



Game Jujutsu kaisen ver. Gojo vs Sukuna

โดย

นายเอกภพ เต็มกระโทก รหัสนักศึกษา 6501157 SEC 03

โครงการนี้เป็นส่วนหนึ่งของ

วิชา CPE 263 การโปรแกรมเชิงวัตถุ

(Objected Oriented Programming)

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2566

บทนำ

การพัฒนาเกมดิจิทัลในยุคปัจจุบันนี้ เป็นอีกหนึ่งด้านที่แสดงถึงความก้าวหน้าของเทคโนโลยีและศิลปะของยุคปัจจุบัน ในโครงการนี้ทางผู้จัดทำขอนำเสนอกระบวนการและแนวทางในการสร้างเกมด้วยภาษา C# และ .NET Framework ผ่านแพลตฟอร์ม Visual Studio โดยเกมที่จะถูกพัฒนานี้มีชื่อว่า "Jujutsu Kaisen ver. Gojo vs Sukuna" ซึ่งได้รับแรงบันดาลใจจากซีรีส์อนิเมะที่มีชื่อเสียงในญี่ปุ่น ณ ขณะนี้

ในการออกแบบและพัฒนาเกมนี้เป็นการผสมผสานความรู้ทางด้านโปรแกรมมิ่ง, กราฟิก, และการออกแบบจากประสบการณ์ เพื่อสร้างเกมที่มีคุณสมบัติตรงตามที่ต้องการของผู้เล่นและผู้สร้าง ซึ่งในโครงการนี้ จะแสดงถึงวิธีการใช้คำสั่งและเครื่องมือต่างๆ ใน Visual Studio เพื่อสร้างและทดสอบเกม รวมทั้งการนำเสนอการออกแบบตัวละครและสภาพแวดล้อมภายในเกม ซึ่งจะช่วยให้ผู้เล่นได้สัมผัสกับประสบการณ์จากการเล่นเกมที่สร้างด้วย C# และ .NET Framework รวมถึงเข้าใจหลักการทำงานของระบบและมีอารมณ์ร่วมไปกับเกมที่ได้เล่นมากยิ่งขึ้นด้วย

โครงการนี้จึงเป็นการผสมรวมความรู้และทักษะจากหลากหลายสาขาวิชาเข้าด้วยกัน ไม่ว่าจะเป็นการเขียนโปรแกรม, การออกแบบกราฟิก, การจัดการระบบเกม, และการปรับปรุงประสิทธิภาพเกม ทั้งนี้ ยังมีการพิจารณาถึงปัจจัยด้านผู้ใช้และประสบการณ์การเล่นเกมเพื่อให้เกม "Jujutsu Kaisen ver. Gojo vs Sukuna" เป็นเกมที่น่าตื่นเต้น ทำทาย และเป็นสิ่งที่พึงพอใจของผู้เล่นในทุกะระดับของประสบการณ์

ผู้จัดทำได้ตระหนักถึงความสำคัญของการทดสอบและการปรับปรุงอย่างต่อเนื่องเพื่อให้เกมมีคุณภาพที่สมบูรณ์ที่สุด ดังนั้น การทดสอบเกมจะเกิดขึ้นอย่างสม่ำเสมอทั้งในระหว่างและหลังกระบวนการพัฒนา เพื่อให้แน่ใจว่าทุกส่วนของเกม ไม่ว่าจะเป็นระบบการเล่น, กราฟิก, และอินเทอร์เฟซผู้ใช้ ล้วนมีประสิทธิภาพที่สมบูรณ์และตรงตามที่ต้องการได้เป็นอย่างดี

นอกจากนี้ โครงการนี้ยังมีเป้าหมายที่จะให้ความรู้และประสบการณ์แก่ผู้จัดทำคนอื่นๆ โดยผ่านกระบวนการเรียนรู้ที่มีความท้าทายและความคิดสร้างสรรค์ ทั้งนี้ ทางผู้จัดทำหวังว่าจะสามารถแบ่งปันความรู้และประสบการณ์เหล่านี้กับชุมชนนักพัฒนาเกม เพื่อเสริมสร้างและพัฒนาอุตสาหกรรมเกมให้เติบโตต่อไป

ขอบเขตของโครงการ

1. สามารถจัดการกับการโหลดรูปภาพตัวละคร พื้นหลัง และทักษะต่างๆของตัวละครได้
2. สามารถจัดการตำแหน่งและขนาดของตัวละครและทักษะต่างๆของตัวละครได้
3. สามารถติดตามและเพิ่มอัตราโน้ต HP (Health Point) ของตัวละคร พร้อมกับ Progress bars เพื่อแสดงค่า HP ของตัวละครทั้ง 2 ตัว
4. สามารถจัดการกับเหตุการณ์ต่างๆได้ภายในเกม เพื่อจัดการทักษะของตัวละคร คูลดาวน์ และการเคลื่อนไหวของตัวละคร
5. สามารถจัดการตรวจจับและจัดการการโจมตีของทักษะตัวละคร

เครื่องมือที่ใช้หรือคำสั่งที่ใช้งานแทนเครื่องมือ

1. กราฟิกและภาพ: จัดการการสร้างและจัดการภาพสำหรับตัวละครและพื้นหลัง โดยใช้คำสั่ง Image
คำสั่ง 'Image' ใน C# ใช้สำหรับจัดการกับภาพในโปรแกรม. ช่วยโหลด, แสดง, และปรับแต่งภาพ, รวมถึงใช้ร่วมกับคลาสอื่นๆ สำหรับการจัดการภาพ.

```
// ประกาศตัวแปรสำหรับรูปภาพของตัวละคร ภาพพื้นหลัง และภาพสกิล
//-----

Image gojo;
Image sukuna;
Image bg;
Image kai;
Image shotmurasaki;
Image shotao;
Image shotarka;
```

2. การจัดตำแหน่งและขนาด: ตั้งค่าตำแหน่งเริ่มต้นและขนาดสำหรับองค์ประกอบกราฟิกด้วยคำสั่ง

e.Graphics.DrawImage

e.Graphics.DrawImage ใน C# เป็นคำสั่งที่ใช้ในการวาดภาพลงบนพื้นผิวกราฟิก, คำสั่งนี้ช่วยให้สามารถแสดงภาพในตำแหน่งที่กำหนดบนแบบฟอร์มหรือคอนโทรลอื่นๆ, และยังให้ความยืดหยุ่นในการกำหนดขนาดหรือสเกลของภาพ

```
e.Graphics.DrawImage(bg, new Rectangle(bgPosition, 0, this.ClientSize.Width, this.ClientSize.Height));
e.Graphics.DrawImage(gojo, new Rectangle(gojoX, gojoY, gojoWidth, gojoHeight));
e.Graphics.DrawImage(sukuna, new Rectangle(sukunaX, sukunaY, sukunaWidth, sukunaHeight));
// -----
```

3. การเคลื่อนไหวของตัวละคร: จัดการการเคลื่อนไหวของตัวละครได้โดยใช้คำสั่งเปลี่ยนรูปภาพ

Image.FromFile

Image.FromFile ใน C# ใช้สำหรับโหลดภาพจากไฟล์. คำสั่งนี้ทำการอ่านไฟล์ภาพจากตำแหน่งที่กำหนดและสร้างออบเจกต์ Image

```
if (Ao)
{
    gojo = Image.FromFile("gojo\\2.png");
    gojoWidth = 550;
    gojoHeight = 800;
}
else if (Arka)
{
    gojo = Image.FromFile("gojo\\3.png");
    gojoWidth = 550;
    gojoHeight = 800;
}
else if (Mugen)
{
    gojo = Image.FromFile("gojo\\4.png");
    gojoWidth = 550;
    gojoHeight = 800;
}
else if (Murasaki)
{
    gojo = Image.FromFile("gojo\\5.png");
    gojoWidth = 550;
    gojoHeight = 800;
}
else
{
    gojo = Image.FromFile("gojo\\1.png");
    gojoWidth = 500;
    gojoHeight = 800;
}

// ถ้า gojoReverse และ isKeyEHeld เป็น true, จะเรียกใช้ GojoReverse
if (gojoReverse && isKeyEHeld)
{
    GojoReverse(50);
}

// อัปเดตหน้าจอเพื่อแสดงการเปลี่ยนแปลง
Invalidate();
```

4. การจัดการ HP ของตัวละคร: ใช้ progress bars ในการแสดงสถานะ HP ของผู้เล่น และของ BOT

C#, progress bars มักใช้เพื่อแสดงความคืบหน้าหรือสถานะของกระบวนการบางอย่างใน GUI.

