ OOP

**ตารางแผนการทำงานเดือนกันยายน-เดือนตุลาคม**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | รายการ | 1ก.ย. – 15ก.ย. | 15ก.ย.-15ต.ค. | 15.ต.ค-28.ต.ค. |
| 1 | หารูปตัวละครในเกม |  |  |  |
| 2 | ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง |  |  |  |
| 3 | เขียนโปรแกรม |  |  |  |
| 4 | จัดทำเอกสาร |  |  |  |
| 5 | ตรวจสอบและแก้ไขข้อผิดพลาด |  |  |  |

**บทที่ 2**

**การพัฒนา**

**เนื้อเรื่องย่อเกม**

ไมร่านักสะสมเหรียญ ตอนนี้เค้าได้เจอเข้ากับเหรียญที่ดูแวววาวและสวยงาม เขาจึงอยากจะไปเอาเหรียญนั้นมาแต่มีเต่าปริศนาขัดขวางเขาอยู่ เขาจึงต้องใช้ทักษะการเคลื่อนที่ของตัวเองเพื่อไปให้ถึงเหรียญโดยที่ไม่สัมผัสกับเต่าปริศนาตัวนั้น เขาจะไม่ยอมไปไหนจนกว่าตัวเองจะได้ครอบครองเหรียญนั้น

**วิธีการเล่น**

กด w/space , a , d เพื่อบังคับทิศทางตัวละคร ปุ่ม a d เพื่อขยับซ้าย ขวา และปุ่ม w/space เพื่อกระโดด เมื่อไมร่าเก็บเหรียญได้จะถือว่าผ่าน

**Story Board**

-ตัวละคร

|  |
| --- |
| **A green and yellow cartoon character  Description automatically generatedA pixelated coin with a letter o  Description automatically generatedA pixelated video game character  Description automatically generated**  ไอเท็มรีเซ็ตเวลา  เต่าปริศนา  เหรียญ  ไมร่า |

**แผนภาพ Class Diagram**

**A screenshot of a computer screen

Description automatically generated**

คลาสหลักๆจะมีอยู่ทั้งหมด 4 คลาสดังนี้

- ProjectMyra มี Main

- MenuPanel หน้าแรกเมื่อกดเริ่มเกม

- DrawArea ด่านภายในเกม

- Actor ตัวละครภายในเกม

โดยคลาส DrawArea จะเป็นตัวที่ใช้วาดด่านนั้นๆซึ่งมี 2 ด่านดังนี้

- DrawForest ด่านป่า

- DrawMountain ด่านภูเขา

และคลาส Actor จะใช้เพิ่มกำหนดค่าต่างๆให้ตัวละครที่ต้องรีแอคกับระบบโดยจะมี 5 อย่างดังนี้

- Myra ตัวละครหลัก

- Turtle ตัวร้าย

- Coin เก็บแล้วจะถือชนะ

- Item เก็บแล้วมีผลกับเกม

- Tile ใช้ทำเป็นแพลตฟอร์มให้ตัวละครเหยียบ

**รูปแบบการพัฒนา**

เกมพัฒนาในรูปแบบ Java Application

**แนวคิดการเขียนโปรแกรมเชิงวัตถุ**

**Constructor**

|  |  |
| --- | --- |
| public Actor(String path, int x, int y, int width, int height) {  URL url = this.getClass().getResource(path);  this.image = new ImageIcon(url).getImage();  this.x = x;  this.y = y;  this.width = width;  this.height = height;  } | กำหนดค่าเริ่มต้นให้ Attribute ต่างๆในclass Actor |
| public ProjectMyra() {  setTitle("Myra");  setSize(1800, 1080);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  add(new MenuPanel(this));  setVisible(true);  } | กำหนดค่าเริ่มต้นให้class ProjectMyra |
| public Myra(String path, int x, int y, int width, int height) {  super(path, x, y, width, height);  this.groundY=y;  walkRightFrames = new Image[3];  walkRightFrames[0] = new ImageIcon(getClass().getResource("images/myra1.png")).getImage();  walkRightFrames[1] = new ImageIcon(getClass().getResource("images/myra2.png")).getImage();  walkRightFrames[2] = new ImageIcon(getClass().getResource("images/myra3.png")).getImage();  walkLeftFrames = new Image[3];  walkLeftFrames[0] = new ImageIcon(getClass().getResource("images/myra4.png")).getImage();  walkLeftFrames[1] = new ImageIcon(getClass().getResource("images/myra5.png")).getImage();  walkLeftFrames[2] = new ImageIcon(getClass().getResource("images/myra6.png")).getImage();  currentFrames = walkRightFrames;  image = currentFrames[0];  } | กำหนดค่าต่างๆให้class Myra แต่เนื่องจาก class Myra extends มาจากclass Actor ทำให้ต้องใช้ super เพื่อกำหนดค่าต่างๆอีกที |
| public DrawArea(JFrame frame) {  this.frame = frame;  setFocusable(true);  addKeyListener(this);  } | กำหนดค่าเริ่มต้นให้ class DrawArea |

