



โค Object-oriented Programming mini project

“ Last Chicken ”

จัดทำโดย

6504062630103 นายณัฏพล ทองน่วม

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชาการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 1/2566

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

โครงการนี้ถูกจัดทำขึ้นเพื่อส่งงาน วิชาการเขียนโปรแกรมเชิงวัตถุ เพื่อวัดผลความรู้และความเข้าใจที่ได้จากการเรียน วิชา Object-oriented Programming ว่าสามารถนำมาประยุกต์ใช้ได้ทางปฏิบัติ ในรูปแบบของเกมที่มีเนื้อหา และการเขียนโปรแกรมแบบ OOP

1.2 ประโยชน์

1.2.1 ฝึกการเขียนโปรแกรม

1.2.2 ฝึกการคิดแบบเป็นหลักการ และเป็นขั้นเป็นตอน

1.2.3 ฝึกวิธีแก้ปัญหา

1.3 ขอบเขตของโครงการ

เกม Last Chicken เป็นเกมกระโดดข้ามสิ่งกีดขวาง โดยที่มีเวลาเป็นตัวนับเต็ม และเมื่อชนกับสิ่งกีดขวางก็จะทำให้พลังชีวิตลดลง และเมื่อพลังชีวิตหมดจนเหลือศูนย์เกมก็จะจบลง

1.3.1 ประเภทของโครงการ

โครงการเป็นประเภทเกมที่สร้างจากภาษา java โดยมีเนื้อหาและการเขียนโปรแกรมแบบ concept OOP

ตารางแผนการทำงานเดือนตุลาคม - เดือนพฤศจิกายน

ลำดับ	รายการ	1 ต.ค. – 15 ต.ค.	16 ต.ค. – 5 พ.ย.	6 พ.ย. – 9 พ.ย.
1	หารูปจัดทำตัวละครในเกม			
2	ศึกษาข้อมูลต่างๆ			
3	เขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบข้อผิดพลาด			

1.4 Storyboardและวิธีการเล่น

รายละเอียดเกม

เจ้าหนู Chicky คือเจ้าไก่ตัวสุดท้ายที่อยู่บนโลกใบนี้ และต้องพยายามเอาชีวิตรอดให้ได้ ไม่ให้ตัวเองสูญพันธุ์ และพยายามตามหาไก่ตัวอื่นที่อ่านเหลือรอดเพื่อ ฟันฟูประชากรชาวไก่ กลับมาอีกครั้ง

วิธีการเล่น

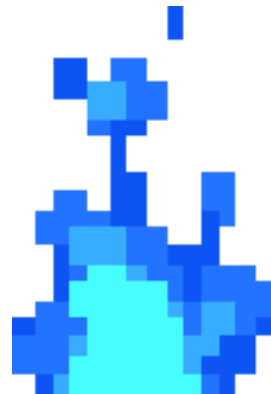
1. กด backspace บน keyboard เพื่อไปกระโดดข้ามสิ่งกีดขวาง
2. เกมจะนับคะแนนจากการอยู่รอดของเจ้าไก่
3. เมื่อคะแนนสูงขึ้นจากก็จะเปลี่ยนไป
4. เกมจะจบก็ต่อเมื่อ พลังชีวิตของเจ้าหนู Chicky หหมดลง

