Карпенко О. КН20-1

**ЛАБОРАТОРНА РОБОТА 5**

Розробити програму для емуляції дисплейного модуля (розширені можливості).

Етапи виконання лабораторної роботи:

1. Розширити можливості програми реалізованою в лабораторній роботі №2 (емулятор дисплейного модуля) шляхом реалізації команди draw\_text тільки з використанням команд малювання ліній. Програма повинна зберегти сумісність з програмами, розробленими в лабораторних роботах №3 і №4.

2. Інтерфейс бібліотеки GraphicsLib не змінювати.

3. Протокол обміну командами не змінювати.

Хід Роботи:

Код:

Тут сервер приймає команду:

case 19:

Console.WriteLine("Command:Draw text as symbols");

TextDecoder(RecievedData, out val1, out val2, out val3, out val4, out RGB, out text);

DrawSymbol(val1, val2, text, val3, RGB);

text = $"Command: Draw text as symbols: x = {val1}, y = {val2}, size = {val3}";

Invoke((MethodInvoker)delegate { listBox1.Items.Add(Text = text); });

break;

Код DrawSymbol:

public void DrawSymbol(short x1, short y1, string text, short size, byte[] RGB)

{

var SymbolLine = new TextAsSymbols(x1, y1, size, RGB);

SymbolLine.TextProcessor(text);

Texts.Add(SymbolLine);

Invalidate();

}

Код Класу TextAsSymbols:

public class TextAsSymbols

{

public int size, x0, y0;

public float[,] coords;

public byte[] RGB;

public List<float[,]> symbols = new List<float[,]>();

public TextAsSymbols(int \_x0, int \_y0, int \_size, Byte[] \_RGB)

{

this.x0 = \_x0;

this.y0 = \_y0;

this.RGB = \_RGB;

this.size = \_size;

}

public void SymbolA(int \_x0, int \_y0)

{

coords = new float[3,4];

coords[0,0] = \_x0;

coords[0,1] = \_y0;

coords[0,2] = (\_x0 + (1 \* size));

coords[0,3] = (\_y0 - (2 \* size));

coords[1, 0] = coords[0, 2];

coords[1, 1] = coords[0, 3];

coords[1, 2] = (\_x0 + (2 \* size));

coords[1, 3] = \_y0;

coords[2,0] = ((\_x0 + coords[1,0]) / 2);

coords[2,1] = ((\_y0 + coords[1, 1]) / 2);

coords[2,2] = ((coords[1,0] + coords[1, 2]) / 2);

coords[2,3] = ((coords[1,1] + coords[1, 3]) / 2);

symbols.Add(coords);

}

public void SymbolB(int \_x0, int \_y0)

{

coords = new float[10,4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = \_y0;

coords[1, 2] = \_x0 + size \* 1.75f;

coords[1, 3] = \_y0;

coords[2, 0] = \_x0;

coords[2, 1] = \_y0 - size;

coords[2, 2] = \_x0 + size \* 1.75f;

coords[2, 3] = \_y0 - size;

coords[3, 0] = \_x0;

coords[3, 1] = \_y0 - size \* 2;

coords[3, 2] = \_x0 + size \* 1.75f;

coords[3, 3] = \_y0 - size \* 2;

coords[4, 0] = coords[1, 2];

coords[4, 1] = coords[1, 3];

coords[4, 2] = \_x0 + size \* 2;

coords[4, 3] = \_y0 - size \* 0.25f;

coords[5, 0] = coords[2, 2];

coords[5, 1] = coords[2, 3];

coords[5, 2] = \_x0 + size \* 2;

coords[5, 3] = \_y0 - size \* 0.75f;

coords[6, 0] = coords[5, 2];

coords[6, 1] = coords[5, 3];

coords[6, 2] = coords[4, 2];

coords[6, 3] = coords[4, 3];

coords[7, 0] = coords[3, 2];

coords[7, 1] = coords[3, 3];

coords[7, 2] = \_x0 + size \* 2;

coords[7, 3] = \_y0 - size \* 1.75f;

coords[8, 0] = coords[2, 2];

coords[8, 1] = coords[2, 3];

coords[8, 2] = \_x0 + size \* 2;

coords[8, 3] = \_y0 - size \* 1.25f;

coords[9, 0] = coords[8, 2];

coords[9, 1] = coords[8, 3];

coords[9, 2] = coords[7, 2];

coords[9, 3] = coords[7, 3];

symbols.Add(coords);

