



รายงาน

เรื่อง หลักการทำงานและการใช้งานโปรแกรม Bouncing ball (ลูกบอลเตี้ง)

จัดทำโดย

นายพงษ์พันธุ์ เลาวพงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2566

รายงาน

เรื่อง หลักการทำงานและการใช้งานโปรแกรม Bouncing ball (ลูกบอลเต้)

จัดทำโดย

นายพงษ์พันธุ์ เลาหวงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2566

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา ENGCE124 โครงสร้างข้อมูลและขั้นตอนวิธี (Data Structures and Algorithms) หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่ ในระดับปริญญาตรีปีที่ 2 โดยมีจุดประสงค์ในอธิบายการทำงานของโปรแกรม Bouncing ball (ลูกบอลเต็ง) ในส่วนของโค้ดการทำงานของโปรแกรม ไม่ว่าจะเป็นโค้ดในส่วนการเคลื่อนไหวของลูกบอล การเคลื่อนไหวของแป้นสะท้อน รวมถึงโค้ดคำสั่งในการสะท้อนลูกบอลเมื่อโดนขอบหน้าจอของโปรแกรม และโดนแป้นสะท้อนเพื่อลูกบอลสัมผัส ส่วนสุดท้ายรายงานจะเป็นผลลัพธ์และวิธีการใช้งานโปรแกรม

ผู้จัดทำรายงานหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้สนใจ หรือนักศึกษาทุกท่านที่กำลังหาศึกษาการสร้างโปรแกรม Bouncing ball (ลูกบอลเต็ง) หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้ และขออภัยมา ณ ที่นี้

ผู้จัดทำ

นายพงษ์พันธุ์ เลาวพงศ์

วันที่ 08/07/2567

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	๗
โค้ดของโปรแกรม Bouncing ball (ลูกบอลเต้) พร้อมคำอธิบาย	1
หลักการทำงานของโปรแกรม Bouncing ball (ลูกบอลเต้)	8
การใช้งานโปรแกรม Bouncing ball (ลูกบอลเต้)	13
บรรณานุกรม	14

โค้ดของโปรแกรม Bouncing ball (ลูกบอลเต้) พร้อมคำอธิบาย

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace Bouncing_ball
{
    public partial class Form1 : Form
    {
        //กำหนดตัวแปรต่างที่จะใช้งานในโปรแกรม

        private int ballWidth = 40; // ความกว้างของลูกบอล มีหน่วยเป็น พิกเซล
        private int ballHeight = 40; // ความสูงของลูกบอล มีหน่วยเป็น พิกเซล
        private float ballPosX = 0; // ตำแหน่งลูกบอลบนแกน X
        private float ballPosY = 0; // ตำแหน่งลูกบอลบนแกน Y
        private float moveStepX = 8; // [ค่ามาตรฐาน : 4] ความเร็วการเลื่อนของลูกบอลบนแกน
        X มีหน่วยเป็น พิกเซล

        private int moveStepY = 8; // [ค่ามาตรฐาน : 4] ความเร็วการเลื่อนของลูกบอลบนแกน Y
        มีหน่วยเป็น พิกเซล

        private System.Windows.Forms.Timer ballTimer; // จับเวลาสำหรับภาพเคลื่อนไหว
        ของลูกบอล

        private int paddleWidth = 300; // ความกว้างของแป้น มีหน่วยเป็น พิกเซล
        private int paddleHeight = 5; // ความสูงของแป้น มีหน่วยเป็น พิกเซล
        private int paddlePosX; // ตำแหน่งแป้นบนแกน X
        private int paddlePosY; // ตำแหน่งแป้นบนแกน Y
        private int paddleMoveStep = 32; // [ค่ามาตรฐาน : 8] ความเร็วการเลื่อนของแป้น มี
        หน่วยเป็น พิกเซล

        private System.Windows.Forms.Timer paddleTimer; // จับเวลาสำหรับภาพเคลื่อนไหว
        ของแป้น

        private bool isLeftPressed = false; // ติดตามว่ากดปุ่มลูกศรซ้ายถูกกด
        private bool isRightPressed = false; // ติดตามว่ากดปุ่มลูกศรขวาถูกกด

        private bool isUpPressed = false; // ติดตามว่ากดปุ่มลูกศรบนถูกกด
        private bool isDownPressed = false; // ติดตามว่ากดปุ่มลูกศรล่างถูกกด
    }
}
```

