# Windows Presentation Foundation

## Kétdimenziós rajzolás WPF-ben

A grafikus megjelenítés módjai WPF-ben:

**System.Windows.Shapes** névtér osztályaival

* + - Magas szintű, rengeteg metódus, tulajdonságok, eseménykezelés, input kezelés (egér, billentyűzet)  lassú; egy sor szabályos geometriai objektum (téglalap, ellipszis, stb.)
    - Hozzunk létre egy Canvast XAML-ben **cvLap** néven, melyre a későbbiekben rajzolunk:

<Canvas Name="cvLap"></Canvas>

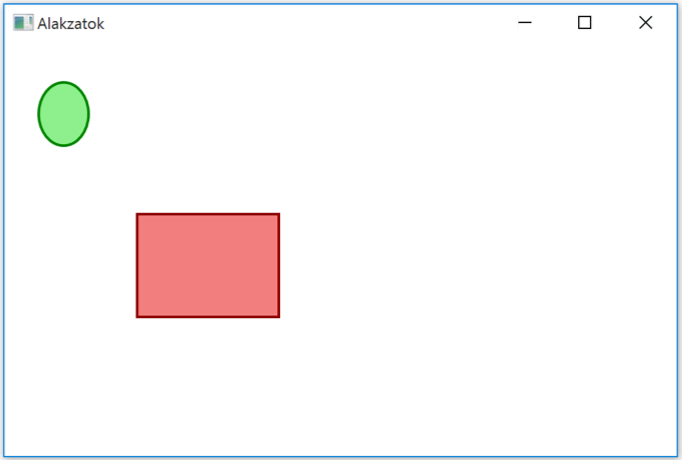
* + - Egyszerűen leírható XAML-ben:

<Ellipse Width="40" Height="50" Stroke="Green" StrokeThickness="2" Fill="LightGreen" Canvas.Left="25" Canvas.Top="30" Name="elLomb" MouseDown="elLomb\_MouseDown" MouseMove="elLomb\_MouseMove"/>

* + - A rajzolás C# kódból is megvalósítható: double tetoMagassag = 30; Rectangle rcHaz = new Rectangle(); rcHaz.Width = 110;

rcHaz.Height = 80; rcHaz.Stroke = Brushes.DarkRed; rcHaz.StrokeThickness = 2;

rcHaz.Fill = Brushes.LightCoral; cvLap.Children.Add(rcHaz); rcHaz.SetValue(Canvas.LeftProperty, (double)100); rcHaz.SetValue(Canvas.TopProperty,

(double)(100 + tetoMagassag));

* + - A tárolóra elhelyezett alakzatok között van egy Z-sorrend, ami azt jelenti, hogy a később feltett alakzatok elfedhetik a korábban feltett alakzatokat (pl. ha az elsőnek feltett alakzatot átmozgatjuk a másodiknak feltett alakzat pozíciójába, akkor az első a második alá kerül.

**System.Windows.Media.Drawing** absztakt osztály leszármazottaival

* + - Vékonyabb réteg (ún. Pehelysúlyú szolgáltatások)  gyorsabb, kisebb erőforrásigény
    - Nincs beépített input kezelés
    - Valamilyen hoszt objektumban kell elhelyezni (pl. DrawingImage, DrawingBrush, DrawingVisual)
    - Több kód szükséges
    - Fontosabb osztályok: GeometryDrawing, ImageDrawing
    - Leírható XAML-ben:

<Image Canvas.Left="0" Canvas.Bottom="0">

<Image.Source>

<DrawingImage>

<DrawingImage.Drawing>

<GeometryDrawing Brush="LightGreen">

<GeometryDrawing.Pen>

<Pen Brush="Red" Thickness="3"/>

</GeometryDrawing.Pen>

<GeometryDrawing.Geometry>

<RectangleGeometry Rect="0,0,30,40"/>

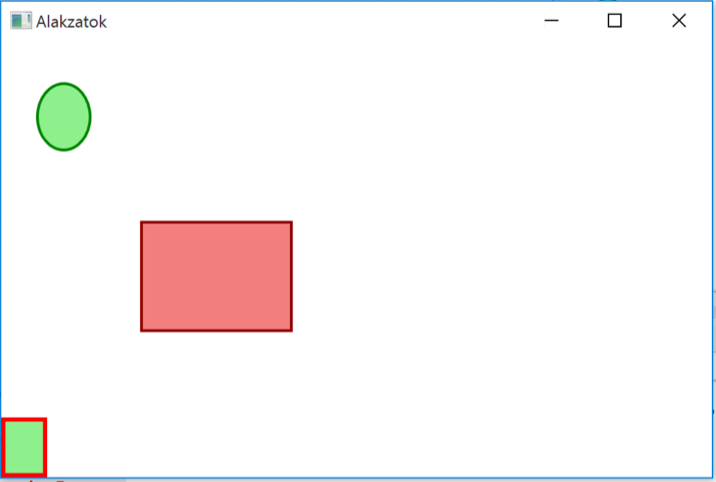
</GeometryDrawing.Geometry>

</GeometryDrawing>

</DrawingImage.Drawing>

</DrawingImage>

</Image.Source>

</Image>

**System.Windows.Media.Visual** absztrakt osztály leszármazottaival

* + - Legvékonyabb réteg  leggyorsabb; csak elemi szolgáltatások, mindenhez meg kell írni a kódot (legtöbb kódolás)
    - nincs input esemény, felületmenedzser, adatkötés, alacsony szintű megközelítés
    - Fontosabb osztályok: DrawingVisual, Viewport3DVisual, ContainerVisual
    - Legkisebb erőforrásigény  Legjobb teljesítmény
    - valamilyen hoszt objektumban kell elhelyezni (pl. DrawingImage, DrawingBrush, DrawingVisual)
    - XAML-ből általában nem oldható meg
    - Rajzolási kapcsolatot/eszközkapcsolatot kell létrehozni és megnyitni, majd a rajzolást követően lezárni (using szerkezet használható)
    - Az új objektumot el kell helyezni a logikai és a vizuális fában.
    - Át kell definiálni a VisualChildrenCount virtuális tulajdonságot.
    - Át kell definiálni a GetVisualChild virtuális metódust.

## Feladat

Készítsen egy WPF alkalmazást, ami

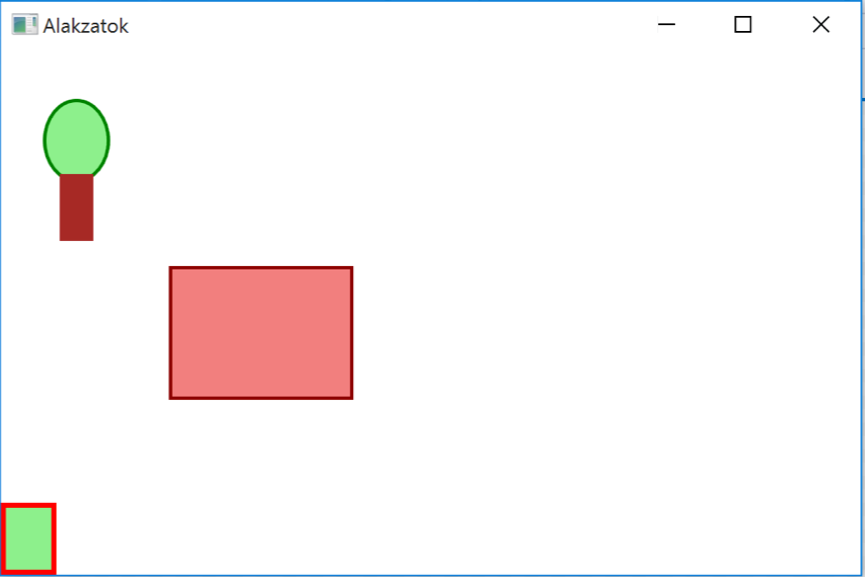
* + Megrajzolja a képen látható fát (XAML-ből) és házat (programból) a

„System.Windows.Shapes” megoldással.

