C++/CLI LAB7 – Animáció

Új projekt: 

-> LAB7

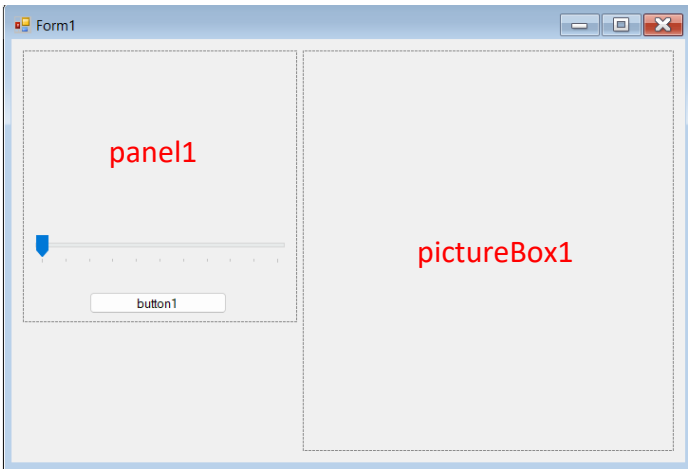
Felület:
1 db Panel

1 db TrackBar

1 db Button

1 db PictureBox (*Anchor: Top, Bottom, Left, Right*)

1 db Timer



Ismeretek:

- Animáció, timer

Form1.h

```
// Globális változók
Graphics^ pgr, ^ gr;
Pen^ toll;
SolidBrush^ ecset;
Bitmap^ bm;

float fok = 90;
float tx = 300; // x tengely helye
float ty = 500; // y tengely helye
float hr = 50;  // hajtókar sugara
float hl = 300; // hajtókar hossza
float dw = 50;  // dugattyú félszélesség
float dh = 50;  // dugattyú magasság

// Kezdeti paraméterek elkészítése
// pictureBox-t horgonyozzuk hozzá az abalak széleihez, automatikusan meg fog nyúlni
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
	this->Text = "Grafika";
	button1->Text = "Stop";

	// trackbar: -30 és +30 fok között mozgat -> 60fok
	trackBar1->Minimum = -30;
	trackBar1->Maximum = 30;

	toll = gcnew Pen(Color::Silver);
	ecset = gcnew SolidBrush(Color::Silver);
```

```
	// az animáció egy bitmap-ben készül (aktuális pictureBox mérettel)
	bm = gcnew Bitmap(pictureBox1->Width, pictureBox1->Height);
	gr = Graphics::FromImage(bm);

	// időzítés beállítása és indítása
	timer1->Interval = 10 + 150 - trackBar1->Value * 5; //min 10 - max 310ms
	timer1->Start();
}

// Rajz frissítése
private: System::Void panel1_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^ e) {
	Rajz();
}

// Sebesség kijelző
void Rajz() {
	// a sebesség kijelzés a panelra készül - gyors mozgatásnál szaggatni fog
	pgr = panel1->CreateGraphics();

	// a folyamatos újrarajzolás miatt mindig törölni kell
	pgr->Clear(System::Drawing::SystemColors::Control);

	toll->Width = 2;
	toll->Color = Color::Blue;

	const float kx = 100; // középpont x
	const float ky = 210; // középpont y
	const float n = 200;  // nagy sugár
	const float k = 100;  // kis sugár

	// középpont elmozgatása
	pgr->TranslateTransform(kx, ky);

	// a két körív (-30 és +30 fok között)
	pgr->DrawArc(toll, -n, -n, 2 * n, 2 * n, -60.0f, -60.0f);
	pgr->DrawArc(toll, -k, -k, 2 * k, 2 * k, -60.0f, -60.0f);

	// a két oldalsó vonal
	float px = (float)(Math::Cos(d2r(60)));
	float py = (float)(Math::Sin(d2r(60)));
	pgr->DrawLine(toll, px * n, -py * n, px * k, -py * k);
	pgr->DrawLine(toll, -px * n, -py * n, -px * k, -py * k);

	// a mutató - a trackbar értéke szerint áll be
	toll->Width = 5;
	toll->Color = Color::Red;
	int szog = trackBar1->Value;
	pgr->RotateTransform(szog - 90);
	pgr->DrawLine(toll, k, 0.0f, n - 20, 0.0f);
}

// Átváltás fokból radiánra a szögfüggvényekhez
public: double d2r(double d) {  return Math::PI / 180 * d; }

// Sebesség átállítása a trackbar változtatásakor
private: System::Void trackBar1_Scroll(System::Object^ sender, System::EventArgs^ e) {
	Rajz();
	timer1->Interval = 10 + 150 - trackBar1->Value * 5; //min 10 - max 310ms
}
```

```
// A dugattyú(k) animációja
// a timer_Tick a timer interval-ban megadott időközönként fut le,
// ha be van kapcsolva timer Enabled értéke
private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e) {

	// törlés Form háttérszínével
	gr->Clear(this->BackColor);

	// koordináta transzformáció törlése
	gr->ResetTransform();

	// koorináta rendszer eltolása az origóba
	gr->TranslateTransform(tx, ty);
	gr->ScaleTransform(1, -1);

	// elforgatás
	gr->RotateTransform(30);

	// dugattyú kirajzoltatása
	Dugattyu();

	// egy másik megrajzolása
	gr->RotateTransform(-60);
	int fazis = 180; //ellenütemben lesznek
	fok += fazis;
	Dugattyu();

	//léptetés
	fok -= fazis - 5;

	// megjelenítés
	pictureBox1->Image = bm;

	// animációhoz
	//bm::Save("kep" + fok + "::png");

	// sebesség kijelzés frissítése
	//Rajz();
}

// Dugattyú kirajzolása
// Az origó a főtengely, y tengely felfelé lett állítva
void Dugattyu() {
	// tengelyek
	toll->Width = 2;
	toll->Color = Color::Silver;
	toll->DashStyle = System::Drawing::Drawing2D::DashStyle::DashDot;
	gr->DrawLine(toll, 0.0f, hl + hr + dh + 20, 0.0f, -2 * hr); //y tengely
	gr->DrawLine(toll, -2 * hr, 0.0f, 2 * hr, 0.0f); //x tengely
	gr->DrawEllipse(toll, -hr, -hr, 2 * hr, 2 * hr); //hajtókar sugara

	// főtengely
	toll->DashStyle = System::Drawing::Drawing2D::DashStyle::Solid;
	toll->Width = 4;
	toll->Color = Color::Black;
	float r = 1.5 * hr;
	gr->DrawEllipse(toll, -r, -r, 2 * r, 2 * r);

	// dugattyúház
	gr->DrawLine(toll, -dw, hl + hr + dh, -dw, hl - hr);
	gr->DrawLine(toll, +dw, hl + hr + dh, +dw, hl - hr);
```

```
        gr->DrawLine(toll, -dw, hl + hr + dh,  dw, hl + hr + dh);

        // számítás
        float forgx = Math::Cos(d2r(fok)) * hr;
        float forgy = Math::Sin(d2r(fok)) * hr;
        float dgx = 0;
        float dgy = forgy + Math::Sqrt(hl * hl - forgx * forgx);

        // hajtókar
        toll->Width = 30;
        gr->DrawLine(toll, forgx, forgy, dgx, dgy);

        // dugattyú
        toll->Width = 2;
        ecset->Color = Color::Silver;
        gr->FillRectangle(ecset, -dw, dgy, 2 * dw, dh);
        gr->DrawRectangle(toll, -dw, dgy, 2 * dw, dh);

        // forgattyú
        r = 20;
        gr->FillEllipse(ecset, forgx - r, forgy - r, 2 * r, 2 * r);
        gr->DrawEllipse(toll, forgx - r, forgy - r, 2 * r, 2 * r);

        // felső kör
        gr->FillEllipse(ecset, -r, dgy - r, 2 * r, 2 * r);
        gr->DrawEllipse(toll, -r, dgy - r, 2 * r, 2 * r);

        // levegő
        float meret = hl + hr - dgy;
        int szin = (int)(255 / (2 * hr) * meret); //a szint a méret szerint állítom
        ecset->Color = Color::FromArgb(150, 255, szin, szin); //az első paraméter az áttetszőség
        gr->FillRectangle(ecset, -dw, dgy + dh, 2 * dw, meret);
        gr->DrawRectangle(toll, -dw, dgy + dh, 2 * dw, meret);
}

// Átméretezéskor mindig az aktuális mérettel dolgozzon
private: System::Void Form1_Resize(System::Object^ sender, System::EventArgs^ e) {
        //az animáció egy bitmap-ben készül (aktuális pictureBox mérettel)
        bm = gcnew Bitmap(pictureBox1->Width, pictureBox1->Height);
        gr = Graphics::FromImage(bm);
}

//A timer indításának és leállításának vezérlése 1 gombbal
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
        if (timer1->Enabled) {
                button1->Text = "Start";
                timer1->Enabled = false;
        } else {
                button1->Text = "Stop";
                timer1->Enabled = true;
        }
}
```