



Razvoj igre »Božiček«

Seminarska naloga: Programiranje 3

Avtor: Urban Celarc

Mentor: viš. pred. mag. Matija Lokar

Asistent: dr. Matjaž Zaveršnik

Ljubljana, marec 2017

Opis ideje

Odločil sem se, da bom pri predmetu Programiranje 3 izdelal igrico za enega igralca. Ker je bil takrat, ko smo dobili temo za seminarsko nalogo božični čas, sem si rekel, da bi izdelal igrico na to temo. Igrica se imenuje Božiček. Igralec, ki igra igro je Božiček na smučeh se smuča levo ter desno po zaslonu ter pobira darila, ki padajo iz neba. Ob tem, ko pobere darilo se mu poveča rezultat za eno točko. Če pa igralec izgubi oz. darila ne pobere pa izgubi življenje. Na začetku ima na voljo 3 življenja. Poleg daril pa iz neba padajo tudi bombe. Če igralec pobere bombo, izgubi vsa življenja in je končal z igro.

Igra je z Božičkom in zimsko idilo v ozadju primerna za zimski čas. Seveda jo lahko samo z zamenjavo 'figur'- slik spremenimo in je aktualna skozi celo leto.



Slika1: Osnovna ideja

Prvi koraki

Igrico sem napisal z programom Visual Studio 2015. In sicer v 'pod programu' Windows Form, ki je namenjena programiranju z Windows okni. Najprej sem narisal okno primerne velikosti. Oknu sem dodal ozadje in na njega naklikal objekte, ki jih bom potreboval pri igri:

Objekti so tipa:

PictureBox

- Figura, ki se bo premikala levo in desno (Božiček)
- Objekt 1, ki bo padal z neba in ga moramo pobrati (darilo)
- Objekt 2, ki pada iz neba in ga ne smemo pobrati (bomba)
- Trije objekti, ki so življenja

Label

- Rezultat
- Navodila za igranje igrice
- Sporočilo ob koncu in izpis rezultata
- Lestvica

Button

- Gumb za začetek igre
- Gumb za pavzo
- Gumb za ponovno igro
- Gumb OK, za potrditev imena

TimeTick

- Čas za premikanje Figure levo in desno
- Čas za padanje daril
- Čas za padanje bombe

TextBox

- Text box za vnos imena igralca

Premikanje figure – Božička

Figuro bomo premikali levo in desno. Višina, ki jo ima figura pa ostane nespremenjena. Torej spreminjamo samo x- koordinato. Da nam figura ne pobegne iz zaslona moramo določiti meje.

Premikanje sem določil z dvema »bool« spremenljivkama (levo in desno). Dokler so označene z 'true' se figura premika. Spremenljivka pa je 'true', dokler držimo tipko za premik. V našem primeru sta to tipki 'a' in 'd' ali pa na Num-Lock števili '4' in '6'. (Sprva sem imel nastavljeno na tipki → in ←, vendar sta mi iz neznanega razloga prenehali delovati.) Skratka, ko pritisneš na tipko se »bool« nastavi na 'true' in ko jo spustiš se nastavi na 'false'. Če smo figuro zapeljali čez mejo se prav tako nastavi na 'false'.

Sedaj pa povejmo štoparici za figuro, da dokler je 'bool' enak 'true' figuri spreminja položaj. Če pritisnemo tipko za v levo, figuri odštevamo neko vrednost, če pa pritisnemo tipko za v desno pa prištevamo. S tem, ko prištejemo neko vrednost spremenimo lokacijo figure v levo oz. desno.

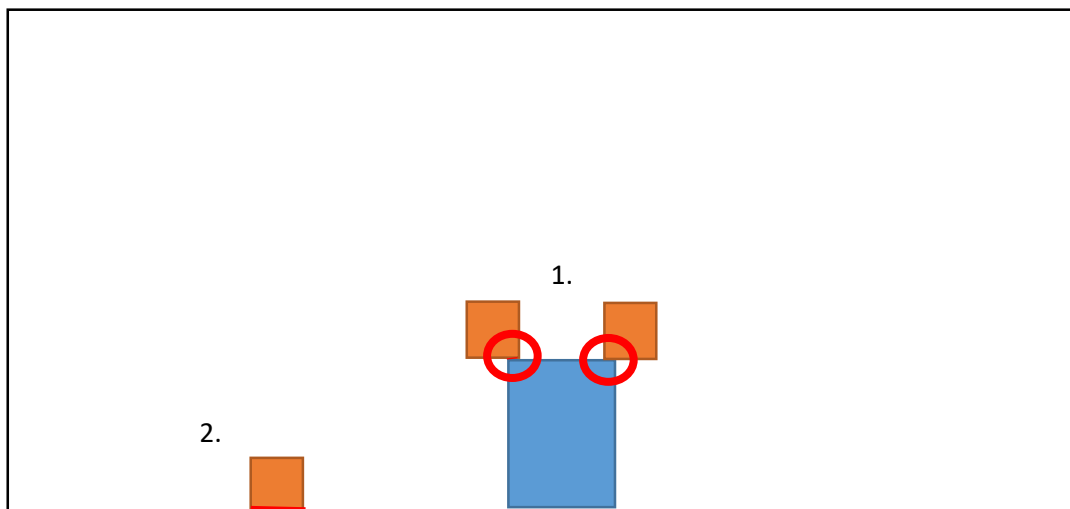
```
//premikanje človečka
private void časBožička_Tick(object sender, EventArgs e)
{
    //da božiček ne zapusti okna
    if (bozicek.Left < ozadje.Left+200) { levo = false; }
    if (bozicek.Right > ozadje.Right - 200) { desno = false; }
    //premik
    if (levo) { bozicek.Left -= hitrostBoz; }
    if (desno) { bozicek.Left += hitrostBoz; }
}
//pritisk na gumb
private void okno_KeyDown(object sender, KeyEventArgs e)
{
    if(e.KeyCode == Keys.A || e.KeyCode==Keys.NumPad4) { levo = true; bozicek.Image =
Properties.Resources.bozicekL; }
    if(e.KeyCode == Keys.D || e.KeyCode == Keys.NumPad6) { desno = true; bozicek.Image =
Properties.Resources.bozicekD; }
}
//ko gumb spustiš
private void okno_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.A || e.KeyCode == Keys.NumPad4) { levo = false; }
    if (e.KeyCode == Keys.D || e.KeyCode == Keys.NumPad6) { desno = false; }
}
}
```