```
// ฟังก์ชัน Form1 เป็นฟังก์ชันหลักของโปรแกรม
public Form1()
{
    InitializeComponent();

    // ใช้งานฟังก์ชันที่มีชื่อว่า Form1_Load
    this.Load += new EventHandler(Form1_Load);

    // ตั้งค่าฟเฟอร์สองเท่าเพื่อลดการสั่นไหว [ใช้เงื่อนไขนี้เพื่อให้ลูกบอลเคลื่อนที่มีความลื่นไหล]
    this.SetStyle(ControlStyles.OptimizedDoubleBuffer |
ControlStyles.AllPaintingInWmPaint | ControlStyles.UserPaint, true);
    this.UpdateStyles();

    // เริ่มต้นจับเวลาลูกบอล เวลาเคลื่อนที่
    // การจับเวลาจะทำให้เกิดการอัปเดตตำแหน่งของลูกบอลเป็นระยะๆ ตามช่วงเวลาที่กำหนด
    ballTimer = new System.Windows.Forms.Timer();
    ballTimer.Interval = 30; // ตั้งค่าช่วงเวลา มีหน่วยเป็น มิลลิวินาที
    ballTimer.Tick += new EventHandler(MoveBall); //การใช้งานฟังก์ชัน Moveball
ที่มีผลกับการจับเวลา

    ballTimer.Start(); //เริ่มต้นการจับเวลา

    // เริ่มต้นจับเวลาแป้น เวลาเคลื่อนที่
    // การจับเวลาจะทำให้เกิดการอัปเดตตำแหน่งของลูกบอลเป็นระยะๆ ตามช่วงเวลาที่กำหนด
    paddleTimer = new System.Windows.Forms.Timer();
    paddleTimer.Interval = 15; // ตั้งค่าช่วงเวลา มีหน่วยเป็น มิลลิวินาที
    paddleTimer.Tick += new EventHandler(MovePaddle); //การใช้งานฟังก์ชัน
Movepaddle ที่มีผลกับการจับเวลา

    // กำหนดตำแหน่งลูกบอลตอนเริ่มต้น
    ballPosX = (this.ClientSize.Width - ballWidth) / 1; // [ค่ามาตรฐาน : 2]
กำหนดตำแหน่งในแนวนอนของลูกบอล

    ballPosY = (this.ClientSize.Height - ballHeight) / 1; // [ค่ามาตรฐาน : 2]
กำหนดตำแหน่งในแนวตั้งของลูกบอล

    // กำหนดตำแหน่งแป้นตอนเริ่มต้น
    paddlePosX = (this.ClientSize.Width - paddleWidth) / 1; // [ค่ามาตรฐาน :
2] กำหนดตำแหน่งในแนวนอนของแป้น

    paddlePosY = this.ClientSize.Height - paddleHeight + 500; // [ค่ามาตรฐาน
: 10] กำหนดตำแหน่งในแนวตั้งของแป้น
```

```

// ใช้งานฟังก์ชัน PaintCircle
this.Paint += new PaintEventHandler(PaintCircle);

// ใช้งานฟังก์ชัน OnKeyDown และ ฟังก์ชัน OnKeyUp
this.KeyDown += new KeyEventHandler(OnKeyDown);
this.KeyUp += new KeyEventHandler(OnKeyUp);
}

// ฟังก์ชัน PaintCircle เป็นฟังก์ชันในการใช้งานการวาดจากฟังก์ชันต่างๆ
private void PaintCircle(object sender, PaintEventArgs e)
{
    e.Graphics.SmoothingMode =
System.Drawing.Drawing2D.SmoothingMode.AntiAlias; // ทำให้กรอบของลูกบอลเรียบขึ้น

    e.Graphics.Clear(this.BackColor); // ล้างพื้นที่วาดด้วยสีพื้นหลังของฟอร์ม

    // เรียกใช้งานเมธอดของการวาดรูปลูกบอล
    PaintBall(e.Graphics);