ตัวอย่างเช่น, ในเกม, ใช้แสดง HP หรือพลังงานของตัวละครภายในเกม

```
// -----  
gojoHP = new ProgressBar();  
gojoName = new Label();  
gojoFont = new Font("Arial", 20, FontStyle.Regular);  
gojoHP.Minimum = 0;  
gojoHP.Maximum = 100;  
gojoHP.Value = 100;  
gojoName.Text = "Gojo Satoru";  
gojoName.Font = gojoFont;  
  
//-----  
  
// สร้างและตั้งค่า Timer สำหรับลด GojoHP ทุก 2 วินาที  
//-----  
  
System.Windows.Forms.Timer gojoHPDecrementTimer = new System.Windows.Forms.Timer();  
gojoHPDecrementTimer.Interval = 2000; // ตั้งเวลาเ็นคอบยหลังเป็น 2000 มิลลิวินาที (2 วินาที)  
gojoHPDecrementTimer.Tick += GojoHPDecrementTimer_Tick; // เชื่อมต่อกับเหตุการณ์ Tick กับฟังก์ชัน GojoHPDecrementTimer_Tick  
gojoHPDecrementTimer.Start(); // เริ่มการทำงานของ Timer  
  
//-----  
  
// สร้างและตั้งค่า Timer สำหรับเพิ่ม GojoHP ทุก 2 วินาที  
//-----  
  
System.Windows.Forms.Timer autoIncreaseGojoTimer;  
autoIncreaseGojoTimer = new System.Windows.Forms.Timer();  
autoIncreaseGojoTimer.Interval = 1000;  
autoIncreaseGojoTimer.Tick += AutoIncreaseTimer_Tick;  
autoIncreaseGojoTimer.Start();  
  
//-----
```

5. การควบคุมด้วย Timer: ใช้ตัวจับเวลาสำหรับการจัดการการเคลื่อนไหวของตัวละครและเวลาในการใช้ทักษะของตัวละครผู้เล่น

C#, Timer มักใช้เพื่อสร้างฟังก์ชันที่ทำงานตามเวลาที่กำหนด เป็นเครื่องมือที่มีประโยชน์ในการจัดการกับเหตุการณ์ที่ต้องเกิดขึ้นเป็นประจำหรือหลังจากช่วงเวลาหนึ่งในเกมหรือแอปพลิเคชัน Timer ใช้เพื่ออัปเดตตำแหน่งของตัวละครจัดการกับเหตุการณ์ภายในเกมหรือเป็นนาฬิกานับถอยหลัง

```
1 reference
private void AoShotMove()
{
    aoMoveTimer = new System.Windows.Forms.Timer();
    aoMoveTimer.Interval = 50; // ตั้งเวลานับถอยหลังเป็น 50 มิลลิวินาที
    aoMoveTimer.Tick += AoMoveTimer_Tick;
}

1 reference
private void ArkaShotMove()
{
    arkaMoveTimer = new System.Windows.Forms.Timer();
    arkaMoveTimer.Interval = 50; // ตั้งเวลานับถอยหลังเป็น 50 มิลลิวินาที
    arkaMoveTimer.Tick += ArkaMoveTimer_Tick;
}

1 reference
private void MurasakiShotMove()
{
    murasakiMoveTimer = new System.Windows.Forms.Timer();
    murasakiMoveTimer.Interval = 50; // ตั้งเวลานับถอยหลังเป็น 50 มิลลิวินาที
    murasakiMoveTimer.Tick += MurasakiMoveTimer_Tick;
}
```

6. การจัดการอินพุตจากคีย์บอร์ด: โค้ดจับการกดปุ่มคีย์บอร์ดเพื่อเรียกใช้ทักษะและการเคลื่อนไหวโดยใช้คำสั่ง e.KeyCode ของ Event Keydown

e.KeyCode ในเหตุการณ์ KeyDown ของ Windows Forms ใน C# ใช้เพื่อตรวจสอบปุ่มที่กดบนคีย์บอร์ด ซึ่งเป็นส่วนหนึ่งของอาร์กิวเมนต์ e ในเหตุการณ์ KeyDown โดยที่โปรแกรมรู้ว่าปุ่มใดถูกกดเป็นเครื่องมือที่สำคัญสำหรับการสร้างปฏิสัมพันธ์ผู้ใช้ในแอปพลิเคชัน, เช่น การใช้งานปุ่มบนคีย์บอร์ดเพื่อควบคุมตัวละครในเกมหรือการเรียกใช้ฟังก์ชันต่างๆ

```
1 reference
private void KeyIsDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Q)
    {
        if (skillQCooldownTimer != null && !skillQCooldownTimer.Enabled)
        {
            Ao = true;
            ShootAo();
            gojoSkills();
            skillQCooldownTimer.Start();
        }
    }
    else if (e.KeyCode == Keys.W)
    {
        if (skillWCooldownTimer != null && !skillWCooldownTimer.Enabled)
        {
            Arka = true;
            gojoSkills();
            ShootArka();
            skillWCooldownTimer.Start();
        }
    }
    else if (e.KeyCode == Keys.E)
    {
        if (skillECooldownTimer != null && !skillECooldownTimer.Enabled)
        {
            Mugen = true;
            gojoReverse = true;
            isKeyEHeld = true;
            gojoSkills();
            skillECooldownTimer.Start();
        }
    }
    else if (e.KeyCode == Keys.R)
    {
        if (skillRCooldownTimer != null && !skillRCooldownTimer.Enabled)
        {
            Murasaki = true;
            gojoSkills();
            ShootMurasaki();
            skillRCooldownTimer.Start();
        }
    }
}
```

7. การจัดการกับระบบการโจมตีของสกล: โค้ดตรวจจับการโจมตีของสกลของตัวละครโดยใช้คำสั่ง Boolean

C#, Boolean หรือ bool เป็นชนิดข้อมูลที่ใช้แทนค่าความจริง ซึ่งรวมถึง true (จริง) และ false (เท็จ) ใช้ในการตัดสินใจเงื่อนไข เช่น ในการควบคุมโครงสร้างการตัดสินใจ (if, while, ฯลฯ) หรือในการตรวจสอบสถานะของเงื่อนไขหรือตัวแปรในเกมหรือแอปพลิเคชัน ตัวแปรประเภท Boolean ใช้ในการติดตามสถานะ เช่น ว่าผู้เล่นกำลังเคลื่อนไหวหรือไม่ หรือหากภารกิจหนึ่งสำเร็จแล้วหรือยัง

```
//-----  
3 references  
private bool DetectCollision(int object1X, int object1Y, int object1Width, int object1Height, int object2X, int object2Y, int object2Width, int object2Height)  
{  
    // ตรวจสอบว่ามีการชนระหว่าง object1 และ object2 หรือไม่  
    if (object1X + object1Width <= object2X || object1X >= object2X + object2Width || object1Y + object1Height <= object2Y || object1Y >= object2Y + object2Height)  
    {  
        return false;  
    }  
    else  
    {  
        return true;  
    }  
}  
//-----
```



```

1 using System;
2 using System.Drawing;
3 using System.Windows.Forms;
4 using System.Diagnostics;
5 using System.Numerics;
6
7 namespace Gojo_Fight
8 {
9     public partial class Form1 : Form
10     {
11         // ประกาศตัวแปรสำหรับรูปภาพของตัวละคร ภาพพื้นหลัง และภาพสกิล
12         //-----
13
14         Image gojo;
15         Image sukuna;
16         Image bg;
17         Image kai;
18         Image shotmurasaki;
19         Image shotao;
20         Image shotarka;
21
22         //-----
23
24         // กำหนดตำแหน่งเริ่มต้นของตัวละคร Gojo Sukuna skill บนแกน X และ Y ขนาดตัวละคร และ ขนาดของ skill
25         //-----
26
27         int gojoX = 0;
28         int gojoY = 250;
29
30         int shotmurasakiX;
31         int shotmurasakiY;
32
33         int shotmurasakiWidth;
34         int shotmurasakiHeight;
35
36
37         int shotaoX;
38         int shotaoY;
39
40         int shotaoWidth;
41         int shotaoHeight;
42
43         int shotarkaX;
44         int shotarkaY;
45
46         int shotarkaWidth;
47         int shotarkaHeight;
48
49         int kaiWidth = 100;
50         int kaiHeight = 100;
51
52
53         int sukunaX = 1450;
54         int sukunaY = 400;
55
56         int gojoWidth = 500;
57         int gojoHeight = 800;
58
59         int gojoCurrentHp = 100;
60         int sukunaCurrentHp = 1000;
61
62         int sukunaWidth = 480;
63         int sukunaHeight = 600;
64
65         int bgPosition = 0;
66
67         //-----
68
69         //ประกาศตัวแปรสำหรับควบคุมการเคลื่อนที่
70         //-----
71
72         bool Ao = false;
73         bool Arka = false;
74         bool Mugen = false;
75         bool Murasaki = false;
76         bool gojoReverse = false;
77
78         bool isKeyEHeld = false;
79         bool isMurasakiMoving = false;
80         bool isAoMoving = false;
81         bool isArkaMoving = false;
82
83         bool isShotaoVisible = true;
84         bool isShotarkaVisible = true;
85
86         bool showImage = true;
87
88         //-----
89
90         //สร้างและตั้งค่า ProgressBar Label Font ของตัวละคร Gojo และ Sukuna
91         //-----
92
93         ProgressBar gojoHP;
94         ProgressBar sukunaHP;
95
96         Label gojoName;
97         Label sukunaName;
98
99         Font gojoFont;
100        Font sukunaFont;
101
102        //-----
103
104        //ประกาศ Timer สำหรับควบคุมการเคลื่อนที่
105        //-----
106
107        System.Windows.Forms.Timer murasakiMoveTimer;
108        System.Windows.Forms.Timer aoMoveTimer;
109        System.Windows.Forms.Timer arkaMoveTimer;
110        System.Windows.Forms.Timer skillQCooldownTimer;
111        System.Windows.Forms.Timer skillWCooldownTimer;
112        System.Windows.Forms.Timer skillECooldownTimer;
113        System.Windows.Forms.Timer skillRCooldownTimer;
114        System.Windows.Forms.Timer blinkTimer;
115        //System.Windows.Forms.Timer shotaoVisibilityTimer;
116        //System.Windows.Forms.Timer shotarkaVisibilityTimer;
117
118        //-----
119
120        public Form1()
121        {
122            //เรียก Methods ต่างๆ
123            //-----
124
125            InitializeComponent();
126            SetUpForm();
127            HPbar();

