

### Alkalmazott Modul III

#### 9. gyakorlat

#### Eseményvezérelt programozás: elemi grafika és egérkezelés

© 2011.11.22. Giachetta Roberto  
groberto@inf.elte.hu  
http://people.inf.elte.hu/groberto

#### Elemi grafika és egérkezelés

##### Rajzolás

- A grafikus felület lehetőséget biztosít 2 dimenziós rajzolásra
  - egyszerű alakzatokat (vonal, kör, szöveg, ...), vagy képeket rajzolhatunk bármely felületre az ablakunkban
- A rajzolásért külön rajzoló osztály (**Graphics**) felel
  - körvonalas alakzatot a **Draw<alakzat>** művelettel készíthetünk, a vonal elkészítéséhez egy tollra (**pen**) van szükségünk
    - lehetőségünk van szöveget kirajzolni a **DrawString** utasítással, ekkor meg kell adnunk a szöveg betűtípusát (**Font**)
  - kitöltött alakzatot a **Fill<alakzat>** művelettel készíthetünk, ehhez egy ecset (**Brush**) szükséges

#### Elemi grafika és egérkezelés

##### Színkezelés

- A rajzoláshoz minden esetben szükséges egy szín (**Color**), amely lehet előre definiált (pl. **Color.White**), vagy a programozó által definiált a piros, kék, zöld színek komponensek alapján (pl. **Color.FromArgb(120, 176, 21)**)
  - ecset létrehozásakor használhatjuk az előre definiált ecsetszíneket (pl. **Brushes.White**)
- pl. egy sárgával kitöltött, piros szélű 100\*100-as négyzet kirajzolása a (10,0) pontba:

```
Graphics gr = <rajzobjektum létrehozása>;
gr.FillRectangle(Brushes.Yellow, 10, 0, 100, 100);
Pen red_pen = new Pen(Color.Red, 2);
// piros toll, ami 2 vastag
gr.DrawRectangle(red_pen, 10, 0, 100, 100);
```

#### Elemi grafika és egérkezelés

##### Rajzobjektumok használata

- A rajzolóobjektumot mindig valamilyen vezérlőre, vagy képre állítjuk rá
  - a direkt rajzolásra készített vezérlő a **Panel**, ennek **Paint** eseménykezelőjében elvégezhetjük a műveleteket, és a rajzobjektumot elérhetjük az eseményparaméterekkel (**e.Graphics**), vagy a panelen keresztül (**Panel.CreateGraphics()**), ekkor a rajzoláshoz frissíteni kell a panelt (**Refresh**)
  - képre (**Bitmap**) rajzolás esetén a képre kell létrehozni a rajzoló (**Graphics.FromImage(<kép objektum>)**), ezt a képet később kirajzolhatjuk a képernyőre (**DrawImage**)
- Ügyeljünk arra, hogy az új rajzot mindig az előzőre készíti el, ezért a háteret időnként újra kell festeni pl. fehér téglalappal

#### Elemi grafika és egérkezelés

##### Rajzobjektumok használata

- Pl.:

```
public DrawForm(){
    ...
    _Panel.Paint +=
        new PaintEventHandler(Panel_Paint);
}
void Panel_Paint(object sender, PaintEventArgs e){
    Graphics gr = e.Graphics;
    // vagy gr = _Panel.CreateGraphics();
    gr.FillRectangle(Brushes.White, 0, 0,
        _Panel.Width, _Panel.Height);
    // a háteret fehérre festjük
    //... további alakzatok rajzolása
}
```

#### Elemi grafika és egérkezelés

##### Feladatok

1. Rajzoljuk egy piros kitöltésű, kék szélű téglalapot a képernyőre.
2. Módosítsuk az előző programot úgy, hogy a két szín váltakozzon egérekattintásra.
3. Módosítsuk az előző programot úgy, hogy a téglalap mindig az egérpozícióban jelenjen meg.
4. Készítsünk egy olyan programot, amely véletlenszerűen szigeteket rajzol ki a képernyőre, valamint minden sziget csúcsára egy zászlót helyez.
  - a sziget generálást és a zászló állítást külön gombnyomásra végezzük, a domborzatot zöld, a tengerszintet kék, a zászlót piros színnel rajzoljuk

## Elemi grafika és egérkezelés

### Feladatok

5. Készítsünk egy olyan programot, amellyel alakzatokat (egyenlőszárú háromszög, négyzet, ellipszis) tudunk rajzolni a képernyőre úgy, hogy az alakzatot az egér lenyomására kezdjük el rajzolni és az egér felengedésével helyezzük el.
  - rajzolás után az alakzat színe a típusnak megfelelően változik
  - legyen egy külön gomb, amellyel kitörölhetjük a képernyő tartalmát
6. Javítsunk az előző program teljesítményén úgy, hogy nem közvetlenül a képernyőre, hanem egy képre rajzolunk, majd ezt helyezzük a képernyőre.