    // เรียกใช้งานเมธอดของการวาดรูปแป้น
    PaintPaddle(e.Graphics);

    // เรียกใช้งานเมธอดของการวาดรูป เขียนชื่อผู้ทำ
    PongpanLaowaphong(e.Graphics);
}

// ฟังก์ชัน PaintBall เป็นฟังก์ชันในการสร้างรูปลูกบอล
private void PaintBall(Graphics graphics)
{
    // วาดลูกบอลขึ้นมาให้เป็นสีแดง
    graphics.FillEllipse(Brushes.Red, ballPosX, ballPosY, ballWidth,
ballHeight);

    // ใส่กรอบสีดำให้กับลูกบอล
    graphics.DrawEllipse(Pens.Black, ballPosX, ballPosY, ballWidth,
ballHeight);
}

// ฟังก์ชัน PongpanLaowaphong เป็นฟังก์ชันในการสร้างชื่อและรหัสนักศึกษา
private void PongpanLaowaphong(Graphics graphics)
{
    // เขียนข้อความ "Pongpan Laowaphong [66543206019-2]"
    string message = "Pongpan Laowaphong [66543206019-2]";
    Font font = new Font("Arial", 24, FontStyle.Bold);
    SizeF textSize = graphics.MeasureString(message, font);

```

```

        // กำหนดตำแหน่งให้กับตัวหนังสือ (กำหนดไว้ซ้ายบน)
        float x = (this.ClientSize.Width - textSize.Width) / 40;
        float y = (this.ClientSize.Height - textSize.Height) / 25;

        graphics.DrawString(message, font, Brushes.Black, x, y);
    }

    // ฟังก์ชัน PaintPaddle เป็นฟังก์ชันในการสร้างแป้นสะท้อน
    private void PaintPaddle(Graphics graphics)
    {
        // วาดแป้นขึ้นมาให้เป็นสีน้ำเงิน
        graphics.FillRectangle(Brushes.Blue, paddlePosX, paddlePosY,
paddleWidth, paddleHeight);
    }

    // ฟังก์ชัน MoveBall เป็นฟังก์ชันในการสร้างการเคลื่อนไหวและการสะท้อนให้ลูกบอล
    private void MoveBall(object sender, EventArgs e)
    {
        // อัปเดตตำแหน่งของลูกบอล
        ballPosX += moveStepX;
        ballPosY += moveStepY;

        // ตรวจสอบการชนของวัตถุกับขอบหน้าจอ และสะท้อนกลับไปในทิศทางตรงกันข้าม
        if (ballPosX < 0 || ballPosX + ballWidth > this.ClientSize.Width)
        {
            moveStepX = -moveStepX;
        }

        if (ballPosY < 0 || ballPosY + ballHeight > this.ClientSize.Height)
        {
            moveStepY = -moveStepY;
        }

        // ตรวจสอบการชนของวัตถุกับแป้น และสะท้อนกลับไปในทิศทางตรงกันข้าม
        if (ballPosX + ballWidth > paddlePosX && ballPosX < paddlePosX +
paddleWidth &&
        ballPosY + ballHeight > paddlePosY && ballPosY < paddlePosY +
paddleHeight)
        {
            moveStepY = -moveStepY;
            moveStepX = -moveStepX;
        }

        // วาดฟอร์มใหม่ขึ้นมา
        this.Invalidate();
    }
}

```



```
// ฟังก์ชัน OnKeyDown เป็นฟังก์ชันในการกำหนดปุ่มต่างๆ ในการเคลื่อนไหว
private void OnKeyDown(object sender, KeyEventArgs e)
{
    // เงื่อนไขในการตรวจสอบการกดปุ่ม ESC
    if (e.KeyCode == Keys.Escape)
    {
        Application.Exit(); // ออกจากแอปพลิเคชัน
        return; // ส่งค่ากลับเพื่อจบการทำงานโปรแกรม
    }

