



รายงาน

เรื่อง Project CS111

สมาชิก

นางสาว กัญญาณัฐ เดชอินทร์ 640961801

นาย ชนเทพ โรจนไพรวงศ์ 6409618029

นาย พันกร ละอออ่อน 6409700074

นางสาว รัชชนก สายันท์ 6409700041

นางสาว ชลวรรณ เรือนปิงวัง 6409700124

เสนอ

อาจารย์ กฤตคม ศรีจิรานนท์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา คพ.111 การโปรแกรมเชิงวัตถุ
ภาคเรียนที่ 2 ปีการศึกษา 2564 มหาวิทยาลัยธรรมศาสตร์ ศูนย์ลำปาง

สารบัญ

เรื่อง	หน้า
หน้าปก	ก
สารบัญ	๗
หน้าที่สมาชิก	1
ส่วนประกอบเกม และขั้นตอนการเล่น	2
อธิบายโค้ด	6

หน้าที่สมาชิก

นางสาว กัญญาณัฐ เดชอินทร์ 640961801

1. ทดสอบเกมเพื่อหาข้อผิดพลาด หรือความผิดปกติ
2. ดีไซน์การจัดวางภายใน panel และออกแบบ character
3. ออกแบบความสัมพันธ์ของ Character (Behind: Cloud และ Ground)

นาย ธนเทพ โรจนไพรวงศ์ 6409618029

1. เขียนโค้ดเพื่อควบคุมการทำงานภายใน panel ต่าง ๆ
2. เขียนโค้ดเพื่อควบคุมการทำงานขณะเล่นเกม
3. ออกแบบความสัมพันธ์ของ Character (Animal: Cat, Bird และ Rabbit)

นาย พันกร ละอออ่อน 6409700074

1. เขียนโค้ดเพื่อจัดการ content ภายใน frame เมื่อเกิด event
2. เขียนโค้ดเพื่อควบคุมการทำงานขณะเล่นเกม
3. ออกแบบความสัมพันธ์ของ Character (Animal: Cat, Bird และ Rabbit)

นางสาว ธัญชนก สายัณห์ 6409700041

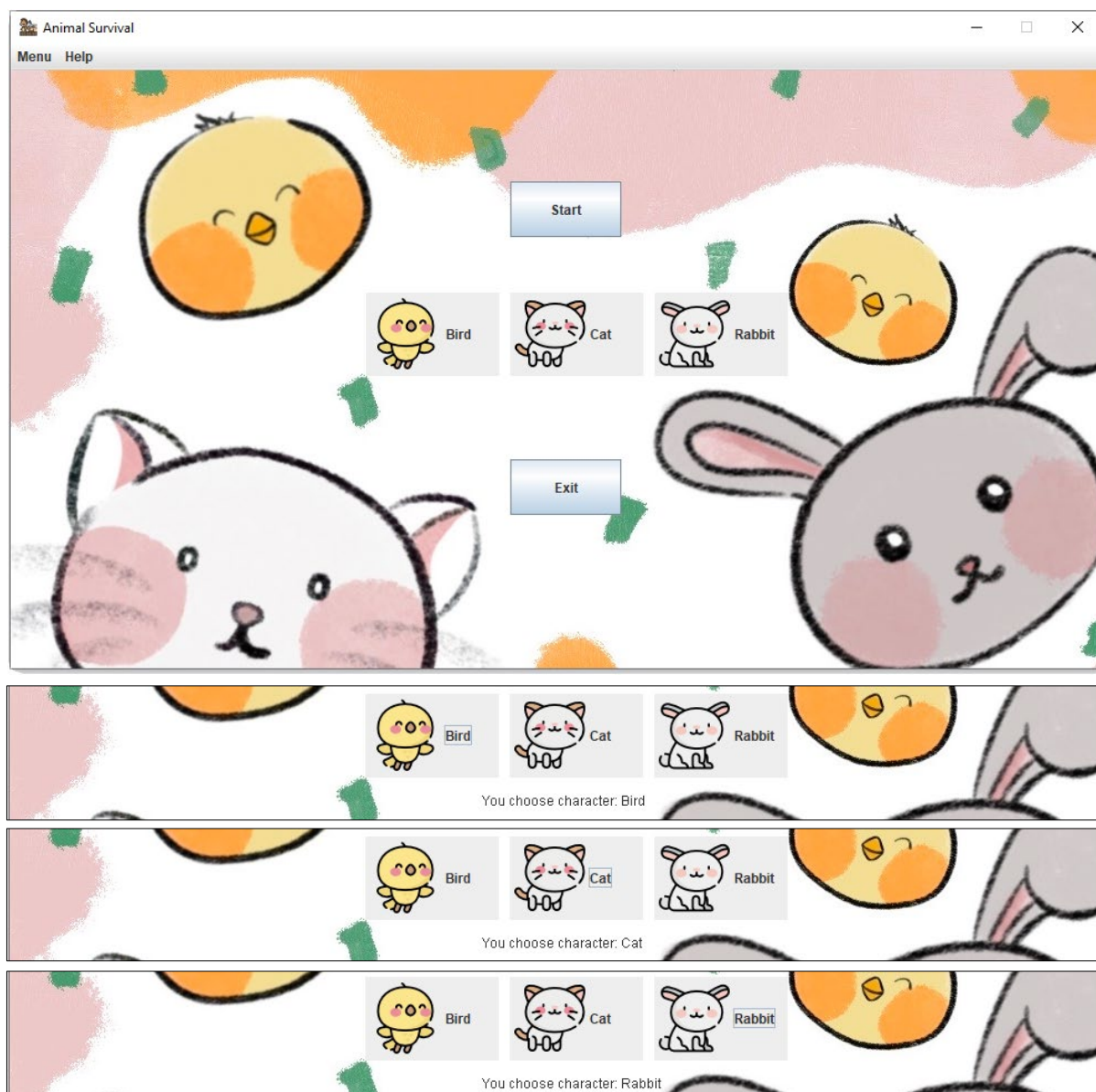
1. เขียนโค้ดเพื่อควบคุมการทำงานภายใน panel ต่าง ๆ
2. เขียนโค้ดเพื่อควบคุมการทำงานขณะเล่นเกม
3. ออกแบบความสัมพันธ์ของ Character (Enemie)