```

```

128     MurasakiShotMove();
129     AoShotMove();
130     ArkaShotMove();
131     Skillcooldown();
132     Blink();
133     //SetUpTimers();
134
135     //-----
136
137     //เช็กรูปภาพและกำหนดความกว้างและยาวสำหรับรูปภาพตัวแปร kai
138     //-----
139
140     kai = Image.FromFile("sukuna\\2.png");
141     kaiWidth = 400;
142     kaiHeight = 400;
143
144     //-----
145
146 }
147
148
149 private void FormPaintEvent(object sender, PaintEventArgs e)
150 {
151     //สร้างภาพพื้นหลังและองศาตัวละคร
152     //-----
153
154     e.Graphics.DrawImage(bg, new Rectangle(bgPosition, 0, this.ClientSize.Width, this.ClientSize.Height));
155     e.Graphics.DrawImage(gojo, new Rectangle(gojoX, gojoY, gojoWidth, gojoHeight));
156     e.Graphics.DrawImage(sukuna, new Rectangle(sukunaX, sukunaY, sukunaWidth, sukunaHeight));
157
158     //-----
159
160     //เช็กลำดับค่าสร้างข้อความ
161     //-----
162
163     using (Font font = new Font("Arial", 50, FontStyle.Bold))
164     {
165
166         using (SolidBrush redBrush = new SolidBrush(Color.Red))
167         {
168             // ตรวจสอบ HP Gojo ถ้า gojoCurrentHp <= 0 ให้แสดงข้อความว่า GAMEOVER โดยเรียกใช้คำสั่งการสร้างข้อความ
169             //-----
170             if (gojoCurrentHp <= 0)
171             {
172                 string gameOverText = "GAME OVER";
173                 SizeF textSize = e.Graphics.MeasureString(gameOverText, font);
174                 PointF textPosition = new PointF((this.ClientSize.Width - textSize.Width) / 2, (this.ClientSize.Height - textSize.Height) / 2);
175                 e.Graphics.DrawString(gameOverText, font, redBrush, textPosition);
176             }
177         }
178
179         using (SolidBrush goldBrush = new SolidBrush(Color.Gold))
180         {
181             // ตรวจสอบ HP Sukuna ถ้า SukunaCurrentHp <= 0 ให้แสดงข้อความว่า WIN โดยเรียกใช้คำสั่งการสร้างข้อความ
182             //-----
183             if (sukunaCurrentHp == 0)
184             {
185                 string winText = "WIN";
186                 SizeF textSize = e.Graphics.MeasureString(winText, font);
187                 PointF textPosition = new PointF((this.ClientSize.Width - textSize.Width) / 2, (this.ClientSize.Height - textSize.Height) / 2);
188                 e.Graphics.DrawString(winText, font, goldBrush, textPosition);
189             }
190         }
191     }
192
193     //-----
194
195     //สร้างภาพขึ้นมาหากกดปุ่มแปรในวงเล็บของ if นั้นเป็นจริง
196     //-----
197
198     if (showImage)
199     {
200         int kaiPositionX = gojoX + (gojoWidth / 2) - (kaiWidth / 2);
201         int kaiPositionY = gojoY + 120;
202         e.Graphics.DrawImage(kai, new Rectangle(kaiPositionX, kaiPositionY, kaiWidth, kaiHeight));
203     }
204
205     if (isMurasakiMoving)
206     {
207         e.Graphics.DrawImage(shotmurasaki, new Rectangle(shotmurasakiX, shotmurasakiY, shotmurasakiWidth, shotmurasakiHeight));
208     }
209
210     if (isAoMoving && isShotaoVisible)
211     {
212         e.Graphics.DrawImage(shotao, new Rectangle(shotaoX, shotaoY, shotaoWidth, shotaoHeight));
213     }
214
215     if (isArkaMoving && isShotarkaVisible)
216     {
217         e.Graphics.DrawImage(shotarka, new Rectangle(shotarkaX, shotarkaY, shotarkaWidth, shotarkaHeight));
218     }
219
220     //-----
221 }
222
223
224
225 /*private void SetUpTimers()
226 {
227     shotaoVisibilityTimer = new System.Windows.Forms.Timer();
228     shotaoVisibilityTimer.Interval = 3000; // Example: 3 seconds
229     shotaoVisibilityTimer.Tick += ShotaoVisibilityTimer_Tick;
230
231     shotarkaVisibilityTimer = new System.Windows.Forms.Timer();
232     shotarkaVisibilityTimer.Interval = 3000; // Example: 3 seconds
233     shotarkaVisibilityTimer.Tick += ShotarkaVisibilityTimer_Tick;
234 }
235
236 private void ShotaoVisibilityTimer_Tick(object sender, EventArgs e)
237 {
238     isShotaoVisible = true;
239     shotaoVisibilityTimer.Stop();
240     Invalidate();
241 }
242
243 private void ShotarkaVisibilityTimer_Tick(object sender, EventArgs e)
244 {
245     isShotarkaVisible = true;
246     shotarkaVisibilityTimer.Stop();
247     Invalidate();
248 }*/
249
250 //-----
251
252 //ตรวจสอบว่า Murasaki Ao Arka ตัวไหนเคลื่อนที่หรือไม่ เช่นตำแหน่ง X ของ Shotmurasaki
253 //ตรวจสอบกับขอบเขตของ Form หรือไม่ หากมีการเคลื่อนที่ให้กับขอบเขตและเรียก CheckMurasakiHit() ถ้ายังไม่เกิน
254 //-----
255

```

```

256 private void MurasakiMoveTimer_Tick(object sender, EventArgs e)
257 {
258
259     if (isMurasakiMoving)
260     {
261         shotmurasakiX += 30; // สก๊อตไปทางขวาที่ 30
262         if (shotmurasakiX > this.Width) // ตรวจสอบถ้ากระสุนออกนอกหน้าจอ
263         {
264             isMurasakiMoving = false;
265             murasakiMoveTimer.Stop();
266         }
267     }
268     else
269     {
270         CheckMurasakiHit(); // ตรวจสอบการชนของกระสุน
271     }
272     Invalidate(); // อัปเดตหน้าจอ
273 }

```

```

//-----
277 private void AoMoveTimer_Tick(object sender, EventArgs e)
278 {
279     if (isAoMoving)
280     {
281         shotaoX += 30;
282         if (shotaoX > this.Width)
283         {
284             isAoMoving = false;
285             aoMoveTimer.Stop();
286         }
287     }
288     else
289     {
290         CheckAoHit();
291     }
292     Invalidate();
293 }

```

```

//-----
297 private void ArkaMoveTimer_Tick(object sender, EventArgs e)
298 {
299     if (isArkaMoving)
300     {
301         shotarkaX += 30;
302         if (shotarkaX > this.Width)
303         {
304             isArkaMoving = false;
305             arkaMoveTimer.Stop();
306         }
307     }
308     else
309     {
310         CheckArkaHit();
311     }
312     Invalidate();
313 }

```

```

//-----
317 //หยุดทุกฟังก์ชันที่เมื่อทุกกลไกมันถึงที่เริ่มต้น Timer
318 //-----

