



Evropská obchodní akademie, Děčín I, Komenského náměstí 2, příspěvková
organizace
IČ 47274611 tel. 412 516 127 email: skola@oadc.cz

Registrační číslo projektu	CZ.1.07/1.5.00/34.0999
Název projektu	Zlepšení podmínek pro vzdělávání na EOA
Šablona	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Sada	Aplikovaná informatika 1. ročník
Označení materiálu	VY_32_INOVACE_19_20
Název materiálu	Timer
Předmět a ročník	AIT, 1.
Autor materiálu	Jan Horais
Stručná anotace	Materiál seznamuje studenty s komponentou Timer.
Metodický pokyn	Materiál je určen jako podpora výkladu do hodiny.
Datum tvorby	20. 4. 2013
Datum ověření ve výuce	29. 4. 2013
Třída	1.A, MIT

Timer

1 Úvod

Pokud potřebujeme vytvořit animaci, případně opakovat konkrétní činnost, využijeme komponentu `Timer`, (časovač). Komponenta `Timer` je součástí záložky *Toolbox* → **Components**. Časovač přetáhneme myší do okna programu a pustíme. Protože se jedná o komponentu, která není v běžícím programu vidět, umístila se na pruh pod oknem aplikace. Když máme `Timer` umístěný v programu, můžeme ho použít. Vytvoříme blikající kolečko, které se bude náhodně objevovat v okně programu. Postup bude následující:

- založíme „generátor“ náhodných čísel,
- vygenerujeme náhodné souřadnice x a y ,
- nakreslíme kolečko na souřadnicích x a y ,
- do události `Tick` komponenty `Timer` napíšeme `Refresh()`, to zajistí znovu vykreslení kolečko na nových souřadnicích.

Z praktických důvodů nebudeme kreslit přímo do okna programu, ale do komponenty `Panel` z toolboxu `Containers`. Zdrojový kód příkladu je zobrazen jako Kód 1.

Kód 1: Blikající kolečko.

```
1 //založíme generátor náhodných čísel
2 Random NahCisl = new Random();
3
4 private void panel1_Paint(object sender, PaintEventArgs e)
5 {
6     //vygeneruje x a y, maximální x a y je šířka a výška panelu
7     //zmenšená o 25, což je průměr kresleného kolečka
8     int x = NahCisl.Next(panel1.Width);
9     int y = NahCisl.Next(panel1.Height);
10    //založíme plátno na které budeme kreslit kolečko
11    Graphics tabule = e.Graphics;
12    //nakreslíme kolečko
13    tabule.FillEllipse(Brushes.Red, x, y, 25, 25);
14 }
15
16 //komponeta timer opakovaně překresluje okno
17 private void timer1_Tick(object sender, EventArgs e)
```

```

18     {
19         Refresh();
20     }

```

Pokud tento kód spustíte, nakreslí se po spuštění jedno kolečko a to bude vše. Chyba je prostá. Časovač `timer1` musíme spustit (nechat tikat). To uděláme tak, že vlastnost `Enabled` nastavíme na `True`. Po spuštění se začne opakovaně překreslovat `panel1` a s ním i kolečko. K úplnému popisu komponenty `Timer` zbývá poslední věc. Jak často (rychle) se `panel1` překresluje? Rychlost se nastavuje ve vlastnosti `Interval` a její hodnota je v milisekundách. To znamená, že při nastavení `Interval=1000` se překreslení provede jednou za sekundu, při nastavení `Interval=250` se překreslení provede čtyřikrát za sekundu.

Náš program ještě trochu vylepšíme. Přidáme tlačítko, které spustí/ukončí překreslování kolečka (viz Kód 2).

Kód 2: Stopnutí/spuštění Timeru.

```

1 private void button1_Click(object sender, EventArgs e)
2     {
3         if (timer1.Enabled == true)
4         {
5             timer1.Enabled = false;
6         }
7         else
8         {
9             timer1.Enabled = true;
10        }
11    }

```

Příklad v Kódu 2 je fakticky správně a dělá přesně co chceme, ale je trochu „ukecený“. Stejnou funkčnost nám poskytne i Kód 3.

Kód 3: Stopnutí/spuštění Timeru – další možnost.

```

1 private void button1_Click(object sender, EventArgs e)
2     {
3         timer1.Enabled = !timer1.Enabled;
4     }

```

2 Cvičení

Vytvořte následující programy:

- Kulička se pohybuje zleva doprava a zpět.
- Kulička se začne pohybovat ze středu panelu. náhodným směrem

Kód 4: Jezdící kulička doprava a zpět.

```

1      //založíme náhodné číslo , pozici kuličky a o kolik se má posunout ↵
      a její poloměr
2      Random nahCisl = new Random();
3      int kulickaX = 0;
4      int kulickaY = 0;
5      int zmena = +1;
6      int polomer = 25;
7
8      //ošetříme posun kuličky
9      private void timer1_Tick(object sender, EventArgs e)
10     {
11         //posunu x-ovou souřadnici o zmena
12         kulickaX = kulickaX + zmena;
13         //zkontroluji jestli x-ova souřadnice není větší než rozměr ↵
            panelu
14         //pokud ano, bude se pohybovat zpět
15         if (kulickaX >= (panel1.Width - polomer))
16             zmena = -1;
17         //zkontroluji jestli x-ova souřadnice není menší než 0
18         //pokud ano, bude se pohybovat vpřed
19         if ((kulickaX) <= 0)
20             zmena = +1;
21         //znovu vykreslíme okno programu
22         Refresh();
23     }
24     //vykreslíme kuličku
25     private void panel1_Paint(object sender, PaintEventArgs e)
26     {
27         Graphics tabule = e.Graphics;
28         tabule.FillEllipse(Brushes.Red, kulickaX, kulickaY, ↵
            kulickaPlomer, kulickaPlomer);
29     }

```

Kód 5: Náhodný směr.

```

1      //založíme náhodné číslo , pozici kuličky, o kolik se má posunout ↵
      na ose X a Y a její poloměr
2      Random NahCisl = new Random();
3      int kulickaX, kulickaY = 0;
4      int zmenaX, zmenaY = 0;
5      int polomer = 25;
6
7      //Nastavíme kuličku na střed Panelu, vygenerujeme posun na ose X ↵
      a Y a spustíme Timer
8      private void btnStart_Click(object sender, EventArgs e)
9      {

```

```

10         kulickaX = panel1.Width / 2;
11         kulickaY = panel1.Height / 2;
12         zmenaX = NahCisl.Next(-10, 11);
13         zmenaY = NahCisl.Next(-10, 11);
14         timer1.Enabled = true;
15         Refresh();
16     }
17     //Ošetříme posun kuličky a zkontrolujeme, jestli neujela z Panelu
18     private void timer1_Tick(object sender, EventArgs e)
19     {
20         kulickaX += zmenaX;
21         kulickaY += zmenaY;
22         if (kulickaY <= 0)
23             timer1.Enabled=false;
24         if (kulickaX <= 0)
25             timer1.Enabled=false;
26         if (kulickaY >= (panel1.Height-polomer))
27             timer1.Enabled = false;
28         if (kulickaX >=(panel1.Width-polomer))
29             timer1.Enabled = false;
30         Refresh();
31     }
32
33     //vykreslíme kuličku
34     private void panel1_Paint(object sender, PaintEventArgs e)
35     {
36         Graphics tabule = e.Graphics;
37         tabule.FillEllipse(Brushes.Red, kulickaX, kulickaY, ←
38             kulickaPlomer, kulickaPlomer);

```

V událost timer1_Tick při kontrole jestli už kulička neujela z panelu můžeme použít i jedno if, viz Kód 6

Kód 6: Náhodný směr – úprava podmínky.

```

1     private void timer1_Tick(object sender, EventArgs e)
2     {
3         kulickaX += zmenaX;
4         kulickaY += zmenaY;
5         if ((kulickaY <= 0)|| (kulickaX <= 0)|| (kulickaY >= (panel1.↵
6             Height-polomer))|| (kulickaX >=(panel1.Width-polomer)))
7             timer1.Enabled = false;
8         Refresh();
9     }

```

Literatura

- [1] VYSTAVĚL, Radek. *Moderní programování učebnice pro začátečníky*. moderníProgramování s.r.o. 2009, ISBN 978-80-903951-6-9
- [2] MICROSOFT CORPORATION *Microsoft Visual C# 2010 Express* [software]. 25. listopadu 2012 [2012-11-25]. Dostupné z: <http://www.microsoft.com/visualstudio/cze/downloads#d-2010-express>. Požadavky na systém: Windows 7 (x86 a x64), Windows 8 (x86 a x64), Windows Server 2008 R2 (x64), Windows Server 2012 (x64)

Použitý SW

1. SPENCER KIMBALL, PETER MATTIS A VÝVOJOVÝ TÝM GIMP *Gimp 2.8.2* [software]. 25. listopadu 2012 [2012-11-25]. Dostupné z: [http://sourceforge.net/projects/gimp-win/files/GIMP%20%2B%20GTK%2B%20\(stable%20release\)/GIMP%202.8.2/gimp-2.8.2-setup-1.exe/download?accel_key=71%3A1354392469%3Ahttp%253Awww.gimp.org/downloads/%3A2e4c478d%24a9f1db49803bd24b7e55cd50aecdc56e10bd140a&click_id=c80d2e80-3bf2-11e2-afaf-0200ac1d1d92&source=accel](http://sourceforge.net/projects/gimp-win/files/GIMP%20%2B%20GTK%2B%20(stable%20release)/GIMP%202.8.2/gimp-2.8.2-setup-1.exe/download?accel_key=71%3A1354392469%3Ahttp%253Awww.gimp.org/downloads/%3A2e4c478d%24a9f1db49803bd24b7e55cd50aecdc56e10bd140a&click_id=c80d2e80-3bf2-11e2-afaf-0200ac1d1d92&source=accel). Požadavky na systém: Windows XP SP3 a novější.
2. TOOLSCENTER *TeXnicCenter 1.0* [software]. 25. listopadu 2012 [2012-11-25]. Dostupné z: <http://texniccenter.en.softonic.com/download>. Požadavky na systém: (La)TeX, Windows XP a novější.
3. MICROSOFT CORPORATION *Microsoft Visual C# 2010 Express* [software]. 25. listopadu 2012 [2012-11-25]. Dostupné z: <http://www.microsoft.com/visualstudio/cze/downloads#d-2010-express>. Požadavky na systém: Windows 7 (x86 a x64), Windows 8 (x86 a x64), Windows Server 2008 R2 (x64), Windows Server 2012 (x64)