Kétdimenziós rajzolás WPF-ben

A grafikus megjelenítés módjai WPF-ben:

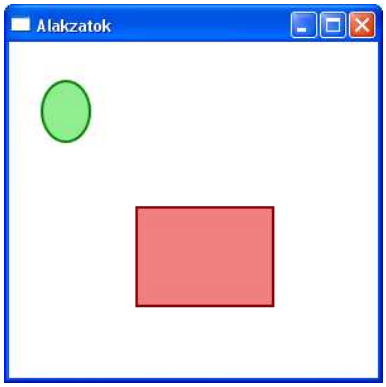
System.Windows.Shapes névtér osztályaival

- magas szintű, rengeteg metódus, tulajdonságok, eseménykezelés, input kezelés (egér, billentyűzet) → lassú; egy sor szabályos geometriai objektum (téglalap, ellipszis, stb.)
- 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 kódból is megvalósítható:

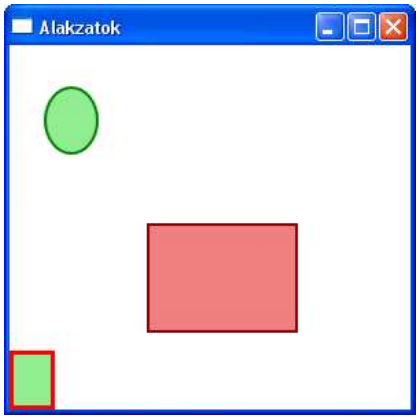
```
Double TetőMagasság=30;
Rectangle rcHáz = new Rectangle();
rcHáz.Width=110;
rcHáz.Height=80;
rcHáz.Stroke = Brushes.DarkRed;
rcHáz.StrokeThickness = 2;
rcHáz.Fill = Brushes.LightCoral;
cvLap.Children.Add(rcHáz);
rcHáz.SetValue(Canvas.LeftProperty,
(double)(100));
rcHáz.SetValue(Canvas.TopProperty,
  (double)(100+TetőMagasság));
```



- 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 absztrakt 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

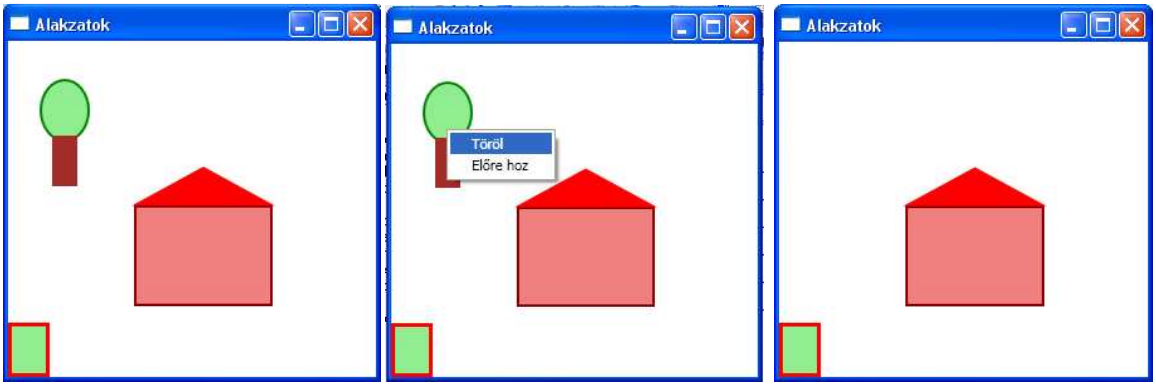
- legvékonyabb réteg → leggyorsabb; csak elemi szolgáltatások, mindenhez meg kell írni a kódot (legtöbb kódolás)
- nincs input esemény, felületmenedzser, adatkötés, alacsony szintű megközelítés
- Fontosabb osztályok: DrawingVisual, Viewport3DVisual, ContainerVisual
- Legkisebb erőforrásigény → Legjobb 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:

```
<Window x:Class="Alakzatok.wndFőablak"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentati
on" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Alakzatok" Height="300" Width="300">
  <Canvas Name="cvLap">
    <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" />
    <Rectangle Width="20" Height="40" Stroke="Brown"
      StrokeThickness="2" Fill="Brown" Canvas.Left="35"
      Canvas.Top="75" Name="rcTörzs"
      MouseDown="elLomb_MouseDown"
      MouseMove="elLomb_MouseMove"/>
  </Canvas>
</Window>
```

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 Ház(double x, double y)
{
  // A tető magassága.
```

```
  Double TetőMagasság = 30;

  // A földszintet leíró téglalap definiálása.
  Rectangle rcHáz = new Rectangle();
  rcHáz.Width = 110;
  rcHáz.Height = 80;
  // Keretvonal színe.
  rcHáz.Stroke = Brushes.DarkRed;
  // Keretvonal vastagsága.
  rcHáz.StrokeThickness = 2;
  // A fal színe.
  rcHáz.Fill = Brushes.LightCoral;
  // A ház helyzetének definiálása.
  rcHáz.SetValue(Canvas.LeftProperty, (double)(x));
  rcHáz.SetValue(Canvas.TopProperty,
    (double)(y + TetőMagasság));
  // Elhelyezés a rajzlapon.
  cvLap.Children.Add(rcHáz);

  // a tetőt leíró háromszög definiálása.
  Polygon pgTető = new Polygon();
  // Keretvonal színe.
  pgTető.Stroke = Brushes.Red;
  // Keretvonal vastagsága.
  pgTető.Fill = Brushes.Red;
  // A háromszög csúcsainak definiálása.
  pgTető.Points = new PointCollection();
  pgTető.Points.Add(new Point(x, y + TetőMagasság));
  pgTető.Points.Add(new Point(x + (rcHáz.Width / 2), y));
  pgTető.Points.Add(new Point(x + rcHáz.Width,
    y + TetőMagasság));
  // Elhelyezés a rajzlapon.
  cvLap.Children.Add(pgTető);
}
```

A gyorsmenüt létrehozó metódus:

```
/// <summary>
/// Létrehozza és a fához rendeli a gyorsmenüt
/// </summary>
private void GyorsMenüLétrehoz()
{
  // Gyorsmenü definiálása.
  ContextMenu cmGyorsMenü = new ContextMenu();
  // Töröl menüpont definiálása.
  MenuItem miTöröl = new MenuItem();
  // Megjelenő szöveg.
  miTöröl.Header = "Töröl";
  // Eseménykezelő hozzárendelése.
```

```
  miTöröl.Click += new RoutedEventHandler(miTöröl_Click);
  // Hozzáadás a gyorsmenühöz.
  cmGyorsMenü.Items.Add(miTöröl);
  // Előre hoz menüpont definiálása.
  MenuItem miElőreHoz = new MenuItem();
  // Megjelenő szöveg.
  miElőreHoz.Header = "Előre hoz";
  // Eseménykezelő hozzárendelése.
  miElőreHoz.Click +=
    new RoutedEventHandler(miElőreHoz_Click);
  // Hozzáadás a gyorsmenühöz.
  cmGyorsMenü.Items.Add(miElőreHoz);
  // Gyorsmenü hozzárendelése a lombot megvalósító
  // objektumhoz.
  elLomb.ContextMenu = cmGyorsMenü;
  // Gyorsmenü hozzárendelése a fatörzset megvalósító
  // objektumhoz.
  rcTörzs.ContextMenu = cmGyorsMenü;
}
```

Az ablak konstruktora:

```
/// <summary>
/// Az ablakosztály konstruktora. Gondoskodik az XAML-ben
/// leírt felület megjelenítéséről, a ház megrajzolásáról és a
/// gyorsmenü létrehozásáról.
/// </summary>
public wndFőablak()
{
  // Az XAML-ben leírt felület megjelenítése.
  InitializeComponent();
  // A ház megrajzolása.
  Ház(100, 100);
  // A gyorsmenü létrehozása.
  GyorsMenüLétrehoz();
}
```

A fa törlésének megvalósítása:

```
/// <summary>
/// Törli a fát alkotó két alakzat objektumot a megjelenítendő
/// objektumok listájáról.
/// </summary>
private void miTöröl_Click(object sender, RoutedEventArgs e)
{
  cvLap.Children.Remove(elLomb);
  cvLap.Children.Remove(rcTörzs);
}
```

```
/// <summary>
/// A Z-sorrend végére helyezi a fát.
/// </summary>
private void miElőreHoz_Click(object sender,
  RoutedEventArgs e)
{
  // Töröljük a fát alkotó két alakzat objektumot a
  // megjelenítendő
  // objektumok listájáról.
  cvLap.Children.Remove(elLomb);
  cvLap.Children.Remove(rcTörzs);
  // Újra felvesszük őket a lista végére.
  cvLap.Children.Add(elLomb);
  cvLap.Children.Add(rcTörzs);
}
```

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 téglalp bal felső sarkának helyzetét.
    - Elmozgatjuk a lombot.
    - Lekérdezzük a fatörzset befoglaló téglalp bal felső sarkának helyzetét.
    - Elmozgatjuk a fatörzset.
    - Tároljuk az egér aktuális helyzetét.

```
/// <summary>
/// Egérpozíció tárolására szolgáló változók.
/// </summary>
double x, y;

/// <summary>
/// Egérgomb lenyomása esemény (a fa területén) kezelője.
/// </summary>
/// <param name="e">Esemény adatai.</param>
private void elLomb_MouseDown(object sender,
MouseButtonEventArgs e)
{
  x = e.GetPosition(cvLap).X;
  y = e.GetPosition(cvLap).Y;
}
/// <summary>
/// Egérgomb mozgatása esemény (a fa területén) kezelője.
/// Gondoskodik a fa elmozdításáról.
/// </summary>
```

```
/// <param name="e">Esemény adatai.</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 újx = e.GetPosition(cvLap).X;
    double újy = e.GetPosition(cvLap).Y;
    // Kiszámoljuk, hogy mennyit mozdult el az előző
    // pozícióhoz képest.
    double dx = újx - x;
    double dy = újy - y;
    // Lekérdezzük a lombot befoglaló téglalp 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 téglalp bal felső
    // sarkának helyzetét.
    double Törzsx =
      (double)rcTörzs.GetValue(Canvas.LeftProperty);
    double Törzsy =
      (double)rcTörzs.GetValue(Canvas.TopProperty);
    // Elmozgatjuk a fatörzset.
    rcTörzs.SetValue(Canvas.LeftProperty, Törzsx + dx);
    rcTörzs.SetValue(Canvas.TopProperty, Törzsy + dy);
    // Tároljuk az egér  aktuális helyzetét.
    x = újx;
    y = újy;
  }
}

/// <summary>
/// Törli a fát alkotó két alakzat objektumot a megjelenítendő
/// objektumok listájáról.
/// </summary>
private void miTöröl_Click(object sender, RoutedEventArgs e)
{
  cvLap.Children.Remove(elLomb);
  cvLap.Children.Remove(rcTörzs);
}

/// <summary>
/// A Z-sorrend végére helyezi a fát.
/// </summary>
```

```
private void miElőreHoz_Click(object sender,
  RoutedEventArgs e)
{
  // Töröljük a fát alkotó két alakzat objektumot a
  // megjelenítendő objektumok listájáról.
  cvLap.Children.Remove(elLomb);
  cvLap.Children.Remove(rcTörzs);
  // Újra felvesszük őket a lista végére.
  cvLap.Children.Add(elLomb);
  cvLap.Children.Add(rcTörzs);
}
```

Egyénileg megoldandó feladat

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