```

```

321 private void SkillCooldownTimer_Tick(object sender, EventArgs e)
322 {
323     skillRCooldownTimer.Stop();
324     skillECooldownTimer.Stop();
325     skillQCooldownTimer.Stop();
326     skillWCooldownTimer.Stop();
327 }

```

```

//-----
331 //ตรวจสอบว่า HP ของ Gojo หรือ Sukuna มีค่าน้อยกว่าหรือเท่ากับ 0 หรือไม่
332 //หยุด Timer ทั้งหมดในส่วนของพาร์ท HP นulled
333 //เปลี่ยนค่า showImage เพื่อทำให้ภาพเปลี่ยนเป็นแสดงชื่อของคนที่ถ่านนulled
334 //-----

```

```

336 private void BlinkTimer_Tick(object sender, EventArgs e)
337 {
338     if (gojoCurrentHp <= 0 || sukunaCurrentHp <= 0)
339     {
340
341         ((System.Windows.Forms.Timer) sender).Stop();
342         return;
343     }
344
345     showImage = !showImage; // สลับสถานะการแสดงผลภาพ
346     Invalidate(); // อัปเดตหน้าจอ
347 }

```

```

349 private void Blink()
350 {
351     blinkTimer = new System.Windows.Forms.Timer();
352     blinkTimer.Interval = 2000;
353     blinkTimer.Tick += BlinkTimer_Tick;
354     blinkTimer.Start();
355 }

```

```

//-----
357 //ตั้งค่าตัวจับเวลาสำหรับการเคลื่อนที่ Ao Arka Murasaki Skillcooldown
358 //-----

```

```

362 private void AoShotMove()
363 {
364     aoMoveTimer = new System.Windows.Forms.Timer();
365     aoMoveTimer.Interval = 50; // ตั้งเวลาให้เคลื่อนที่ครั้งเป็น 50 มิลลิวินาที
366     aoMoveTimer.Tick += AoMoveTimer_Tick;
367 }

```

```

369 private void ArkaShotMove()
370 {
371     arkaMoveTimer = new System.Windows.Forms.Timer();
372     arkaMoveTimer.Interval = 50; // ตั้งเวลาให้เคลื่อนที่ครั้งเป็น 50 มิลลิวินาที
373     arkaMoveTimer.Tick += ArkaMoveTimer_Tick;
374 }

```

```

377 private void MurasakiShotMove()
378 {
379     murasakiMoveTimer = new System.Windows.Forms.Timer();
380     murasakiMoveTimer.Interval = 50; // ตั้งเวลาให้เคลื่อนที่ครั้งเป็น 50 มิลลิวินาที
381     murasakiMoveTimer.Tick += MurasakiMoveTimer_Tick;
382 }

```

```

384 /
385
386 private void Skillcooldown()
387 {
388     skillRCooldownTimer = new System.Windows.Forms.Timer();
389     skillRCooldownTimer.Interval = 60000; // ตั้งเวลาปล่อยสกิลเป็น 60,000 มิลลิวินาที (60 วินาที)
390     skillRCooldownTimer.Tick += SkillCooldownTimer_Tick;
391
392     skillECooldownTimer = new System.Windows.Forms.Timer();
393     skillECooldownTimer.Interval = 20000; // ตั้งเวลาปล่อยสกิลเป็น 20,000 มิลลิวินาที (20 วินาที)
394     skillECooldownTimer.Tick += SkillCooldownTimer_Tick;
395
396     skillQCooldownTimer = new System.Windows.Forms.Timer();
397     skillQCooldownTimer.Interval = 3000; // ตั้งเวลาปล่อยสกิลเป็น 3,000 มิลลิวินาที (3 วินาที)
398     skillQCooldownTimer.Tick += SkillCooldownTimer_Tick;
399
400     skillWCooldownTimer = new System.Windows.Forms.Timer();
401     skillWCooldownTimer.Interval = 5000; // ตั้งเวลาปล่อยสกิลเป็น 5,000 มิลลิวินาที (5 วินาที)
402     skillWCooldownTimer.Tick += SkillCooldownTimer_Tick;
403 }
404
405 //-----
406
407 //
408 private void HPBar()
409 {
410     // สร้าง ProgressBar, Label, และ Font สำหรับ Gojo
411     // -----
412
413     gojoHP = new ProgressBar();
414     gojoName = new Label();
415     gojoFont = new Font("Arial", 20, FontStyle.Regular);
416     gojoHP.Minimum = 0;
417     gojoHP.Maximum = 100;
418     gojoHP.Value = 100;
419     gojoName.Text = "Gojo Satoru";
420     gojoName.Font = gojoFont;
421
422     //-----
423
424     // สร้างและตั้งค่า Timer สำหรับลด GojoHP ทุก 2 วินาที
425     // -----
426
427     System.Windows.Forms.Timer gojoHPDecrementTimer = new System.Windows.Forms.Timer();
428     gojoHPDecrementTimer.Interval = 2000; // ตั้งเวลาปล่อยสกิลเป็น 2000 มิลลิวินาที (2 วินาที)
429     gojoHPDecrementTimer.Tick += GojoHPDecrementTimer_Tick; // เชื่อมต่อกับเหตุการณ์ Tick กับฟังก์ชัน GojoHPDecrementTimer_Tick
430     gojoHPDecrementTimer.Start(); // เริ่มการทำงานของ Timer
431
432     //-----
433
434     // สร้างและตั้งค่า Timer สำหรับเพิ่ม GojoHP ทุก 2 วินาที
435     // -----
436
437     System.Windows.Forms.Timer autoIncreaseGojoTimer;
438     autoIncreaseGojoTimer = new System.Windows.Forms.Timer();
439     autoIncreaseGojoTimer.Interval = 1000;
440     autoIncreaseGojoTimer.Tick += AutoIncreaseTimer_Tick;
441     autoIncreaseGojoTimer.Start();
442
443     //-----
444
445     // กำหนดตำแหน่งและขนาดของ GojoHP และ GojoName
446     // -----
447
448     gojoHP.Location = new Point(70, 50);
449     gojoName.Location = new Point(70, 100);
450     gojoHP.Size = new Size(800, 30);
451     gojoName.Size = new Size(170, 32);
452
453     //-----
454
455     // สร้าง ProgressBar, Label, และ Font สำหรับ Sukuna
456     // -----
457
458     sukunaHP = new ProgressBar();
459     sukunaName = new Label();
460     sukunaFont = new Font("Arial", 20, FontStyle.Regular);
461     sukunaHP.Minimum = 0;
462     sukunaHP.Maximum = 1000;
463     sukunaHP.Value = 1000;
464     sukunaName.Text = "Ryomen Sukuna (Yuji Form)";
465     sukunaName.Font = sukunaFont;
466
467     //-----
468
469     // สร้างและตั้งค่า Timer สำหรับเพิ่ม SukunaHP ทุก 2 วินาที
470     // -----
471
472     System.Windows.Forms.Timer autoIncreaseSukunaTimer;
473     autoIncreaseSukunaTimer = new System.Windows.Forms.Timer();
474     autoIncreaseSukunaTimer.Interval = 2000;
475     autoIncreaseSukunaTimer.Tick += AutoIncreaseTimer_Tick;
476     autoIncreaseSukunaTimer.Start();
477
478     //-----
479
480     // กำหนดตำแหน่งและขนาดของ SukunaHP และ SukunaName
481     // -----
482
483     sukunaHP.Location = new Point(1000, 50);
484     sukunaName.Location = new Point(1430, 100);
485     sukunaHP.Size = new Size(800, 30);
486     sukunaName.Size = new Size(370, 32);
487
488     //-----
489
490     //เพิ่ม GojoHP, GojoName, SukunaHP, และ SukunaName ลงใน Controls
491     // -----
492
493     Controls.Add(gojoHP);
494     Controls.Add(gojoName);
495     Controls.Add(sukunaHP);
496     Controls.Add(sukunaName);
497
498     //-----
499 }
500 // โหลดรูปภาพสำหรับตัวละครและพื้นหลัง
501 //-----
502 private void SetUpForm()
503 {
504     SetStyle(ControlStyles.AllPaintingInWmPaint | ControlStyles.OptimizedDoubleBuffer | ControlStyles.UserPaint, true);
505
506     gojo = Image.FromFile("gojo\\1.png");
507     sukuna = Image.FromFile("sukuna\\1.png");
508     bg = Image.FromFile("background.jpg");
509 }
510 //-----