Storyboard

ตัวละคร



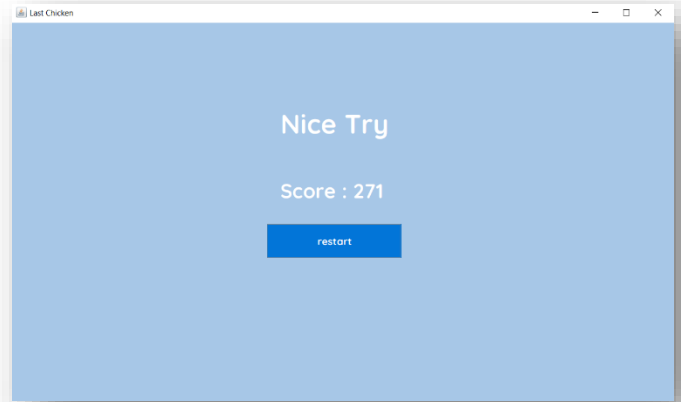
เจ้าหนู Chicky



ไฟ

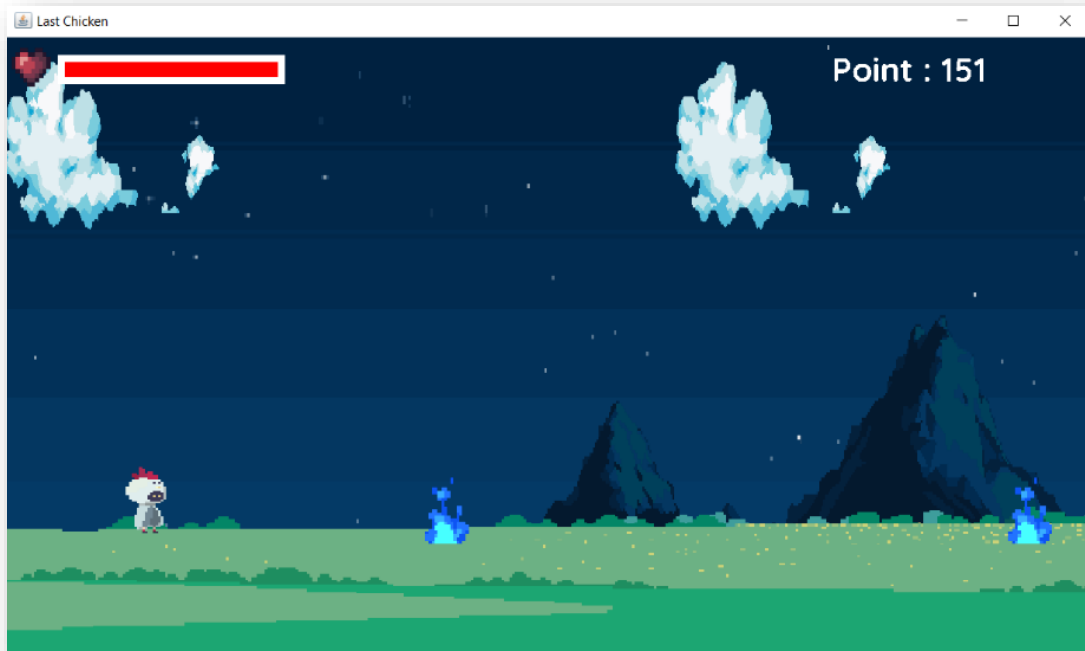
ฉากภายในเกม

1. หน้าเริ่มต้น และ หน้าจบเกม

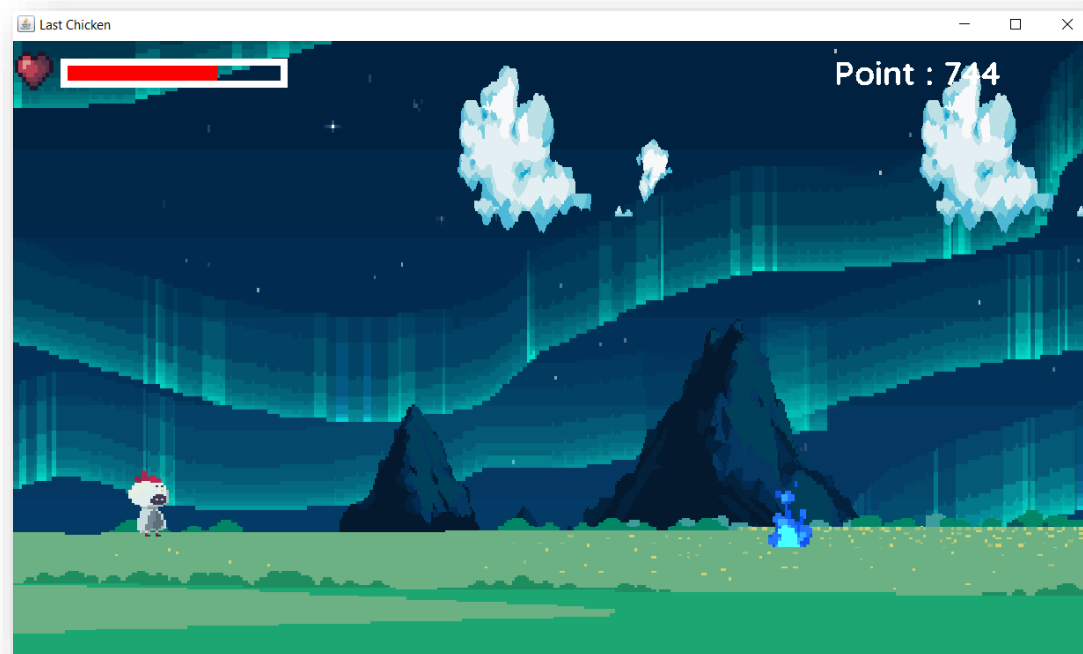


2. ฉากภายในเมื่อเริ่มเกม

2.1 เมื่อคะแนน มากกว่าที่กำหนดไว้ฉากหลังก็จะเปลี่ยน



2.1 เมื่อคะแนน มากกว่าที่กำหนดไว้ฉากหลังก็จะเปลี่ยน



บทที่ 2

ส่วนของการพัฒนา

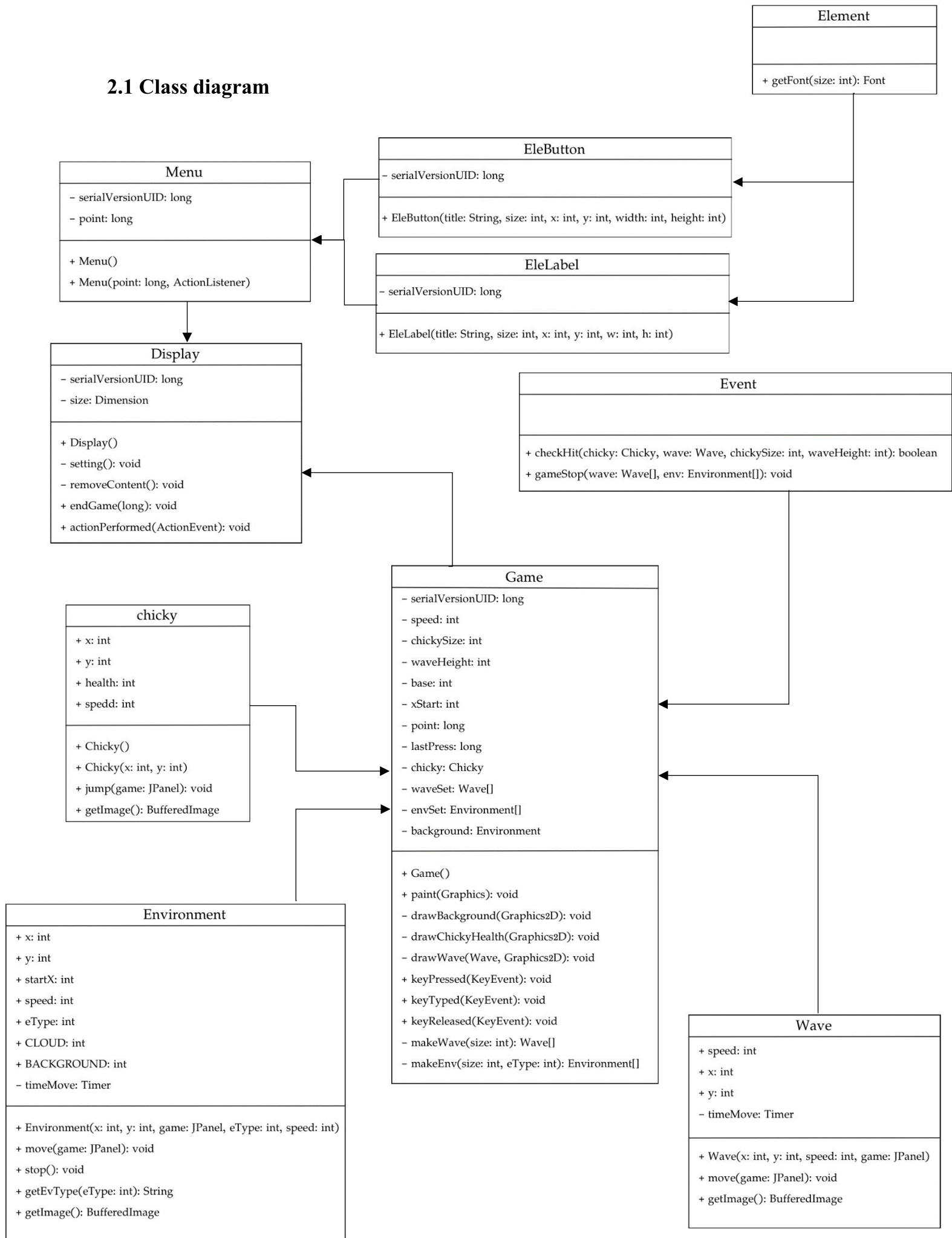
เนื้อเรื่องย่อ

เจ้าหนู Chicky คือเจ้าไก่ตัวสุดท้ายที่อยู่บนโลกใบนี้ และต้องพยายามเอาชีวิตรอดให้ได้ไม่ให้ตัวเองสูญพันธุ์ และพยายามตามหาไก่ตัวอื่นที่อ่านหนังสือรอดเพื่อฟื้นฟูประชากรชาวไก่กลับมาอีกครั้ง

