



## โครงการ

Mini Project

## จัดทำโดย

6504062610226 นาย วรวิทย์ กิมเฮงหลี

## เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## เกี่ยวกับโครงงาน

ชื่อโปรเจค : Tarzan Escape

นำเสนอโดย : นาย วรวิทย์ กิมเฮงหลี

อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

## บทที่ 1 บทนำ

### ที่มาและความสำคัญของโปรเจค

โปรเจคเกม Tarzan Escape นี้สร้างขึ้นเพื่อนำความรู้ที่ได้จากการเรียนในวิชา Object Oriented Programming ตลอดหนึ่งเทอมการศึกษามาประยุกต์ใช้ในการสร้างเป็นชิ้นงานในรูปแบบเกม

### ประเภทโครงงาน

โปรเจคเกม

### ประโยชน์

1. ช่วยให้เกิดความผ่อนคลายและสนุกสนาน
2. เพื่อฝึกความรอบรู้ในการตัดสินใจ
3. เพื่อฝึกสมาธิในการจดจ่อกับสิ่ง ๆ หนึ่ง

### ของเขตของโครงงาน

ลำดับ	รายการ	ก.ย			ต.ค	
		11-17	18-24	25-30	1-7	8-9
1	ออกแบบตัวละคร					
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง					
3	ลงมือเขียนโปรแกรม					
4	จัดทำเอกสาร					
5	ตรวจสอบและแก้ไขข้อผิดพลาด					