นางสาว ชมลวรรณ เรือนปิงวัง 6409700124

1. ทดสอบเกมเพื่อหาข้อผิดพลาด หรือความผิดปกติ
2. ดีไซน์การจัดวางภายใน panel และออกแบบ character
3. ออกแบบความสัมพันธ์ของ Character (Behind: Cloud และ Ground)

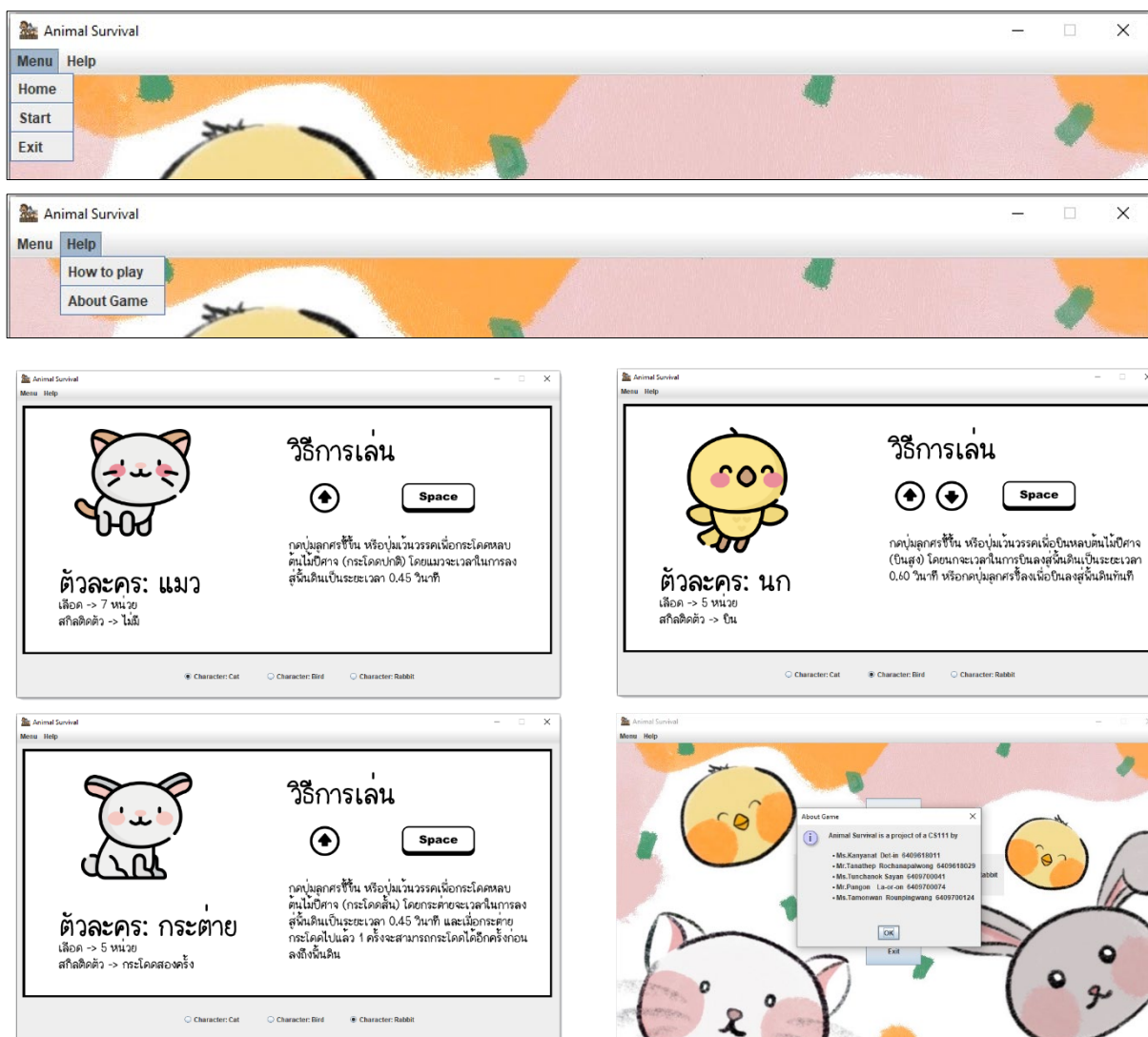
ส่วนประกอบเกม และขั้นตอนการเล่น

1. หน้าแรก (Home)



หน้าแรกเป็นหน้าเริ่มต้นที่ผู้เล่นจะทำการเลือกตัวละครเพื่อเล่นเกม โดยหลักจากเลือกตัวละครจะแสดงสถานะที่เลือกดังตัวอย่างรูปข้างต้น หากผู้เล่นกดเล่นเกมโดยไม่ได้เลือกตัวละครก่อนเล่นจะเป็นการสุ่มตัวละครให้ผู้เล่น

2. แถบเมนู (Menu bar)