* + Az egér segítségével mozgathatóvá teszi a fát attól függetlenül, hogy a törzsön vagy a lombnál fogjuk-e meg.
  + A fán (törzsön vagy lombon) kattintva jobb egérgombbal egy gyorsmenü jelenik meg (Töröl és Előre hoz menüpontokkal).
  + Töröl: törli a fát.
  + Előre hoz: a Z sorrend végére helyezi a fát, ami azt eredményezi, hogy amikor a ház területére húzzuk az egérrel, akkor elfedi a házat.

## Megoldás

Az ablakot leíró XAML kód:

<Rectangle Width="20" Height="40" Stroke="Brown" StrokeThickness="2" Fill="Brown" Canvas.Left="35" Canvas.Top="75" Name="rcTorzs" MouseDown="rcTorzs\_MouseDown" MouseMove="rcTorzs\_MouseMove"/>

A ház megrajzolását végző metódus:

/// <summary>

/// Megrajzolja a házat.

/// </summary>

/// <param name="x">A házat befoglaló téglalap bal felső sarkának X

/// koordinátája</param>

/// <param name="y">A házat befoglaló téglalap bal felső sarkának Y

/// koordinátája</param>

public void Haz(double x, double y){

//A tető magassága

double tetoMagassag = 30;

//A földszintet leíró téglalap definiálása. Rectangle rcHaz = new Rectangle();

rcHaz.Width = 110;

rcHaz.Height = 80;

//Keretvonal színe rcHaz.Stroke = Brushes.DarkRed;

//A fal színe rcHaz.StrokeThickness = 2;

//A fal színe

rcHaz.Fill = Brushes.LightCoral;

//A ház helyzetének definiálása rcHaz.SetValue(Canvas.LeftProperty, (double)100); rcHaz.SetValue(Canvas.TopProperty,

(double)(100 + tetoMagassag));

//Elhelyezés a rajzlapon cvLap.Children.Add(rcHaz);

//A tetőt leíró háromszög definiálása Polygon pgTeto = new Polygon();

//Keretvonal színe pgTeto.Stroke = Brushes.Red;

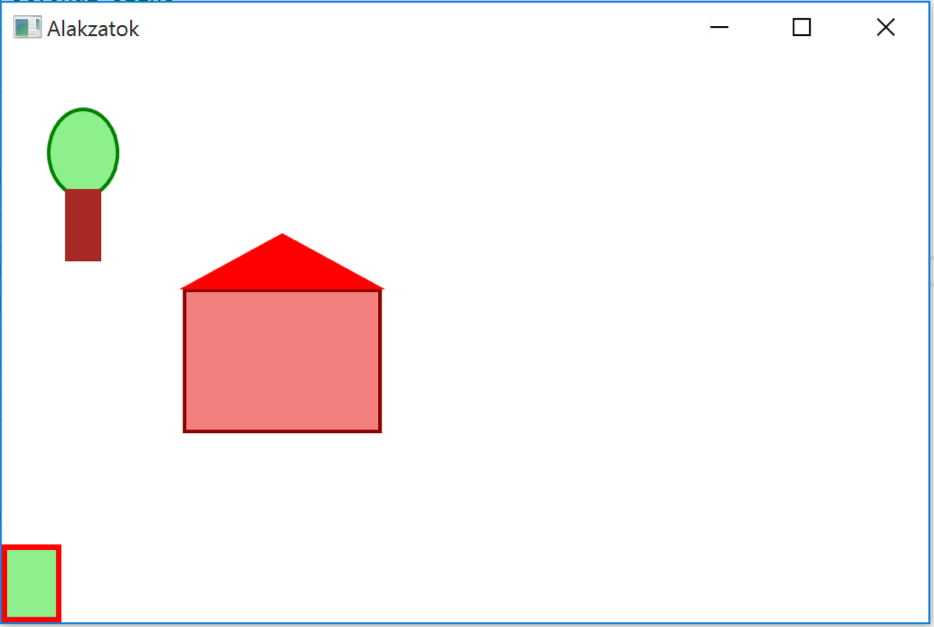
//Keretvonal vastagsága pgTeto.Fill = Brushes.Red;

