

İçindekiler

1. FlappyBird Oyunu.....	2
1.1 Amaç.....	2
2. Proje Oluşturma.....	2
3. Elementler.....	4
3.1 Form.....	4
3.2 PictureBox.....	8
3.3 Timer.....	21
3.4 Panel.....	22
3.5 Label.....	24
3.6 Button.....	31
3.7 CustomFont.....	35
3.8 Hiyarşı.....	42
3.9 Gizlilik.....	44
3.10 Ses Dosyaları.....	46
4. Kodlama.....	47
4.1 Global Değişkenler.....	49
4.2 PipesReset.....	50
4.3 PlaySound.....	50
4.4 FormLoad.....	51
4.5 From KeyDown.....	52
4.6 Form KeyUp.....	53
4.7 StartGame.....	55
4.8 btnStart.....	56
4.9 btnRestart.....	59
4.10 EndGame.....	62
4.11 TimerTick.....	63

1. FlappyBird Oyunu

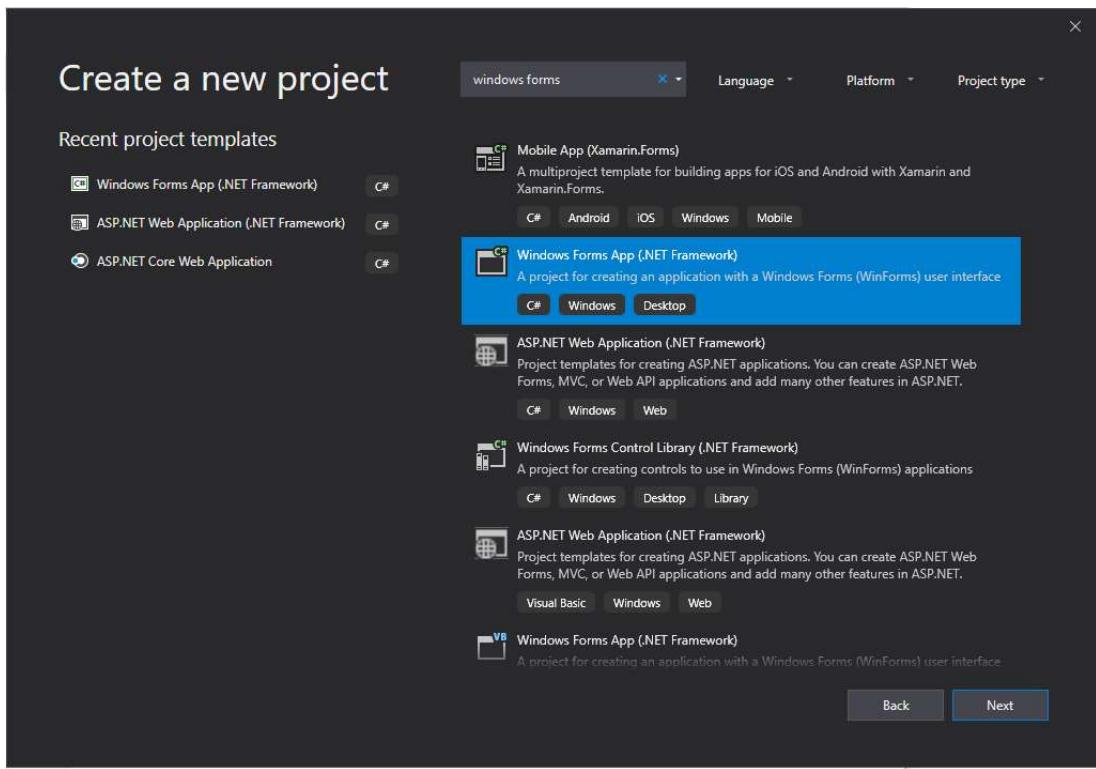
C# Windows Forms ile masaüstü oyun programıdır. Bu program ile resim ve ses nesnelerinin bir oyun algoritması içerisinde nasıl kullanılabildeği anlatılmıştır.

