TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

ĐỒ ÁN MÔN LẬP TRÌNH TRỰC QUAN
(THỰC HÀNH)

BÁO CÁO LẬP TRÌNH TRỰC QUAN

SV thực hiện : Lê Hữu Thắng
MSSV : 16521098
Lớp : IT008.I21.1
Giảng viên hướng dẫn : Huỳnh Hồ Thị Mộng Trinh

Thành phố Hồ Chí Minh – ngày 04 tháng 06 năm 2018

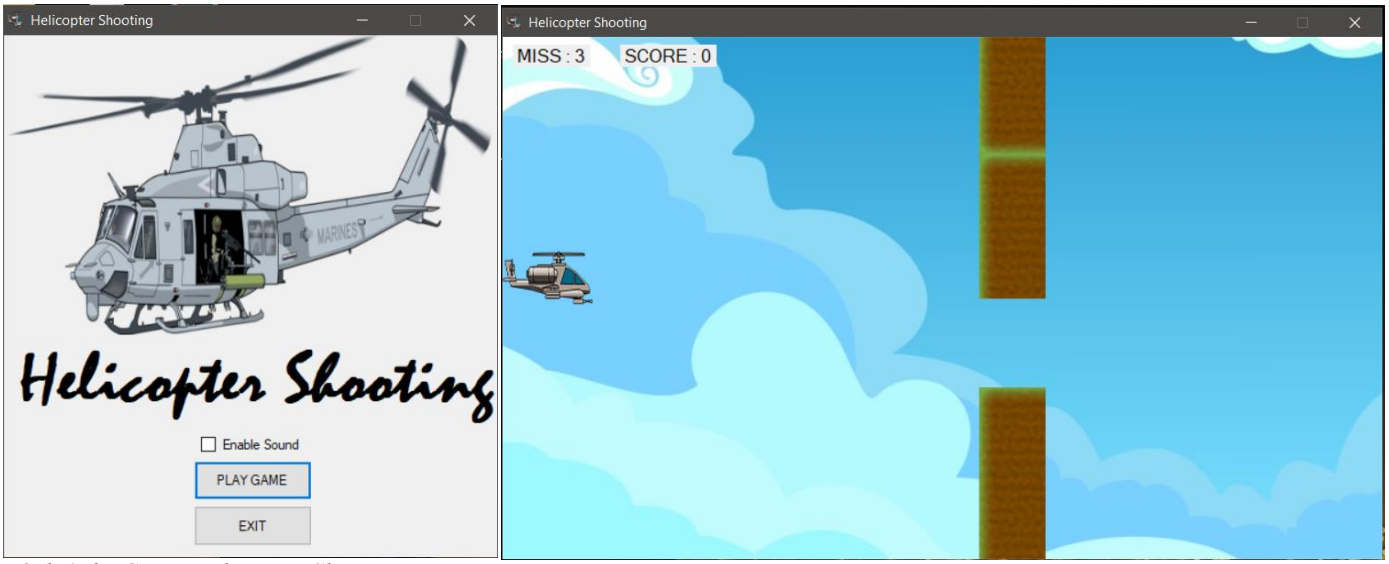
Mục lục

Lời cảm ơn :

Em xin gửi lời cảm ơn Khoa Công nghệ phần mềm cũng như cô Huỳnh Hồ Mộng Trinh đã tạo điều kiện cho em được làm đồ án môn học Lập trình trực quan.Và em cũng xin chân thành cảm ơn cô vì đã nhiệt tình hướng dẫn để giúp em có thể hoàn thành tốt đồ án

I.Giới thiệu đề tài

Ý tưởng : Dựa trên 1 game cực kì nổi tiếng về chú chim đáng ghét "Flappy Bird" do anh Nguyễn Hà Đông lập trình nên vào năm 2013.Trò chơi chính là niềm tự hào của người Việt khi có trên 50 triệu lượt tải xuống và cũng là một trong những trò chơi tiên phong về phong cách lập trình những game đơn giản mà gây ức chế, khó chịu cho người chơi.Trên ý tưởng đó,em đã tạo nên một game có ý tưởng tương tự nhưng dễ chịu hơn.Đó là game Helicopter Shooting



Hình ảnh :Game Helicopter Shooting

II.Hiện thực hóa :

1.Chuẩn bị

1.1 Tìm hiểu về GDI+ trong .NET

a.Tổng quan

-GDI là một giao diện lập trình ứng dụng (API) của Window đặc trưng cho việc vẽ các đối tượng và tương tác với các thiết bị đầu ra như màn hình và máy in

-GDI+ là một phiên bản phát triển của GDI giúp giảm độ phức tạp của GDI và làm tăng tính linh hoạt trong việc vẽ các đối tượng

-Các lớp GDI+ được cung cấp bởi .NET được bao gói lại và định nghĩa trong System,Drawing.dll

-GDI+ bao gồm 3 nhóm dịch vụ chính:

+2D vector graphic : cho phép tạo hình từ các hình cong cơ bản :đường thẳng, tròn,eclipse,…

+Imaging : làm việc với các tệp tin hình ảnh(bitmap,metafile)

+Typhography : vẽ chữ

-Một số đối tượng cơ bản của GDI+

+Color

+Point,PointF

+Rectangle,RectangleF

+Size,SizeF

b.Các lớp trong GDI+

-Ở đây chúng ta sẽ tìm hiểu về lớp Graphics.Vì đây là lớp quan trọng của GDI+.Mọi thao tác vẽ đều được thực hiện trên đối tượng Graphics của lớp này

-Bất kì lớp control nào cũng đều có thuộc tính Graphic dùng để vẽ chính nó

*Lấy đối tượng Graphic

-Chúng ta không thể khai báo theo constructor 1 cách thông thường như sau được :

```
Graphics g=new Graphics(); //Chương trình sẽ báo lỗi
```