//A háromszög csúcsainak definiálása pgTeto.Points = new PointCollection(); pgTeto.Points.Add(new Point(x, y + tetoMagassag));

pgTeto.Points.Add(new Point(x + (rcHaz.Width / 2), y)); pgTeto.Points.Add(new Point(x + rcHaz.Width,

y + tetoMagassag));

//Elhelyezés a rajzlapon cvLap.Children.Add(pgTeto);

}

A gyorsmenüt létrehozó metódus:

/// <summary>

/// Létrehozza és a fához rendeli a gyorsmenüt

/// </summary>

private void GyorsMenuLetrehoz(){

//Gyorsmenü definiálása

ContextMenu cmGyorsMenu = new ContextMenu();

//Töröl menüpont definiálása MenuItem miTorol = new MenuItem();

//Megjelenő szöveg miTorol.Header = "Töröl";

//Eseménykezelő hozzárendelése

miTorol.Click += new RoutedEventHandler(miTorol\_Click);

//Hozzááadás a gyorsmenühöz cmGyorsMenu.Items.Add(miTorol);

//Előre hoz menüpont definiálása MenuItem miEloreHoz = new MenuItem();

//Megjelenő szöveg miEloreHoz.Header = "Előre hoz";

//Eseménykezelő hozzárendelése

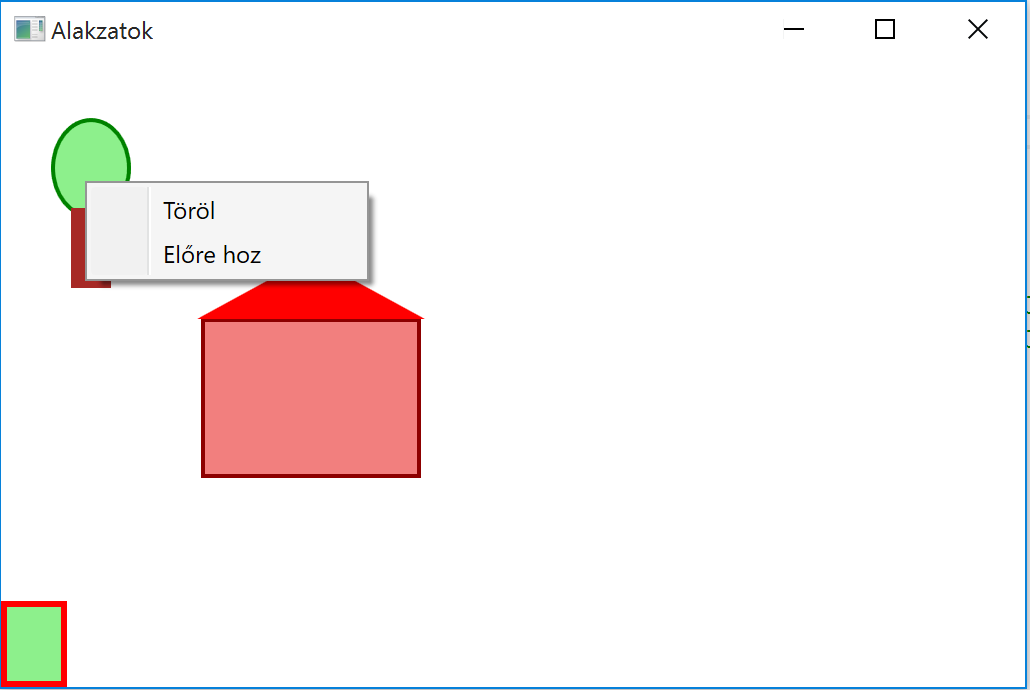
miEloreHoz.Click += new RoutedEventHandler(miEloreHoz\_Click);

//Hozzáadás a gyorsmenühöz cmGyorsMenu.Items.Add(miEloreHoz);

//Gyorsmenü hozzárendelése a lombot megvalósító objektumhoz elLomb.ContextMenu = cmGyorsMenu;

//Gyorsmenü hozzárendelése a fatörzset megvalósító objektumhoz rcTorzs.ContextMenu = cmGyorsMenu;

}



Az ablak konstruktora:

/// <summary>

/// Az ablakosztály konstruktora. Gondoskodik az XAML-ben

/// leírt felület megjelenítéséről, a ház megrajzolásáról és a

/// gyorsmenü létrehozásáról.