## บทที่ 2 ส่วนการพัฒนา

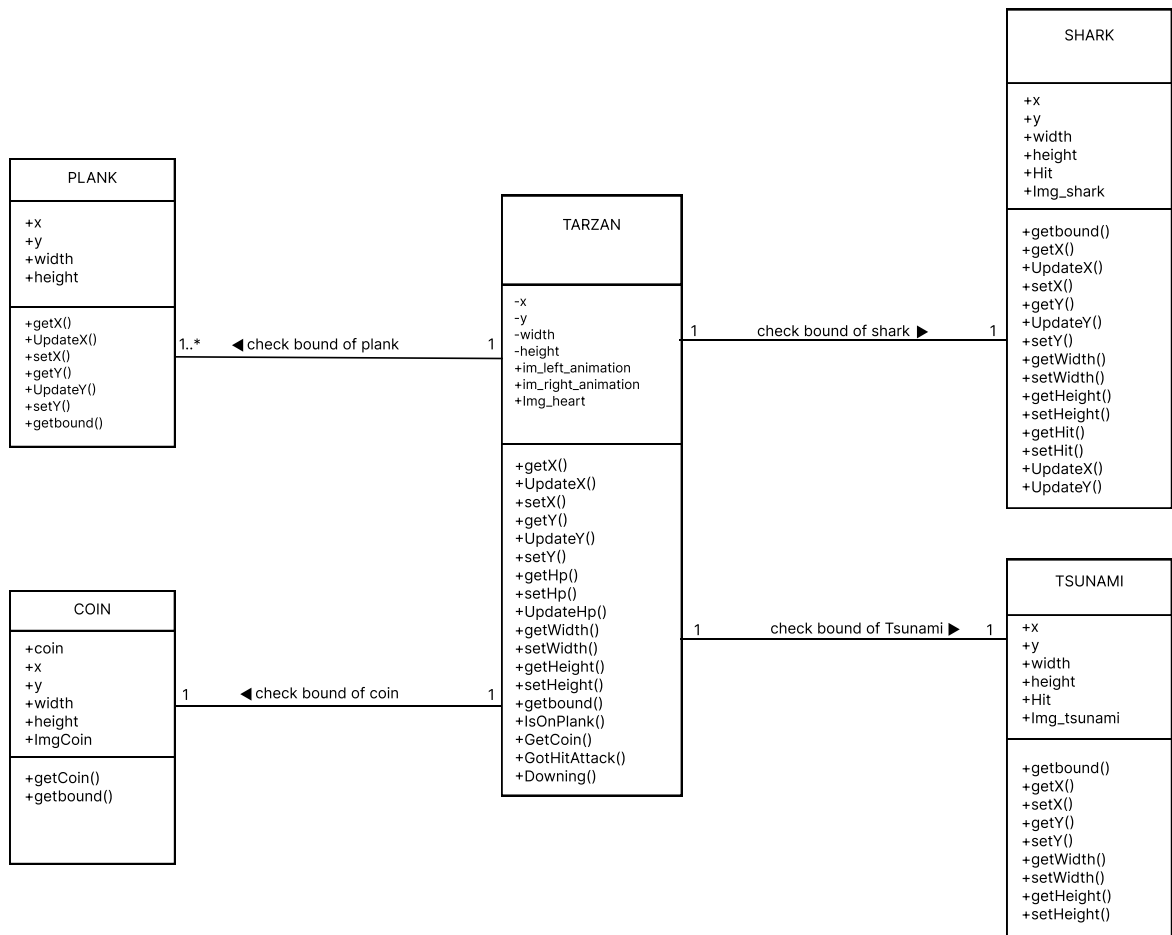
### เนื้อเรื่องย่อของเกม Tarzan Escape

เมื่อโลกเกิดภัยพิบัติน้ำท่วมโลกครั้งยิ่งใหญ่ ทาร์ซานผู้อาศัยอยู่ในป่าต้องทำการหลบหนีภัยพิบัติน้ำท่วมในครั้งนี้ แต่เรื่องที่ไม่คาดคิดก็เกิดขึ้น ภัยพิบัติครั้งนี้ได้นำพาฉลามตัวใหญ่ยักษ์มาด้วย ทำให้ทาร์ซานนั้นต้องเอาชีวิตรอดจากทั้งภัยพิบัติและฉลามนักล่าผู้หิวโหย หนทางเดียวที่ทาร์ซานจะรอดจากภัยพิบัติในครั้งนี้ได้ต้องเก็บเหรียญพิเศษจนครบ 10 เหรียญเพื่อขอพรให้พระเจ้าช่วยหยุดภัยพิบัติในครั้งนี้

### วิธีการเล่น

ใช้ปุ่มลูกศรบนแป้นพิมพ์เพื่อบังคับทิศทางของทาร์ซานในการหลบหนีฉลามและภัยพิบัติน้ำท่วม โดยใช้ลูกศรซ้ายและขวาเพื่อบังคับตัวละครไปในทิศทางซ้ายและขวาและใช้ลูกศรบนเพื่อกระโดดไปบนแผ่นไม้และใช้ลูกศรล่างเพื่อลงจากแผ่นไม้ และเมื่อตัวละครวิ่งผ่านเหรียญก็จะเก็บเหรียญโดยอัตโนมัติ เมื่อเก็บครบ 10 เหรียญก็จะรอดจากการโดนไล่ล่าของฉลามและภัยพิบัติน้ำท่วม แต่ถ้าทาร์ซานตกจากแผ่นไม้ลงน้ำ หรือ โดนฉลามกัดครบ 3 ครั้ง ก็จะแพ้

# คลาสไดอะแกรม



## รูปแบบการพัฒนา Application / Applet

1. ออกแบบ User Interface
  - 1.1 มีการเรียกใช้แพ็คเกจใน java.awt และ javax.swing และ java.net.URL เพื่อใช้ในการออกแบบ User Interface
2. เพิ่มฟังก์ชันการคำนวณเพื่อให้ตัวละครและสิ่งของในเกมเคลื่อนไหว
  - 2.1 มีการเรียกใช้ java.util.ArrayList เพื่อใช้ในการเคลื่อนไหวของตัวละคร
  - 2.2 มีการเรียกใช้ java.util.Random เพื่อใช้ในการสุ่มจุดเกิดของเหรียญบนแผ่นไม้
  - 2.3 มีการเรียกใช้ java.awt.geom.Rectangle2D เพื่อใช้ในการคำนวณการโจมตีของตัวละครในเกม
3. เพิ่มฟังก์ชันการควบคุมตัวละครจากแป้นพิมพ์ โดยการเรียกใช้ Keylistener จากแพ็คเกจ java.awt

### อธิบายส่วนของโปรแกรม

#### ในส่วนของคุณา TARZAN

```
1  import java.awt.geom.Rectangle2D;
2  import javax.swing.ImageIcon;
3
4  public class TARZAN
5  {
6      private int x;
7      private int y;
8      private int width = 60 , height = 64;
9      private int hp = 3;
10     public ImageIcon[] im_left_animation = new ImageIcon[4];
11     public ImageIcon[] im_right_animation = new ImageIcon[4];
12     public ImageIcon img_heart = new ImageIcon(this.getClass().getResource(name:"Image/Heart.png"));
13
14     TARZAN(int x , int y)
15     {
16         this.x = x;
17         this.y = y;
18
19         // add Tarzan left animation to array
20         for(int i=0;i<im_left_animation.length;i++)
21         {
22             im_left_animation[i] = new ImageIcon(this.getClass().getResource("Image/left_"+(i+1)+".PNG"));
23         }
24         // add Tarzan right animation to array
25         for(int i=0;i<im_right_animation.length;i++)
26         {
27             im_right_animation[i] = new ImageIcon(this.getClass().getResource("Image/right_"+(i+1)+".PNG"));
28         }
29     }
30 }
```

