CODE GAME RẮN SĂN MỒI

<http://nhocpun.tk/>

//Đọc kỹ trc khi copy code nhé!

//Bài này tạo 3 Class nên bạn chú ý cả copy nhầm sẽ ko chạy đc !

//Bạn them 1 Class có tên là Class CDiemKT nhé!

//Chú ý: Bạn nên đặt tên trc Class CDiemKT (Chỉ Class này thôi nhé!, còn 2 Class còn lại bạn muốn đặt sao đặt (Mình đặt là GameRanSanMoi) ) là:

namespace HuongDoiTuong

//Còn đây là code của Class đó!

//Sau khi đã thêm xong 3 Class hoàn chỉnh bạn hãy thêm vào đầu 2 Class CSnack và Program 2 dòng này nữa nhé !

using System.Threading.Tasks;

using System.Threading;

using System.Threading;

//Cái nào có rồi thì bỏ đi nhé!

//Ok Chúc bạn thành công !!

public class CDiemKT

{

//Khai báo lớp random, sử dụng

//tạo số ngẫu nhiên

Random r = new Random();

private int x, y;

char c;

//Nhóm này chỉ dành cho khởi tạo và cập nhật

#region Nhóm kiểm tra ràng buộc

/// <summary>

/// Kiểm tra tính hợp lệ của toạ độ x

/// </summary>

/// <param name="xx">giá trị toạ độ x</param>

/// <returns>true/fasle</returns>

public bool KiemTraX(int xx)

{

return ((xx >= 0) && (xx < Console.WindowWidth));

}

/// <summary>

/// Kiểm tra tính hợp lệ của toạ độ y

/// </summary>

/// <param name="yy">Giá trị toạ độ y</param>

/// <returns>true/false</returns>

public bool KiemTraY(int yy)

{

return ((yy >= 0) && (yy < Console.WindowHeight));

}

#endregion

#region Nhóm khởi tạo

/// <summary>

/// Nhập toạ độ x,y, kí tự c với hàm tạo sẵn

/// </summary>

public void Nhap()

{

do

{

//Nhập x, y với do while, rời ra, kiểm tra từng cái

try

{

Console.Write("Nhap toa do x: ");

x = int.Parse(Console.ReadLine());

}

catch

{

Console.WriteLine("Nhap khong hop le "

+ ",nhap lai");

}

} while (!KiemTraX(x));

do

{

//Nhập x, y với do while, rời ra, kiểm tra từng cái

try

{

Console.Write("Nhap toa do Y: ");

y = int.Parse(Console.ReadLine());

}

catch

{

Console.WriteLine("Nhap khong hop le "

+ ",nhap lai");

}

} while (!KiemTraY(y));

bool hl = false;

do

{

try

{

Console.Write("Nhap k i tu C: ");

c = Convert.ToChar(Console.ReadLine());

hl = true;

}

catch

{

Console.WriteLine("Nhap khong hop le");

hl = true;

}

} while (!hl);

}

/// <summary>

/// Tạo thông số mặc định

/// </summary>

public void KhoiTao()

{

x = 0;

y = 2;

c = 'A';

}

/// <summary>

/// Nhập toạ độ x,y kí tự c với tham số truyền vào

/// </summary>

/// <param name="xx">toạ độ x</param>

/// <param name="yy">toạ độ y</param>

/// <param name="cc">kí tự c</param>

/// <returns></returns>

public bool KhoiTao(int xx, int yy, char cc)

{

if (!KiemTraX(xx) || !KiemTraY(yy))

return false;

x = xx;

y = yy;

c = cc;

return true;

}

/// <summary>

/// Phát sinh ngẫu nhiên x,y, char

/// </summary>

public void PhatSinh()

{

//r là lớp Random

//x=[0..WindowWidth - 1]

x = r.Next(Console.WindowWidth);

//y=[0..WindowHeight - 1]

y = r.Next(Console.WindowHeight);

// lấy kí tự trong phạm vi 33..254

// theo mã ascii

c = Convert.ToChar(r.Next(33, 254));

}

#endregion

#region Nhóm cập nhật

//Trực tiếp

/// <summary>

/// Cập nhật toạ độ x

/// </summary>

/// <param name="xx">giá trị mới cần cập nhật</param>

/// <returns>true nếu cập nhật thành công</returns>

public bool CapNhatX(int xx)

{

if (!KiemTraX(xx))

return false;

x = xx;

return true;

}

/// <summary>

/// Cập nhật giá trị mới cho toạ độ y

/// </summary>

/// <param name="yy">Giá trị toạ độ y mới</param>

/// <returns></returns>

public bool CapNhatY(int yy)

{

if (!KiemTraY(yy))

return false;

y = yy;

return true;

}

/// <summary>

/// Cập nhật kí tự khác

/// </summary>

/// <param name="c">kí tự cần cập nhật</param>

/// <returns></returns>

public bool CapNhatCh(char cc)