วิธีการเล่น

1. กด backspace บน keyboard เพื่อไปกระโดดข้ามสิ่งกีดขวาง
2. เกมจะนับคะแนนจากการอยู่รอดของเจ้าไก่
3. เมื่อคะแนนสูงขึ้นฉากก็จะเปลี่ยนไป
4. เกมจะจบก็ต่อเมื่อ พลังชีวิตของเจ้าหนู Chicky หมดลง

2.1 Class diagram



2.1.1 Class Game

คลาส Game ในโปรแกรมเกมนี้เป็นหน่วยควบคุมหลักที่รวบรวมและจัดการกับตัวละคร, องค์ประกอบ, และเหตุการณ์ในเกม. มันดำเนินการวาดและควบคุมการทำงานของตัวละคร, ความสูงของวัตถุ, การตอบสนองต่อคีย์บอร์ด, และการตรวจสอบการชน. นอกจากนี้, มันยังรับผิดชอบในการจัดการคะแนน, การแสดงผลที่ต่างไปตามความคืบหน้าของเกม, และการจัดทำแบบจำลองทางสิ่งแวดล้อม

2.1.2 Class Display

Display มีบทบาทสำคัญในการเรียกเกมและเมนู โดยจัดการการเปลี่ยนหน้าต่าง, ลบเนื้อหา, และตอบสนองเหตุการณ์เมื่อผู้เล่นเลือกเริ่มเกมใหม่หรือจบเกม. นอกจากนี้, คลาสนี้ยังดำเนินการตาม ActionListener เพื่อตรวจสอบการกดปุ่ม "Restart" และดำเนินการตามมันโดยการสร้างเกมใหม่

2.1.3 Class Menu

คลาส Menu ใช้ในการแสดงผลลัพธ์หลังจากเกม, รวมถึงคะแนนที่ได้, และมีปุ่มที่เป็นตัวเลือกให้ผู้เล่นเริ่มเกมใหม่หลังจากตายในเกมก่อนหน้านี้

2.1.4 Class Chicky

คลาส Chicky มีหน้าที่แทนตัวละครในเกม โดยจัดการตำแหน่ง, สุขภาพ, และการกระโดดของตัวละคร รวมถึงการโหลดรูปภาพ. สามารถใช้ในบทบาทของตัวละครในเกมหรือแอปพลิเคชันที่เกี่ยวข้องกับกราฟิก