มีการสร้างตัวแปรในคลาส TARZAN เพื่อเก็บค่าพิกัดของตัวละคร ณ พิกัด x และ y และกำหนดความกว้างและความสูงของตัวละคร และมีการเก็บรูปภาพของตัวละครเพื่อใช้ในการทำภาพเคลื่อนไหวของตัวละคร โดยแบ่งออกเป็น 2 แบบ คือ ภาพเคลื่อนไหวขณะตัวละครวิ่งไปทางซ้าย และ ภาพเคลื่อนไหวขณะตัวละครวิ่งไปทางขวา

ภายใน Constructor ของ TARZAN มีการรับพารามิเตอร์พิกัด x และ y เพื่อเป็นจุดเกิดของ

TARZAN

```
130
131     //Tarzan
132     // Coordinate (x,y) of Tarzan Spawn at plank floor 3 plank 1
133
134     private int xStart = fl3_pl1.getX();
135     private int yStart = fl3_pl1.getY();
136
137     TARZAN tarzan = new TARZAN(xStart, yStart);
138
```

โดยมีการสร้าง Object ของคลาส Tarzan ไว้ที่ไฟล์ InGame.java โดยจะส่งค่าอาร์กิวเมนต์พิกัด x และ y ของแผ่นไม้ที่ 1 ชั้นที่ 3 เข้าไปเพื่อกำหนดจุดเกิด

จากนั้นใน Constructor ของ Tarzan ทำการวนลูปเพื่อเก็บรูปภาพที่ใช้ในการทำการเคลื่อนไหวของตัวละคร

โดย method สำคัญในคลาส Tarzan จะมีดังนี้

```
73     public void UpdateHp(int hp) {
74         this.hp -= hp;
75     }
```

Method UpdateHp นี้ใช้ในการลด hp ของตัวละครเมื่อโดนโจมตี

```
76     public Rectangle2D getbound(){
77         return (new Rectangle2D.Double(x,y,width,height));
78     }
```

Method getbound นี้ใช้ในการกำหนดขอบของตัวละคร TARZAN, SHARK, PLANK และ

TSUNAMI

```
80     public boolean IsOnPlank(PLANK p) {
81         if(((p.getbound()).intersects(getbound()))
82         {
83             return true;
84         } else {
85             return false;
86         }
87     }
```

Method IsOnPlank นี้ใช้ในการตรวจสอบตัวละครว่าอยู่บนแผ่นไม้หรือไม่โดยการใช้ฟังก์ชัน intersect จากแพ็คเกจ java.awt.geom.Rectangle2D

```

public boolean GetCoin(COIN c)
{
    if((c.getbound()).intersects(getbound()))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

Method GetCoin นี้ใช้ในการตรวจสอบว่าตัวละครนั้นได้เดินมาทับกับเหรียญแล้วหรือไม่โดยใช้ฟังก์ชัน intersect มาใช้ในการตรวจสอบ

```

public boolean GotHitAttack(SHARK sh)
{
    if((sh.getbound()).intersects(getbound()))
    {
        return true;
    }
    else
    {
        return false;
    }
}

public boolean Downing(TSUNAMI tsm)
{
    if((tsm.getbound()).intersects(getbound()))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

ทั้ง 2 Method นี้ใช้เพื่อเช็คได้ว่าตัวละคร TARZAN นั้นโดนโจมตีหรือไม่โดยใช้ฟังก์ชัน intersect มาใช้ในการตรวจสอบ

## ในส่วนของคลาส PLANK

```
3 public class PLANK
4 {
5     public int x;
6     public int y;
7     public int width = 210;
8     public int height = 30;
9
10    PLANK(int x , int y)
11    {
12        this.x = x;
13        this.y = y;
14    }
```

มีการสร้างตัวแปรในคลาส PLANK เพื่อเก็บค่าพิกัดของแผ่นไม้ ณ พิกัด x และ y และกำหนดความกว้างและความสูงของแผ่นไม้