{

try

{

c = cc;

return true;

}

catch

{

return false;

}

}

//Gián tiếp

/// <summary>

/// Dịch qua phải k kí tự

/// </summary>

/// <param name="k">số kí tự muốn dịch qua thêm</param>

/// <returns></returns>

public bool DichPhai()

{

return CapNhatX(x + 1);

}

public bool DichTrai()

{

return CapNhatX(x - 1);

}

public bool DichLen()

{

return CapNhatY(y - 1);

}

public bool DichXuong()

{

return CapNhatY(y + 1);

}

public bool DichXienPhaiXuong()

{

if (!CapNhatX(x + 1) || !CapNhatY(y + 1))

return false;

return true;

}

public bool DichXienPhaiLen()

{

if (!CapNhatX(x + 1) || !CapNhatY(y - 1))

return false;

return true;

}

public bool DichXienTraiXuong()

{

if (!CapNhatX(x - 1) || !CapNhatY(y + 1))

return false;

return true;

}

public bool DichXienTraiLen()

{

if (!CapNhatX(x - 1) || !CapNhatY(y - 1))

return false;

return true;

}

public bool DichPhai(int k)

{

return CapNhatX(x + k);

}

public bool DichTrai(int k)

{

return CapNhatX(x - k);

}

public bool DichLen(int k)

{

return CapNhatY(y - k);

}

public bool DichXuong(int k)

{

return CapNhatY(y + k);

}

public bool DichXienPhaiXuong(int k)

{

if (!CapNhatX(x + k) || !CapNhatY(y + k))

return false;

return true;

}

public bool DichXienPhaiLen(int k)

{

if (!CapNhatX(x + k) || !CapNhatY(y - k))

return false;

return true;

}

public bool DichXienTraiXuong(int k)

{

if (!CapNhatX(x - k) || !CapNhatY(y + k))

return false;

return true;

}

public bool DichXienTraiLen(int k)

{

if (!CapNhatX(x - k) || !CapNhatY(y - k))

return false;

return true;

}

#endregion

//Từ thông tin ban đầu đưa ra được thông tin mới

#region Nhóm xử lý tính toán

public double KhoangCach(CDiemKT M)

{

return Math.Sqrt(Math.Pow((double)(x - M.x), 2) + Math.Pow((double)(y - M.y), 2));

}

public int KhoangCachX(CDiemKT M)

{

return (Math.Abs(x - M.x));

}

public int KhoangCachY(CDiemKT M)

{

return (Math.Abs(y - M.y));

}

public void Xoa()

{

Console.SetCursorPosition(x, y);

Console.Write(" ");

}

#endregion

#region Nhóm cung cấp thông tin

/// <summary>

/// Xuất thông tin x,y, char

/// </summary>

public void Xuat()

{

Console.SetCursorPosition(x, y);

Console.Write(c);

}

public int GiaTriX()

{

return x;

}

public int GiaTriY()

{

return y;

}

public char GiaTriCh()

{

return c;

}

#endregion

#region Constructor

public CDiemKT()

{

PhatSinh();

}

public CDiemKT(CDiemKT d)

{

this.x = d.x;

this.y = d.y;

this.c = d.c;

}

//Phương thức thiết lập có tham số

public CDiemKT(int xx, int yy, char cc)

{

if (!KiemTraX(xx) || !KiemTraY(yy))//Không hợp lệ

KhoiTao();//Khởi tạo mặc định

else

{

this.x = xx;

this.y = yy;

this.c = cc;

}

}

public CDiemKT(int xx, int yy)

{

if (!KiemTraX(xx) || !KiemTraY(yy))//Không hợp lệ

KhoiTao();//Khởi tạo mặc định

else

{

this.x = xx;

this.y = yy;

this.c = Convert.ToChar(r.Next(65, 91));

}

}

public CDiemKT(char cc)

{

PhatSinh();

this.c = cc;

}

#endregion

#region Property

public int X

{

get { return x; }

set

{

if (KiemTraX(value))

x = value;

}

}

public int Y

{

get { return y; }

set

{

if (KiemTraY(value))

y = value;

}

}

public char C

{

get { return c; }

set { c = value; }

}

#endregion

#region Operator

public static bool operator ==(CDiemKT d1, CDiemKT d2)

{

return ((d1.X == d2.X) && (d1.Y == d2.Y));

}

public static bool operator !=(CDiemKT d1, CDiemKT d2)

{

return !((d1.X == d2.X) && (d1.Y == d2.Y));

}

public override bool Equals(object obj)

{

return base.Equals(obj);

}

public override int GetHashCode()

{

return base.GetHashCode();

}

public override string ToString()

{

string ss = "X= " + this.x.ToString() +

" Y= " + this.y.ToString() + " C= " + this.c.ToString();

return ss;

}

#endregion

}

//Bạn thêm vào 1 Class CSnack

//Code của e nó đây!