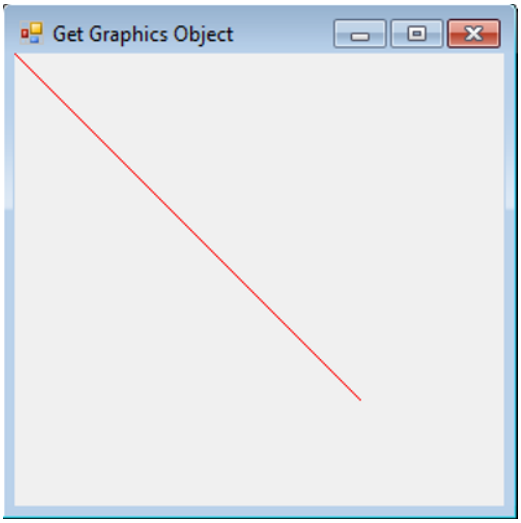
-Giải pháp: có thể lấy đối tượng Graphic từ tham số PaintEventArgs của sự kiện Paint từ Form hoặc từ phương thức OnPaint của Form

Ví dụ: Sự kiện Paint

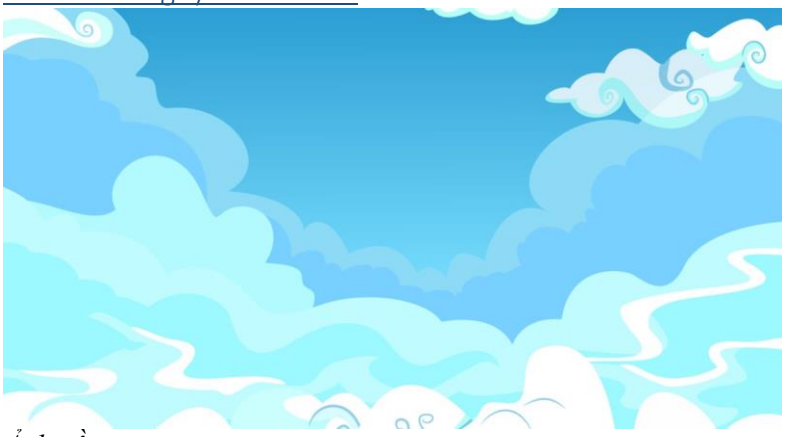
```
Private void Form1_Paint(object sender,PaintEventArgs e)
{
        Graphics g=e.Graphics;
        Pen pen=new Pen(Color.Red);
        g.DrawLine(pen,0,0,200,300);
}
```

Hoặc ví dụ : override OnPaint

```
Protected override void OnPaint(PaintEventArgs e)
{
        Graphics g=e.Graphics;
        Pen pen=new Pen(Color.Red);
        g.DrawLine(pen,0,0,200,300);
}
```



1.2 Các tài nguyên hình ảnh



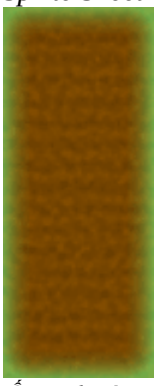
Ảnh nền

Sprite Sheet Helicopter

Sprite Sheet Enemy (type 1)

Sprite Sheet Enemy (type 2)

Sprite Sheet Enemy (type 3)



Ống chướng ngại vật

1.3.Các class trong Game :

a.Class Sprites (Class trực thăng)

*Chức năng : Là đối tượng chính mà người chơi sẽ điều khiển trong game

Thuộc tính	
Private Graphics g	
Private Image image	Ảnh hiển thị chuyển động của đối tượng trực thăng
Rectangle rect	Hình chữ nhật để vẽ đối tượng trực thăng
PictureBox bullet	Hiển thị đạn do người chơi bắn ra
Private int speed	Tốc độ di chuyển của người chơi
Private int healpoint	Số lượng kẻ địch người chơi có thể bỏ qua
Private int index	Biến số để tùy chọn ảnh hiển thị
Phương thức	
Public Sprites()	Constructor khởi tạo đối tượng trực thăng
Public void Draw(PaintEventArgs e)	Hàm vẽ đối tượng người chơi bằng graphic thông qua sự kiện Paint của form
Public void setRecUp()	Hàm di chuyển lên của trực thăng
Public void setRecDown()	Hàm di chuyển xuống của trực thăng
Public void MakeBullet(Form f1)	Hàm tạo đạn từ trực thăng được hiển thị lên form f1
Public Rectangle Rectangle()	Hàm trả về thuộc tính Rectangle
Public PictureBox getbullet()	Hàm trả về thuộc tính Bullet
Public Graphics graphics()	Hàm trả về thuộc tính Graphics g
Public int healpoint()	Hàm trả về thuộc tính healpoint

Public void healpoint(int x)	Hàm trừ healpoint mỗi khi bỏ qua 1 đối tượng Enemy

b.Class Pillar (Cột ngại vật trong Game)

*Chức năng

Là các vật cản xuất hiện gây khó khăn cho người chơi khi di chuyển

*Hiện thực hóa

Thuộc tính	
Private PictureBox pillar	Hiển thị đối tượng ống chướng ngại vật
Private Int speed	Tốc độ di chuyển của ống chướng ngại vật
Phương thức	
Public Pillar(Form f1,int x,int y)	Constructor khởi tạo đối tượng ống chướng ngại vật ở tọa độ (x,y) đưa vào hiện thị trên Form f1
Public void PillarMove()	Hàm di chuyển của ống chướng ngại vật
Public void Appearance()	Hàm khi ống chướng ngại vật vượt qua khỏi khung hình Form thì sẽ xuất hiện lại
Public int getHeigth()	Hàm trả về chiều cao của ống chướng ngại vật
Public int getWidth()	Hàm trả về chiều rộng của ống chướng ngại vật
Public int getLeft()	Hàm trả về vị trí x của ống chướng ngại vật so với trục Oxy
Public int getTop()	Hàm trả về vị trí y của ống chướng ngại vật so với trục Oxy
Public PictureBox()	Trả về đối tượng ống chướng ngại vật là 1 PictureBox
Public void Dispose()	Hảm hủy PictureBox "ống chướng ngại vật"

c.Class Enemy (Kẻ địch trong game)