}

public void SymbolC(int \_x0, int \_y0)

{

coords = new float[8, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - (size \* 0.33f);

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 1.66f;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0] + (size \* 0.33f);

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.66f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[0, 2];

coords[3, 1] = coords[0, 3];

coords[3, 2] = coords[1, 2];

coords[3, 3] = \_y0 - (size \* 2f);

coords[4, 0] = coords[3, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[2, 2];

coords[4, 3] = coords[4, 1];

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = coords[5, 0] + (size \* 0.33f);

coords[5, 3] = coords[0, 3];

coords[6, 0] = coords[2, 2];

coords[6, 1] = coords[2, 3];

coords[6, 2] = coords[6, 0] + (size \* 0.33f);

coords[6, 3] = coords[0, 1];

symbols.Add(coords);

}

public void SymbolD(int \_x0, int \_y0)

{

coords = new float[6, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - (2 \* size);

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size;

coords[1, 3] = coords[0, 3];

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = coords[1, 2] + size;

coords[2, 3] = coords[1, 3] + (size \* 0.33f);

coords[3, 0] = \_x0;

coords[3, 1] = \_y0;

coords[3, 2] = coords[1, 2];

coords[3, 3] = coords[3, 1];

coords[4, 0] = coords[3, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[3, 2] + size;

coords[4, 3] = coords[3, 3] - (size \* 0.33f);

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = coords[2, 2];

coords[5, 3] = coords[2, 3];

symbols.Add(coords);

}

public void SymbolE(int \_x0, int \_y0)

{

coords = new float[4, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0 + (size \* 2);

coords[0, 3] = \_y0;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0];

coords[1, 3] = coords[0, 1] - (size \* 2);

coords[2, 0] = coords[0, 0];

coords[2, 1] = coords[1, 3];

coords[2, 2] = coords[0, 2];

coords[2, 3] = coords[1, 3];

coords[3, 0] = coords[0, 0];

coords[3, 1] = coords[0, 1] - size;

coords[3, 2] = coords[0, 0] + size;

coords[3, 3] = coords[3, 1];

symbols.Add(coords);

}

public void SymbolF(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = coords[0, 0];

coords[0, 3] = coords[0, 1] - (size \* 2);

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 3];

coords[1, 2] = coords[0, 0] + (size \* 2);

coords[1, 3] = coords[0, 3];

coords[2, 0] = coords[0, 0];

coords[2, 1] = coords[0, 1] - size;

coords[2, 2] = coords[0, 0] + size;

coords[2, 3] = coords[2, 1];

symbols.Add(coords);

}

public void SymbolG(int \_x0, int \_y0)

{

coords = new float[9,4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - (size \* 0.33f);

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size\*1.66f;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0] + (size \* 0.33f);

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.66f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[0, 2];

coords[3, 1] = coords[0, 3];

coords[3, 2] = coords[1, 2];

coords[3, 3] = \_y0 - (size \* 2f);

coords[4, 0] = coords[3, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[2, 2];

coords[4, 3] = coords[4, 1];

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = coords[5, 0] + (size \* 0.33f);

coords[5, 3] = coords[0, 3];

coords[6, 0] = coords[2, 2];

coords[6, 1] = coords[2, 3];

coords[6, 2] = coords[6, 0] + (size \* 0.33f);

coords[6, 3] = coords[0, 1];

coords[7, 0] = coords[6, 2];

coords[7, 1] = coords[6, 3];

coords[7, 2] = coords[7, 0];

coords[7, 3] = coords[7, 1] - (size \* 0.5f);

coords[8, 0] = coords[7, 2];

coords[8, 1] = coords[7, 3];

coords[8, 2] = coords[8, 0] - size;

coords[8, 3] = coords[8, 1];

symbols.Add(coords);