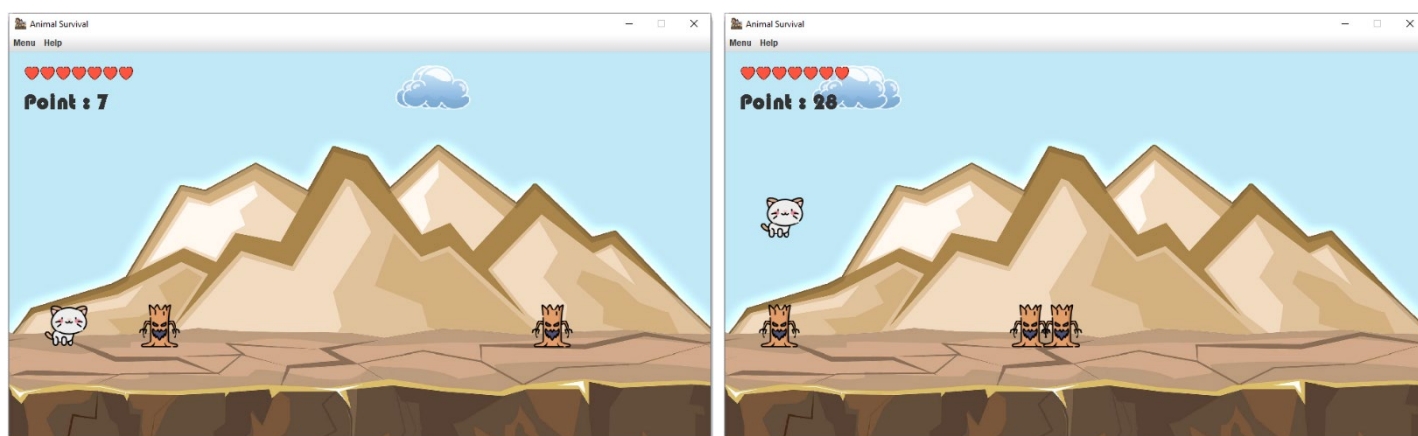


แถบเมนูจะมีเมนูหลัก 2 เมนูคือ Menu และ Help ซึ่งมีเมนูย่อยดังนี้

1. Home: เปิดหน้าแรกเพื่อทำการเลือกตัวละครก่อนเริ่มเกม
2. Start: เริ่มเกม โดยหากผู้เล่นกดปุ่ม Start ก่อนเริ่มเกมจะเป็นการเล่นเกมตามตัวละครที่ผู้เล่นเลือก หรือสุ่มตัวละคร แต่ถ้าหากผู้เล่นกดขณะกำลังเล่นเกมอยู่จะเป็นการเริ่มเกมใหม่โดยใช้ตัวละครเดิมที่ผู้เล่นกำลังเล่นอยู่ ณ ขณะนั้น
3. Exit: ออกจากโปรแกรม หรือเกม
4. How to play: เป็นหน้าที่แสดงวิธีการเล่นของตัวละครต่าง ๆ ภายในเกมดังตัวอย่างรูปข้างต้น
5. About game: แสดงข้อมูลเกี่ยวกับเกมดังตัวอย่างรูปข้างต้น

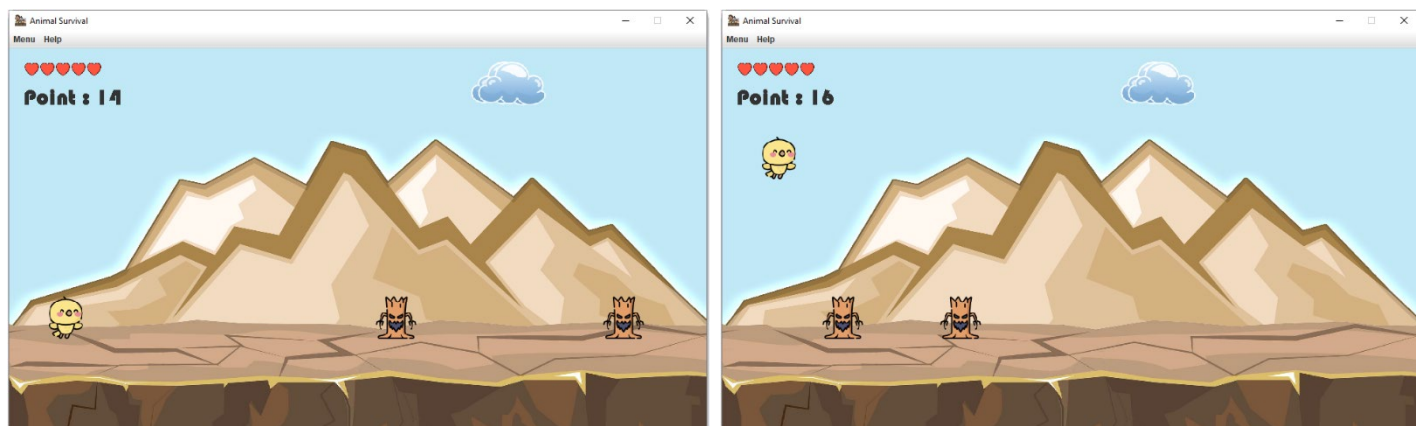
3. วิธีการเล่น

3.1 แมว



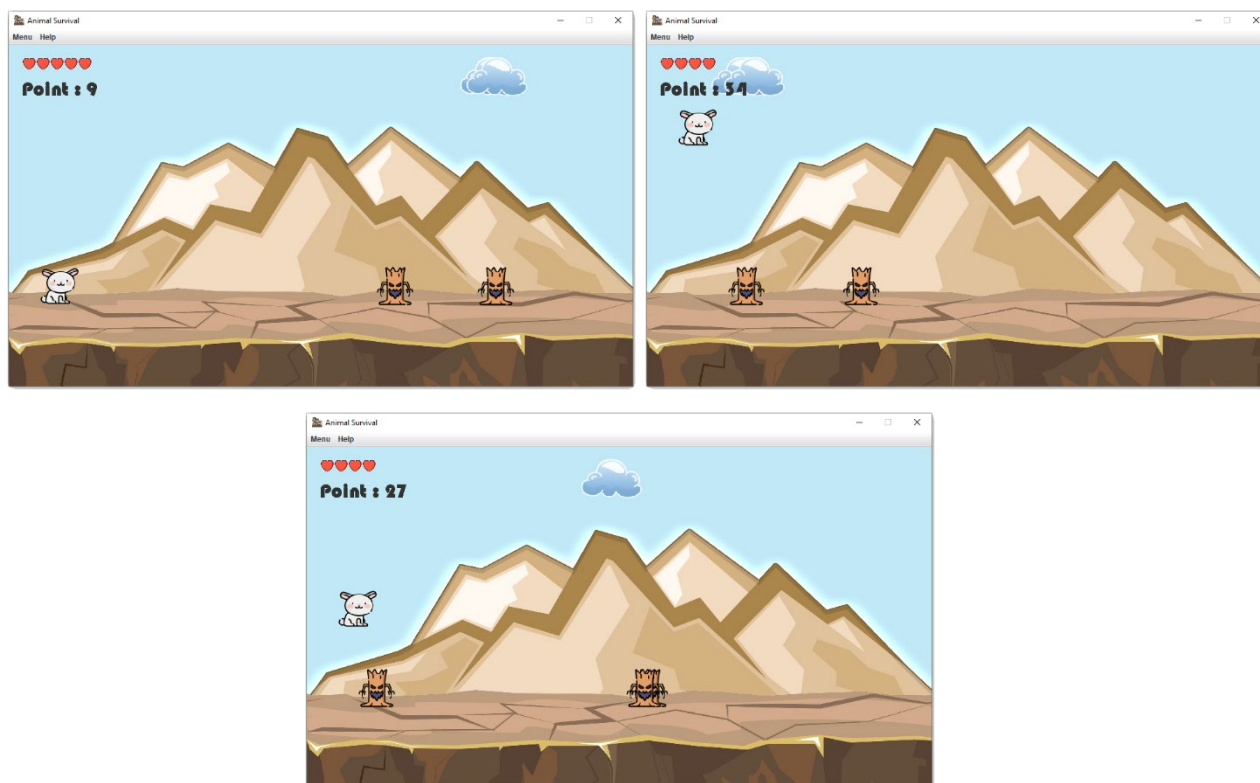
แมวเป็นตัวละครพื้นฐาน ไม่มีสกิลติดตัว แต่มีจำนวนเลือดมากกว่าตัวละครปกติ 2 หน่วย (รวม 7 หน่วย) โดยการบังคับตัวละครแมวนั้นใช้เพียงปุ่มเว้นวรรค หรือปุ่มลูกศรขึ้น เพื่อทำการกระโดดหลบต้นไม้ ปีสาง (150 หน่วย) หลังจากกระโดดแล้วเมื่อเวลาผ่านไป 0.45 วินาที แมวจะดิ่งลงสู่พื้นดินดังตำแหน่งเดิม