*Chức năng

Là mục tiêu cho người chơi ghi điểm khi bắn hạ,và cũng là 1 vật cản

Thuộc tính	
Private Graphics g	
Private Image image	Ảnh hiện thị chuyển động của đối tượng Enemy
Private Rectangle rect	Hình chữ nhật để vẽ đối tượng trực thăng
Private int speed	Tốc độ di chuyển của Enemy
Private int type	Loại Enemy
Private int index	Biến số để tùy chọn ảnh hiển thị
Phương thức	
Public Enemy(int x,int y)	Constructor khởi tạo đối tượng Enemy ở tọa độ x,y với type ngẫu nhiên

Public void Draw(PaintEventArgs e)	Hàm vẽ đối tượng người chơi bằng graphic thông qua sự kiện Paint của form
Public void Move()	Hàm di chuyển của Enemy
Public void Appearance(pillar p)	Hàm khi Enemy vượt khỏi khung hình Form thì sẽ xuất hiện lại ở vị trí bất kỳ sau Ống chướng ngại vật
Public int getHeigth()	Hàm trả về chiều cao của rectangle vẽ đối tượng Enemy
Public int getLeft()	Hàm trả về vị trí x của rectangle vẽ đối tượng Enemy
Public Rectangle Rectangle()	Hàm trả về thuộc tính rect
Public Image Image()	Hàm trả về thuộc tính image
Public Graphics graphics()	Hàm trả về thuộc tính graphics g
Public void SetLeft(int x)	Hàm di chuyển Enemy đến tọa độ (x,0)

d.Class Game(Class quản lý trò chơi)

Thuộc tính	
Private Sprites player	Trực thăng (nhân vật chính trong game)
Private Enemy enemy	Kẻ địch trong game
Private Pillar pillar1	Ống trên 1
Private Pillar pillar2	Ống dưới 1
Private int DoChenhLech	Khoảng cách giữa 2 ống chướng ngại vật
Private bool up	Thuộc tính nhận sự kiện ấn phím lên
Private bool down	Thuộc tính nhận sự kiện ấn phím xuống
Private bool shoot	Thuộc tính nhận sự kiện ấn phím Space (bắn đạn)
Private int score	Số điểm người chơi
Private bool enable	Thuộc tính gán cờ để gọi hàm vẽ đối tượng Enemy
Private bool sound	Thuộc tính bật/tắt âm thanh Game
Phương thức	
Public Game(Form f1)	Constructor khởi tạo class Game với tham số truyền vào là 1 Form
Private void TaoPillar(Form f1)	Tạo ống chướng ngại vật được hiển thị trên form f1
Private void TaoEnemy()	Tạo đối tượng Enemy
Private bool CheckVaCham()	Hàm xét sự va chạm giữa người chơi với chướng ngại vật/Enemy
Public void SetUp(bool x)	Hàm gán cờ thuộc tính up
Public void SetDown(bool x)	Hàm gán cờ thuộc tính down
Public void SetShoot(bool x)	Hàm gán cờ thuộc tính shoot
Public void SetSound(bool x)	Hàm gán cờ thuộc tính sound
Private void PlaySound(string s)	Hàm gọi sound mở lên
Public void NewGame(Form f1,Label label1,Label label2,Timer timer1)	Hàm chính của game/Hàm tạo game mới

Public void EndGame(Form f1)	Hàm kết thúc game,dọn dẹp tài nguyên khi game kết thúc

2. Xây dựng game

2.1 Xây dựng các Class

a.Class Sprites.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    class Sprites
    {
        #region Thuoc tinh
        Graphics g;
        int speed;
        Image image;
        Rectangle rect;
        PictureBox bullet;
        int index;
        int healthpoint;
        #endregion

        #region Constructor
        public Sprites()
        {
            healthpoint = 3;
            image = Properties.Resources.heli1;
            rect = new Rectangle(0, 50, 100, 54);
            speed = 10;
            index = 0;
        }
        #endregion

        #region Draw
        public void Draw(PaintEventArgs e)
        {
            ++index;
            //Tùy chọn ảnh hiển thị của đối tượng
            if (index == 0) image = Properties.Resources.heli1;
            if (index == 1) image = Properties.Resources.heli2;
            if (index == 3) image = Properties.Resources.heli3;
            if (index == 4) image = Properties.Resources.heli4;
            if (index > 4) index = 0;
            g = e.Graphics;
            g.DrawImage(image, rect);
        }
        #endregion
```

```
        #region Move
        public void setRecUp()
        {
            rect.Y -= speed;
        }
        public void setRecDown()
        {
            rect.Y += speed;
        }
        #endregion

        #region MakeBullet
        public void MakeBullet(Form f1)
        {
            bullet = new PictureBox();
            bullet.BackColor = System.Drawing.Color.DarkOrange;
            bullet.Height = 5;
            bullet.Width = 10;
            bullet.Tag = "bullet";
            bullet.Location = new Point(rect.X+80, rect.Y + 54 / 2) ;
            f1.Controls.Add(bullet);
        }
        #endregion