}

public void SymbolH(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0 + size \* 2;

coords[1, 1] = \_y0;

coords[1, 2] = coords[1, 0];

coords[1, 3] = coords[0, 3];

coords[2, 0] = \_x0;

coords[2, 1] = \_y0 - size;

coords[2, 2] = coords[1, 0];

coords[2, 3] = coords[2, 1];

symbols.Add(coords);

}

public void SymbolI(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0 + size;

coords[0, 1] = \_y0;

coords[0, 2] = coords[0, 0];

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 2;

coords[1, 3] = coords[0, 3];

coords[2, 0] = coords[1, 0];

coords[2, 1] = \_y0;

coords[2, 2] = coords[1, 2];

coords[2, 3] = \_y0;

symbols.Add(coords);

}

public void SymbolJ(int \_x0, int \_y0)

{

coords = new float[5, 4];

coords[0, 0] = \_x0 + size \* 1.66f;

coords[0, 1] = \_y0 - size \* 0.33f;

coords[0, 2] = coords[0, 0];

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 1.66f;

coords[1, 3] = coords[0, 3];

coords[2, 0] = coords[0, 0];

coords[2, 1] = coords[0, 1];

coords[2, 2] = coords[2, 0] - size \* 0.5f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[2, 2];

coords[3, 1] = coords[2, 3];

coords[3, 2] = \_x0 + size \* 0.5f;

coords[3, 3] = \_y0;

coords[4, 0] = coords[3, 2];

coords[4, 1] = \_y0;

coords[4, 2] = \_x0;

coords[4, 3] = \_y0 - size\*0.5f;

symbols.Add(coords);

}

public void SymbolK(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = \_y0 - size;

coords[1, 2] = \_x0 + size \* 2;

coords[1, 3] = \_y0;

coords[2, 0] = \_x0;

coords[2, 1] = coords[1, 1];

coords[2, 2] = coords[1, 2];

coords[2, 3] = y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolL(int \_x0, int \_y0)

{

coords = new float[2, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0 + size \* 2;

coords[0, 3] = \_y0;

coords[1, 0] = \_x0;

coords[1, 1] = \_y0;

coords[1, 2] = \_x0;

coords[1, 3] = \_y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolM(int \_x0, int \_y0)

{

coords = new float[4, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0 + size \* 2;

coords[1, 1] = \_y0;

coords[1, 2] = coords[1, 0];

coords[1, 3] = coords[0, 3];

coords[2, 0] = \_x0;

coords[2, 1] = coords[0, 3];

coords[2, 2] = \_x0 + size;

coords[2, 3] = \_y0 - size;

coords[3, 0] = coords[1, 0];

coords[3, 1] = coords[0, 3];

coords[3, 2] = coords[2, 2];

coords[3, 3] = coords[2, 3];

symbols.Add(coords);

}

public void SymbolN(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0 + size \* 2;

coords[1, 1] = \_y0;

coords[1, 2] = coords[1, 0];

coords[1, 3] = coords[0, 3];

coords[2, 0] = \_x0;

coords[2, 1] = coords[0, 3];

coords[2, 2] = coords[1, 0];

coords[2, 3] = \_y0;

symbols.Add(coords);

}

public void SymbolO(int \_x0, int \_y0)

{

coords = new float[8, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - (size \* 0.33f);

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 1.66f;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0] + (size \* 0.33f);

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.66f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[0, 2];

coords[3, 1] = coords[0, 3];

coords[3, 2] = coords[1, 2];

coords[3, 3] = \_y0 - (size \* 2f);

coords[4, 0] = coords[3, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[2, 2];

coords[4, 3] = coords[4, 1];

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = \_x0 + size \* 2;

coords[5, 3] = coords[0, 3];

coords[6, 0] = coords[2, 2];

coords[6, 1] = coords[2, 3];

coords[6, 2] = coords[6, 0] + (size \* 0.33f);

coords[6, 3] = coords[0, 1];

coords[7, 0] = \_x0 + size \* 2;

coords[7, 1] = coords[0, 1];

coords[7, 2] = coords[7, 0];

coords[7, 3] = coords[0, 3];

symbols.Add(coords);