2.1.5 Class Wave

คลาส Wave ทำหน้าที่ควบคุมคลื่นเคลื่อนที่ในเกม โดยเลื่อนคลื่นไปทางซ้ายและนำกลับไปตำแหน่งเริ่มต้นเมื่ออยู่ที่ตำแหน่งที่กำหนด. คลาสนี้ยังสามารถโหลดรูปภาพของคลื่นได้, นำไปใช้ในเกมหรือแอปพลิเคชันที่ต้องการควบคุมคลื่นที่เคลื่อนที่.

2.1.6 Class Environment

คลาส Environment ทำหน้าที่ควบคุมการเคลื่อนที่และการแสดงภาพของสภาพแวดล้อมในเกม ซึ่งรวมถึงคลาวด์และพื้นหลัง. มีฟังก์ชันสำหรับการเคลื่อนที่ต่อเนื่อง, หยุดการเคลื่อนที่, โหลดรูปภาพ, และเลือกรูปภาพตามประเภทของสภาพแวดล้อม.

2.1.7 Class Event

คลาส Event มีเมทอดสำหรับเหตุการณ์ในเกม, รวมถึงการตรวจสอบการชนของตัวละครและคลื่น

2.1.8 Class Element

คลาส Element มีหน้าที่ดึง Font จากไฟล์ฟอนต์ที่ระบุและคืนค่า Font ที่ถูกสร้างขึ้นโดยกำหนดขนาดตามที่ระบุ.

2.1.9 Class EleLabel

คลาส EleLabel คือตัวที่ช่วยในการแสดงข้อความในโปรแกรม โดยจะกำหนดรูปแบบของตัวหนังสือและตำแหน่งของข้อความนั้นในหน้าต่างโปรแกรม

2.1.10 Class EleButton

คลาส EleButton ใช้ในการสร้างปุ่มที่เราสามารถกดได้ในโปรแกรม ซึ่งเมื่อผู้ใช้คลิกที่ปุ่มนี้ โปรแกรมสามารถทำงานตามที่ได้กำหนดไว้ เช่น เปลี่ยนหน้าจอ, ทำการคำนวณ, หรือดำเนินการใด ๆ ตามที่โปรแกรมต้องการ

2.2 รูปแบบการพัฒนา Application / Applet

1. Application Format (รูปแบบการพัฒนาแอปพลิเคชัน):

ประกอบด้วยหลายๆ คลาสที่เกี่ยวข้องกัน เพื่อสร้างและดำเนินการต่างๆ ในแอปพลิเคชัน ในที่นี้มีคลาสหลัก คือ Game, Menu` ที่ถูกสร้างขึ้นเพื่อจัดการกับเกมและหน้าเมนูต่างๆ ของแอปใช้ Swing ในการสร้าง UI elements เช่น JFrame, JPanel, JButton, JLabel เพื่อสร้างหน้าต่างและองค์ประกอบต่างๆ ในแอป

2. Composition (การใช้งานสำหรับการรวมองค์ประกอบ):

- Gameคือหน้าเกมหลักที่สร้างและควบคุมเกมมี JPanel เพื่อสร้างพื้นที่ที่สามารถวาดภาพได้ ประกอบด้วยองค์ประกอบต่างๆ เช่น Chicky, Wave, Environment
- Menu` คือหน้าที่แสดงหลังจากเกมเสร็จสิ้นหรือต้องการเรียกหน้าเมนูประกอบด้วย JButton, JLabel เพื่อแสดงข้อความและปุ่มที่ให้ผู้เล่นทำการ Restart เกม

2.3 คลาสที่มีการใช้คุณสมบัติของ OOP

2.3.1 คลาสที่มีการใช้ Constructor

```
2 usages
public Game(){
    this.setBounds( x: 0, y: 0, width: 1000, height: 600);
    this.addKeyListener( l: this);
    this.setLayout(null);
    this.setFocusable(true);
}
```