        #region Get/Set cac thuoc tinh

        public Rectangle Rectangle()
        {
            return rect;
        }
        public PictureBox getbullet()
        {
            return bullet;
        }

        public int Healpoint()
        {
            return healthpoint;
        }
        public void Healpoint(int a)
        {
            healthpoint -= a;
        }

        public Graphics graphics()
        {
            return g;
        }
        #endregion
    }
}
```

b.Class Pillar.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    class Pillar
    {
        #region Thuoc tinh
        private PictureBox pillar;
        int rong = 60;
        int dai = 500;
        int speedPillar = 6;
        #endregion

        #region Constructor
        public Pillar(Form f1, int x, int y)
        {
            pillar = new PictureBox();
            pillar.BackgroundImage = Properties.Resources.pillar;
            pillar.BackColor = Color.Transparent;
            pillar.Size = new Size(rong, dai);
            pillar.Location = new Point(x, y);
            f1.Controls.Add(pillar);
        }
        #endregion

        #region Draw
        public void Appearance()
        {
            if (pillar.Left < -100)
            {
                Random rnd = new Random();
                int x = rnd.Next(600, 700);

                pillar.Left = x;
            }
        }
        #endregion

        //Ham di chuyen pillar
        public void PillarMove()
        {
            pillar.Left -= speedPillar;
        }
        #region Get cac thuoc tinh
        public int getHeight()
        {
            return pillar.Height;
        }
        public int getWidth()
        {
            return pillar.Width;
        }
        public int getLeft()
        {
```

```
            return pillar.Left;
        }
        public PictureBox getPic()
        {
            return pillar;
        }
        #endregion

        //Huy doi tuong pillar
        public void Dispose()
        {
            pillar.Dispose();
        }
    }
}
```

c.Class Enemy.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    class Enemy
    {
        #region Thuoc tinh
        Image image;
        Graphics g;
        Rectangle rect;
        int index;
        int speed =6;
        int type;
        #endregion

        #region Constructor
        public Enemy(int x,int y)
        {
            Random rnd = new Random();
            type = rnd.Next(1,4);
            System.Diagnostics.Debug.WriteLine("So duoc tao "+type);
            rect = new Rectangle(x, y, 100, 60);
            index = 0;
        }
        #endregion

        #region Draw
        public void Draw(PaintEventArgs e)
        {
            //Vẽ các đối tượng Enemy
            if (type ==1)
            {
                ++index;
                if (index == 0) image = Properties.Resources.alien1_1;
                if (index == 1) image = Properties.Resources.alien1_2;
                if (index == 2) image = Properties.Resources.alien1_3;
                if (index == 3) image = Properties.Resources.alien1_4;
```

```
            if (index > 4) index = 0;
        }
        if(type==2)
        {
            ++index;
            if (index == 0) image = Properties.Resources.alien2_1;
            if (index == 1) image = Properties.Resources.alien2_2;
            if (index == 2) image = Properties.Resources.alien2_3;
            if (index == 3) image = Properties.Resources.alien2_4;
            if (index == 4) image = Properties.Resources.alien2_5;
            if (index > 4) index = 0;
        }
        if(type==3)
        {
            ++index;
            if (index == 0) image = Properties.Resources.alien3_1;
            if (index == 1) image = Properties.Resources.alien3_2;
            if (index == 2) image = Properties.Resources.alien3_3;
            if (index == 3) image = Properties.Resources.alien3_4;
            if (index == 4) image = Properties.Resources.alien3_5;
            if (index > 4) index = 0;
        }
        g = e.Graphics;
        g.DrawImage(image, rect);

    }
    #endregion

    #region Move
    public void Move()
    {
        rect.X -= speed;
    }
    #endregion


    #region Get cac thuoc tinh
    public int getHeigth()
    {
        return rect.Height;
    }


    public int getLeft()
    {
        return rect.Left;
    }
    public Image Image()
    {
        return image;
    }

    public Rectangle Rectangle()
    {
        return rect;
    }
    public void SetLeft(int x)
    {
```

```
            rect.Location = new Point(x, 0);
        }
        public Graphics graphics()
        {
            return g;
        }
        #endregion
    }
}
```

d.Class Game.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    class Game
    {
        #region Thuoc tinh
        int score;
        Sprites player;
        Pillar pillar1;
        Pillar pillar2;
        bool enable;
        Random rnd;
        Enemy enemy;
        int DoChenhLech = 80;
        bool up;
        bool down;
        bool shoot;
        #endregion

        #region Constructor
        public Game()
        {

        }
        public Game(Form f1)
        {
            rnd = new Random();
            player = new Sprites();
            pl = new SoundPlayer();
            enable = true;
            TaoPillar(f1);
            TaoEnemy();
        }
        #endregion

        #region Tao doi tuong trong game
        void TaoPillar(Form f1)
        {

            int y = rnd.Next(-470, -150); // tạo độ dài ngẫu nhiên cho ống
            int x = 700;
            pillar1 = new Pillar(f1, x, y);
            pillar2 = new Pillar(f1, x, pillar1.getHeight() + y + DoChenhLech);
```

```
            enable = true;

        }

        void TaoEnemy()
        {
            //Tạo enemy ở vị trí sau 2 ống ngại vật tại 1 tọa độ random
            enemy = new Enemy(pillar1.getLeft() + pillar1.getWidth() + rnd.Next(200,
400), rnd.Next(100, 350) - 48);
        }
        #endregion

        #region Ve doi tuong
        public void Draw(PaintEventArgs e)
        {
            player.Draw(e);
            if (enable == true) enemy.Draw(e); //Chỉ khi thuộc tính enable=true mới được
vẽ Enemy
        }
        #endregion

        #region NewGame
        public void NewGame(Form f1,Label label1,Label label2,Timer timer1)
        {
            if (up) //Nếu nhận key từ bàn phím là mũi tên lên thì di chuyển lên
            {
                if (player.Rectangle().Y >= 0)
                    player.setRecUp();
            }
            if (down) //Nếu nhận key từ bàn phím là mũi tên lên thì di chuyển xuống

            {
                if ((player.Rectangle().Y + player.Rectangle().Height) <= f1.Size.Height
- player.Rectangle().Height)
                    player.setRecDown();
            }

            if (shoot) //Nếu nhận key từ bàn phím là Space thì bắn đạn

                player.MakeBullet(f1);
            enemy.Move();
            f1.Invalidate(); //refesh sự kiện vẽ
            //Xet di chuyen cua vien dan trong form
            foreach (Control temp in f1.Controls)
            {
                if (temp is PictureBox && temp.Tag == "bullet")
                {
                    temp.Left += 10;
                    if (temp.Left >= 800) //Dan ra khoi khung hinh
                    {
                        f1.Controls.Remove(temp);
                        temp.Dispose();
                    }
      //Đạn trúng Enemy
                    if ((temp.Left + temp.Width) >= enemy.getLeft() && temp.Location.Y >=
enemy.Rectangle().Y && temp.Location.Y <= enemy.Rectangle().Y + enemy.Rectangle().Height)
                    {
                        score++;
```