3.2 นก



นกเป็นตัวละครที่มีสกิลพื้นฐานคือ “บิน” และมีจำนวนเลือด 5 หน่วย โดยตัวละครนกนั้นสามารถเคลื่อนที่ได้ 2 แบบคือบินขึ้น (225 หน่วย) เพื่อทำการบินหลบต้นไม้ ปีสาง โดยใช้ปุ่มเว้นวรรค หรือปุ่มลูกศรขึ้น และบินลงเพื่อร่อนลงสู่พื้นดิน โดยใช้ปุ่มลูกศรชี้ลง หรือหลังจากบินแล้วผ่านไป 0.6 วินาที

3.3 กระต่าย



กระต่ายเป็นตัวละครที่มีสกิลติดตัวคือ “กระโดดสองครั้ง” และมีจำนวนเลือด 5 หน่วยโดยการบังคับตัวละครกระต่ายนั้นใช้เพียงปุ่มเว้นวรรค หรือปุ่มลูกศรขึ้นเพื่อทำการกระโดดหลบต้นไม้ปีศาจ (125 หน่วย) และหลังจากกระโดดไปแล้ว 1 ครั้งจะสามารถกระโดดได้อีกครั้งก่อนลงถึงพื้นดิน หลังจากกระโดดแล้วเมื่อเวลาผ่านไป 0.45 วินาที แมวจะดิ่งลงสู่พื้นดินดังตำแหน่งเดิม

อธิบายโค้ด

package character

Character.java

เป็นคลาสชนิด abstract ออกแบบไว้เพื่อให้คลาสอื่นสืบทอดนำไปสร้างคลาสประเภท character เช่น สัตว์ต่าง ๆ สิ่งกีดขวางต่าง ๆ หรือฉากประกอบ เป็นต้น มีตัวแปรวัตถุประเภท integer จำนวน 3 ตัว คือ positionX, positionY และ size เพื่อเก็บตำแหน่ง และขนาดของวัตถุประเภท character มี getter และ setter เมทอดสำหรับทุกตัวแปรวัตถุ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าเริ่มต้นให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา และมีเมทอดประเภท abstract อยู่ 2 เมทอด คือ move(JPanel panel): void และ getImage(): BufferedImage เพื่อใช้กำหนด actions ให้กับ character และใช้คืนค่ารูปภาพของคลาสนั้น ๆ

```
int positionX, positionY, size;
```

Character 1: ตัวแปรวัตถุของคลาส

```
public int getPositionX() {
    return positionX;
}
public void setPositionX(int x) {
    positionX = x;
}
public int getPositionY() {
    return positionY;
}
public void setPositionY(int y) {
    positionY = y;
}
public int getSize() {
    return size;
}
public void setSize(int size) {
    this.size = size;
}
// actions of animal
public abstract void move(JPanel panel);
// image of animal
public abstract BufferedImage getImage();
```

Character 2: getter, setter และ abstract เมทอดของคลาส

Animal.java

เป็นคลาสชนิด abstract ออกแบบไว้เพื่อให้คลาสอื่นสืบทอดนำไปสร้างคลาสประเภทตัวละครสัตว์ เช่น แมว นก กระต่าย เป็นต้น มีตัวแปรวัตถุประเภท integer จำนวน 2 ตัวคือ health และ jumpHigh เพื่อเก็บจำนวนเลือด และความสูงในการกระโดดของสัตว์นั้น ๆ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา และมี getter และ setter สำหรับทุกตัวแปรวัตถุ

```
private int health, jumpHigh;
```

Animal 1: ตัวแปรวัตถุของคลาส

```
public int getHealth() {  
    return health;  
}  
public void setHealth(int health) {  
    this.health = health;  
}  
public void setJumpHigh(int jumpHigh) {  
    this.jumpHigh = jumpHigh;  
}  
public int getJumpHigh() {  
    return jumpHigh;  
}
```

Animal 2: getter และ setter สำหรับตัวแปรวัตถุของคลาส

Behind.java

เป็นคลาสชนิด abstract ออกแบบไว้เพื่อให้คลาสอื่นสืบทอดนำไปสร้างคลาสประเภทพื้นหลัง (ที่มีการเคลื่อนไหว) เช่น ก้อนเมฆ เป็นต้น มีตัวแปรวัตถุประเภท integer จำนวน 1 ตัว คือ speed เพื่อเก็บความเร็วในการเคลื่อนที่ของวัตถุนั้นๆ และประเภท Timer1 ตัวคือ speed ไว้จัดการเวลาเพื่อเคลื่อนที่มีเมที่อดตัวสร้างเพื่อใช้กำหนดค่าเริ่มต้นให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา และมี getter และ setter เมที่อดสำหรับทุกตัวแปรวัตถุ

```
protected Timer timer;
private int speed;
```

Behind 1: ตัวแปรวัตถุของคลาส

```
public int getSpeed() {
    return speed;
}
public void setSpeed(int speed) {
    this.speed = speed;
}
```