โดยใน Constructor ของ PLANK จะรับค่าพารามิเตอร์พิกัด x และ y เข้ามาเก็บไว้เพื่อไปใช้ในการสร้างกรอบของแผ่นไม้ใน method getbound

```
108 // Coordinate (x,y) of plank
109 // floor 1
110 PLANK fl1_p11 = new PLANK(PLANK_MARGIN_LEFT, PLANK_PADDING + (6 * MARGIN));
111 PLANK fl1_p12 = new PLANK(PLANK_MARGIN_LEFT + (2 * PLANK_PADDING) + MARGIN, PLANK_PADDING + (6 * MARGIN));
112 PLANK fl1_p13 = new PLANK(PLANK_MARGIN_LEFT + (5 * PLANK_PADDING) + MARGIN, PLANK_PADDING + (6 * MARGIN));
113
114 // floor 2
115 PLANK fl2_p11 = new PLANK(PLANK_MARGIN_LEFT + (PLANK_PADDING/2) + MARGIN, PLANK_PADDING + (4 * MARGIN));
116 PLANK fl2_p12 = new PLANK(PLANK_MARGIN_LEFT + TWOPLANK_PADDING + (PLANK_PADDING/2) + (4 * MARGIN), PLANK_PADDING + (4 * MARGIN));
117
118 // floor 3
119 PLANK fl3_p11 = new PLANK(PLANK_MARGIN_LEFT, PLANK_PADDING + (2 * MARGIN));
120 PLANK fl3_p12 = new PLANK(PLANK_MARGIN_LEFT + (2 * PLANK_PADDING) + MARGIN, PLANK_PADDING + (2 * MARGIN));
121 PLANK fl3_p13 = new PLANK(PLANK_MARGIN_LEFT + (5 * PLANK_PADDING) + MARGIN, PLANK_PADDING + (2 * MARGIN));
122
123 // floor 4
124 PLANK fl4_p11 = new PLANK(PLANK_MARGIN_LEFT + (PLANK_PADDING/2) + MARGIN, PLANK_PADDING);
125 PLANK fl4_p12 = new PLANK(PLANK_MARGIN_LEFT + TWOPLANK_PADDING + (PLANK_PADDING/2) + (4 * MARGIN), PLANK_PADDING);
```

โดยมีการสร้าง Object ไว้ในไฟล์ InGame.java โดยจะส่งค่าอาร์กิวเมนต์พิกัด x และ y เข้าไปซึ่งจะมีพิกัดแตกต่างกันไปในแต่ละชั้น โดยสร้างทั้งหมด 4 ชั้น ซึ่งชั้นที่ 1 กับ ชั้นที่ 3 จะมี 3 แผ่นไม้ และชั้นที่ 2 กับ 4 จะมี 2 แผ่นไม้



## ในส่วนของคลาส TSUNAMI

```
4 public class TSUNAMI
5 {
6     private int x;
7     private int y;
8     private int width = 960;
9     private int height = 162;
10    public ImageIcon[] Img_tsunami = new ImageIcon[5];
11
12    TSUNAMI(int x , int y)
13    {
14        this.x = x;
15        this.y = y;
16        for(int i=0;i<Img_tsunami.length;i++)
17        {
18            Img_tsunami[i] = new ImageIcon(this.getClass().getResource("Image/Tsunami_"+(i+1)+".png"));
19        }
20    }
21 }
```

มีการสร้างตัวแปรในคลาส TSUNAMI เพื่อเก็บค่าพิกัดของTSUNAMI ณ พิกัด x และ y และกำหนดความกว้างและความสูงของTSUNAMI และมีการเก็บรูปภาพของTSUNAMI โดยใน Constructor ของ TSUNAMI จะรับค่าพารามิเตอร์พิกัด x และ y เข้ามาเก็บไว้เพื่อไปใช้ในการสร้างกรอบของ TSUNAMI ใน method getbound และทำการเก็บรูปภาพเคลื่อนไหวของคลื่น TSUNAMI

```
84 // -----
85 //Tsunami
86 TSUNAMI tsunami = new TSUNAMI(x:0, SCREEN_HEIGHT - (9* (MARGIN/4)));
87 int TsunamiMotion = 0;
88 // -----
89
```

โดยมีการสร้าง Object ไว้ในไฟล์ InGame.java โดยจะส่งค่าอาร์กิวเมนต์พิกัด x และ y เข้าไป

## ในส่วนของคลาส SHARK

```
4 public class SHARK
5 {
6     public int x;
7     public int y;
8     public int width = 200;
9     public int height = 140;
10    public int Hit = 1;
11    public ImageIcon[] Img_shark = new ImageIcon[3];
12
13    SHARK(int x , int y)
14    {
15        for(int i=0;i<Img_shark.length;i++)
16        {
17            Img_shark[i] = new ImageIcon(this.getClass().getResource("Image/Shark_"+(i+1)+".PNG"));
18        }
19        this.x = x;
20        this.y = y;
21    }
```

มีการสร้างตัวแปรในคลาส SHARK เพื่อเก็บค่าพิกัดของฉลาม ณ พิกัด x และ y และกำหนดความกว้างและความสูงของฉลาม และมีการเก็บรูปภาพของTSUNAMI