ในรูปแบบเป็นตัวอย่างจากคลาสGame ซึ่งในโปรเจกมีการใช้ Constructor เกือบทุกคลาส

2.3.2 คลาสที่มีการใช้Encapsulation

```
public class Game extends JPanel implements KeyListener{

    no usages
    private static final long serialVersionUID = 1L;

    // set เวลาการปล่อยของ object //set size player // set ความสูงของ object
    1 usage
    private static int speed = 50, chickySize = 60, waveHeight = 50;
    4 usages
    private static int base=400, xStart = 1000; // set position ground // จุดที่ให้เจอนแมวในการเลื่อน
    6 usages
    private long point = 0, lastPress=0; // variable for keep score

    // new object chicky and set position
    9 usages
    private Chicky chicky = new Chicky( x: 100, y: base-10);
```

ในรูปแบบเป็นตัวอย่างจากคลาสGame ซึ่งในโปรเจกมีการใช้ Encapsulation ในทุกคลาส

2.3.3 คลาสที่มีการใช้Polymorphism

ไม่พบการใช้PolymorphismในทุกคลาสของโปรเจกPolymorphism มักจะเกี่ยวข้องกับการใช้ interface

2.3.4 คลาสที่มีการใช้ Inheritance

```
public class Game extends JPanel implements KeyListener{
```

1 usage

```
public class Menu extends JPanel {
```

คลาสที่มีการใช้ Inheritance คือ Menu class ที่สืบทอดจาก JPanel class และ Game class ที่สืบทอดจาก JPanel และ KeyListener โดยส่วนนี้เป็นการใช้ Inheritance เพื่อสร้างความสัมพันธ์ระหว่างคลาสแม่ (JPanel) และคลาสลูก (Menu, Game)

2.3.5 คลาสที่มีการใช้ interface-implement

2 usages

```
public class Display extends JFrame implements ActionListener{
```

```
@Override
public void actionPerformed(ActionEvent e) {
    //check ว่ากดrestartไหม ไข่ก็gameใหม่
    if(e.getActionCommand().equals("restart")) {
        removeContent();
        Game game = new Game();
        this.getContentPane().add(game);
        game.requestFocus();
    }
}
```

คลาสที่มีการใช้ interface-implement ในโค้ดที่คุณให้มานั้นคือ Display ที่ implement ActionListener. นอกจากนี้, EleButton และ Menu ก็ใช้ interface ActionListener ในการทำงานกับ ActionListener ด้วย

2.3.5 คลาสที่มีการใช้ Composition

```
9 usages
private Chicky chicky = new Chicky( x: 100, y: base-10);
2 usages
static Display display;

//////// size wave
1 usage
private Wave[] waveSet = makeWave( size: 4); // set n wave

// for cloud, background
1 usage
private Environment[] envSet = makeEnv( size: 2, Environment.CLOUD);
9 usages
private Environment background = new Environment( x: xStart-100, y: base-150, game: this, Environment.BACKGROUND, speed: 4);
```

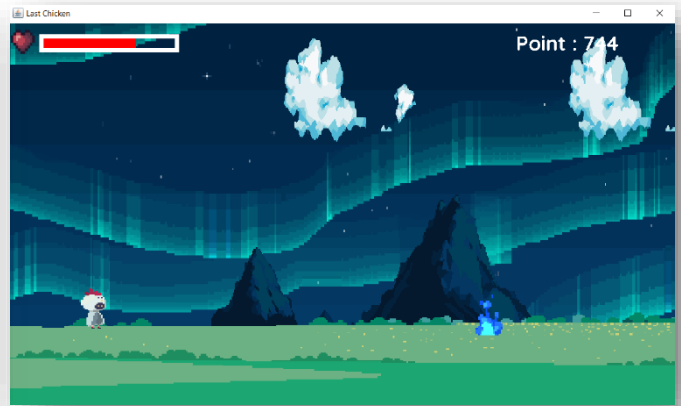
ในตัวอย่างนี้, Game มี Chicky, Wave, Environment, และ Display เป็นองค์ประกอบ (objects) ซึ่งถูกสร้างและใช้งานภายใน Game class.