Behind 2: getter และ setter ของตัวแปร speed ของคลาส

Cat.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทสัตว์ (แมว) โดยสืบทอดมาจากคลาส Animal มีตัวแปรคลาสประเภท String 1 ตัว เพื่อใช้ออกสถานะสัตว์ของคลาสว่าเป็นสัตว์อะไร และตัวแปรวัตถุประเภท BufferedImage เพื่อใช้สำหรับจัดการรูปภาพ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา โดยคลาส Cat จะทำการกำหนดรายละเอียดให้กับเมทอด move(JPanel panel): void และ getImage(): BufferedImage โดยจะกำหนดการเคลื่อนไหวของ Cat ที่เมทอด move() และกำหนดรูปภาพให้กับเมทอด getImage() การเคลื่อนไหวของ Cat นั้นจะเป็นการกระโดดพื้นฐานทั่วไปโดยทำการลดแกนค่า y เมื่อทำการกระโดด และหลังจากกระโดดผ่านไปแล้ว 0.45 วินาที จะเพิ่มแกนค่า y กลับมายังตำแหน่งเดิมก่อนการกระโดด

```
@Override
public BufferedImage getImage() {
    try {
        image = ImageIO.read(new File("image\\cat.png"));
        return image;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return image;
}
```

Cat 1: กำหนดที่อยู่ของรูปภาพแมว

```
@Override
public void move(JPanel panel) {
    positionY -= getJumpHigh();
    panel.repaint();
    Timer timer = new Timer(450, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(positionY < Game.startY) {
                positionY += getJumpHigh();
                panel.repaint();
            }
        }
    });
    timer.setRepeats(false);
    timer.start();
}
```

Cat 2: กำหนดการเคลื่อนที่ของแมว (กระโดด)

Bird.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทสัตว์ (นก) โดยสืบทอดมาจากคลาส Animal มีตัวแปรคลาสประเภท String 1 ตัว เพื่อใช้บอกสถานะสัตว์ของคลาสว่าเป็นสัตว์อะไร และตัวแปรวัตถุประเภท BufferedImage เพื่อใช้สำหรับจัดการรูปภาพ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา โดยคลาส Bird จะทำการกำหนดรายละเอียดให้กับเมทอด move(JPanel panel): void และ getImage(): BufferedImage โดยจะกำหนดการเคลื่อนไหวของ Cat ที่เมทอด move() และกำหนดรูปภาพให้กับเมทอด getImage() การเคลื่อนไหวของ Bird นั้นจะเป็นการบินขึ้นโดยการลดแกนค่า y เมื่อทำการบิน และสามารถบังคับให้บินลงได้ หรือหลังจากบินไปแล้ว 0.60 วินาทีจะเพิ่มค่าแกน y กลับมายังตำแหน่งเดิมก่อนบิน

```
@Override
public BufferedImage getImage() {
    try {
        image = ImageIO.read(new File("image\\bird.png"));
        return image;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return image;
}
```

Bird 1: กำหนดที่อยู่ของรูปภาพนก

```
@Override
public void move(JPanel panel) {
    positionY -= getJumpHigh();
    panel.repaint();
    Timer timer = new Timer(600, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(positionY < Game.startY) {
                positionY += getJumpHigh();
                panel.repaint();
            }
        }
    });
    timer.setRepeats(false);
    timer.start();
}

public void move(JPanel panel, String status) {
    positionY += Game.startY - positionY;
}
```

Bird 2: กำหนดการเคลื่อนที่ของนก (บินขึ้น และบินลง)

Rabbit.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทสัตว์ (นก) โดยสืบทอดมาจากคลาส Animal มีตัวแปรคลาสประเภท String 1 ตัว เพื่อใช้ออกสถานะสัตว์ของคลาสว่าเป็นสัตว์อะไร และตัวแปรวัตถุประเภท BufferedImage เพื่อใช้สำหรับจัดการรูปภาพ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา โดยคลาส Bird จะทำการกำหนดรายละเอียดให้กับเมทอด move(JPanel panel): void และ getImage(): BufferedImage โดยจะกำหนดการเคลื่อนไหวของ Cat ที่เมทอด move() และกำหนดรูปภาพให้กับเมทอด getImage() การเคลื่อนไหวของ Rabbit นั้นจะเป็นการกระโดดคล้ายกระโดดพื้นฐานแต่จะมีระยะที่สั้นกว่า โดยเมื่อกระโดดไปแล้ว 1 ครั้งสามารถกระโดดได้อีกครั้งก่อนลงถึงพื้น และหลังจากกระโดดไปแล้ว 0.45 วินาที จะเพิ่มค่าแกน y ตามค่าระยะความสูงของการกระโดดแต่ละครั้ง

```
@Override
public BufferedImage getImage() {
    try {
        image = ImageIO.read(new File("image\\rabbit.png"));
        return image;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return image;
}
```

Rabbit 1: กำหนดที่อยู่ของรูปภาพนก

```
@Override
public void move(JPanel panel) {
    amountJump++;
    if(amountJump < 3) {
        positionY -= getJumpHigh();
        panel.repaint();
        Timer timer = new Timer(450, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if(positionY < Game.startY) {
                    positionY += getJumpHigh();
                    panel.repaint();
                }
            }
        });
        timer.setRepeats(false);
        timer.start();
    }
}
```

Rabbit 2: กำหนดการเคลื่อนที่ของกระต่าย (กระโดด) โดยมีการตรวจสอบจำนวนครั้งของการกระโดดโดยห้ามเกิน 2 ครั้ง

Enemy.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทสิ่งกีดขวาง (ในตอนนี้สร้างแค่แบบเดียวคือ ต้นไม้ป่า) โดยสืบทอดคลาสมาจาก Character มีตัวแปรวัตถุประเภท Timer 1 ตัว คือ timer ไว้จัดการเวลาเพื่อเคลื่อนที่ ประเภท integer 1 ตัว คือ speedEnemy ไว้เก็บค่าความเร็วของสิ่งกีดขวาง และมีประเภท boolean 1 ตัว คือ checked ไว้เก็บค่าเท็จ/จริง ว่าสิ่งกีดขวางนั้น ๆ ได้ถูกตรวจสอบการชนไปแล้วหรือยัง มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามค่าที่รับเข้ามา มี getter และ setter สำหรับตัวแปรวัตถุ checked การเคลื่อนไหวของ Enemy นั้นจะเป็นการเคลื่อนไหวจากทางด้านขวาไปซ้ายโดยการลดแกน x ตามค่าของตัวแปร speedEnemy เมื่อตำแหน่งของวัตถุหลุดขอบ frame ทางด้านซ้ายจะทำการกำหนดตำแหน่งแกน x ใหม่เป็นค่าเริ่มต้นของกรอบ frame บวกกับค่าที่ทำการสุ่มได้ (1000 + (0 ถึง 1500))

```
private int speedEnemy = 30;
//---- for check enemy is cash ----
private boolean checked;
```