โดยใน Constructor ของ SHARK จะรับค่าพารามิเตอร์พิกัด x และ y เข้ามาเก็บไว้เพื่อในไปใช้ในการสร้างกรอบของฉลาม ใน method getbound และทำการเก็บรูปภาพเคลื่อนไหวของฉลาม โดยจะแบ่งออกเป็น 3 ภาพ คือ ภาพที่ฉลามกระโดดขึ้น ภาพที่ฉลามกระโดดอยู่ ณ จุดสูงสุด และภาพที่ฉลามกำลังตกลงทะเล

```
142 // -----
143
144 // Shark
145 int Shark_Max_Jump = 500 , Shark_jump = 10 , Shark_count_jump = 0 , Shark_forward = 60, Shark_count_forward = 0 , Shark_Image = 0 ;
146 boolean shark_jumping = true;
147 SHARK shark = new SHARK(SCREEN_WIDTH - MARGIN- (2 * MARGIN) - (MARGIN / 2),SCREEN_HEIGHT - 100);
148
```

โดยมีการสร้าง Object ไว้ในไฟล์ InGame.java โดยจะส่งค่าอาร์กิวเมนต์พิกัด x และ y เข้าไป

## ในส่วนของคลาส COIN

```
4 public class COIN
5 {
6     public int coin;
7     public int x;
8     public int y;
9     public int width = 70;
10    public int height = 70;
11    public ImageIcon ImgCoin = new ImageIcon(this.getClass().getResource(name:"Image/Coins.png"));
12
13    COIN(int x, int y , int c)
14    {
15        this.x = x;
16        this.y = y;
17        this.coin = c;
18    }
```

มีการสร้างตัวแปรในคลาส COIN เพื่อเก็บค่าพิกัดของเหรียญ ณ พิกัด x และ y และกำหนดความกว้างและความสูงของเหรียญ และมีการเก็บรูปภาพของเหรียญ

โดยใน Constructor ของ COIN จะรับค่าพารามิเตอร์พิกัด x และ y เข้ามาเก็บไว้เพื่อในไปใช้ในการสร้างกรอบของเหรียญ ใน method getbound และทำการเก็บรูปภาพของเหรียญ

```
---
290 // Draw Coin
291 if(Spawn == false)
292 {
293     Random rand = new Random();
294     do
295     {
296         random = rand.nextInt(P planklist.size());
297     }while(random == last_plank_spawn); // compare between new coin spawn and last coin spawn. it check to not spawn at same place
298     x_random = Planklist.get(random).getX() + 60;
299     y_random = Planklist.get(random).getY();
300     Spawn = true;
301     last_plank_spawn = random;
302 }
303 COIN coin = new COIN(x_random, y_random,coint_amount);
304
```

โดยมีการสร้าง Object ไว้ในไฟล์ InGame.java โดยจะส่งค่าอาร์กิวเมนต์พิกัด x และ y และจำนวนเหรียญที่ Tarzan เก็บได้เข้าไป

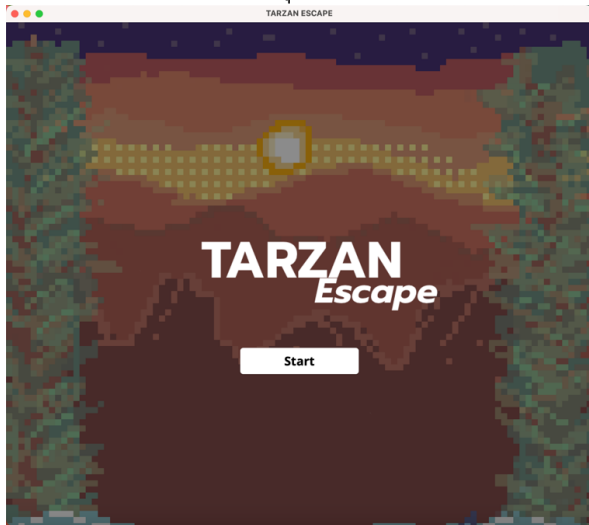
## โครงสร้างของ GUI

ประกอบด้วย

1. หน้าจอ Home
2. หน้าจอขณะเล่นเกม
3. หน้าจอเวลาเกมจบ

### หน้าจอ Home

ประกอบด้วยปุ่มเริ่มเกม และพื้นหลังของเกม



โดยในหน้าจอ Home จะมี Event handling ในส่วนของปุ่ม Start

```
21     home.startButton.addActionListener(new ActionListener() {
22         @Override
23         public void actionPerformed(ActionEvent e)
24         {
25             if(e.getSource() == home.startButton)
26             {
27                 Game.this.add(ingame);
28                 ingame.setFocusable(focusable:true);
29                 ingame.requestFocus();
30                 Game.this.revalidate();
31             }
32         }
33     });
```