    // เงื่อนไขการควบคุมการเคลื่อนไหวของแป้นโดยใช้ปุ่มลูกศรซ้าย และปุ่มลูกศรขวา สำหรับ
    การเคลื่อนที่ซ้ายและขวา
    if (e.KeyCode == Keys.Left)
    {
        isLeftPressed = true;
        paddleTimer.Start();
    }
    else if (e.KeyCode == Keys.Right)
    {
        isRightPressed = true;
        paddleTimer.Start();
    }

    // เงื่อนไขการควบคุมการเคลื่อนไหวของแป้นโดยใช้ปุ่มลูกศรบน และปุ่มลูกศรล่าง สำหรับ
    การเคลื่อนที่บนและล่าง
    if (e.KeyCode == Keys.Up)
    {
        isUpPressed = true;
        paddleTimer.Start();
    }
    else if (e.KeyCode == Keys.Down)
    {
        isDownPressed = true;
        paddleTimer.Start();
    }
}

// ฟังก์ชัน OnKeyUp เป็นฟังก์ชันในการกำหนดปุ่มต่างๆ ในการเคลื่อนไหว
private void OnKeyUp(object sender, KeyEventArgs e)
{
    // เงื่อนไขในการตรวจสอบว่าให้แป้นหยุดเคลื่อนไหว ถ้าหากปล่อยปุ่มกดของลูกศร
    if (e.KeyCode == Keys.Left)
    {
        isLeftPressed = false;
        if (!isRightPressed)
        {
            paddleTimer.Stop();
        }
    }
}
```

```

else if (e.KeyCode == Keys.Right)
{
    isRightPressed = false;
    if (!isLeftPressed)
    {
        paddleTimer.Stop();
    }
}
else if (e.KeyCode == Keys.Up)
{
    isUpPressed = false;
    if (!isDownPressed)
    {
        paddleTimer.Stop();
    }
}
else if (e.KeyCode == Keys.Down)
{
    isDownPressed = false;
    if (!isUpPressed)
    {
        paddleTimer.Stop();
    }
}
}

// ฟังก์ชัน MovePaddle เป็นฟังก์ชันในการสร้างการเคลื่อนไหวให้กับแป้น
private void MovePaddle(object sender, EventArgs e)
{
    // อัปเดตตำแหน่งของลูกบอลเมื่อกดปุ่มลูกศรซ้าย และปุ่มลูกศรขวา การเคลื่อนที่ซ้ายและขวา
    if (isLeftPressed && paddlePosX > 0)
    {
        paddlePosX -= paddleMoveStep;
    }
    else if (isRightPressed && paddlePosX + paddleWidth <
this.ClientSize.Width)
    {
        paddlePosX += paddleMoveStep;
    }

    // อัปเดตตำแหน่งของลูกบอลเมื่อกดปุ่มลูกศรบน และปุ่มลูกศรล่าง การเคลื่อนที่บนและล่าง
    if (isUpPressed && paddlePosY > 0)
    {
        paddlePosY -= paddleMoveStep;
    }
    else if (isDownPressed && paddlePosY + paddleWidth <
this.ClientSize.Width)
    {
        paddlePosY += paddleMoveStep;
    }

    // วาดฟอร์มใหม่ขึ้นมา
    this.Invalidate();
}

```

```
// ฟังก์ชัน Form1_Load เป็นฟังก์ชันในการกำหนดหน้าต่าง Form ของโปรแกรม
private void Form1_Load(object sender, EventArgs e)
{
    // ตั้งค่าให้หน้าต่างฟอร์มเป็นแบบเต็มหน้าจอ [full screen]