2.4 หน้าจอ GUI

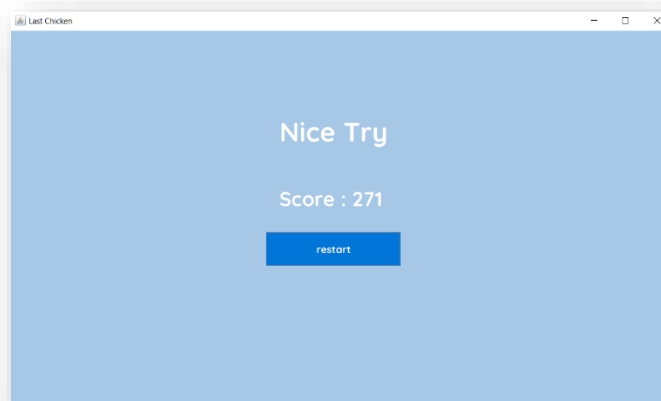
2.4.1 ส่วนประกอบของ GUI

1. JPanel (Game, Menu) สร้าง JPanel ที่ใช้เป็นพื้นที่ในการทำงานหลักของGUI ซึ่งในที่นี้ถูกใช้เพื่อแสดงภาพ,สร้างการแสดงผล และจัดการกับเหตุการณ์ต่างๆ เช่น KeyEvents
- 2.JFrame(Display) สร้าง JFrame ที่เป็นหน้าต่างหลักของแอปพลิเคชัน GUI โดยที่มีการกำหนดขนาด, ชื่อ และ ตำแหน่งของหน้าต่างเสมือนเป็นกรอบรูป และนำ JPanel มาใส่
- 3.Graphic2D สำหรับการวาดภาพและกราฟิกต่างๆ บนPanel
- 4.Character package มีคลาส Chicky, Environment, Wave ที่ใช้ในการสร้างวัตถุ ตัวละครต่างๆในเกม
- 5.Element package ประกอบด้วยองค์ประกอบที่ใช้ในการสร้างข้อความและปุ่มที่ขึ้นแสดงโชว์บนตัวเกม
- 6.Button มีการใช้ปุ่ม Button โดยการใช้ Keylistener ในการรับการกดปุ่ม

-หน้าเล่นเกมก็จะมีการวาง HP อยู่บนซ้าย Score อยู่บนขวาและมีการวางตัวละคร และ object อยู่ที่พื้นของฉากแล้วก็จะเปลี่ยน Background และ สไลด์ Background และฉากหลังก็จะเปลี่ยนไปตามค่าที่กำหนดโดยยึดจากคะแนน



-หน้าเมื่อจบเกม จะมีการจัดการวางข้อความตาม Layout ที่ได้กำหนดไว้ มีคะแนน และปุ่ม Restart เมื่อกดตัวเกมก็จะกลับมา Run ใหม่อีกครั้ง



2.5 Event handling

- Class Menu: สืบทอดจาก JPanel และมีการใช้ 'ActionListener' เพื่อตรวจจับการคลิกปุ่ม Restart ที่ใช้ในการรีเซ็ตเกมหลังจากที่ผู้เล่นสิ้นสุดเกม เมื่อปุ่ม Restart ถูกคลิก actionPerformed method ของ 'Display' Class ถูกเรียกเพื่อสร้างเกมใหม่

- Game Class: มีการใช้ KeyListener เพื่อตรวจจับการกดปุ่มบนคีย์บอร์ด โดยส่วน keyPressed method ที่ตรวจสอบการกดปุ่มสำหรับการกระโดดของตัวละคร เมื่อผู้เล่นกดปุ่ม spacebar หรือปุ่มลูกศรขึ้น ทำให้ตัวละครกระโดดขึ้น

- รูปภาพและไอคอน: การโหลดรูปภาพและไอคอนที่ใช้ในการแสดงผลบนหน้าจอ

- การจัดการกับการแสดงผล: การใช้ Graphics และ Graphics2D เพื่อวาดรูปภาพและแสดงผลบนหน้าจอ GUI