```

```

511
512 //ตัวชี้เวลาสำหรับเพิ่ม HP ยังไม่มี
513 //-----
514
515 private void AutoIncreaseTimer_Tick(object sender, EventArgs e)
516 {
517     // หยุดชี้เวลาถ้า HP ของ Gojo หรือ Sukuna เป็น 0 หรือน้อยกว่า
518     if (gojoCurrentHp <= 0 || sukunaCurrentHp <= 0)
519     {
520         ((System.Windows.Forms.Timer)sender).Stop();
521         return;
522     }
523     // เพิ่ม HP ของ Sukuna และ Gojo ตามเวลา
524     SukunaReverse(10); // เพิ่ม HP ของ Sukuna 10 หน่วย
525     IncrementGojoHealth(1); // เพิ่ม HP ของ Gojo 1 หน่วย
526 }
527
528 //-----
529
530 //ตัวชี้เวลาสำหรับลด HP ของ Gojo ตามเวลา
531 //-----
532
533 private void GojoHPDecrementTimer_Tick(object sender, EventArgs e)
534 {
535     // หยุดชี้เวลาถ้า HP ของ Gojo หรือ Sukuna เป็น 0 หรือน้อยกว่า
536     if (gojoCurrentHp <= 0 || sukunaCurrentHp <= 0)
537     {
538         ((System.Windows.Forms.Timer)sender).Stop();
539         return;
540     }
541     // ลด HP ของ Gojo ถ้าปุ่ม E ไม่ถูกกดค้าง
542     if (!isKeyHeld)
543     {
544         if (gojoCurrentHp <= 0 || sukunaCurrentHp <= 0)
545         {
546             ((System.Windows.Forms.Timer)sender).Stop();
547             return;
548         }
549         DecrementGojoHealth(10);
550     }
551 }
552
553 //-----
554
555 //เพิ่ม HP ของทั้ง Gojo และ Sukuna
556 //-----
557
558 private void GojoReverse(int increment)
559 {
560     gojoCurrentHp += increment;
561     // ตรวจสอบและจำกัด HP ให้อยู่ในช่วงที่กำหนด
562     if (gojoCurrentHp < gojoHP.Minimum)
563         gojoCurrentHp = gojoHP.Minimum;
564
565     if (gojoCurrentHp > gojoHP.Maximum)
566         gojoCurrentHp = gojoHP.Maximum;
567
568     gojoHP.Value = gojoCurrentHp;
569 }
570
571 private void IncrementGojoHealth(int increment)
572 {
573     // ตรวจสอบก่อนเพิ่ม HP
574     if (gojoCurrentHp <= 0 || sukunaCurrentHp <= 0)
575     {
576         return;
577     }
578
579     gojoCurrentHp += increment;
580     // จำกัด HP ให้อยู่ในช่วงที่กำหนด
581     if (gojoCurrentHp > gojoHP.Maximum)
582     {
583         gojoCurrentHp = gojoHP.Maximum;
584     }
585
586     gojoHP.Value = gojoCurrentHp;
587 }
588
589 private void SukunaReverse(int increment2)
590 {
591     if (sukunaCurrentHp <= 0 || gojoCurrentHp <= 0)
592     {
593         return;
594     }
595
596     sukunaCurrentHp += increment2;
597     if (sukunaCurrentHp > sukunaHP.Maximum)
598     {
599         sukunaCurrentHp = sukunaHP.Maximum;
600     }
601
602     sukunaHP.Value = sukunaCurrentHp;
603 }
604
605 //-----
606
607 //ลด HP ของ Gojo และจัดการลด HP
608 //-----
609
610 private void DecrementGojoHealth(int decrement)
611 {
612     gojoCurrentHp -= decrement;
613
614     if (gojoCurrentHp < gojoHP.Minimum)
615     {
616         gojoCurrentHp = gojoHP.Minimum;
617         this.Invalidate();
618     }
619
620     if (gojoCurrentHp > gojoHP.Maximum)
621         gojoCurrentHp = gojoHP.Maximum;
622
623     gojoHP.Value = gojoCurrentHp;
624 }
625
626 //-----
627
628 // ตรวจสอบสถานะของสเกลต่างๆ และกำหนดภาพและขนาดของตัวละคร Gojo ตามสเกลที่ถูกละเลือก
629 //-----
630
631 private void gojoSkills()
632 {
633     if (Ao)
634     {
635         gojo = Image.FromFile("gojo\\2.png");
636         gojoWidth = 550;
637         gojoHeight = 800;
638     }
639 }

```

```

639     }
640     else if (Arka)
641     {
642         gojo = Image.FromFile("gojo\\3.png");
643         gojoWidth = 550;
644         gojoHeight = 800;
645     }
646     else if (Mugen)
647     {
648         gojo = Image.FromFile("gojo\\4.png");
649         gojoWidth = 550;
650         gojoHeight = 800;
651     }
652     else if (Murasaki)
653     {
654         gojo = Image.FromFile("gojo\\5.png");
655         gojoWidth = 550;
656         gojoHeight = 800;
657     }
658     else
659     {
660         gojo = Image.FromFile("gojo\\1.png");
661         gojoWidth = 500;
662         gojoHeight = 800;
663     }
664
665     // ถ้า gojoReverse และ isKeyEHeld เป็น true, จะเรียก GojoReverse
666     if (gojoReverse && isKeyEHeld)
667     {
668         GojoReverse(50);
669     }
670     // อัปเดตหน้าจอเพื่อแสดงการเปลี่ยนแปลง
671     Invalidate();
672 }
673
674 //-----
675
676 // ตรวจสอบเมื่อมีการกดปุ่มบนแป้นพิมพ์และทำการเปิดใช้สกิลตามปุ่มที่ถูกกด
677 //-----
678
679 private void KeyIsDown(object sender, KeyEventArgs e)
680 {
681     if (e.KeyCode == Keys.Q)
682     {
683         if (skillQCooldownTimer != null && !skillQCooldownTimer.Enabled)
684         {
685             Ao = true;
686             ShootAo();
687             gojoSkills();
688             skillQCooldownTimer.Start();
689         }
690     }
691     else if (e.KeyCode == Keys.W)
692     {
693         if (skillWCooldownTimer != null && !skillWCooldownTimer.Enabled)
694         {
695             Arka = true;
696             gojoSkills();
697             ShootArka();
698             skillWCooldownTimer.Start();
699         }
700     }
701     else if (e.KeyCode == Keys.E)
702     {
703         if (skillECooldownTimer != null && !skillECooldownTimer.Enabled)
704         {
705             Mugen = true;
706             gojoReverse = true;
707             isKeyEHeld = true;
708             gojoSkills();
709             skillECooldownTimer.Start();
710         }
711     }
712     else if (e.KeyCode == Keys.R)
713     {
714         if (skillRCooldownTimer != null && !skillRCooldownTimer.Enabled)
715         {
716             Murasaki = true;
717             gojoSkills();
718             ShootMurasaki();
719             skillRCooldownTimer.Start();
720         }
721     }
722 }
723
724 //-----
725
726 //-----
727
728 // ตรวจสอบเมื่อปล่อยปุ่มบนแป้นพิมพ์และหยุดการใช้สกิลที่เกี่ยวข้อง
729 //-----
730 private void KeyIsUp(object sender, KeyEventArgs e)
731 {
732     if (e.KeyCode == Keys.Q)
733     {
734         Ao = false;
735         gojoSkills();
736     }
737     else if (e.KeyCode == Keys.W)
738     {
739         Arka = false;
740         gojoSkills();
741     }
742     else if (e.KeyCode == Keys.E)
743     {
744         Mugen = false;
745         gojoReverse = false;
746         isKeyEHeld = false;
747         gojoSkills();
748     }
749     else if (e.KeyCode == Keys.R)
750     {
751         Murasaki = false;
752         gojoSkills();
753     }
754 }
755
756 //-----
757
758 // ฟังก์ชันที่จัดการกับการใช้สกิล Murasaki Ao Arka
759 //-----
760
761 private void ShootMurasaki()
762 {
763     shotmurasaki = Image.FromFile("gojo\\shotmurasaki.png");
764     shotmurasakiX = gojoX + gojoWidth;
765     shotmurasakiY = 200;
766     shotmurasakiWidth = 750;