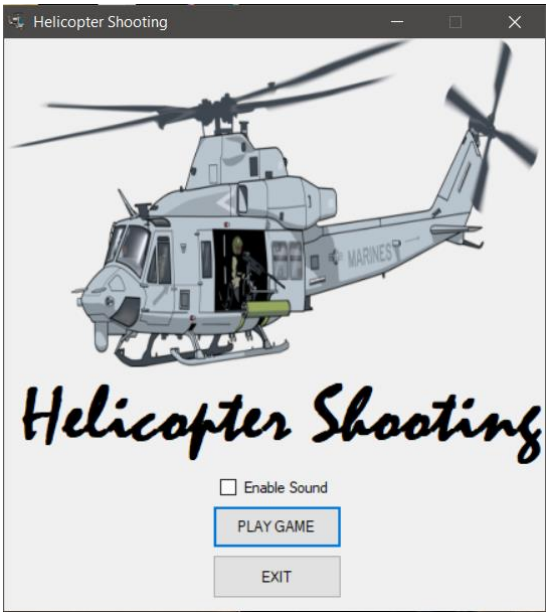
```
                    player.Healpoint(-1);
                    enemy.SetLeft(-120);
                    f1.Controls.Remove(temp);
                    temp.Dispose();//Hủy bullet
                }

            }
        }

        //Ham ong ngai vat di chuyen
        pillar1.PillarMove();
        pillar2.PillarMove();
        //Ham huy doi tuong Enemy khi di khoi khung hinh
        if (enemy.getLeft() < -100)
        {
            player.Healpoint(1);
            enemy.graphics().Dispose();
            if (enable == false) enable = true; //Enemy đã được hủy nên gán cờ true
để có thể vẽ tiếp Enemy
            TaoEnemy();
        }

        //Ham huy doi tuong ong ngai vat khi di khoi khung hinh
        if (pillar1.getLeft() < -100)
        {
            f1.Controls.Remove(pillar1.getPic());
            f1.Controls.Remove(pillar2.getPic());
            pillar1.getPic().Dispose();
            pillar2.getPic().Dispose();
            enable = false; //khi ống ngại vật ra khỏi khung hình thì ngăn chưa cho
vẽ tiếp đối tượng Enemy (vì Enemy vẫn còn trong form)
            TaoPillar(f1);
        }

        label1.Text = "MISS : " + player.Healpoint();
        label2.Text = "SCORE : " + score.ToString();
        //Xu ly va cham
        if (CheckVaCham() == true || player.Healpoint() == 0)
        {
            timer1.Stop();
            MessageBox.Show("Kết thúc game! Bạn đã đạt được số điểm là "+score);
            FormMain fm = new FormMain();
            fm.Show();
            EndGame(f1);
        }
    }
    #endregion

    #region Ket thuc game
    public void EndGame(Form f1)
    {
        f1.Dispose();
    }
    #endregion


    #region Check va cham cua cac doi tuong
    private bool CheckVaCham()
```

```
        {
            int x1 = 0;
            int x2 = x1 + player.Rectangle().Width;
            int y1 = player.Rectangle().Location.Y;
            int y2 = y1 + player.Rectangle().Height;
            //Check player va cham voi Enemy
            if (player.Rectangle().IntersectsWith(enemy.Rectangle()) &&
(player.Rectangle().Left + player.Rectangle().Width) >= (enemy.Rectangle().Left +
20)&&((player.Rectangle().Location.Y<=enemy.Rectangle().Location.Y+enemy.Rectangle().Heig
ht-
10)||(player.Rectangle().Location.Y+player.Rectangle().Height>=enemy.Rectangle().Location
.Y-20))) return true;
            //TH1 : Player vừa ở trước 2 cột
            if (x2 - 15 == pillar1.getPic().Location.X)
            {
                if (y1 <= pillar1.getPic().Location.Y + pillar1.getPic().Height || y2 >=
pillar2.getPic().Location.Y)
                    return true;
            }
            //TH2 : Player nằm giữa 2 cột
            if (x2 - 15 > pillar1.getPic().Location.X && x2 - 15 <=
pillar1.getPic().Location.X + pillar1.getPic().Width)
            {
                if (y1 <= pillar1.getPic().Location.Y + pillar1.getPic().Height || y2 >=
pillar2.getPic().Location.Y)
                    return true;
            }
            //TH3 : Play đã đi qua 1 nửa giữa 2 cột
            if (x2 - 15 > pillar1.getPic().Location.X + pillar1.getPic().Width)
            {
                if (x1 <= pillar1.getPic().Location.X + pillar1.getPic().Width)
                {
                    if (y1 <= pillar1.getPic().Location.Y + pillar1.getPic().Height || y2
- 10 >= pillar2.getPic().Location.Y)
                        return true;
                }
            }
            return false;
        }
        #endregion

        #region Gan co cho su kien an phim len/xuong/Space
        public void SetUp(bool x)
        {
            up = x;
        }
        public void SetDown(bool x)
        {
            down = x;
        }
        public void SetShoot(bool x)
        {
            shoot = x;
        }
        #endregion
        void PlaySound(string s)
        {
            pl.SoundLocation = Application.StartupPath + @"\Sounds\" + s;
```

```
            pl.Play();
        }
 //Bật tắt âm thanh
        public void SetSound(bool x)
        {
            sound = x;
        }
    }
}
```