Padanje objektov iz 'neba'

Na podoben način kot za premikanje človečka naredimo pri padanju objektov. Razlika je v tem, da se te objekti premikajo v vertikalni smeri. V našem primeru so objekti darila in bomba. Na vsak 'tick' štoparice, spremenimo lokacijo darila »darilo.Top += hitrostDaril;« S tem dosežemo, da darilo pada.

Sedaj pa pogledamo, če je Božiček pobral darilo. Torej pogledamo, če so koordinate figure enake koordinatam objekta:

```
1. if (darilo.Top > bozicek.Top - darilo.Height  
    && darilo.Left + darilo.Width > bozicek.Left  
    && darilo.Left < bozicek.Left + bozicek.Width)
```



Če ga pobere, prištejemo točko, povečamo hitrosti, ter postavimo objekt (novo darilo) na novo lokacijo. To je nekam nad zaslon in naključno nekam, levo ali desno:

```
darilo.Top = ozadje.Top - darilo.Top;  
darilo.Left = nakl.Next(ozadje.Left + 300, ozadje.Right - 300);
```

2. Če pa darila ni pobral, torej če je lokacija darila pod figuro, izgubimo življenje. Če je življenj enako 0 potem je konec igre, drugače pa odštejemo življenje in darilo postavimo na novo lokacijo, kot je opisano zgoraj.

Bomba

Na zelo podoben način delamo pri bombi. Edina razlika je, da pri pobiranju bombe izgubimo življenje, če pa pade 'na tal' pa se postavi nekam visoko nad ozadje.

```
//če je bomba na tleh  
if (bomba.Top + bomba.Height > ozadje.Height)  
{  
    //naključna višina bombe  
    bomba.Top = ozadje.Top - nakl.Next(800, 2500);  
    bomba.Left = nakl.Next(ozadje.Left + 300, ozadje.Right - 300);  
}
```

Začetek in konec igre

Za začetek in konec igre sem napisal pomožne funkcije, ker jih na različnih delih kode večkrat pokličem.

Začetek:

To funkcijo pokličemo, ko kliknemo start gumb, ter ko znova igramo igro.

V njej se skrivajo vsi objekti, ki jih ne potrebujemo pri igri (gumb start, gumb znova, ...) in pokažejo vsi, ki jih potrebujemo (Božiček, darilo, bomba, rezultat, ...).

Ob začetku zaženemo tudi vse štoparice.

Ter poenostavimo vse vrednosti, kot sta hitrosti daril in premikanje Božička, življenje in rezultat na začetne vrednosti.

Lokacijo darila in bombe pa postavimo nekam nad okno.

```
private void zacetekIgre()
{
    //nastavimo vse objekte, da so vidni
    bozicek.Visible = true;
    rezultat.Visible = true;
    bomba.Visible = true;
    darilo.Visible = true;

    //oziroma nevidni
    START.Visible = false;
    konec.Visible = false;
    znova.Visible = false;

    //zaženemo vse štoparce, da se objekti premikajo
    časBožička.Start();
    padanjeDaril.Start();
    padanjeBombe.Start();

    //poenostavimo življenja in rezultat
    zivljenja = 3;
    stTock = 0;
    zivljenjeSlike(zivljenja);
    rezultat.Text = stTock.ToString();

    //hitrost gre nazaj na začetno
    hitrostBoz = zacHitrostBoz;
    hitrostDaril = zacHitrostDaril;

    //darilo in bombo postavimo nekam nad okno
    darilo.Left = nakl.Next(ozadje.Left + 300, ozadje.Right - 300);
    darilo.Top = ozadje.Top;

    bomba.Top = ozadje.Top - nakl.Next(800, 2500);
    bomba.Left = nakl.Next(ozadje.Left + 300, ozadje.Right - 300);
}
```

Konec:

Konec igre se zgodi, ko izgubimo vsa tri življenja ali ko pobremo bombo.

S klicanjem te funkcije se skrijejo vsi objekti potrebni pri igri (darilo, Božiček, ...). Prikaže se pa gumb za ponovno igro, ter rezultat.

Štoparice ustavimo.

Izpišemo rezultat.

```
private void konecIgre()
{
    //ustavimo stoparice
    časBožička.Stop();
    padanjeDaril.Stop();
    padanjeBombe.Stop();

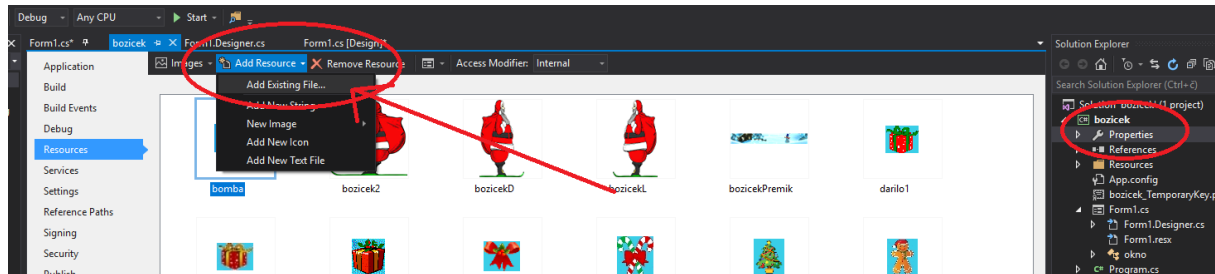
    //prikažemo/skrijemo objekte
    božicek.Visible = false;
    konec.Visible = true;
    znova.Visible = true;
    rezultat.Visible = false;
    bomba.Visible = false;
    darilo.Visible = false;
    // izpis rezultata
    konec.Text = konec.Text + "\n" + stTock.ToString();

    //slike življenj se izbrišejo
    zivljenjeSlike(0);

    //zvok ho ho ho
    zvokHoHo.Play();
}
```

Resources files

Igrica bo vsebovala več slik in zvokov. Te datoteke, ki sem jih uvozil sem shranil v 'resources'. Kako sem prišel do tega? Z desnim klikom na **Properties** → **Open** → **Add resource**.



Zakaj shraniti datoteke v resource files? Zato, da lahko programirash na katerem koli računalniku, in te datoteke 'nosiš sabo' s programom. Saj, če navedeš samo pot, kjer imaš sliko shranjeno, se na drugem računalniku nebo odprla, saj je lahko drugačna pot do te datoteke, ali pa jo sploh ni na računalniku.

Vse slike sem z programom C# ustrezno pomanjšal, jim odrezal ozadje in jih prezrcalil, če je bilo potrebno.

V resources file sem shranil tudi zvočne datoteke.

Menjava slik

Ker se za darila menjava slike, sem za to napisal funkcijo, ki naključno vrača sliko. Vse slike sem uvozil v seznam slik. Potem za z random funkcijo dobil naključni indeks med začetkom in koncem seznama. Ter vrnil to sliko.

Zvok

V igrico sem dodal tudi zvok. Zvok sem shranil v resource files. Za predvajanje zvoka sem uporabil paket »System.Media«. Zvok pa je razreda »SoundPlayer«. Se pravi, da nov zvočni efekt uvoziš nekako takole:

```
private SoundPlayer zvok1 = new SoundPlayer(Properties.Resources.kaplja);
```

Zvok predvajaš z metodo: `zvok1.Play();` , ustaviš pa z `zvok1.Stop();`

Življenja

Za prikaz življenj – srčkov desno zgoraj, sem napisal svojo funkcijo, ki za parameter dobi število življenj. Funkcija ne vrača nič, ampak samo prikazuje objekte oz. jih skriva. Poklicana je ob izgubi življenja ali pa novi igri.



Lestvica

Na koncu sem dodal še lestvico z rezultati. Ko izgubiš igro se ti pojavi napis konec in število doseženih točk. Ob tem napisu pa je tudi TextBox, kamor vpišeš svoj ime . S klikom na gumb OK, se tvoj ime postavi na pravo mesto v lestvici.

Podatki z imenom in številom točk so shranjeni v seznamu dvojic (niz 'ime', število točk). Ko kliknemo na gumb OK, se podatek doda v seznam. Potem seznam uredimo po drugi komponenti in obrnemo, da je seznam urejen padajoče. Nato pa v zanki 'foreach' dobljene podatke izpišemo v niz. Niz pa izpišemo na zaslon.

Viri:

(vse spletne strani so bile nazadnje obiskane 18. 3. 2017)

- <https://www.freesound.org/>
- [https://msdn.microsoft.com/en-us/library/7k989cfy\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/7k989cfy(v=vs.90).aspx)
- <https://stackoverflow.com/>
- <https://images.google.com/>
- https://www.youtube.com/results?search_query=c%23+game+
- <https://www.icoconverter.com/>

... in še mnogo drugih spletnih strani, ki so mi pomagale pri reševanju tekočih problemov