    this.FormBorderStyle = FormBorderStyle.None; // ลบขอบของหน้าจอ
    this.WindowState = FormWindowState.Maximized; // กำหนดหน้าจอของฟอร์มให้
    ใหญ่สุด
}
}
}
```

หลักการทำงานของโปรแกรม Bouncing ball (ลูกบอลเต็ง)

หลักการทำงานของโปรแกรม Bouncing ball (ลูกบอลเต็ง) คือการสร้างแอนิเมชันของบอลเต็งที่เคลื่อนที่ไปมาบนหน้าจอและสามารถควบคุมแป้นเพื่อรับลูกบอลได้ โดยใช้ฟังก์ชันจับเวลาสำหรับอัปเดตตำแหน่งของลูกบอลและแป้น พร้อมกับการตรวจจับการกดปุ่มลูกศรเพื่อควบคุมการเคลื่อนที่ของแป้น ซึ่งในหลักการทำงานของโปรแกรม Bouncing ball (ลูกบอลเต็ง) จะอธิบายโค้ดอยู่สามส่วนหลักๆที่จำเป็น คือ ในส่วนของการที่ลูกบอลมีการเคลื่อนที่ และมีการสะท้อนกลับไปที่เดิม ส่วนที่สองจะอธิบายในส่วนของการเคลื่อนที่และการกำหนดทิศทางของแป้น และส่วนที่สามคือการกำหนดหน้าจอของโปรแกรม (Form) โดยจะแบ่งการอธิบายได้ดังต่อไปนี้

1. โค้ดในส่วนฟังก์ชัน MoveBall สามารถอธิบายการทำงานคร่าวๆได้ดังนี้

```
// ฟังก์ชัน MoveBall เป็นฟังก์ชันในการสร้างการเคลื่อนไหวและการสะท้อนให้ลูกบอล
private void MoveBall(object sender, EventArgs e)
{
    // อัปเดตตำแหน่งของลูกบอล
    ballPosX += moveStepX;
    ballPosY += moveStepY;

    // ตรวจสอบการชนของวัตถุกับขอบหน้าจอ และสะท้อนกลับไปทิศทางตรงกันข้าม
    if (ballPosX < 0 || ballPosX + ballWidth > this.ClientSize.Width)
    {
        moveStepX = -moveStepX;
    }

    if (ballPosY < 0 || ballPosY + ballHeight > this.ClientSize.Height)
    {
        moveStepY = -moveStepY;
    }

    // ตรวจสอบการชนของวัตถุกับแป้น และสะท้อนกลับไปทิศทางตรงกันข้าม
    if (ballPosX + ballWidth > paddlePosX && ballPosX < paddlePosX +
paddleWidth &&
        ballPosY + ballHeight > paddlePosY && ballPosY < paddlePosY +
paddleHeight)
    {
        moveStepY = -moveStepY;
        moveStepX = -moveStepX;
    }

    // วาดฟอร์มใหม่ขึ้นมา
    this.Invalidate();
}
```

ฟังก์ชัน MoveBall เป็นฟังก์ชันที่ทำหน้าที่อัปเดตตำแหน่งของลูกบอลและตรวจสอบการชนกับขอบหน้าจอและแป้น เพื่อให้ลูกบอลเคลื่อนที่และสะท้อนได้อย่างถูกต้องตามกฎฟิสิกส์

2. โค้ดในส่วนฟังก์ชัน MovePaddle สามารถอธิบายการทำงานคร่าวๆได้ดังนี้

```
// ฟังก์ชัน MovePaddle เป็นฟังก์ชันในการสร้างการเคลื่อนไหวให้กับแป้น
private void MovePaddle(object sender, EventArgs e)
{
    // อัปเดตตำแหน่งของลูกบอลเมื่อกดปุ่มลูกศรซ้าย และปุ่มลูกศรขวา การเคลื่อนที่ซ้ายและ
    ขวา
    if (isLeftPressed && paddlePosX > 0)
    {
        paddlePosX -= paddleMoveStep;
    }
    else if (isRightPressed && paddlePosX + paddleWidth <
this.ClientSize.Width)
    {
        paddlePosX += paddleMoveStep;
    }

    // อัปเดตตำแหน่งของลูกบอลเมื่อกดปุ่มลูกศรบน และปุ่มลูกศรล่าง การเคลื่อนที่บนและ
    ล่าง
    if (isUpPressed && paddlePosY > 0)
    {
        paddlePosY -= paddleMoveStep;
    }
    else if (isDownPressed && paddlePosY + paddleWidth <
this.ClientSize.Width)
    {
        paddlePosY += paddleMoveStep;
    }