Enemy 1: ตัวแปรวัตถุของคลาส

```
public boolean getChecked() {
    return checked;
}
public void setChecked(boolean checked) {
    this.checked = checked;
}
```

Enemy 2: getter และ setter ของตัวแปร checked ไว้ตรวจสอบสถานะการชน

```
public void move(JPanel panel) {
    timer = new Timer(50, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(positionX <= 0) {
                //the end of frame + new start point
                positionX = Game.startX + (int)(Math.random() * 1500);
                checked = false;
            }
            positionX -= speedEnemy;
            panel.repaint();
        }
    });
    timer.start();
}
```

Enemy 3: กำหนดการเคลื่อนที่ของ Enemy โดยมีการกำหนดตำแหน่ง x ใหม่เมื่อมีการหลุดกรอบ

Cloud.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทพื้นหลัง (ก้อนเมฆ) โดยสืบทอดมาจากคลาส Behind มีตัวแปรวัตถุประเภท BufferedImage เพื่อใช้สำหรับจัดการรูปภาพ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสม โดยคลาส Cloud จะทำการกำหนดรายละเอียดให้กับเมทอด move(JPanel panel): void และ getImage(): BufferedImage โดยการเคลื่อนไหวนของ Cloud จะเหมือนกับการเคลื่อนไหวนของ Enemy ทุกอย่าง

```
@Override
public void move(JPanel panel) {
    timer = new Timer(10, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(positionX <= 0) {
                //the end of frame + new start point
                positionX = Game.startX + (int)(Math.random() * 1500);
            }
            positionX -= getSpeed();
            panel.repaint();
        }
    });
    timer.start();
}
```

Cloud 1: การเคลื่อนไหวนของ Cloud เหมือนกับการเคลื่อนไหวนของ Enemy

Groud.java

เป็นคลาสที่ใช้สร้างวัตถุประเภทพื้นหลัง (ก้อนเมฆ) โดยสืบทอดมาจากคลาส Behind มีตัวแปรวัตถุประเภท BufferedImage เพื่อใช้สำหรับจัดการรูปภาพ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสม โดยคลาส Cloud จะทำการกำหนดรายละเอียดให้กับเมทอด move(JPanel panel): void และ getImage(): BufferedImage โดยที่เมทอด move() จะไม่มีการทำงานใด ๆ ภายในเมทอดเพราะวัตถุที่สร้างขึ้นจาก Groud จะไม่มีการเคลื่อนไหวใด ๆ มีแต่เพียงกำหนดรายละเอียดรูปภาพให้กับเมทอด getImage(): BufferedImage เท่านั้น

```
@Override
public void move(JPanel panel) {
    // not move
}
@Override
public BufferedImage getImage() {
    try {
        image = ImageIO.read(new File("image\\ground.png"));
        return image;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return image;
}
```

Groud 1: คลาส Groud มีเพียงการกำหนดที่อยู่ของรูปภาพ ไม่มีการกำหนดการเคลื่อนไหว

package event

Clash.java

เป็นคลาสที่ใช้ตรวจสอบว่าตัวละคร (Animal) มีการชนกันกับสิ่งกีดขวาง (Enemy) หรือไม่ โดยตรวจสอบจากจุดอันดับ (x, y) ของตัวละคร และสิ่งกีดขวางว่ามีนัยของการชนกัน หรือการทับกันหรือไม่ ซึ่งแต่ละรอบของการวนมาของสิ่งกีดขวางสามารถชนกับตัวละครได้เพียง 1 ครั้ง/สิ่งกีดขวาง

```
public static boolean isClash(Animal animal, Enemy tree) {
    // every move in round can clash animal 1 time/enemy (getChecker())
    if(animal.getPositionX() + animal.getSize() >= tree.getPositionX() && animal.getPositionX() <= tree.getPositionX() &&
        animal.getPositionY() + animal.getSize() >= tree.getPositionY() - tree.getSize() && !tree.getChecked()) {
        return true;
    }
    return false;
}
```

Clash 1: คลาส Clash มีเพียงเมทอด isClash เพื่อใช้ในการตรวจสอบการชนกันของตัวละครกับสิ่งกีดขวาง

package game

Welcome.java

เป็นคลาสที่สร้าง panel ขึ้นเพื่อรอให้ frame นำไปเพิ่มและแสดงให้กับผู้เล่น โดยคลาส Welcome จะเป็นหน้าเริ่มต้นของเกม (Home) เพื่อให้ผู้เล่นทำการเลือกตัวละครก่อนเล่น มีตัวแปรวัตถุเพื่อสร้างปุ่มต่าง ๆ, เก็บค่าของรูปภาพของสัตว์แต่ละแบบ, มีตัวแปร choose: String เพื่อใช้เก็บค่าว่าผู้ใช้เลือกตัวละครอะไร, มีกลุ่มของตัวแปรเพื่อเก็บค่าของชื่อสัตว์แต่ละตัว และตัวแปรเพื่อใช้สุ่มค่า มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสมโดยมีการเรียกใช้เมทอด createButton(ActionListener display): void เพื่อสร้างปุ่มตัวเลือกให้กับผู้เล่น มีเมทอด paintComponent(Graphics g): void เพื่อใช้วาดข้อความแสดงสถานะการเลือกตัวละคร และมี getter และ setter สำหรับตัวแปร choose และมีเมทอด getStatus(): String เพื่อใช้คืนค่าสถานะการเลือกของผู้เล่น