**Encapsulation**

ในทุกๆคลาสจะมีการประกาศตัวแปรแบบต่างๆ

|  |  |
| --- | --- |
| public abstract class Actor {  protected Image image;  protected int x, y, width, height;  …  } | ตัวอย่างการประกาศตัวแปรในclass Actor มีการประกาศโดยใช้ protected หมายถึงอนุญาตให้เข้าเฉพาะคลาสลูกที่สืบทอดไปเท่านั้น |
| public class Item extends Actor {  private boolean active;  public boolean isActive() { return active; }  public void setActive(boolean value) { active = value; }  } | Class item มีการใช้ตัวแปร private พร้อมกับ getter และ setter เพื่อเข้าถึงตัวแปรนั้น |

**Composition**

|  |  |
| --- | --- |
| public class DrawForest extends DrawArea {  private Turtle turtle;  private Coin coin;  private Image bg, fullHeart, emptyHeart;  private Item timeResetItem;  …  } | Class DrawForest มี Myra(อยู่ในDrawArea) Turtle , Coin , Item เป็นสมาชิกภายในของคลาส |
| public class DrawMountain extends DrawArea{  private Turtle turtle1,turtle2;  private Coin coin;  private Image bg, fullHeart, emptyHeart;  private Tile[] platform1, platform2;  ...  } | Class DrawMountain มี Myra , Turtle , Coin , Tile เป็นสมาชิกภายในของคลาส |

**Polymorphism**

คลาสลูกของ Actor และ DrawArea สามารถOverride เมธอดได้เพื่อให้แต่ละObjectมีพฤติกรรมเฉพาะของตัวเอง

|  |  |
| --- | --- |
| class Turtle extends Actor {  …  @Override  public void update() {  x = x + dir \* 3; //500 + 1\*3 when reach right = 1200 + (-1)\*3  if (x >= maxX || x <= minX){dir = dir \* -1;}  if (dir == 1) {currentImage = turtleRight;}  else {currentImage = turtleLeft;}  } | Class Turtle มีการ Override เมธอด update() ส่งผลให้พฤติกรรมแตกต่างไปจากคลาสแม่ |

**Abstract**

|  |  |
| --- | --- |
| public abstract class Actor {  …  public abstract void update();  …  } | บังคับให้คลาสลูกของ Actor ต้องมีเมธอด update() |
| public abstract class DrawArea extends JPanel implements KeyListener {  …  protected abstract void initActors(); // Myra, enemy, coin  protected abstract void initTimers(); // frameTimer gameTimer  protected abstract void resetGame(); // reset map  …  } | บังคับให้คลาสลูกของ DrawArea ต้องมีเมธอด initActor() initTimer() resetGame() |

**Inheritance**

|  |  |
| --- | --- |
| class Myra extends Actor {…} | Myra,Turtle,Coin,Tile,Item สืบทอดมาจากคลาส Actor |
| class Turtle extends Actor {…} |
| class Coin extends Actor {…} |
| class Tile extends Actor {…} |
| class Item extends Actor {…} |
| public class DrawForest extends DrawArea {…} | DrawForest , DrawMountain สืบทอดมาจากคลาส DrawArea |
| class DrawMountain extends DrawArea{…} |

**GUI**

1.หน้า Menu

A screenshot of a video game

Description automatically generated

ประกอบไปด้วย

Image Background 1 Component (menu bg.png)

Button Component 2 Component (StartBtn , ExitBtn)

2.หน้าเลือก Map

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generatedA video game screen with trees and a brick wall

Description automatically generated

ประกอบไปด้วย

Image Background Component 3 Component (menu bg.png , game bg.png , mountains bg.png)

Button Component 4 Component (BackBtn , ForestBtn , MountainBtn , StartGameBtn)

3.Map Forest

A video game screen with a cartoon character and a stone wall

Description automatically generated

ประกอบไปด้วย

Image Background 1 Component (game bg.png)

Heart Component 3 Component

Actor Component 4 Component (Myra,Turtle,Item,Coin)

Label 1 Component (Time , Score)

4.Map Mountain

A video game screen with cartoon characters

Description automatically generated

ประกอบไปด้วย

Image Background 1 Component (mountains bg.png)

Heart Component 3 Component

Actor Component 6 Component (Myra,turtle1,turtle2,Coin,Tile1,Tile2)

Label 1 Component (Time , Score)