}

public void SymbolP(int \_x0, int \_y0)

{

coords = new float[6, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 1.66f;

coords[1, 3] = coords[0, 3];

coords[2, 0] = \_x0;

coords[2, 1] = \_y0 - size;

coords[2, 2] = coords[1, 2];

coords[2, 3] = coords[2, 1];

coords[3, 0] = coords[1, 2];

coords[3, 1] = coords[1, 3];

coords[3, 2] = \_x0 + size \* 2;

coords[3, 3] = \_y0 - size \* 1.67f;

coords[4, 0] = coords[1, 2];

coords[4, 1] = coords[2, 1];

coords[4, 2] = \_x0 + size \* 2;

coords[4, 3] = \_y0 - size \* 1.33f;

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = coords[3, 2];

coords[5, 3] = coords[3, 3];

symbols.Add(coords);

}

public void SymbolQ(int \_x0, int \_y0)

{

coords = new float[9, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - (size \* 0.33f);

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 1.66f;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0] + (size \* 0.33f);

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.66f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[0, 2];

coords[3, 1] = coords[0, 3];

coords[3, 2] = coords[1, 2];

coords[3, 3] = \_y0 - (size \* 2f);

coords[4, 0] = coords[3, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[2, 2];

coords[4, 3] = coords[4, 1];

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = \_x0 + size \* 2;

coords[5, 3] = coords[0, 3];

coords[6, 0] = coords[2, 2];

coords[6, 1] = coords[2, 3];

coords[6, 2] = coords[6, 0] + (size \* 0.33f);

coords[6, 3] = coords[0, 1];

coords[7, 0] = \_x0 + size \* 2;

coords[7, 1] = coords[0, 1];

coords[7, 2] = coords[7, 0];

coords[7, 3] = coords[0, 3];

coords[8, 0] = \_x0 + size \* 2;

coords[8, 1] = \_y0;

coords[8, 2] = \_x0 + size \* 1.66f;

coords[8, 3] = \_y0 - size \* 0.5f;

symbols.Add(coords);

}

public void SymbolR(int \_x0, int \_y0)

{

coords = new float[7, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0;

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 1.66f;

coords[1, 3] = coords[0, 3];

coords[2, 0] = \_x0;

coords[2, 1] = \_y0 - size;

coords[2, 2] = coords[1, 2];

coords[2, 3] = coords[2, 1];

coords[3, 0] = coords[1, 2];

coords[3, 1] = coords[1, 3];

coords[3, 2] = \_x0 + size \* 2;

coords[3, 3] = \_y0 - size \* 1.67f;

coords[4, 0] = coords[1, 2];

coords[4, 1] = coords[2, 1];

coords[4, 2] = \_x0 + size \* 2;

coords[4, 3] = \_y0 - size \* 1.33f;

coords[5, 0] = coords[4, 2];

coords[5, 1] = coords[4, 3];

coords[5, 2] = coords[3, 2];

coords[5, 3] = coords[3, 3];

coords[6, 0] = coords[1, 2] - size \* 0.66f;

coords[6, 1] = coords[2, 1];

coords[6, 2] = \_x0 + size \* 2;

coords[6, 3] = \_y0;

symbols.Add(coords);

}

public void SymbolS(int \_x0, int \_y0)

{

coords = new float[11, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size \* 0.25f;

coords[0, 2] = \_x0 + size \* 0.5f;

coords[0, 3] = \_y0;

coords[1, 0] = coords[0, 2];

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 1.5f;