HowToPlay.java

เป็นคลาสที่สร้าง panel ขึ้นเพื่อรอให้ frame นำไปเพิ่มและแสดงให้กับผู้เล่น โดยคลาส HowToPlay จะเป็นหน้าเพื่อบอกวิธีการเล่นของตัวละครต่าง ๆ ให้กับผู้ใช้ มีตัวแปรวัตถุเพื่อสร้างปุ่มต่าง ๆ และมีตัวแปร filename: String เพื่อใช้เก็บที่อยู่ของตำแหน่งรูปภาพวิธีการเล่นของตัวละครนั้น ๆ มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสมโดยมีการเรียกใช้เมทอด createButton(ActionListener display): void เพื่อสร้างปุ่มตัวเลือกให้กับผู้เล่น มีเมทอด paintComponent(Graphics g): void เพื่อใช้วาดรูปภาพวิธีการเล่นตัวละครนั้น ๆ ตามตำแหน่งที่ได้จากตัวแปร fileName และมีเมทอด setFileName(String fileName): void เพื่อใช้ตั้งค่าตำแหน่งของรูปภาพตามค่าที่รับเข้ามา

GameOver.java

เป็นคลาสที่สร้าง panel ขึ้นเพื่อรอให้ frame นำไปเพิ่มและแสดงให้กับผู้เล่น โดยคลาส GameOver จะเป็นหน้าที่แสดงขึ้นเมื่อเกมจบลงโดยจะแสดง score ของผู้เล่น และปุ่มทางเลือกเล่นอีกครั้ง กับออกเกม มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสม โดยเรียกใช้เมทอด createLabel(int score): void เพื่อสร้างข้อความแสดงสถานะจบเกม และแสดง score ของผู้เล่น และเรียกใช้เมทอด createButton(ActionListener display): void เพื่อสร้างปุ่มทางเลือกให้กับผู้เล่น และมีเมทอด paintComponent(Graphics g): void เพื่อใช้วาดรูปพื้นหลัง

Game.java

เป็นคลาสที่สร้าง panel ขึ้นเพื่อใช้สำหรับเล่นเกม มีการ implements KeyListener มีเมทอดตัวแปร คลาส 3 ตัว คือ display ไว้สร้าง Frame เพื่อเพิ่ม panel ลงไปและเริ่มเล่นเกม, StartX และ StartY ไว้เก็บค่า ตำแหน่งเริ่มต้นของแกน x และแกน y มีตัวแปร hasReleasedKey ไว้เก็บค่าเท็จ/จริงว่าผู้เล่นมีการปล่อยคีย์ที่ กดอยู่หรือยัง มีตัวแปร score ไว้เก็บคะแนนของผู้เล่น และตัวแปรอื่น ๆ เพื่อสร้าง Character ที่จำเป็นภายใน เกม เช่น สัตว์ สิ่งกีดขวาง ก้อนเมฆ พื้นดิน เป็นต้น มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตาม ค่าที่รับเข้ามา และมีการเรียกใช้เมทอด makeEnemies() และmakeClouds() เพื่อสร้างสิ่งกีดขวาง และก้อนเมฆ มีเมทอด paint(Graphics g): void เพื่อใช้วาดพื้นหลัง drawBackGround(g2D): void, วาดจำนวนเลือดที่ เหลืออยู่ drawHealth(g2D): void, วาดคะแนนของผู้เล่น, วาดตัวละคร, วาดสิ่งกีดขวาง และมีการตรวจสอบ การชนกันของตัวละครกับสิ่งกีดขวาง พร้อมทั้งมีการลดเลือดตัวละครเมื่อมีการชนกัน โดยหากจำนวนเลือด ของตัวละครเหลือ 0 จะทำการจบเกมลงทันที มีเมทอด keyReleased(KeyEvent e): void เพื่อใช้กำหนดค่า ให้กับ hasReleasedKey เมื่อผู้เล่นมีการปล่อยคีย์ที่กดอยู่ และมีเมทอดkeyPressed(KeyEvent e): void เพื่อใช้ ตรวจสอบการกดคีย์ของผู้เล่น โดยหากคีย์นั้นตรงกับคีย์การบังคับตัวละครนั้น ๆ จะมีการเรียกใช้เมทอด move() ของตัวละครนั้น ๆ เพื่อให้เกิดการเคลื่อนไหว หรือมีกำหนดค่าของตัวแปรวัตถุในตัวละครเพื่อให้ ถูกต้อง ตรงตามสถานะการเคลื่อนไหวของตัวละครอย่างปกติ (คีย์ 32 คือเว้นวรรค, คีย์ 38 คือลูกศรขึ้น และ คีย์ 40 คือลูกศรลง)

```
public void paint(Graphics g) {
    super.paint(g);
    Graphics2D g2D = (Graphics2D) g;
    drawBackGround(g2D);
    drawHealth(g2D);
    g2D.setFont(new Font("Bauhaus 93", Font.BOLD, 30));
    g2D.drawString("Point : " + Math.round(score), 20, 80);
    g2D.drawImage(animal.getImage(), animal.getPositionX(), animal.getPositionY(), animal.getSize(), animal.getSize(), null);
    for(Enemy enemy: tree) {
        g2D.drawImage(enemy.getImage(), enemy.getPositionX(), enemy.getPositionY(), enemy.getSize(), enemy.getSize(), null);
        if(Clash.isClash(animal, enemy)) {
            animal.setHealth(animal.getHealth() - 1);
            enemy.setChecked(true);
            if(animal.getHealth() == 0) {
                display.endGame((int)score);
            }
        }
    }
    else {
        score += 0.05;
    }
}
```

Game 1: เมทอด paint ใช้วาดส่วนประกอบต่าง ๆ ที่จำเป็นภายในเกม เช่น ตัวละคร สิ่งกีดขวาง จำนวนเลือด คะแนน พื้นดิน เมฆ เป็นต้น

```

@Override
public void keyPressed(KeyEvent e) {
    //32 is space bar, 38 is up arrow and 40 is down arrow
    if(animal instanceof Cat) {
        if(animal.getPositionY() == Game.startY && hasReleasedKey) {
            if(e.getKeyCode() == 32 || e.getKeyCode() == 38) {
                animal.move(this);
                hasReleasedKey = false;
            }
        }
    }
    }else if(animal instanceof Bird) {
        if(animal.getPositionY() == Game.startY && hasReleasedKey) {
            if(e.getKeyCode() == 32 || e.getKeyCode() == 38) {
                animal.move(this);
                hasReleasedKey = false;
            }
        }else if(e.getKeyCode() == 40) {
            ((Bird)animal).move(this, "down");
            hasReleasedKey = false;
        }
    }else if(animal instanceof Rabbit) {
        if(hasReleasedKey) {
            if(animal.getPositionY() == Game.startY){
                ((Rabbit) animal).resetAmountJump();
            }
            if(e.getKeyCode() == 32 || e.getKeyCode() == 38) {
                animal.move(this);
                hasReleasedKey = false;
            }
        }
    }
}

@Override
public void keyReleased(KeyEvent e) {
    hasReleasedKey = true;
}

public static void main(String[] args) {
    display = new Display();
}