**Event handling**

|  |  |
| --- | --- |
| public class MenuPanel extends JPanel {  …  startBtn.addActionListener(e -> showMapButtons(frame));  exitBtn.addActionListener(e -> System.exit(0));  forestBtn.addActionListener(e -> selectMap("forest",frame));  mountainBtn.addActionListener(e -> selectMap("mountain",frame));  backBtn.addActionListener(e -> backToMenu(frame));  startGameBtn.addActionListener(e -> startGame((JFrame) getTopLevelAncestor(), selectedMap));  backBtn.addActionListener(e -> backToMapSelection(frame));  } | Class MenuPanel มีการดักจับการกดปุ่มต่างๆ |
| public abstract class DrawArea extends JPanel implements KeyListener {  …  @Override  public void keyPressed(KeyEvent e) {  switch (e.getKeyCode()) {  case KeyEvent.VK\_A -> myra.setLeftPressed(true);  case KeyEvent.VK\_D -> myra.setRightPressed(true);  case KeyEvent.VK\_W, KeyEvent.VK\_SPACE -> myra.jump();  }  }  @Override  public void keyReleased(KeyEvent e) {  switch (e.getKeyCode()) {  case KeyEvent.VK\_A -> myra.setLeftPressed(false);  case KeyEvent.VK\_D -> myra.setRightPressed(false);  }  }  ...  } | Class Draw มีการดักจับการกดปุ่มบนคีย์บอร์ดซึ่งคลาสลูกที่สืบทอดไปก็จะได้รับส่วนการดักจับนี้ไปด้วย |

**อัลกอริทึมที่สำคัญในโปรแกรม**

|  |  |
| --- | --- |
| //Collision Item  if(timeResetItem.isActive() && myra.getBounds().intersects(timeResetItem.getBounds())) {  timeResetItem.setActive(false);  timeLeft = 60;  score = 1000;  }  // Collision Coin  if (myra.getBounds().intersects(coin.getBounds())) {  stopAllTimers();  int choice = JOptionPane.showOptionDialog(  this,  "You got the coin!\nYour score: " + score,  "Victory!",  JOptionPane.YES\_NO\_OPTION,  JOptionPane.INFORMATION\_MESSAGE,  null,  new String[]{"Exit", "Play Again"},  "Exit"  );  if (choice == JOptionPane.YES\_OPTION) System.exit(0);  else {  resetGame();  backToMenu();  }  }  // Collision Turtle  if (myra.getBounds().intersects(turtle.getBounds())) {  if (myra.getY() + myra.getHeight() <= turtle.getY() + 50) {  // over turtle head  } else {  myra.life--;  if (myra.life <= 0) {  stopAllTimers();  int choice = JOptionPane.showOptionDialog(  this,  "You Lose!",  "Game Over",  JOptionPane.YES\_NO\_OPTION,  JOptionPane.ERROR\_MESSAGE,  null,  new String[]{"Exit", "Play Again"},  "Exit"  );  if (choice == JOptionPane.YES\_OPTION) System.exit(0);  else {  resetGame();  backToMenu();  }  } else {  myra.resetPosition(150, 600);  }  }  } | เป็นอัลกอริทึมของแผนที่ Forestตรวจจับว่าตัวละครของเราแตะกับactorตัวอื่นๆหรือไม่ โดยจะทำงานทุกครั้งที่มีการอัพเดทเฟรม  ถ้าแตะไอเท็มจะทำการรีเซ็ตเวลาและคะแนน  ถ้าแตะเต่าจะเช็คว่ายังเหลือชีวิตหรือไม่ ถ้าเหลือให้ลดหัวใจลง 1 และรีเซ็ตตำแหน่งตัวละคร ถ้าไม่เหลือจะนับว่าGameOver แล้วจะมีOptionให้เลือกระหว่างปิดเกมหรือเริ่มเล่นใหม่โดยจะกลับไปหน้าmenu  ถ้าแตะเหรียญจะนับว่าชนะเกม  แล้วจะมีOptionให้เลือกระหว่างปิดเกมหรือเริ่มเล่นใหม่โดยจะกลับไปหน้าmenu |

**บทที่ 3**

**สรุป**

**ปัญหาที่พบระหว่างการพัฒนา**

ตัวละครเดินหลุดกรอบ,เรียกภาพไม่ขึ้น,timerทำงานผิดปกติ,ตัวละครลอยได้ ระหว่างทางยังคงพบปัญหาอยู่เรื่อยๆ

**จุดเด่นของโปรแกรม**

ในแผนที่ Mountainเมื่อตัวละครขึ้นไปบนแพลตฟอร์มระบบจะกำจัดการเดินให้อยู่แค่บนแพลตฟอร์มนั้นเท่านั้น โดยถ้าผู้เล่นต้องการจะออกจากแพลตฟอร์มนั้นต้องกด w , space เพื่อกระโดดออกเท่านั้น

**คำแนะนำสำหรับผู้สอนที่อยากให้อธิบายหรือที่เรียนแล้วไม่เข้าใจหรืออยากให้เพิ่มสำหรับน้องๆรุ่นต่อไป**

อยากให้อาจารย์มีGuideLineในแลปข้อต่างๆว่าควรเริ่มประมาณไหนหรือว่าอัลกอริทึมหลักของข้อนั้นๆว่าใช้แนวคิดlogicประมาณไหนและอยากให้ช่วยสอนพื้นฐาน HTML ให้ด้วยคับ;-;