2.2 Xây dựng Form

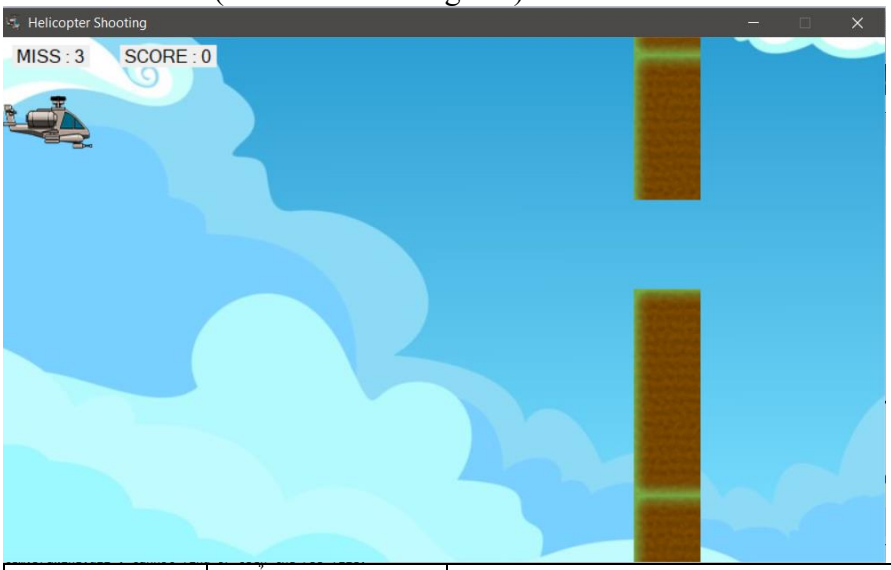
a.FormMain.cs




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    public partial class FormMain : Form
    {
        public FormMain()
        {
            InitializeComponent();
            pictureBox1.Image = Properties.Resources.HelicopterShooting;
        }

        //Hàm gọi form chính của Game
        private void button1_Click(object sender, EventArgs e)
        {
```

```
            MessageBox.Show("Nhiệm vụ của bạn là hãy lái chiếc trực thăng vượt chướng
ngại vật và tiêu diệt kẻ thù càng nhiều càng tốt \nNhấn phím mũi tên lên/xuống để di
chuyển trực thăng và nhấn phím Space để bắn đạn", "Hướng dẫn");
            this.Hide();
            MainGame g = new MainGame();
     if (checkBox1.Checked == true) g.EnableSound(true);
            g.Show();//Show form MainGame


        }


        private void FormMain_FormClosing(object sender, FormClosingEventArgs e)
        {
            Application.Exit();//Thoát ứng dụng
        }

        private void FormMain_Load(object sender, EventArgs e)
        {

        }

        //Hàm đóng Application khi nhấn button Exit
        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

b.MainGame.cs (Form chính chơi game)



Tên	Kiểu	Chức năng
Label1	Label	Thể hiện số lượt bỏ qua còn lại
Label2	Label	Thể hiện số điểm của người chơi
Timer1	Timer	Thời gian thực trong game

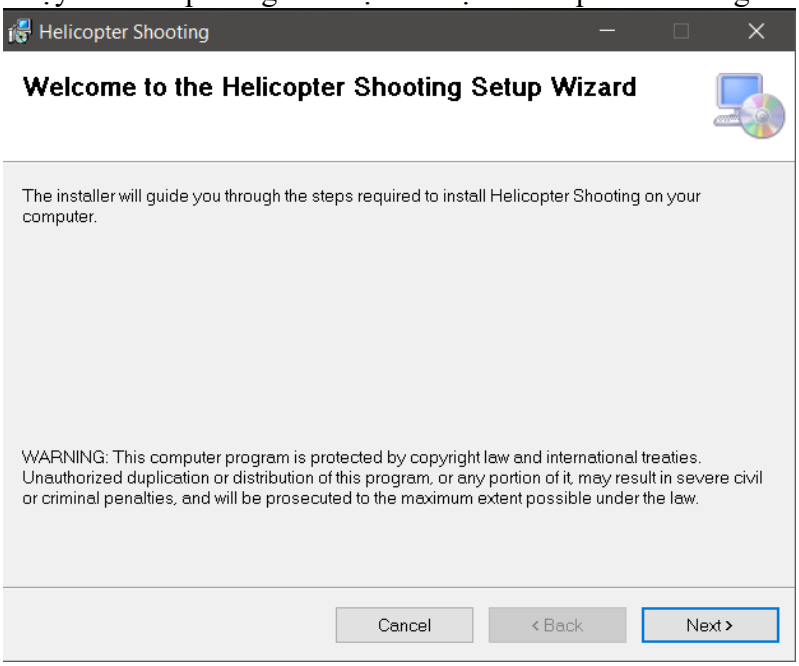
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace HelicopterShooting
{
    public partial class MainGame : Form
    {
        Game game;
        public MainGame()
        {
            InitializeComponent();
            this.Size = new Size(800, 500);
            timer1.Start();
            game = new Game(this);
            this.BackgroundImage = Properties.Resources.background;

