Problem 13

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Common;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Problem13

{

public class CActorBlock

{

public int X, Y, W, H, position;

public Color cl;

}

public class CActorColumn

{

public int X, Y, W, H;

public Color cl;

}

public partial class Form1 : Form

{

List<CActorColumn> LColumns = new List<CActorColumn>();

List<CActorBlock> Blocks1 = new List<CActorBlock>();

List<CActorBlock> Blocks2 = new List<CActorBlock>();

List<CActorBlock> Blocks3 = new List<CActorBlock>();

List<CActorBlock> NewList = null;

bool isDrag = false;

int xOld = -1, yOld = -1;

int XmoveBlock = -1;

int YmoveBlock = -1;

int blockList = -1;

int newblockList = -1;

Bitmap off;

public Form1()

{

this.WindowState = FormWindowState.Maximized;

this.Paint += new PaintEventHandler(Form1\_Paint);

this.Load += new EventHandler(Form1\_Load);

this.MouseDown += Form1\_MouseDown;

this.MouseMove += Form1\_MouseMove;

this.MouseUp += Form1\_MouseUp;

}

private void Form1\_MouseUp(object sender, MouseEventArgs e)

{

if (blockList != -1)

{

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

{

if (e.X > LColumns[i].X && e.X < (LColumns[i].X + LColumns[i].W) && e.Y > LColumns[i].Y && e.Y < (LColumns[i].Y + LColumns[i].H))

{

newblockList = i;

break;

}

}

if (newblockList == 0)

{

NewList = Blocks1;

}

else if (newblockList == 1)

{

NewList = Blocks2;

}

else if (newblockList == 2)

{

NewList = Blocks3;

}

if (blockList == 0)

{

if (newblockList == -1 || newblockList == blockList) //block didnot move or get dragged

{

Blocks1[Blocks1.Count - 1].X = XmoveBlock;

Blocks1[Blocks1.Count - 1].Y = YmoveBlock;

}

else

{

moveBlock(Blocks1, NewList, newblockList, Blocks1[Blocks1.Count - 1]);

}

}

else if (blockList == 1)

{

if (newblockList == -1 || newblockList == blockList)

{

Blocks2[Blocks2.Count - 1].X = XmoveBlock;

Blocks2[Blocks2.Count - 1].Y = YmoveBlock;

}

else

{

moveBlock(Blocks2, NewList, newblockList, Blocks2[Blocks2.Count - 1]);

}

}

else if (blockList == 2)

{

if (newblockList == -1 || newblockList == blockList)

{

Blocks3[Blocks3.Count - 1].X = XmoveBlock;

Blocks3[Blocks3.Count - 1].Y = YmoveBlock;

}

else

{

moveBlock(Blocks3, NewList, newblockList, Blocks3[Blocks3.Count - 1]);

}

}

}

isDrag = false;

blockList = -1;

DrawDubb(this.CreateGraphics());

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

if (isDrag)

{

int dx = e.X - xOld;

int dy = e.Y - yOld;

if (blockList != -1)

{

if (blockList == 0)

{

Blocks1[Blocks1.Count - 1].X += dx;

Blocks1[Blocks1.Count - 1].Y += dy;

}

else if (blockList == 1)

{

Blocks2[Blocks2.Count - 1].X += dx;

Blocks2[Blocks2.Count - 1].Y += dy;

}

else if (blockList == 2)

{

Blocks3[Blocks3.Count - 1].X += dx;

Blocks3[Blocks3.Count - 1].Y += dy;

}

xOld = e.X;

yOld = e.Y;

}

DrawDubb(this.CreateGraphics());

}

}

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

blockList = -1;

whichBlockList(e.X, e.Y);

}

void whichBlockList(int xMouse, int yMouse)

{

if (Blocks1.Count > 0)

{

if (xMouse > Blocks1[Blocks1.Count - 1].X && xMouse < (Blocks1[Blocks1.Count - 1].X + Blocks1[Blocks1.Count - 1].W) && yMouse > Blocks1[Blocks1.Count - 1].Y && yMouse < (Blocks1[Blocks1.Count - 1].Y + Blocks1[Blocks1.Count - 1].H))

{

isDrag = true;

xOld = xMouse;

yOld = yMouse;

XmoveBlock = Blocks1[Blocks1.Count - 1].X;

YmoveBlock = Blocks1[Blocks1.Count - 1].Y;

blockList = 0;

}

}

if (Blocks2.Count > 0)

{

if (xMouse > Blocks2[Blocks2.Count - 1].X && xMouse < (Blocks2[Blocks2.Count - 1].X + Blocks2[Blocks2.Count - 1].W) && yMouse > Blocks2[Blocks2.Count - 1].Y && yMouse < (Blocks2[Blocks2.Count - 1].Y + Blocks2[Blocks2.Count - 1].H))