คือ เมื่อกดปุ่ม Start จะทำการเข้าสู่หน้าจอ Ingame เพื่อเล่นเกมทันที

## หน้าจอขณะเล่นเกม

ประกอบด้วย ตัวละคร Tarzan , แผ่นไม้ทั้ง 4 ชั้น , ฉลาม , คลื่น Tsunami , เหยี่ยว , พื้นหลังเกม , หน้าจอการแสดงผลหัวใจและเหรียญที่ Tarzan เก็บได้



โดยในหน้าจอขณะเล่นเกม จะมี Event handling ในส่วนของการเคลื่อนไหวของ Tarzan โดยจะรับการเคลื่อนไหวจากแป้นพิมพ์

```
167 addKeyListener(new KeyAdapter() {
168     @Override
169     public void keyPressed(KeyEvent e)
170     {
171         switch (e.getKeyCode()) {
172             // when Tarzan Jump
173             case KeyEvent.VK_UP:
174                 if (tarzan.getY() >= MARGIN/2)
175                 {
176                     // jump
177                     if(jumbable == true)
178                     {
179                         for(int i = 0 ; i < 240 ; i++)
180                         {
181                             tarzan.UpdateY(-1);
182                             repaint();
183                         }
184                         repaint();
185                     }
186                 }
187                 break;
188             // when Tarzan going left side
189             case KeyEvent.VK_LEFT:
190                 if (tarzan.getX() >= 0 + MARGIN)
191                 {
192                     tarzan.UpdateX(-20);
193                     // x -= 30;
194                 }
195                 right = false; left = true;
196                 break;
197             // when Tarzan going right side
198             case KeyEvent.VK_RIGHT:
199                 if (tarzan.getX() + MARGIN < SCREEN_WIDTH)
200                 {
201                     tarzan.UpdateX(x:20);
202                     // x += 30;
203                 }
204                 right = true; left = false;
205                 break;
206             case KeyEvent.VK_DOWN:
207                 if(tarzan.getY() < SCREEN_HEIGHT - (5 * MARGI
208                 {
209                     tarzan.UpdateY(y:100);
210                 }
211             default:
212                 break;
213         }
214         repaint();
215     }
216 }
```

โดยเมื่อกดลูกศรขึ้น (VK\_UP) Tarzan จะทำการกระโดด  
และเมื่อกดลูกศรลง (VK\_DOWN) Tarzan จะทำการลงจากแผ่นไม้  
และเมื่อกดลูกศรซ้าย (VK\_LEFT) Tarzan จะทำการเคลื่อนที่ไปทางซ้าย  
และเมื่อกดลูกศรขวา (VK\_RIGHT) Tarzan จะทำการเคลื่อนที่ไปทางขวา

## หน้าจอขณะจบเกม

โดยแบ่งออกเป็น 2 หน้าคือ หน้าจบเกมแบบชนะ และ หน้าจบเกมแบบแพ้ ซึ่งทั้ง 2 แบบ ประกอบด้วย ปุ่ม Home และ พื้นหลัง

### หน้าจอแบบชนะ



### หน้าจอแบบแพ้



โดยในหน้าจอขณะจบเกม จะมี Event handling ในส่วนของปุ่ม Home

```
34     ingame.homeButton.addActionListener(new ActionListener() {
35         @Override
36         public void actionPerformed(ActionEvent e)
37         {
38             if(e.getSource() == ingame.homeButton)
39             {
40                 ingame.RestartGame();
41                 Game.this.remove(ingame);
42                 Game.this.add(home);
43                 home.setFocusable(focusable:true);
44                 home.requestFocus();
45                 Game.this.revalidate();
46                 Game.this.repaint();
47             }
48         }
49     });
50 }
```

คือ เมื่อกดปุ่ม Home จะทำการเข้าสู่หน้า Home

## อัลกอริทึมที่สำคัญในโปรแกรม

### อัลกอริทึมการสุ่มเกิดของเหรียญ

```
290 // Draw Coin
291 if(Spawn == false)
292 {
293     Random rand = new Random();
294     do
295     {
296         random = rand.nextInt(Planklist.size());
297     }while(random == last_plank_spawn); // compare between new coin spawn and last coin spawn. it check to not spawn at same place
298     x_random = Planklist.get(random).getX() + 60;
299     y_random = Planklist.get(random).getY();
300     Spawn = true;
301     last_plank_spawn = random;
302 }
303 COIN coin = new COIN(x_random, y_random, coin_amount);
304
305 if(Spawn == true)
306 {
307     coin.y = Planklist.get(random).getY();
308     g.drawImage(coin.ImgCoin.getImage(), x_random, coin.y - 60, coin.width, coin.height, this);
309 }
310
311 if(tarzan.GetCoin(coin) == true)
312 {
313     coin_amount++;
314     Spawn = false;
315 }
316 //
```