```

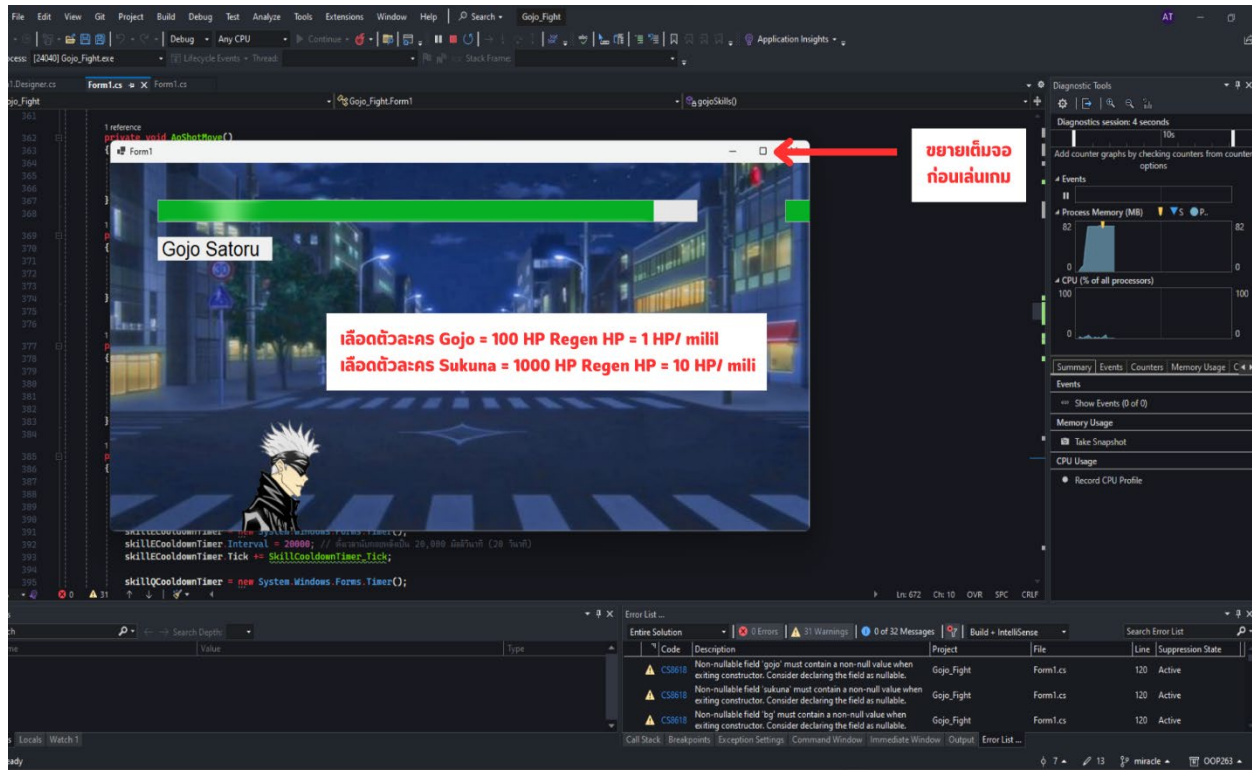
```

767         shotMurasakiHeight = 750;
768         isMurasakiMoving = true;
769         murasakiMoveTimer.Start();
770     }
771
772     private void ShootAo()
773     {
774         shotao = Image.FromFile("gojo\\shotao.png");
775         shotaoX = gojoX + 350;
776         shotaoY = 400;
777         shotaoWidth = 250;
778         shotaoHeight = 270;
779         isAoMoving = true;
780         aoMoveTimer.Start();
781     }
782     private void ShootArka()
783     {
784         shotarka = Image.FromFile("gojo\\shotarka.png");
785         shotarkaX = gojoX + 350;
786         shotarkaY = 400;
787         shotarkaWidth = 250;
788         shotarkaHeight = 270;
789         isArkaMoving = true;
790         arkaMoveTimer.Start();
791     }
792
793     //-----
794
795     //ฟังก์ชันที่ตรวจสอบว่าลูก Murasaki Ao Arka ถูกตัวละคร Sukuna หรือไม่
796     //-----
797
798     private void CheckMurasakiHit()
799     {
800         // ตรวจสอบลูก Murasaki โดนตัวละคร Sukuna
801         bool collision = DetectCollision(shotMurasakiX, shotMurasakiY, shotMurasakiWidth, shotMurasakiHeight, sukunaX, sukunaY, sukunaWidth, sukunaHeight);
802
803         if (collision)
804         {
805             // ลดค่า HP ของ Sukuna ตามการชน
806             sukunaCurrentHp -= 20;
807             // ตรวจสอบไม่ให้ HP ต่ำกว่า 0
808             if (sukunaCurrentHp < 0) sukunaCurrentHp = 0;
809             // อัปเดตค่า HP ของ Sukuna
810             sukunaHP.Value = sukunaCurrentHp;
811         }
812     }
813
814     private void CheckAoHit()
815     {
816         // ตรวจสอบลูก Ao โดนตัวละคร Sukuna
817         bool collision = DetectCollision(shotaoX, shotaoY, shotaoWidth, shotaoHeight, sukunaX, sukunaY, sukunaWidth, sukunaHeight);
818
819         if (collision)
820         {
821             sukunaCurrentHp -= 8;
822             if (sukunaCurrentHp < 0) sukunaCurrentHp = 0;
823             sukunaHP.Value = sukunaCurrentHp;
824             // isShotaoVisible = false;
825             // shotaoVisibilityTimer.Start();
826         }
827     }
828
829     private void CheckArkaHit()
830     {
831         // ตรวจสอบลูก Arka โดนตัวละคร Sukuna
832         bool collision = DetectCollision(shotarkaX, shotarkaY, shotarkaWidth, shotarkaHeight, sukunaX, sukunaY, sukunaWidth, sukunaHeight);
833
834         if (collision)
835         {
836             sukunaCurrentHp -= 10;
837             if (sukunaCurrentHp < 0) sukunaCurrentHp = 0;
838             sukunaHP.Value = sukunaCurrentHp;
839             // isShotarkaVisible = false;
840             // shotarkaVisibilityTimer.Start();
841         }
842     }
843
844     //-----
845
846     //ฟังก์ชันที่ตรวจสอบการชนระหว่างวัตถุ
847     //-----
848
849     private bool DetectCollision(int object1X, int object1Y, int object1Width, int object1Height, int object2X, int object2Y, int object2Width, int object2Height)
850     {
851         // ตรวจสอบว่ามีการชนระหว่าง object1 และ object2 หรือไม่
852         if (object1X + object1Width <= object2X || object1X >= object2X + object2Width || object1Y + object1Height <= object2Y || object1Y >= object2Y + object2Height)
853         {
854             return false;
855         }
856         else
857         {
858             return true;
859         }
860     }
861
862     //-----
863
864     }
865 }
866

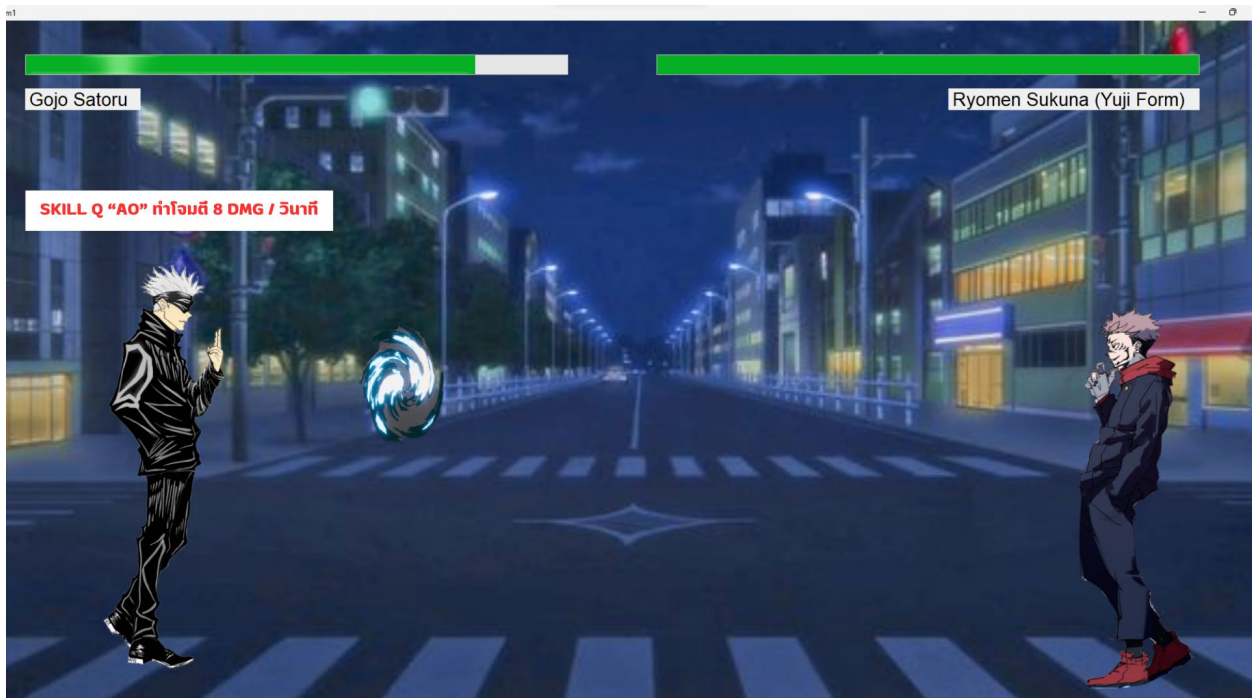
```