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = \_y0;

coords[2, 2] = \_x0 + size \* 2;

coords[2, 3] = coords[0, 1];

coords[3, 0] = coords[2, 2];

coords[3, 1] = coords[2, 3];

coords[3, 2] = coords[2, 2];

coords[3, 3] = \_y0 - size \* 0.75f;

coords[4, 0] = coords[2, 2];

coords[4, 1] = coords[3, 3];

coords[4, 2] = coords[1, 2];

coords[4, 3] = \_y0 - size;

coords[5, 0] = coords[0, 2];

coords[5, 1] = \_y0 - size;

coords[5, 2] = coords[1, 2];

coords[5, 3] = coords[5, 1];

coords[6, 0] = coords[0, 2];

coords[6, 1] = coords[5, 3];

coords[6, 2] = \_x0;

coords[6, 3] = \_y0 - size \* 1.25f;

coords[7, 0] = \_x0;

coords[7, 1] = \_y0 - size \* 1.25f;

coords[7, 2] = \_x0;

coords[7, 3] = \_y0 - size \* 1.75f;

coords[8, 0] = \_x0;

coords[8, 1] = coords[7, 3];

coords[8, 2] = coords[0, 2];

coords[8, 3] = \_y0 - size \* 2;

coords[9, 0] = coords[8, 2];

coords[9, 1] = coords[8, 3];

coords[9, 2] = coords[1, 2];

coords[9, 3] = coords[9, 1];

coords[10, 0] = coords[9, 2];

coords[10, 1] = coords[9, 3];

coords[10, 2] = \_x0 + size \* 2;

coords[10, 3] = coords[7, 3];

symbols.Add(coords);

}

public void SymbolT(int \_x0, int \_y0)

{

coords = new float[2, 4];

coords[0, 0] = \_x0 + size;

coords[0, 1] = \_y0;

coords[0, 2] = coords[0, 0];

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = \_x0;

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0 + size \* 2;

coords[1, 3] = coords[0, 3];

symbols.Add(coords);

}

public void SymbolU(int \_x0, int \_y0)

{

coords = new float[5, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - (size \* 0.33f);

coords[0, 2] = \_x0;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = coords[0, 0];

coords[1, 1] = coords[0, 1];

coords[1, 2] = coords[0, 0] + (size \* 0.33f);

coords[1, 3] = \_y0;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.66f;

coords[2, 3] = \_y0;

coords[3, 0] = coords[2, 2];

coords[3, 1] = coords[2, 3];

coords[3, 2] = \_x0 + size\* 2;

coords[3, 3] = coords[0, 1];

coords[4, 0] = \_x0 + size \* 2;

coords[4, 1] = coords[0, 1];

coords[4, 2] = coords[4, 0];

coords[4, 3] =\_y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolV(int \_x0, int \_y0)

{

coords = new float[2, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size\*2;

coords[0, 2] = \_x0 + size;

coords[0, 3] = \_y0;

coords[1, 0] = \_x0 + size;

coords[1, 1] = \_y0;

coords[1, 2] = \_x0 + size \* 2;

coords[1, 3] = \_y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolW(int \_x0, int \_y0)

{

coords = new float[4, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size \* 2;

coords[0, 2] = \_x0 + size \* 0.5f;

coords[0, 3] = \_y0;

coords[1, 0] = coords[0, 2];

coords[1, 1] = \_y0;

coords[1, 2] = \_x0 + size;

coords[1, 3] = \_y0 - size;

coords[2, 0] = coords[1, 2];

coords[2, 1] = coords[1, 3];

coords[2, 2] = \_x0 + size \* 1.5f;

coords[2, 3] = \_y0;

coords[3, 0] = \_x0 + size \* 1.5f;

coords[3, 1] = \_y0;

coords[3, 2] = \_x0 + size \* 2f;

coords[3, 3] = \_y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolX(int \_x0, int \_y0)

{

coords = new float[2, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size \* 2;

coords[0, 2] = \_x0 + size\*2;

coords[0, 3] = \_y0;

coords[1, 0] = \_x0;

coords[1, 1] = \_y0;

coords[1, 2] = \_x0 + size \* 2;

coords[1, 3] = \_y0 - size \* 2;

symbols.Add(coords);

}

public void SymbolY(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size \* 2;

coords[0, 2] = \_x0 + size;

coords[0, 3] = \_y0 - size;

coords[1, 0] = \_x0 + size \* 2;

coords[1, 1] = \_y0 - size \* 2;

coords[1, 2] = \_x0 + size;

coords[1, 3] = \_y0 - size;

coords[2, 0] = \_x0 + size;

coords[2, 1] = \_y0 - size;

coords[2, 2] = \_x0 + size;

coords[2, 3] = \_y0;

symbols.Add(coords);