/// </summary> public MainWindow(){

//Az XAML-ben leírt felület megjelenítése InitializeComponent();

//A ház megrajzolása Haz(100,100);

//A gyorsmenü létrehozása GyorsMenuLetrehoz();

}

A fa törlésének megvalósítása:

/// <summary>

/// Törli a fát megjelenítő két alakzat objektumot a megjelenítendő

/// objektumok listájáról

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void miTorol\_Click(object sender, RoutedEventArgs e){ cvLap.Children.Remove(elLomb); cvLap.Children.Remove(rcTorzs);

}

/// <summary>

/// A Z-sorrend végére helyezi a fát

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void miEloreHoz\_Click(object sender, RoutedEventArgs e){

//Töröljük a fát megjelenítő két alakzat objektumot a

//megjelenítendő objektumok listájáról cvLap.Children.Remove(elLomb); cvLap.Children.Remove(rcTorzs);

//Újra felvesszük őket a lista végére cvLap.Children.Add(elLomb); cvLap.Children.Add(rcTorzs);

}

A fa mozgatásának megvalósítása:

* Az egérgomb lenyomásakor (ha az a fa területén történik) tároljuk az egér helyzetét.
* Egér mozgatás eseménykor (ha az a fa területén történik) ha a bal oldali egérgomb le van nyomva
  + Lekérdezzük az egér helyzetét.
  + Kiszámoljuk, hogy mennyit mozdult el az előző pozícióhoz képest.
  + Lekérdezzük a lombot befoglaló téglalap bal felső sarkának helyzetét.
  + Elmozgatjuk a lombot.
  + Lekérdezzük a fatörzset befoglaló téglalap bal felső sarkának helyzetét.
  + Elmozgatjuk a fatörzset.
  + Tároljuk az egér aktuális helyzetét.

/// <summary>

/// Egérgomb mozgatása esemény (a fa területén) kezelője.

/// Gondoskodik a fa elmozdításáról.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void elLomb\_MouseMove(object sender, MouseEventArgs e){

//Ha a bal egérgomb le van nyomva

if(e.LeftButton == MouseButtonState.Pressed){

//Lekérdezzük az egér helyzetét double ujx = e.GetPosition(cvLap).X; double ujy = e.GetPosition(cvLap).Y;

//Kiszámoljuk, hogy mennyit mozdult el az előző pozícióhoz

//képest

double dx = ujx - x; double dy = ujy - y;

//Lekérdezzük a lombot befoglaló téglalap bal felső

//sarkának helyzetét. double lombx =

(double)elLomb.GetValue(Canvas.LeftProperty); double lomby = (double)elLomb.GetValue(Canvas.TopProperty);

//Elmozgatjuk a lombot elLomb.SetValue(Canvas.LeftProperty, lombx + dx); elLomb.SetValue(Canvas.TopProperty, lomby + dy);

//Lekérdezzük a fatörzset befoglaló téglalap bal felső

//sarkának helyzetét. double torzsx =

(double)rcTorzs.GetValue(Canvas.LeftProperty); double torzsy = (double)rcTorzs.GetValue(Canvas.TopProperty);

//Elmozgatjuk a torzset rcTorzs.SetValue(Canvas.LeftProperty, torzsx + dx); rcTorzs.SetValue(Canvas.TopProperty, torzsy + y);

//Tároljuk az egér aktuális helyzetét x = ujx;

y = ujy;

}

}

## Egyénileg megoldandó feladat

* + Készítsen a programból ablakot, ajtót és kéményt a házhoz.
  + Egészítse ki a gyorsmenüt “Hátra visz” menüponttal, és valósítsa meg a hozzá tartozó funkcionalitást.