ผลการทำงาน

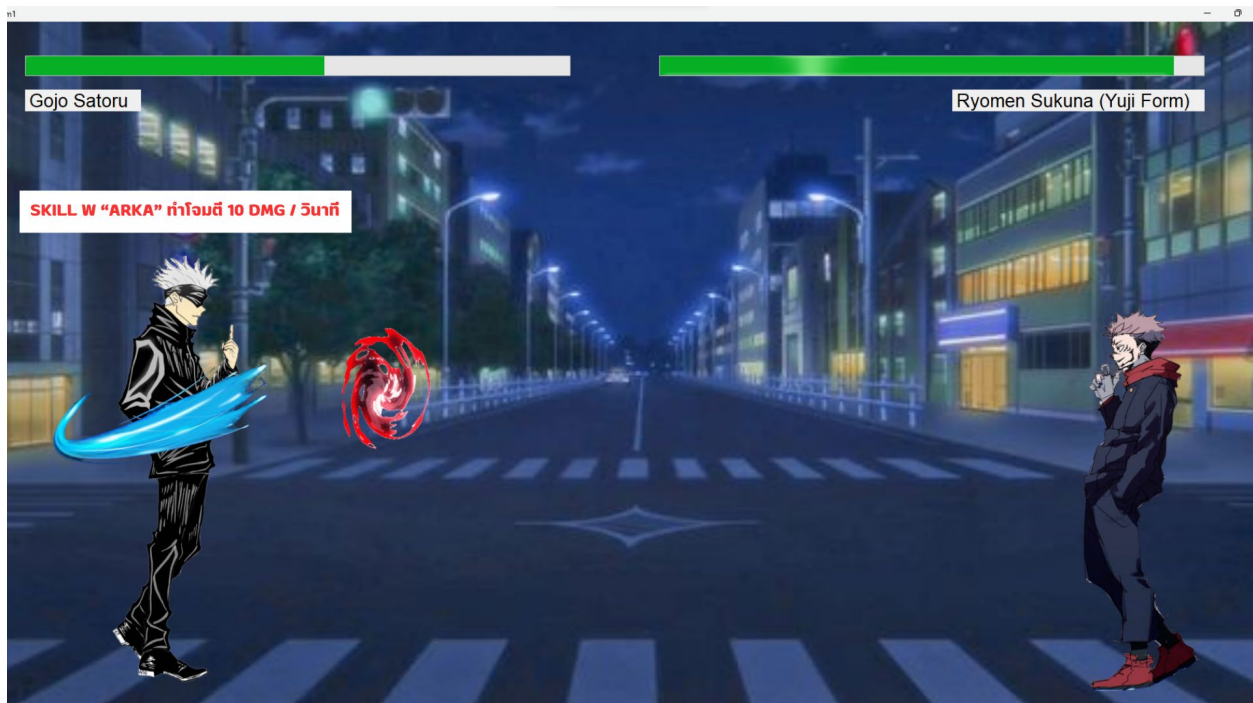
1.



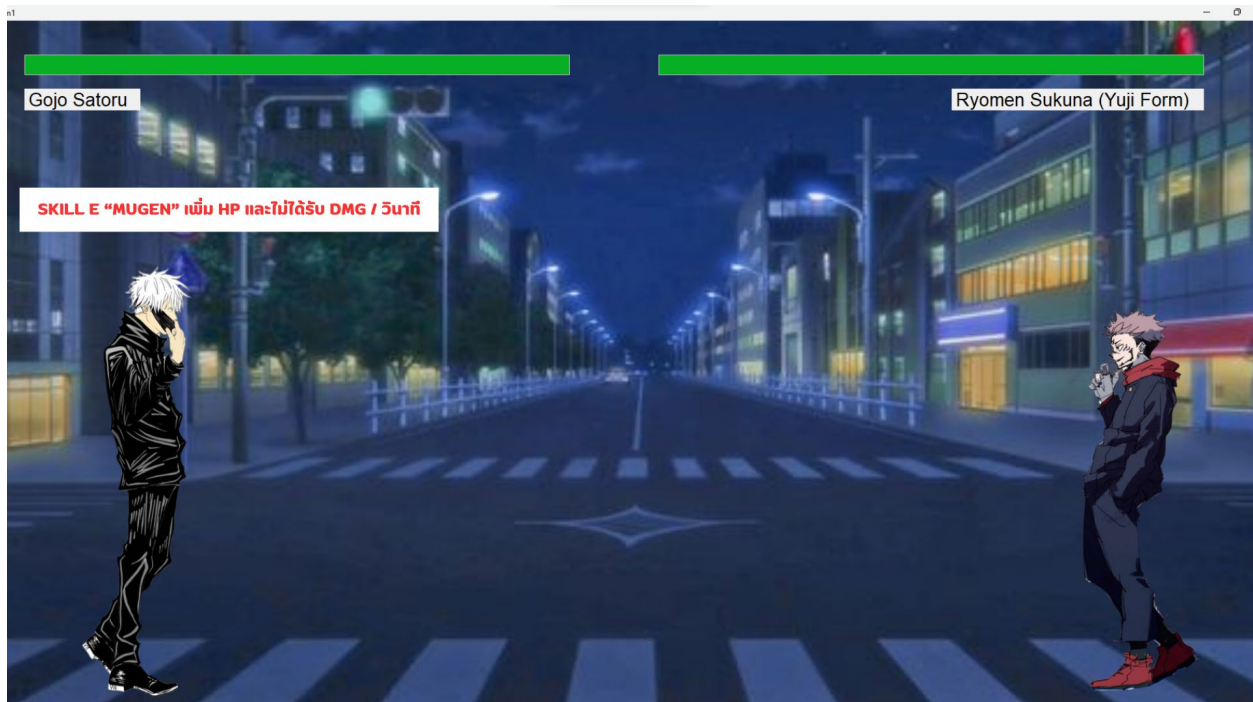
2.



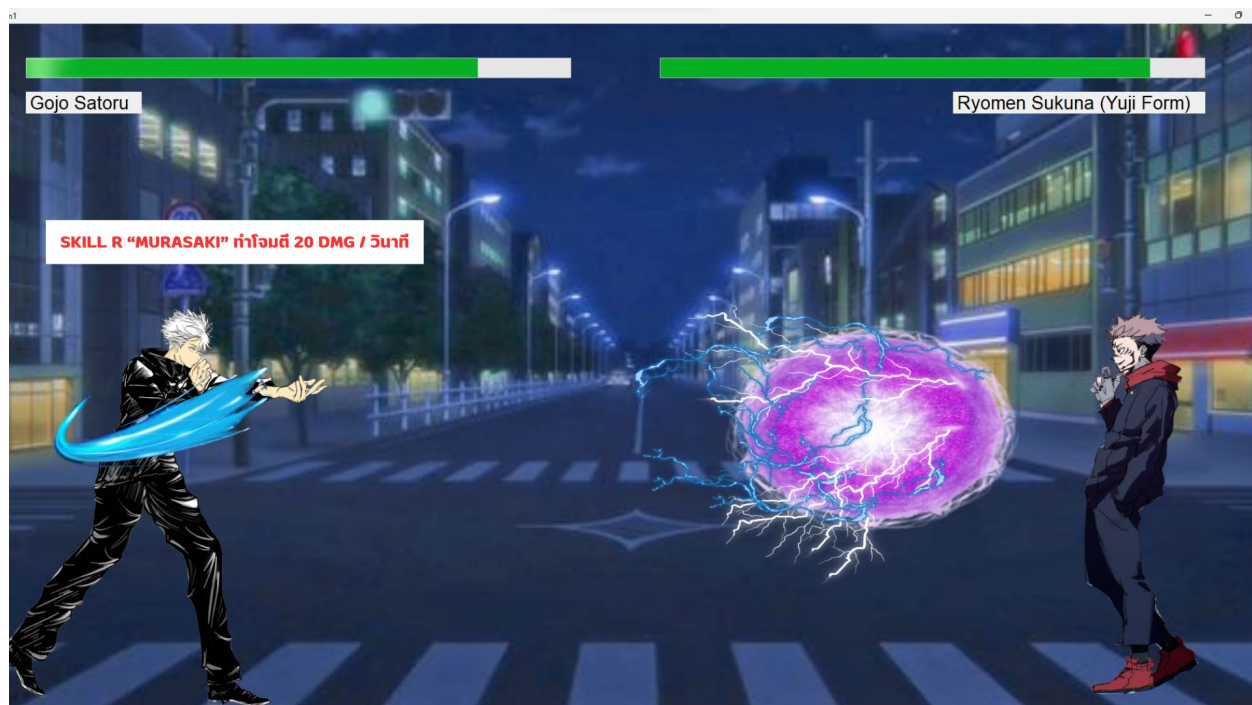
3.