โดยที่เริ่มต้นเกมจะตั้ง สถานะการเกิดของเหรียญเป็น False และเมื่อเช็คเงื่อนไขก็จะทำการวนดูป  
สุ่มการเกิดของเหรียญโดยที่จะไม่ให้ซ้ำกับจุดเกิดในรอบที่แล้ว เพื่อป้องกันการเกิดซ้ำทำให้เกมง่าย  
เกินไป โดยเริ่มต้นตัวแปร last\_plank\_spawn จะเก็บจุดเกิดของ Tarzan ไว้ เพื่อเวลาเริ่มเกมมา  
เหรียญจะได้ไม่เกิด ณ แผ่นไม้ที่ Tarzan เกิด เมื่อเหรียญเกิดแล้วก็จะตั้งสถานะเกิดของเหรียญ  
เป็น True เพื่อไม่ให้ซ้ำเงื่อนไขสถานะเกิดของเหรียญเป็น False เพื่อให้เหรียญยังคงอยู่ ณ  
ตำแหน่งเดิมจนกว่า Tarzan จะเก็บเหรียญ ซึ่งเหรียญจะกลายเป็น False อีกครั้งเมื่อ Tarzan เก็บ  
เหรียญ

## อัลกอริทึมการกระโดดของฉลาม

```
352 // -----
353 // Draw Shark
354 // Update Shark Jump
355 if (Shark_count_jump < Shark_Max_Jump && shark_jumping == true) // jump
356 {
357     shark.UpdateY(Shark_jump);
358     Shark_count_jump += Shark_jump;
359 }
360 else if (Shark_count_jump == Shark_Max_Jump && shark_jumping == true) // when shark at highest jump
361 {
362     Shark_Image = 1;
363     shark.UpdateX(Shark_forward);
364     shark_jumping = false;
365     Shark_count_jump += Shark_jump;
366     Shark_count_forward += Shark_forward;
367 }
368 else if (shark_jumping == false && Shark_count_jump > 0) // fall
369 {
370     Shark_count_jump -= Shark_jump;
371     Shark_Image = 2;
372     shark.UpdateY(-Shark_jump);
373 }
374 else if (Shark_count_jump == 0)
375 {
376     Shark_Image = 0;
377     shark_jumping = true;
378     if (shark.getHit() == 0)
379     {
380         shark.setHit(hit:1);
381     }
382 }
383 if (Shark_count_forward == 600 && Shark_count_jump == 0)
384 {
385     shark.setX(SCREEN_WIDTH - (2*MARGIN));
386     shark.setY(SCREEN_HEIGHT - 100);
387     Shark_count_forward = 0;
388 }
389
390 g.drawImage(shark.Img_shark[Shark_Image].getImage(), shark.getX(), shark.getY(), shark.getWidth(), shark.getHeight(), this);
391 // outline of shark
392 // g.drawRect(shark.getX(), shark.getY(), shark.width, shark.height);
393 // -----
```

โดยจะสร้างตัวแปรในการเริ่มต้นเพื่อใช้ในการทำการเคลื่อนไหวของฉลาม

```
144 // Shark
145 int Shark_Max_Jump = 500 , Shark_jump = 10 , Shark_count_jump = 0 , Shark_forward = 60 , Shark_count_forward = 0 , Shark_Image = 0 ;
146 boolean shark_jumping = true;
```

โดยเริ่มต้นจะเข้าเงื่อนไขว่าฉลามกำลังกระโดดนั้นเป็นจริง จากนั้นทำการเปลี่ยนพิกัด y ของฉลามทีละ 10 เพื่อทำ Animation การเคลื่อนไหวของฉลาม และนับระยะทางทาง y ที่ฉลามกระโดดไปด้วย จากนั้นเมื่อนับระยะทางทาง y ที่ฉลามกระโดดจนครบกำหนดคือ 500 หมายถึงตอนนี้ฉลามกระโดดอยู่ ณ จุดสูงสุดของการกระโดด ซึ่งจะทำ Animation การเคลื่อนไหวของการที่ฉลามโน้มตัวมาด้านหน้า โดยการเปลี่ยนภาพใน Array ของฉลามเป็นภาพที่ 2 และทำการเปลี่ยนพิกัด x ของฉลามไป 60 และนับระยะทางทาง x ไป 60 และเปลี่ยนสถานการณ์กระโดดเป็นเท็จเพื่อให้เข้าเงื่อนไขการทำ Animation การตกของฉลาม โดยจะทำการเปลี่ยนพิกัด y ของฉลามให้ลดลงทีละ 10 และลดการนับระยะทางทาง y ลงทีละ 10 และเมื่อระยะทางของฉลามที่นับเหลือ 0 จะเปลี่ยนสถานะการกระโดดของฉลามเป็นจริงอีกครั้งเพื่อกระโดดใหม่ และเมื่อการนับระยะทางทาง x ครบ 600 จะทำการย้ายพิกัดของฉลามไปที่จุดเกิดใหม่อีกครั้งเพื่อทำการกระโดดใหม่