    // วาดฟอร์มใหม่ขึ้นมา
    this.Invalidate();
}
}
```

ฟังก์ชัน MovePaddle มีหน้าที่ในการอัปเดตตำแหน่งของแป้นเมื่อมีการกดปุ่มลูกศรบนแป้นพิมพ์ เพื่อให้แป้นสามารถเคลื่อนที่ไปในทิศทางที่ต้องการได้ โดยใช้การบังคับแป้นจะมีการใช้งานฟังก์ชัน OnKeyUp และฟังก์ชัน OnKeyDown

2.1 โค้ดในส่วนฟังก์ชัน OnKeyDown

```
// ฟังก์ชัน OnKeyDown เป็นฟังก์ชันในการกำหนดปุ่มต่างๆ ในการเคลื่อนไหว
private void OnKeyDown(object sender, KeyEventArgs e)
{
    // เงื่อนไขในการตรวจสอบการกดปุ่ม ESC
    if (e.KeyCode == Keys.Escape)
    {
        Application.Exit(); // ออกจากแอปพลิเคชัน
        return; // ส่งค่ากลับเพื่อจบการทำงานโปรแกรม
    }

    // เงื่อนไขการควบคุมการเคลื่อนไหวของแป้นโดยใช้ปุ่มลูกศรซ้าย และปุ่มลูกศรขวา
    สำหรับการเคลื่อนที่ซ้ายและขวา
    if (e.KeyCode == Keys.Left)
    {
        isLeftPressed = true;
        paddleTimer.Start();
    }
    else if (e.KeyCode == Keys.Right)
    {
        isRightPressed = true;
        paddleTimer.Start();
    }

    // เงื่อนไขการควบคุมการเคลื่อนไหวของแป้นโดยใช้ปุ่มลูกศรบน และปุ่มลูกศรล่าง
    สำหรับการเคลื่อนที่บนและล่าง
    if (e.KeyCode == Keys.Up)
    {
        isUpPressed = true;
        paddleTimer.Start();
    }
    else if (e.KeyCode == Keys.Down)
    {
        isDownPressed = true;
        paddleTimer.Start();
    }
}
```

2.2 โค้ดในส่วนฟังก์ชัน OnKeyUp

```
// ฟังก์ชัน OnKeyUp เป็นฟังก์ชันในการกำหนดปุ่มต่างๆ ในการเคลื่อนไหว
private void OnKeyUp(object sender, KeyEventArgs e)
{
    // เงื่อนไขในการตรวจสอบว่าให้แป้นหยุดเคลื่อนไหว ถ้าหากปล่อยปุ่มกดของลูกศร
    if (e.KeyCode == Keys.Left)
    {
        isLeftPressed = false;
        if (!isRightPressed)
        {
            paddleTimer.Stop();
        }
    }
    else if (e.KeyCode == Keys.Right)
    {
        isRightPressed = false;
        if (!isLeftPressed)
        {
            paddleTimer.Stop();
        }
    }
    else if (e.KeyCode == Keys.Up)
    {
        isUpPressed = false;
        if (!isDownPressed)
        {
            paddleTimer.Stop();
        }
    }
    else if (e.KeyCode == Keys.Down)
    {
        isDownPressed = false;
        if (!isUpPressed)
        {
            paddleTimer.Stop();
        }
    }
}
```

3. โค้ดในส่วนฟังก์ชัน Form1_Load สามารถอธิบายการทำงานคร่าวๆได้ดังนี้

```
// ฟังก์ชัน Form1_Load เป็นฟังก์ชันในการกำหนดหน้าต่าง Form ของโปรแกรม
private void Form1_Load(object sender, EventArgs e)
{
    // ตั้งค่าให้หน้าต่างฟอร์มเป็นแบบเต็มหน้าจอ [full screen]