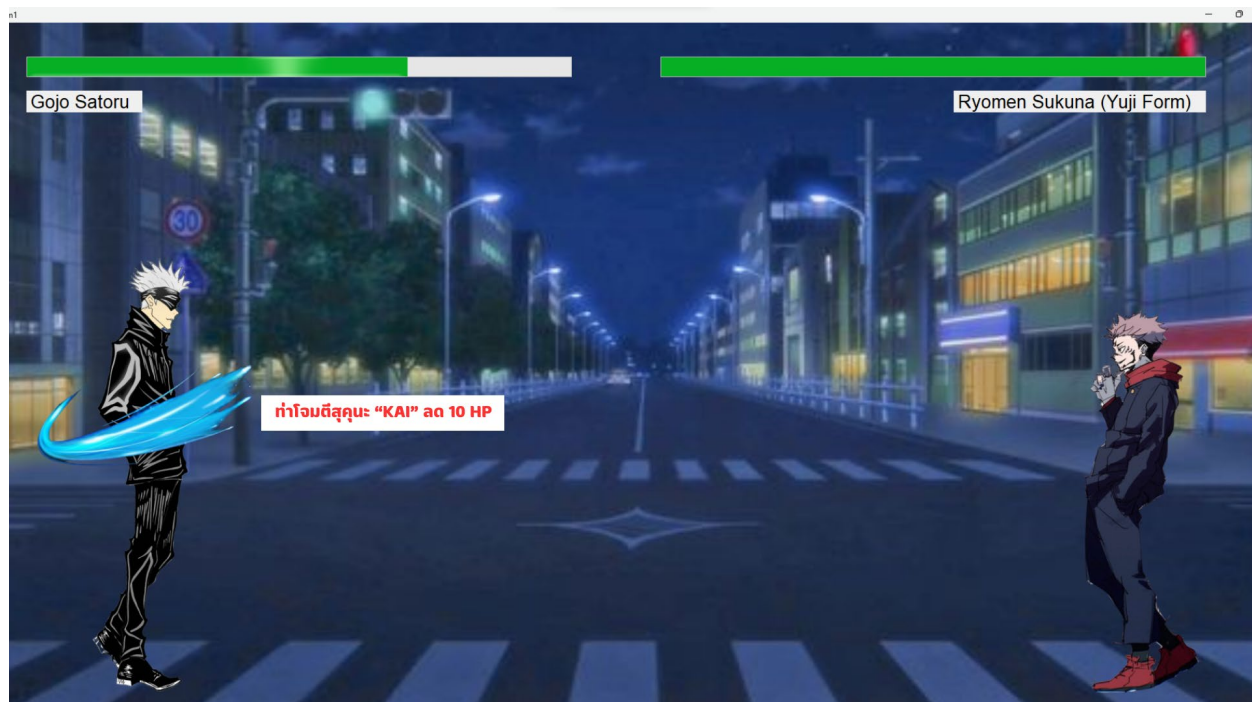
4.



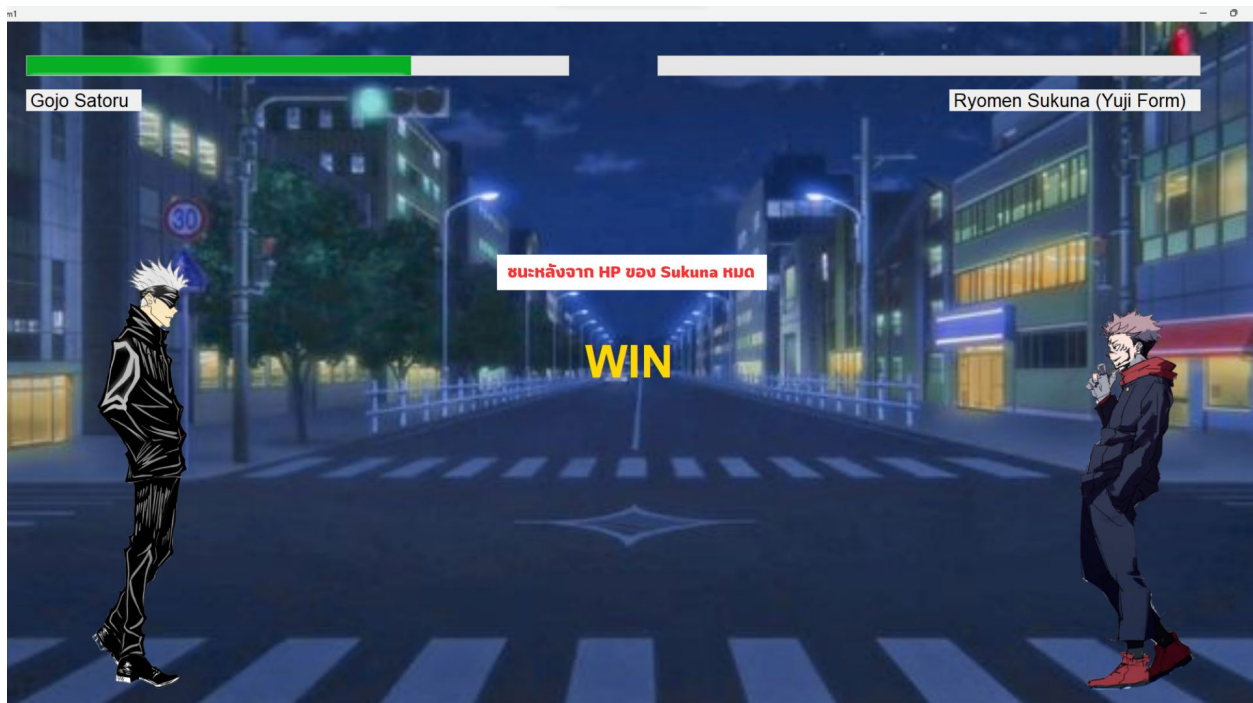
5.



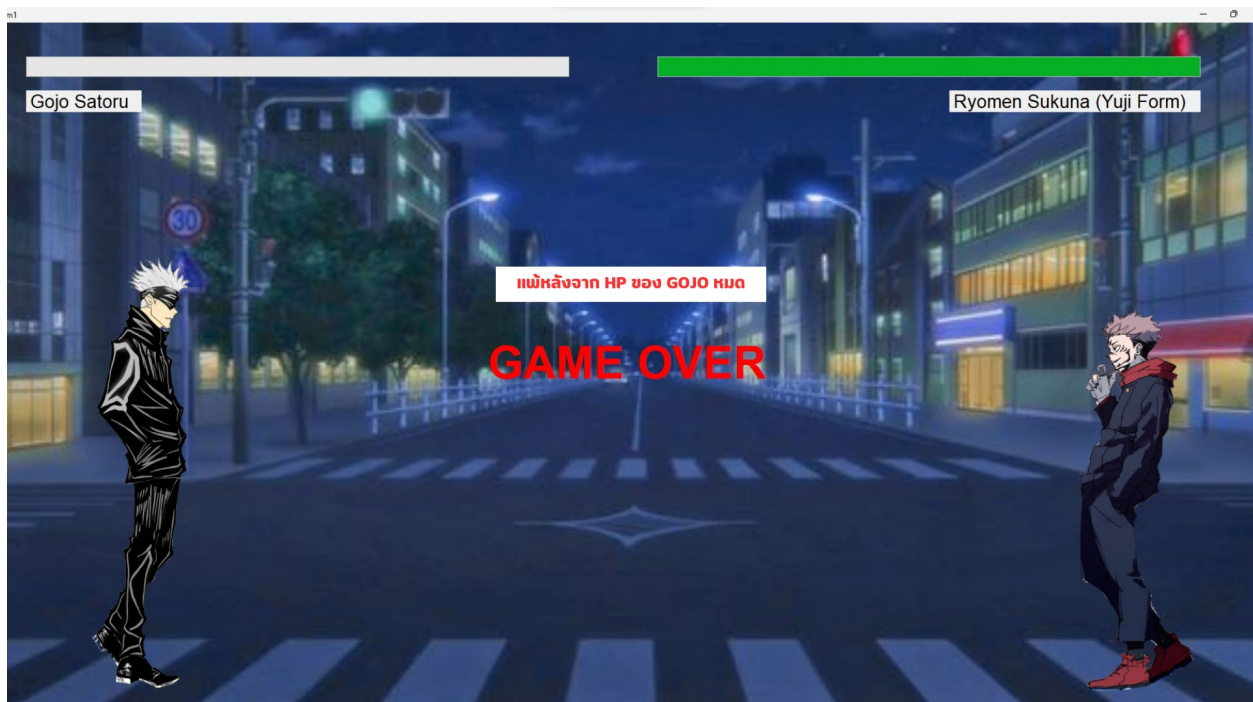
6.



7.



8.



ดาวน์โหลดเกม หรือ ทดลองเล่นเกมได้ที่นี้



https://github.com/miraclegalaxys/project263_game