```

Game 2: เมื่อกด keyPressed และ keyReleased ใช้ตรวจสอบการกดคีย์ของผู้เล่นหากเป็นคีย์ที่ตรงกับที่ตรงกับคีย์ที่กดตัวละครั้งนั้น ๆ จะทำการเรียกใช้เมทอด move ของตัวละครเพื่อทำการเคลื่อนไหว ซึ่งมีข้อกำหนดว่าเมื่อทำการเคลื่อนไหวทุกครั้งผู้เล่นทำการปล่อยคีย์ก่อนการเคลื่อนไหวครั้งถัดไปโดยตรวจสอบจากตัวแปร hasKeyReleasedKey

Display.java

เป็นคลาสที่สร้าง Frame ขึ้นเพื่อให้ panel ต่าง ๆ ตาม event ที่เกิดขึ้นจากผู้เล่น มีการจัดการกับ event ที่เกิดขึ้นโดย implements ActionListener มาใช้ มีตัวแปรวัตถุเพื่อสร้าง panel ต่าง ๆ, เพื่อเก็บค่าตัวละครที่ผู้ใช้เลือก, เพื่อสร้างเมนูต่าง ๆ และเพื่อเก็บค่าของรูปภาพไอคอนของ frame มีเมทอดตัวสร้างเพื่อใช้กำหนดค่าให้กับตัวแปรวัตถุตามความเหมาะสม โดยมีการเรียกเมทอด createMenu(): void เพื่อสร้างแถบเมนู และเมทอด startGame(): void เพื่อเริ่มหน้าแรกให้ผู้ใช้เลือกตัวละครในการเล่นเกมี มีตัวแปร removeContent():void ในการจัดการล้างรายละเอียดต่าง ๆ ที่ถูกเพิ่มลงในเฟรมเพื่อให้พร้อมต่อการเพิ่ม panel ใหม่ลงไป มีเมทอด startGame():void, howToPlayGame():void และ gameOver(): void เพื่อใช้เพิ่มหน้า panel ต่าง ๆ ลงไปตาม event ที่เกิดจากผู้ใช้ เช่น เริ่มเกม, แสดงวิธีการเล่นเกม, จบเกม เป็นต้น ซึ่งมีเมทอดที่สำคัญในการจัดการกับ event คือ actionPerformed(ActionEvent e): void เช่น การเปลี่ยน panel ต่าง ๆ, การปิดโปรแกรม, การเลือกตัวละครของผู้เล่น, การเลือกหน้าแสดงวิธีการเล่น, แสดงหน้าต่าง About Game เป็นต้น

```
public void removeContent() {
    this.getContentPane().removeAll();
    this.getContentPane().repaint();
}
public void startGame() {
    this.getContentPane().add(welcome);
}
public void howToPlayGame() {
    removeContent();
    this.getContentPane().add(htp);
}
public void endGame(int score) {
    removeContent();
    this.getContentPane().add(new GameOver(score, this));
}
```

Display 3: เมทอดที่ใช้ในการจัดการล้าง content หรือเพิ่ม content ลงใน frame

```

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equalsIgnoreCase("Restart") || e.getActionCommand().equalsIgnoreCase("Start")
    || e.getActionCommand().equalsIgnoreCase("Home")) {
        removeContent();
        if(e.getActionCommand().equalsIgnoreCase("Start")) {
            if(character == "") {
                welcome.setChoose();
                character = welcome.getChoose();
                this.getContentPane().add(game = new Game(character));
                game.addKeyListener(game);
                game.setFocusable(true);
            }else {
                this.getContentPane().add(game = new Game(character));
                game.addKeyListener(game);
                game.setFocusable(true);
            }
        }else {
            welcome.setChoose("");
            character = "";
            startGame();
        }
        if(game != null) {
            game.requestFocus();
        }
    }else if(e.getActionCommand().equalsIgnoreCase("Exit")) {
        System.exit(0);
    }else if(e.getActionCommand().equalsIgnoreCase("Cat")) {
        welcome.setChoose("Cat");
        this.character = "Cat";
        welcome.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("Bird")) {
        welcome.setChoose("Bird");
        this.character = "Bird";
        welcome.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("Rabbit")) {
        welcome.setChoose("Rabbit");
        this.character = "Rabbit";
        welcome.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("How to play")) {
        howToPlayGame();
    }else if(e.getActionCommand().equalsIgnoreCase("Character: Cat")) {
        http.setFileName("image\\howToPlayCat.png");
        http.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("Character: Bird")) {
        http.setFileName("image\\howToPlayBird.png");
        http.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("Character: Rabbit")) {
        http.setFileName("image\\howToPlayRabbit.png");
        http.repaint();
    }else if(e.getActionCommand().equalsIgnoreCase("About Game")) {
        JOptionPane.showMessageDialog(null, "Animal Survival is a project of a CS111 by\n"
            +"\n • Ms.Kanyanat Det-in 6409618011"
            +"\n • Mr.Tanathep Rochanapaiwong 6409618029"
            +"\n • Ms.Tunchanok Sayan 6409700041"
            +"\n • Mr.Pangon La-or-on 6409700074"
            +"\n • Ms.Tamonwan Rounpingwang 6409700124\n\n\n",
            "About Game", JOptionPane.INFORMATION_MESSAGE);
    }
}

```

Display 4: เมื่อกด actionPerformed ใช้จัดการกับ event ต่าง ๆ ที่เกิดขึ้นจากผู้ใช้ ทั้งภายในส่วนของเมนู, คลาส Welcome, คลาส HowToPlay และคลาส GameOver