{

isDrag = true;

xOld = xMouse;

yOld = yMouse;

XmoveBlock = Blocks2[Blocks2.Count - 1].X;

YmoveBlock = Blocks2[Blocks2.Count - 1].Y;

blockList = 1;

}

}

if (Blocks3.Count > 0)

{

if (xMouse > Blocks3[Blocks3.Count - 1].X && xMouse < (Blocks3[Blocks3.Count - 1].X + Blocks3[Blocks3.Count - 1].W) && yMouse > Blocks3[Blocks3.Count - 1].Y && yMouse < (Blocks3[Blocks3.Count - 1].Y + Blocks3[Blocks3.Count - 1].H))

{

isDrag = true;

xOld = xMouse;

yOld = yMouse;

XmoveBlock = Blocks3[Blocks3.Count - 1].X;

YmoveBlock = Blocks3[Blocks3.Count - 1].Y;

blockList = 2;

}

}

}

void moveBlock(List<CActorBlock> previousColumn, List<CActorBlock> newColumn, int newList, CActorBlock block)

{

if (newColumn.Count == 0)

{

block.X = LColumns[newList].X + LColumns[newList].W / 2 - block.W / 2;

block.Y = LColumns[newList].Y + LColumns[newList].H - block.H;

newColumn.Add(block);

previousColumn.RemoveAt(previousColumn.Count - 1); //remove last block dragged

}

else

{

if (block.position > newColumn[newColumn.Count - 1].position)

{

previousColumn[previousColumn.Count - 1].X = XmoveBlock;

previousColumn[previousColumn.Count - 1].Y = YmoveBlock;

}

else

{

block.Y = newColumn[newColumn.Count - 1].Y - block.H;

block.X = LColumns[newList].X + LColumns[newList].W / 2 - block.W / 2;

newColumn.Add(block);

previousColumn.RemoveAt(previousColumn.Count - 1); //remove last block dragged

}

}

}

void CreateColumns()

{

int xColumn = 150;

int yColumn = this.ClientSize.Height - 520;

for (int i = 0; i < 3; i++)

{

CActorColumn pnn = new CActorColumn();

pnn.X = xColumn;

pnn.Y = yColumn;

pnn.W = 25;

pnn.H = 500;

pnn.cl = Color.Gray;

LColumns.Add(pnn);

xColumn += 300;

}

}

void CreateBlocks()

{

int blockWidth = 200;

int blockHeight = 30;

int pos = 9;

int xBlock = 62;

int yBlock = this.ClientSize.Height - 20 - blockHeight;

for (int i = 0; i < 10; i++, pos--)

{

CActorBlock pnn = new CActorBlock();

pnn.X = xBlock;

pnn.Y = yBlock;

pnn.cl = Color.Yellow;

pnn.W = blockWidth;

pnn.H = blockHeight;

pnn.position = pos;

yBlock -= blockHeight;

blockWidth -= 20;

xBlock += 10;

Blocks1.Add(pnn);

}

}

void Form1\_Load(object sender, EventArgs e)

{

CreateColumns();

CreateBlocks();

off = new Bitmap(this.ClientSize.Width, this.ClientSize.Height);

}

void Form1\_Paint(object sender, PaintEventArgs e)

{

DrawDubb(e.Graphics);

}

void DrawScene(Graphics g)

{

g.Clear(Color.White);

Pen p=new Pen(Color.Black);

SolidBrush brush = new SolidBrush(Color.Gray);

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

{

CActorColumn pTrav = LColumns[i];

g.FillRectangle(brush, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

g.DrawRectangle(p, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

}

brush=new SolidBrush(Color.Yellow);

for (int i = 0; i < Blocks1.Count; i++)

{

CActorBlock pTrav = Blocks1[i];

pTrav = Blocks1[i];

g.FillRectangle(brush, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

g.DrawRectangle(p, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

}

for (int i = 0; i < Blocks2.Count; i++)

{

CActorBlock pTrav = Blocks2[i];

g.FillRectangle(brush, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

g.DrawRectangle(p, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

}

for (int i = 0; i < Blocks3.Count; i++)

{

CActorBlock pTrav = Blocks3[i];

g.FillRectangle(brush, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

g.DrawRectangle(p, pTrav.X, pTrav.Y, pTrav.W, pTrav.H);

}

}

void DrawDubb(Graphics g)

{

Graphics g2 = Graphics.FromImage(off);

DrawScene(g2);

g.DrawImage(off, 0, 0);

}

}

}