- การสร้างอ็อบเจกต์เกม: การสร้างอ็อบเจกต์ที่ใช้ในการควบคุมเกม เช่น ตัวละคร, วัตถุภายในเกม, และกระบวนการที่เกี่ยวข้องกับเกม

- การจัดการกับเหตุการณ์การชน: การตรวจสอบการชนของอ็อบเจกต์ในเกม จะตรวจสอบของตัวละครกับอุปสรรค

2.6 Algorithm

1. การจัดการตำแหน่งและการเคลื่อนที่ของวัตถุ: ในคลาส Wave และ Environment, มี Algorithm ที่ใช้ Timer เพื่อให้วัตถุเคลื่อนที่ตามเวลาที่กำหนด.
2. การจัดการการกระโดดของตัวละคร: ในคลาส Chicky, มี Algorithm ที่ใช้ Timer เพื่อให้ตัวละครกระโดดขึ้นและลง โดยกำหนดตำแหน่งต่างๆ ในระหว่างกระโดด.
3. การตรวจสอบการชน: ใน Event มี Algorithm ที่ใช้ในการตรวจสอบการชนระหว่าง Chicky และ Wave, หากมีการชนกัน checkHit จะคืนค่า true.
4. การวาด UI และสร้างกราฟิก: ในคลาส Game, มี Algorithm ที่ใช้ในการวาด UI และสร้างกราฟิกบนหน้าต่าง, เช่นการวาดตัวละคร, สถานะเกม, และวัตถุต่างๆ.
5. การจัดการ Event Handling: ใน Display และ Game, มี Algorithm ที่ใช้ในการจัดการ Event Handling เช่นการตรวจสอบการคลิกปุ่ม Restart และการตรวจจับการกดปุ่มบนคีย์บอร์ด.
6. การจัดการการจบเกม: ใน Game, มี Algorithm ที่ใช้ในการตรวจสอบเมื่อตัวละครตกต่ำเหนือพื้นหรือสุดชั้นหรือไม่, และหากเงื่อนไขนี้เป็นจริง, จะมีการเรียก method เพื่อแสดงหน้าจอเมนู.
7. การจัดการกับการเคลื่อนที่ของตัวละครและวัตถุ: ใน Game, มี Algorithm ที่ใช้ในการควบคุมการเคลื่อนที่ของตัวละครและวัตถุต่างๆ เช่น move method ใน Wave และ Environment.

บทที่ 3

สรุป

3.1 ปัญหาที่พบระหว่างการพัฒนา

1. ไม่มีความรู้ความเข้าใจมากพอ ที่จะทำให้นมันออกมาได้ดี
2. เวลาที่มีไม่มากพอ ตารางในการทำงานใกล้เคียงกับตารางสอบ Final
3. ด้านกราฟฟิก บางอย่างเราก็ไม่สามารถวาดขึ้นมาได้

3.2 จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

-มีสตอรี่ที่น่าสนใจ และตัวละครที่ดี

3.2 คำแนะนำ

อยากให้อาจารย์มีเวลาพักเบรกบ้าง 5-10 นาที และก็เวลาสอนอยากให้ อาจารย์พิมพ์ให้ดู
ละให้นักศึกษาพิมพ์ตาม มันเข้าใจดีครับ บางเรื่องอาจารย์สอนรวดเดียว แล้วให้ทำ Lap มันยังไม่
เข้าใจมากพอที่จะทำ Lap ได้

และการทำเกม ก็ต้องเรียนให้จบก่อนถึงจะ เข้าใจว่าอะไรคืออะไร พอเรียนจบ เริ่มทำเกม
การสอบ Final ก็ใกล้เข้ามาแล้ว เลยเวลามันอาจจะน้อยไปหน่อย ก็อาจผิดที่ผมเองด้วยที่ไม่หาศึกษา
เองก่อน

อาจารย์สอนดีครับไม่มีปัญหาอะไรเลย