class CSnack

{

enum eHuongDi { len, xuong, trai, phai };

private List<CDiemKT> lstConRan; //lưu CDiemKT cua con ran

eHuongDi huongDi = eHuongDi.phai; //hướng ban đầu là dich phải

private List<CDiemKT> lstMoi; // lưu mồi của con rắn

#region Constructor

public CSnack()

{

this.lstConRan = new List<CDiemKT>();

//Tạo con rắn ban đầu 5 kí tự

for (int i = 5; i < 20; i++)

{

CDiemKT cKT = new CDiemKT(5, i, '\*');

lstConRan.Add(cKT);

cKT.Xuat();

}

lstMoi = new List<CDiemKT>();

}

#endregion

#region Property

public List<CDiemKT> LstConRan

{

get { return this.lstConRan; }

set { this.lstConRan = value; }

}

#endregion

#region other method

public void DichChuyenConRan()

{

bool song = true;

while (true)

{

while (!Console.KeyAvailable)

{

//Nếu ko nhấn phím thì dịch chuyển con rắn mặc định theo hướng đã có

CDiemKT dauRanMoi = new CDiemKT(lstConRan.Last());

switch (huongDi)

{

case eHuongDi.phai:

song = dauRanMoi.DichPhai();

break;

case eHuongDi.trai:

song = dauRanMoi.DichTrai();

break;

case eHuongDi.len:

song = dauRanMoi.DichLen();

break;

case eHuongDi.xuong:

song = dauRanMoi.DichXuong();

break;

}

//In đầu rắn ở vị trí mới;

dauRanMoi.Xuat();

//Nếu nó tự ăn nó

if (TimDiemKT(dauRanMoi, lstConRan) != -1)

{

song = false;

}

//Nếu chết

if (!song)

{

Console.Clear();

Console.WriteLine("Game over");

return;

}

//Thêm đầu rắn mới vào cuối list

lstConRan.Add(dauRanMoi);

//Khi ăn

int vitri = TimDiemKT(dauRanMoi, lstMoi);

//Ăn được thì ko cần xoá đuôi rắn, coi như mồi đã được ăn

if (vitri != -1)

{

//xoá mồi đã bị ăn trong list mồi

lstMoi.RemoveAt(vitri);

}

//Ăn ko duoc thi xoá đuôi

else

{

lstConRan.First().Xoa();

lstConRan.RemoveAt(0);

}

//tốc độ chạy

Thread.Sleep(100);

}

//Nếu có nhấn phím, thì lấy phím đưa vào k

ConsoleKeyInfo k = Console.ReadKey();

switch (k.Key)

{

//Tuỳ theo phím được nhấn mà thay đổi hướng đi

case ConsoleKey.UpArrow:

if (huongDi != eHuongDi.xuong)

huongDi = eHuongDi.len;

break;

case ConsoleKey.DownArrow:

if (huongDi != eHuongDi.len)

huongDi = eHuongDi.xuong;

break;

case ConsoleKey.LeftArrow:

if (huongDi != eHuongDi.phai)

huongDi = eHuongDi.trai;

break;

case ConsoleKey.RightArrow:

if (huongDi != eHuongDi.trai)

huongDi = eHuongDi.phai;

break;

}

}

}

public void TaoMoi()

{

CDiemKT moi;

int vitri;

while (true)

{

//Tránh việc mồi bị tạo trùng vào thân con rắn

do

{

moi = new CDiemKT('@');

vitri = TimDiemKT(moi, lstConRan);

if (vitri == -1)

{

//ko trùng thì môi moi duoc tạo va add vao list

lstMoi.Add(moi);

moi.Xuat();

}

//lặp cho đên khi nao tao duoc diem ko trung

} while (vitri != -1);

Thread.Sleep(2000);

}

}

/// <summary>

/// Tìm vi tri của 1 diem ki tu trong list diem kit tu

/// </summary>

/// <param name="cKT">diem ki tu mau</param>

/// <param name="lstDiemKT">;ist diem ki tu</param>

/// <returns>vitri neu tim thay, -1 neu ko</returns>

public int TimDiemKT(CDiemKT cKT, List<CDiemKT> lstDiemKT)

{

for (int i = 0; i < lstDiemKT.Count; i++)

{

if (cKT == lstDiemKT[i])//operator so sánh trong class CDiemKT

return i;

}

return -1;

}

public void Run()

{

Thread tTaoMoi = new Thread(TaoMoi);

tTaoMoi.IsBackground = true;//DichChuyenConRan chạy trên Thread foreground nếu DichChuyenConRanDung

//thì tạo mội ở Thread Background cũng dừng theo

tTaoMoi.Start();

DichChuyenConRan();

}

#endregion

}

//Code của phần Class Program nhé!

class Program

{

static void Main(string[] args)

{

Console.CursorVisible = false;

CSnack game = new CSnack();

game.Run();

}

}