## อัลกอริทึมการตกตามแรงโน้มถ่วงและการกระโดดของตัวละคร Tarzan

```
317 // Draw Tarzan
318 int down;
319 if(tarzan.IsOnPlank(fl1_pl1) == true ||
320 tarzan.IsOnPlank(fl1_pl2) == true ||
321 tarzan.IsOnPlank(fl1_pl3) == true ||
322 tarzan.IsOnPlank(fl2_pl1) == true ||
323 tarzan.IsOnPlank(fl2_pl2) == true ||
324 tarzan.IsOnPlank(fl3_pl1) == true ||
325 tarzan.IsOnPlank(fl3_pl2) == true ||
326 tarzan.IsOnPlank(fl3_pl3) == true ||
327 tarzan.IsOnPlank(fl4_pl1) == true ||
328 tarzan.IsOnPlank(fl4_pl2) == true)
329 {
330     jumbable = true;
331     down = 0;
332 }
333 else
334 {
335     jumbable = false;
336     down = 20;
337     tarzan.UpdateY(down);
338 }
339
340 if(right == true)
341 {
342     g.drawImage(tarzan.im_right_animation[TarzanMotion].getImage(), tarzan.getX(), tarzan.getY() + down ,tarzan.getWidth(),tarzan.getHeight(),this);
343 }
344 else
345 {
346     g.drawImage(tarzan.im_left_animation[TarzanMotion].getImage(),tarzan.getX(), tarzan.getY() + down ,tarzan.getWidth(),tarzan.getHeight(),this);
347 }
348 TarzanMotion++;
349 TarzanMotion %= 4;
350 // outline of tarzan
351 // g.drawRect(tarzan.getX(), tarzan.getY() + down ,tarzan.getWidth(),tarzan.getHeight());
...
```

โดยเริ่มต้นจะตรวจสอบว่า Tarzan อยู่บนแผ่นไม้หรือไม่โดยการตรวจสอบจาก method IsOnPlank ในคลาส Tarzan ถ้า Tarzan อยู่บนแผ่นไม้ก็จะไม่ตกลงไปตามแรงโน้มถ่วง และสามารถกระโดดได้ แต่ถ้า Tarzan ไม่ได้อยู่บนแผ่นไม้ก็จะตกตามแรงโน้มถ่วงและไม่สามารถกระโดดได้

## บทที่ 3 สรุป

### ปัญหาที่พบเจอระหว่างการพัฒนา

1. การทำ Animation ของตัวละคร Tarzan และฉลาม และแผ่นไม้ ให้มีการเคลื่อนไหวที่ลื่นไหล เป็นไปตามกฎฟิสิกส์
2. การทำฟังก์ชันการสุ่มจุดเกิดของเหรียญที่ต้องทำให้เหรียญไม่เกิดจุดเกิดที่เดิม เนื่องจากการเกิดของเหรียญจะเกิดบนแผ่นไม้ ซึ่งแผ่นไม้มี 4 ชั้นรวมกันทั้งสิ้น 10 แผ่น ทำให้โอกาสการเกิดของเหรียญที่จะเกิดจุดเดิมมีโอกาส 1 ใน 10 ซึ่งในระหว่างที่ทำการทดสอบเกมนั้น เหรียญเกิดจุดเดิมซ้ำ 1 ถึง 2 รอบ ในเกมๆเดียว ซึ่งเป็นโอกาสการเกิดที่มาก ทำให้เกมนั้นง่ายเกินไป

### จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

1. มีการทำการเคลื่อนไหวทางฟิสิกส์เข้ามาเกี่ยวข้องในการทำเกมเช่น การกระโดด และการตกตามแรงโน้มถ่วงของตัวละคร
2. การสุ่มเกิดของเหรียญที่ทำให้ผู้เล่นไม่สามารถรู้ได้เลยว่าเหรียญจะเกิด ณ จุดไหนของแผ่นไม้
3. การที่ทำให้ผู้เล่นต้องมีสมาธิในการเล่นระหว่างเก็บเหรียญและไม่ตกลงไปในทะเลหรือโดนฉลามกัดขณะเก็บเหรียญ