}

public void SymbolZ(int \_x0, int \_y0)

{

coords = new float[3, 4];

coords[0, 0] = \_x0;

coords[0, 1] = \_y0 - size \* 2;

coords[0, 2] = \_x0 + size \* 2;

coords[0, 3] = \_y0 - size \* 2;

coords[1, 0] = coords[0, 2];

coords[1, 1] = coords[0, 3];

coords[1, 2] = \_x0;

coords[1, 3] = \_y0;

coords[2, 0] = \_x0;

coords[2, 1] = \_y0;

coords[2, 2] = \_x0 + size \* 2;

symbols.Add(coords);

}

public void TextProcessor(string text)

{

foreach(char c in text)

{

switch(c)

{

case 'A':

case 'a':

SymbolA(x0, y0);

break;

case 'B':

case 'b':

SymbolB(x0, y0);

break;

case 'C':

case 'c':

SymbolC(x0, y0);

break;

case 'D':

case 'd':

SymbolD(x0, y0);

break;

case 'E':

case 'e':

SymbolE(x0, y0);

break;

case 'F':

case 'f':

SymbolF(x0, y0);

break;

case 'G':

case 'g':

SymbolG(x0, y0);

break;

case 'H':

case 'h':

SymbolH(x0, y0);

break;

case 'I':

case 'i':

SymbolI(x0, y0);

break;

case 'J':

case 'j':

SymbolJ(x0, y0);

break;

case 'K':

case 'k':

SymbolK(x0, y0);

break;

case 'L':

case 'l':

SymbolL(x0, y0);

break;

case 'M':

case 'm':

SymbolM(x0, y0);

break;

case 'N':

case 'n':

SymbolN(x0, y0);

break;

case 'O':

case 'o':

SymbolO(x0, y0);

break;

case 'P':

case 'p':

SymbolP(x0, y0);

break;

case 'Q':

case 'q':

SymbolQ(x0, y0);

break;

case 'R':

case 'r':

SymbolR(x0, y0);

break;

case 'S':

case 's':

SymbolS(x0, y0);

break;

case 'T':

case 't':

SymbolT(x0, y0);

break;

case 'U':

case 'u':

SymbolU(x0, y0);

break;

case 'V':

case 'v':

SymbolV(x0, y0);

break;

case 'W':

case 'w':

SymbolW(x0, y0);

break;

case 'X':

case 'x':

SymbolX(x0, y0);

break;

case 'Y':

case 'y':

SymbolY(x0, y0);

break;

case 'Z':

case 'z':

SymbolZ(x0, y0);

break;

default:

break;

}

x0 += Convert.ToInt32(2 \* size \* 1.3f);

}

}

}

Код Відрисовки:

if (Texts != null)

{

foreach (var text in Texts)

{

Pen Pen = new Pen(Color.FromArgb(text.RGB[0], text.RGB[1], text.RGB[2]), 1);

foreach(var character in text.symbols)

{

for (int i = 0; i < character.GetLength(0); i++)

{

g.DrawLine(Pen, character[i, 0] + xMiddle, character[i,1] + yMiddle, character[i,2] + xMiddle, character[i,3] + yMiddle);

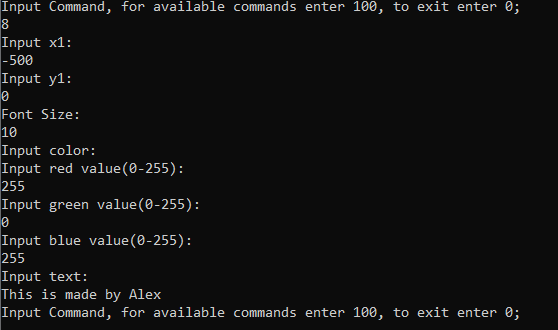
}

}

}

}

Клієнт:



Сервер:

