Osnove 2D grafike

Skripta objašnjenja primere za gradivo 2. nedelje:

- **Grafičke primitive**: Drawing, Visual, Shape i
- podsećanje na upotrebu **komandi** u WPF-u

U Visual Studio projektu **Primer_1.2**, radi implementacije komandi, formirana je klasa *RoutedCommands*. U njoj su formirani objekti klase *RoutedUICommand*, koji definišu komande kroz njihov opis, ime za deklarisanje i kombinacije tastera koje će predstavljati prečicu kojom se poziva komanda. Primer definicije objekta ove klase naveden je u listingu koda ispod.

```
public static readonly RoutedUICommand Komanda = new RoutedUICommand(
      "Komanda",
      "Komanda",
      typeof(RoutedCommands),
      new InputGestureCollection()
      {
          new KeyGesture(Key.K, ModifierKeys.Control),
      }
      );
```

Listing 1. Primer definicije objekta klase RoutedUICommand

Da bi se komanda povezala sa interfejsom, koristiće se *CommandBindings* kolekcija koja je property glavnog prozora, u kojoj se definiše *CommandBinding* koji se povezuje sa samim objektom *RoutedUICommand*. U listingu ispod je navedena definicija ovog povezivanja.

```
<CommandBinding Command="local:RoutedCommands.Komanda" CanExecute="Komanda_CanExecute"
Executed="Komanda_Executed"></CommandBinding>
```

Listing 2. Definicija CommandBinding-a

Još jedan način implementacije komandi je pomoću klase *ApplicationCommands*, koja je upotrebljena u prozoru *UgradjeneKomande.xaml*. Klasa *ApplicationCommands* obezbeđuje standardni set komandi koje su dostupne u svim Windows aplikacijama, od kojih su iskorišćene Cut, Copy i Paste. Ovom dodelom komande, ona je automatski već implementirana.

Osnove 2D grafike čine **grafičke primitive**. Prva od njih je opisana u prozoru *DrawingKontrola*. Ono što će biti praksa u WPF računarskoj grafici je upotreba *Canvas*-a kao sadržaja prozora u kojem će se iscrtavati grafički elementi. Pozicije crteža se definišu u odnosu na gornji levi ugao *Canvas*-a.

Svi ovi elementi se iscrtavaju pomoću klase *Image*. Instanci klase *Image* se kao property dodeljuje property ***Source*** koji određuje šta će biti sadržaj slike. Kao sadržaj se bira objekat klase *DrawingImage*, koja predstavlja *ImageSource* objekat, čiji će sadržaj biti objekat klase *Drawing*. *Drawing* klasa može da bude predstavljena kao geometrijski crtež, crtež koji čine „glifovi“ (slova), crtež na osnovu slike ili videa ili grupa više crteža.

Geometrijski crtež se definiše kroz klasu *GeometryDrawing*. Toj klasi se definišu kao property-ji ***Geometry***, koji određuje same geometrijske oblike koji formiraju crtež, property ***Brush***, koji određuje boje kojima je crtež obojen i property ***Pen***, koji predstavlja konturu oblika.

Geometrija definiše kao jedan ili više oblika (da bi je činilo više oblika, potrebno je da se oni definišu kao kolekciju *GeometryGroup*). Jedan od oblika kojim se definiše geometrija je elipsa (klasa *EllipseGeometry*). Kao i generalno u geometriji, elipsa se definiše sa tačkom centra (***Center***) i dva poluprečnika (***RadiusX***, ***RadiusY***).

Bojenje se vrši pomoću objekata klase *Brush*, od kojih jedan može biti *LinearGradientBrush*. Gradijentna boja se definiše kroz tačke koje čine prelaze između boja. U skladu sa pravilima računarske grafike **WPF**-a, koordinatni početak se nalazi u gornjem levom uglu.

Prva tačka (0,0) se povezuje sa bojom kroz property ***Color***. Između svake dve tačke koje su vezane sa bojom formira se prelaz između tih boja. Po osnovnoj definiciji, gradijent se iscrtava po dijagonali od koordinatnog početka, osim ako se ne definišu početne i krajnje tačke, na osnovu kojih se određuje u kom pravcu se prostire gradijent.

Kada se iscrtava crtež na osnovu slike, upotrebljava se klasa *ImageDrawing*, kojoj se dodeljuje property ***ImageSource***, koji određuje putanju do slike koja se prikazuje. Uz to definiše se property ***Rect***, koji određuje pravougaonu površinu (npr, 100x100) koja će biti popunjena datom slikom.

Crteži koje čine glifovi (slova) se iscrtavaju pomoću objekata klase *GlyphRunDrawing*, koji omogućava renderovanje teksta kao slike. Jedan od property-ja za koji se dodeljuje vrednost je ***ForegroundBrush***, koji određuje boju teksta koji se „iscrtava“. Još jedan property je objekat klase ***GlyphRun***, kojim se određuje koji se tekst „iscrtava“ i kako.

Najvažniji property-ji ***GlyphRun*** definicije su ***GlyphIndices***, koji predstavlja brojevni kod svakog slovnog znaka. Da bi se uneli brojevni kodovi, postoji tabela u kojoj su oni definisani za svaki karakter (**Slika 1**).

	1	2	3	4	5	6	7	8	9	10
0x				!	"	#	$	%	&	'
1x	(	)	*	+	,	-	.	/	0	1
2x	2	3	4	5	6	7	8	9	:	;
3x	<	=	>	?	@	A	B	C	D	E
4x	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y
6x	Z	[	\	]	^	_	`	a	b	c
7x	d	e	f	g	h	i	j	k	l	m
8x	n	o	p	q	r	s	t	u	v	w
9x	x	y	z	{	|	}	~			

Slika 1. Tabela sa brojevnim kodovima za svaki karakter

Property ***FontRenderingEmSize***, predstavlja koliko će prostora po širini biti zauzeto na ekranu za slovo „M“ koje je najširi karakter. ***AdvanceWidths*** property će služiti za definiciju koliko prostora na ekranu će biti zauzeto za svaki karakter koji je definisan preko ***GlyphIndices***. Pored toga, da bi se tekst prikazao, još jedan property ***GlyphRun*** objekta je ***GlyphTypeface*** za koji se definiše ***FontUri***, koji predstavlja putanju do fonta kojim će tekst biti ispisan.

Grupa više crteža se definiše pomoću klase ***DrawingGroup***. U sadržaju objekta se navode ***Drawing*** objekti, kao oni objajšnjeni u prethodnim pasusima. Sekvenca navođenja ovih objekata će biti i redosled u kojem se iscrtavaju, odnosno koji objekat će biti iscrtan preko kog.

Još grafičkih primitiva je definisano u prozoru *ShapeKontrola*. Linije se iscrtavaju pomoću klase ***Line***, kojoj se dodeljuju pozicije početne i krajnje tačke linije (***X1,Y1*** i ***X2,Y2***), boja (***Stroke***) i debljina (***StrokeThickness***).

Elipsa se može iscrtati i pomoću klase ***Ellipse***. Njoj se kao property-ji definišu boja površine (***Fill***), boja konturne linije (***Stroke***), debljina konturne linije (***StrokeThickness***) i visina (***Height***) i širina (***Width***), koje će se manifestovati kao prečnici elipse.

Još jedna mogućnost iscrtavanja je upotreba klase ***Polygon***, koji se manifestuje kao mnogougao. Da bi se on iscrtao, definišu se tačke između kojih se iscrtavaju linije (property ***Points***). Najmanja površina koja se može nacrtati pomoću ovog objekta je trougao (4 tačke).

VisualKontrola i *VisualAdvanced* prezentuju klasu ***VisualBrush***, koja omogućava renderovanje svih elemenata smeštenih u njenom property-ju tipa ***Visual***. Ovo za posledicu ima da se interakcijski elementi prikazuju kao renderi, odnosno nemaju mogućnost interakcije već su samo slike. Visual može biti bilo kakav XAML kod, odnosno bilo kakva XAML hijerarhija elemenata.

Na nju se, kao i na svaki grafički objekat, mogu primeniti transformacije (više na sledećem terminu). Da bi se objekat okrenuo naopako (kao u ogledalu) primenjuje se transformacija skaliranja i zavisno od smera u kojem treba da se prostire slika u ogledalu po toj osi se kao parametar skaliranja prosleđuje vrednost -1, dok po drugoj osi ostaje vrednost 1 kako bi vrednost ostala ista.

Nad renderovanim objektima može se na *Alpha kanalu* (određuje providnost boje) dodati *OpacityMask*. Ova maska može da se upotrebi kao još jedan sloj boje nad renderovanim objektom i očekuje instancu klase ***Brush***, koja se može definisati kroz sve klase koje je nasleđuju (***GradientBrush***, ***ImageBrush***...).

Objektima se mogu dodeljivati i efekti, gde je jedna opcija ***BlurBitmapEffect*** i simulira da se objekat gleda kao kroz nefokusirani objektiv. Property ***Radius*** određuje koliko je veliko zamućenje.