        }

        //Chỉnh bật tắt âm thanh
        public void EnableSound(bool x)
        {
            game.SetSound(x);
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
        //Vẽ
        private void Form1_Paint(object sender, PaintEventArgs e)
        {

            game.Draw(e);
        }

        private void timer1_Tick(object sender, EventArgs e)
        {

            game.NewGame(this, label1, label2, timer1);
        }

        //Nhận sự kiện từ bàn phím để xử lý tương ứng
        private void Form1_KeyDown(object sender, KeyEventArgs e)
        {
            switch (e.KeyCode)
            {
                case Keys.Down:
                    game.SetDown(true);
                    break;
                case Keys.Up:
                    game.SetUp(true);
                    break;
```

```
                case Keys.Space:
                    game.SetShoot(true);
                    break;
                default:
                    break;
            }
        }

        private void Form1_KeyUp(object sender, KeyEventArgs e)
        {
            switch (e.KeyCode)
            {
                case Keys.Down:
                    game.SetDown(false);
                    break;
                case Keys.Up:
                    game.SetUp(false);
                    break;
                case Keys.Space:
                    game.SetShoot(false);
                    break;
                default:
                    break;
            }
        }

        private void MainGame_FormClosed(object sender, FormClosedEventArgs e)
        {


        }

        private void MainGame_FormClosing(object sender, FormClosingEventArgs e)
        {
            timer1.Stop();
            if(MessageBox.Show("Bạn có muốn thoát ra màn chính? ","Thông