    this.FormBorderStyle = FormBorderStyle.None; // ลบขอบของหน้าจอ
    this.WindowState = FormWindowState.Maximized; // กำหนดหน้าจอของฟอร์ม
    ให้ใหญ่สุด
}
```

ฟังก์ชัน Form1_Load เป็นฟังก์ชันที่ถูกเรียกเมื่อฟอร์มถูกโหลด (ในกรณีนี้คือเมื่อโปรแกรมเริ่มทำงาน) โดยทำหน้าที่ในการกำหนดหน้าต่างฟอร์มของโปรแกรมให้เป็นแบบเต็มหน้าจอ (full screen)

การใช้งานโปรแกรม Bouncing ball (ลูกบอลเต้)

1. เข้าในโฟลเดอร์ “66543206019-2_Bouncing_Ball”

Name	Date modified	Type	Size
66543206019-2_Bouncing_Ball	7/7/2024 10:41 PM	File folder	

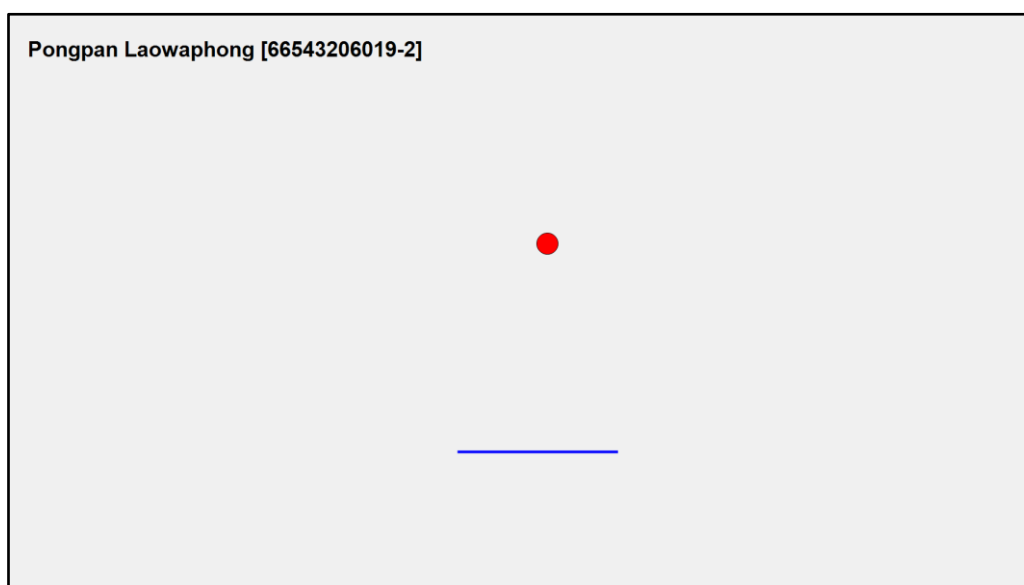
2. จากนั้นเข้าสู่โฟลเดอร์ “Application”

Name	Date modified	Type	Size
.vs	7/6/2024 7:04 PM	File folder	
Application	7/7/2024 10:42 PM	File folder	
Bouncing_ball	7/7/2024 11:48 PM	File folder	
Bouncing_ball.sln	7/5/2024 2:19 PM	Visual Studio Solut...	2 KB

3. จากนั้นเลือกไฟล์ “Bouncing_ball” ที่เป็นนามสกุล .exe [Application]

Name	Date modified	Type	Size
Bouncing_ball.deps.json	7/7/2024 10:42 PM	Adobe.AfterEffect...	1 KB
Bouncing_ball.dll	7/7/2024 10:42 PM	Application extens...	10 KB
Bouncing_ball	7/7/2024 10:42 PM	Application	140 KB
Bouncing_ball.pdb	7/7/2024 10:42 PM	Program Debug D...	14 KB
Bouncing_ball.runtimeconfig.json	7/7/2024 10:42 PM	Adobe.AfterEffect...	1 KB

4. เมื่อเข้าสู่หน้าต่างของโปรแกรม จะมีรายละเอียดการใช้งานโปรแกรม ดังนี้



- 4.1 สามารถใช้ปุ่มลูกศรขึ้น ลง ซ้าย และ ขวา ในการบังคับแป้น
- 4.2 สามารถกดปุ่ม ESC เพื่อปิดการทำงานของโปรแกรม

บรรณานุกรม

AllTech. (2561). Bouncing ball with winforms c#. สืบค้น 5 กรกฎาคม 2567,
จาก <https://youtu.be/54GnbNEvLTk?si=ABr96fcb8SWDgqjY>

ChatGPT. (-). How to make Bouncing ball with C# in Visual Studio 2022. สืบค้น 5 กรกฎาคม 2567,
จาก <https://chatgpt.com/>