báo",MessageBoxButtons.YesNo,MessageBoxIcon.Question)==DialogResult.No)
            {
                e.Cancel = true;
                timer1.Start();
            }
            else
            {
                FormMain fm = new FormMain();
                fm.Show();
            }
        }
    }
```

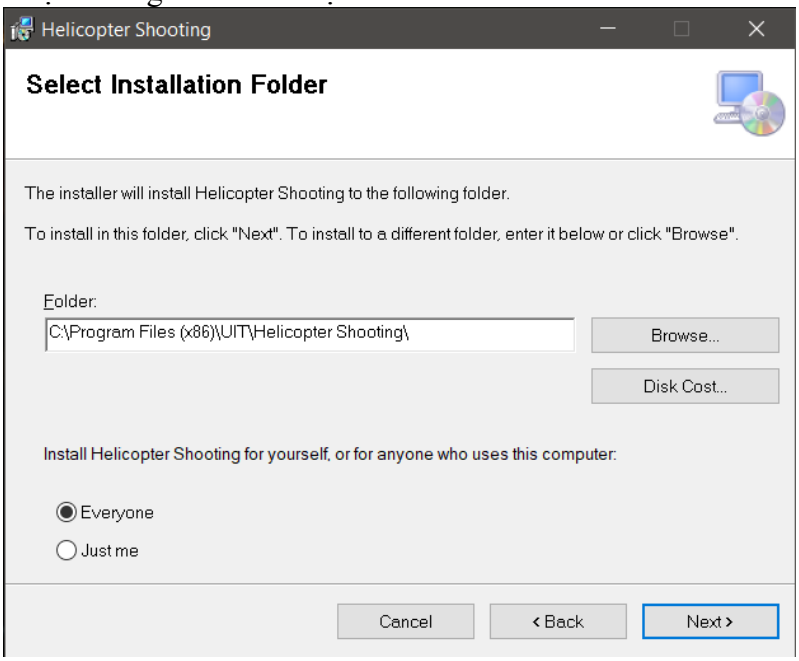
III.Cài đặt - Cách chơi

1/Cài đặt

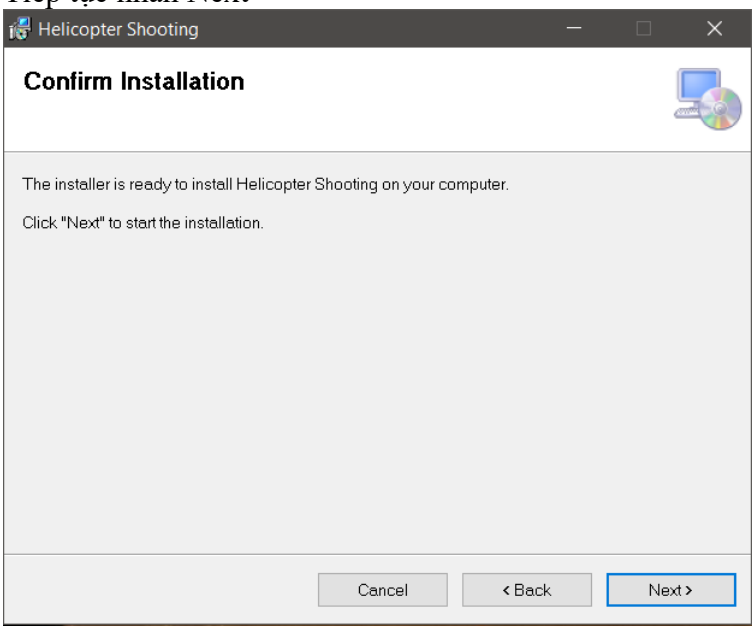
Chạy file Setup trong thư mục cài đặt Helicopter Shooting.Nhấn next để tiếp tục



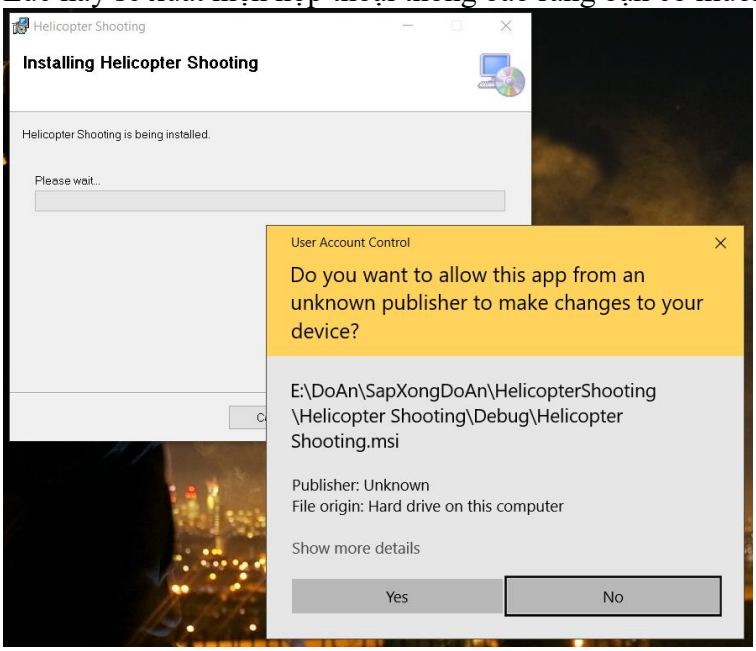
Chọn đường dẫn để cài đặt Game.Nhấn next



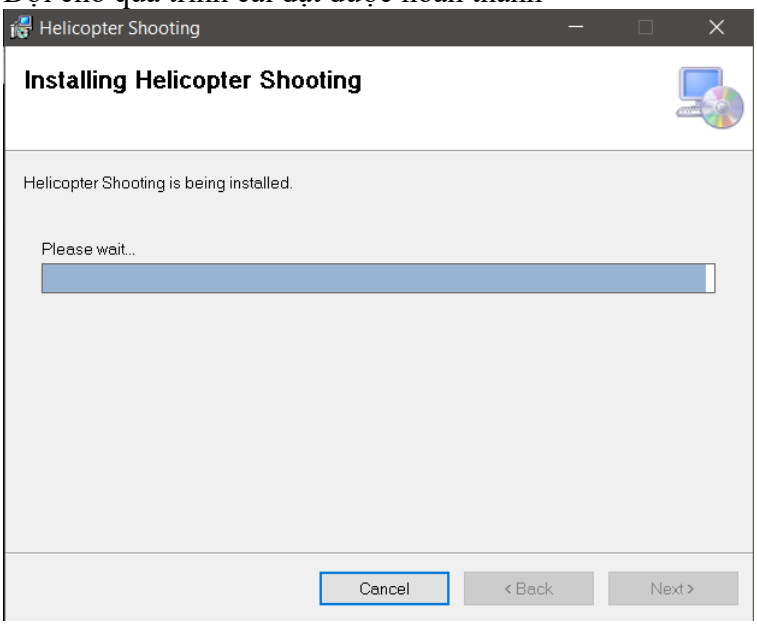
Tiếp tục nhấn Next



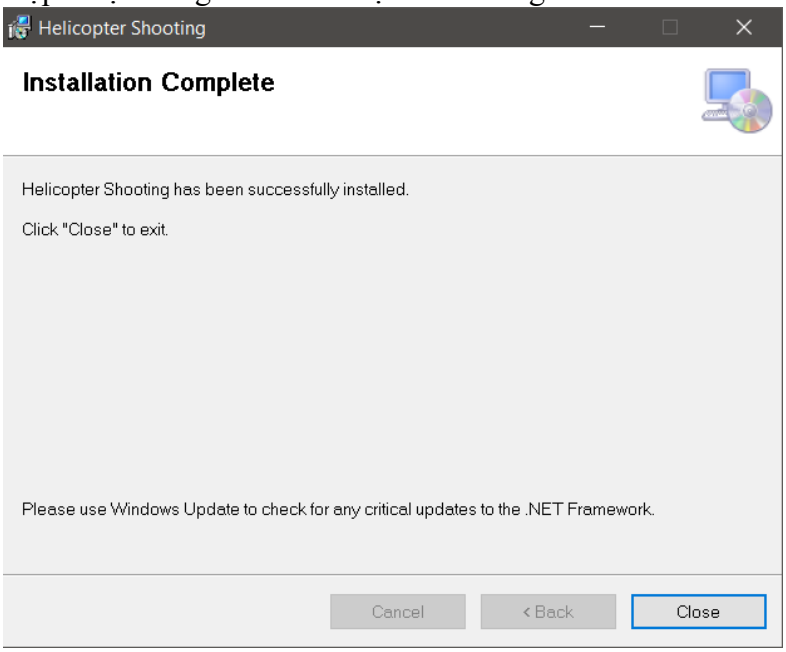
Lúc này sẽ xuất hiện hộp thoại thông báo rằng bạn có muốn thực hiện cài đặt.Nhấn Yes



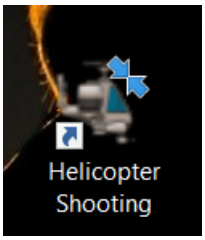
Đợi cho quá trình cài đặt được hoàn thành



Hộp thoại thông báo đã cài đặt thành công



Lúc này trên Desktop (màn hình chính) sẽ xuất hiện Short cut của trò chơi.Các bạn chỉ việc nhấn vào biểu tượng là đã có thể chơi được



Chúc các bạn thành công!

2/Cách chơi

Nhiệm vụ : Người chơi sẽ điều khiển chiếc trực thăng vượt qua các chướng ngại vật và cố gắng hạ càng nhiều kẻ địch càng tốt
Hướng dẫn : Bấm phím mũi tên lên/xuống để điều khiển trực thăng bay lên/xuống và nhấn Space để bắn đạn

Demo:
https://www.youtube.com/watch?v=3a1ctf-Q-aQ&feature=youtu.be

IV.Tài liệu tham khảo

[1] Giáo trình Lập trình hướng đối tượng - Đại học Quốc gia TPHCM – trường Đại học Công Nghệ Thông Tin – Biên soạn : Ths.Trương Hải Bằng
[2] Giáo trình Lập trình trực quan-Slide chương 5 :Lập trình GDI+
[3] https://www.mooict.com/c-tutorial-create-a-helicopter-flying-and-shooting-game-in-visual-studio/ - lần cuối truy cập : 3/5/2018