1.1 Amaç

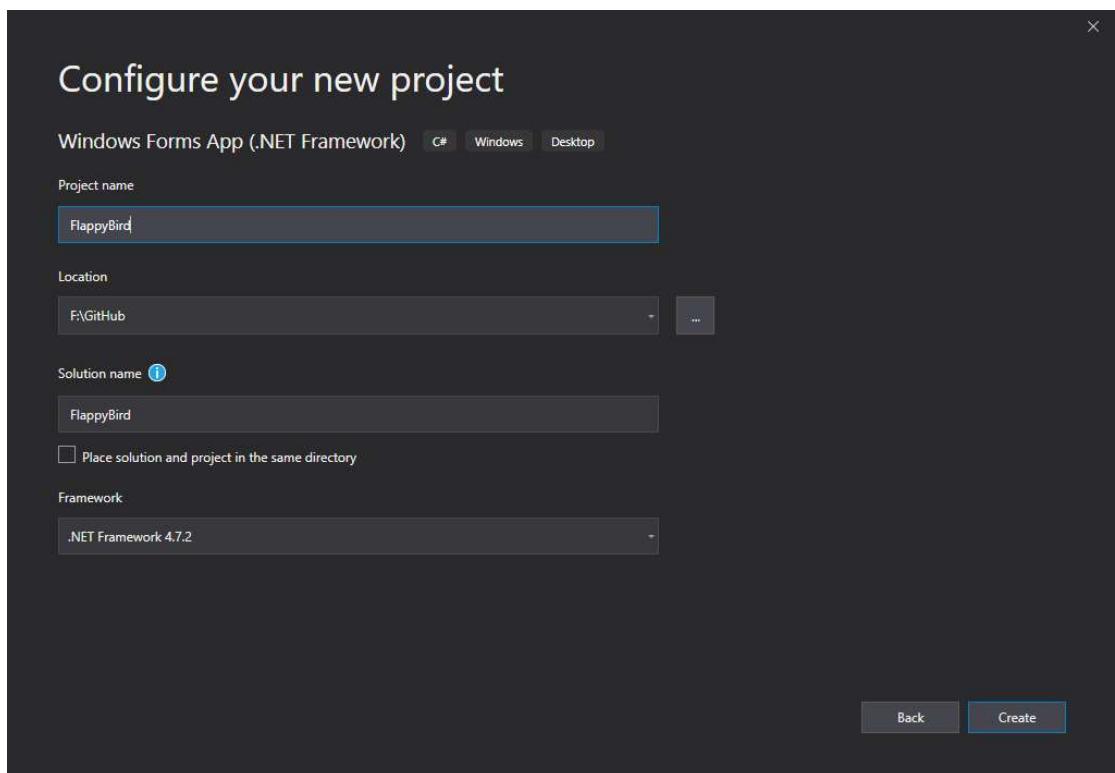
Oyunun içerisinde bir kuş ve engeller bulunur. Amaç; kuş yere düşmeden ve objelere çarpmadan düzgün bir şekilde en yüksek skoru yapmak.

2. Proje Oluşturma

- Visual Studio da "Windows Forms App" projesi seçilir.



Proje adı olarak "FlappyBird" yazılarak proje oluşturulur.

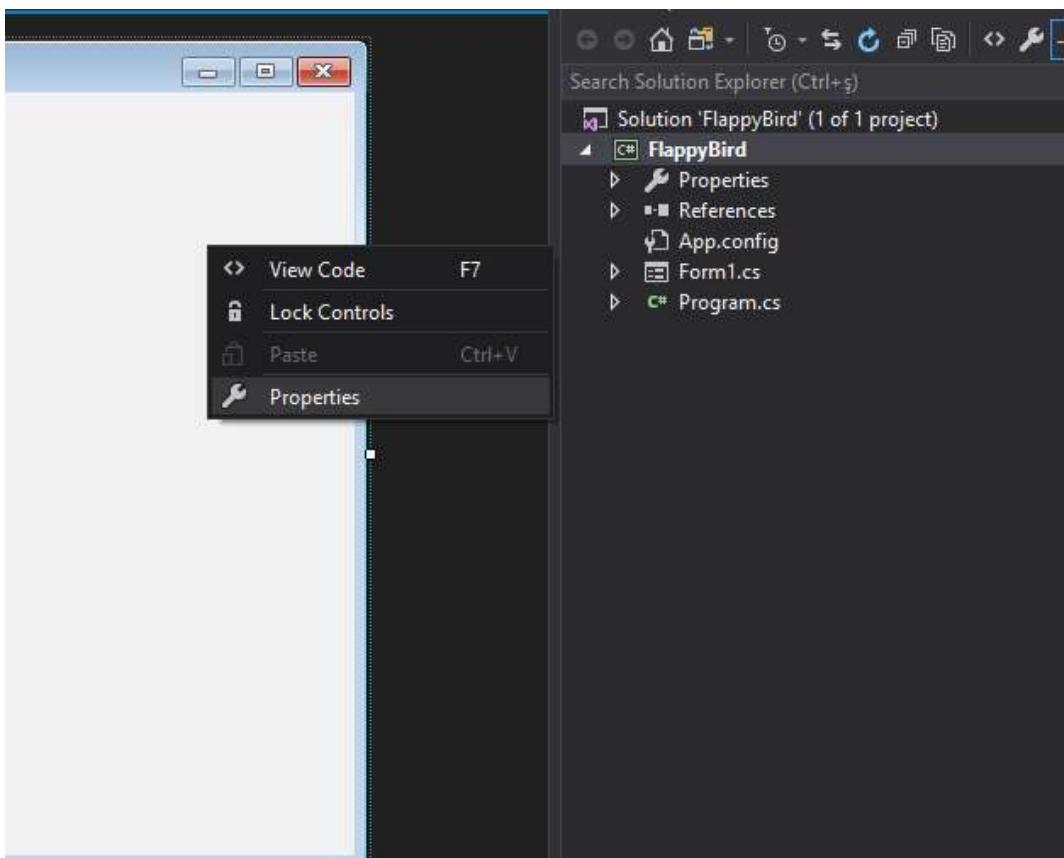


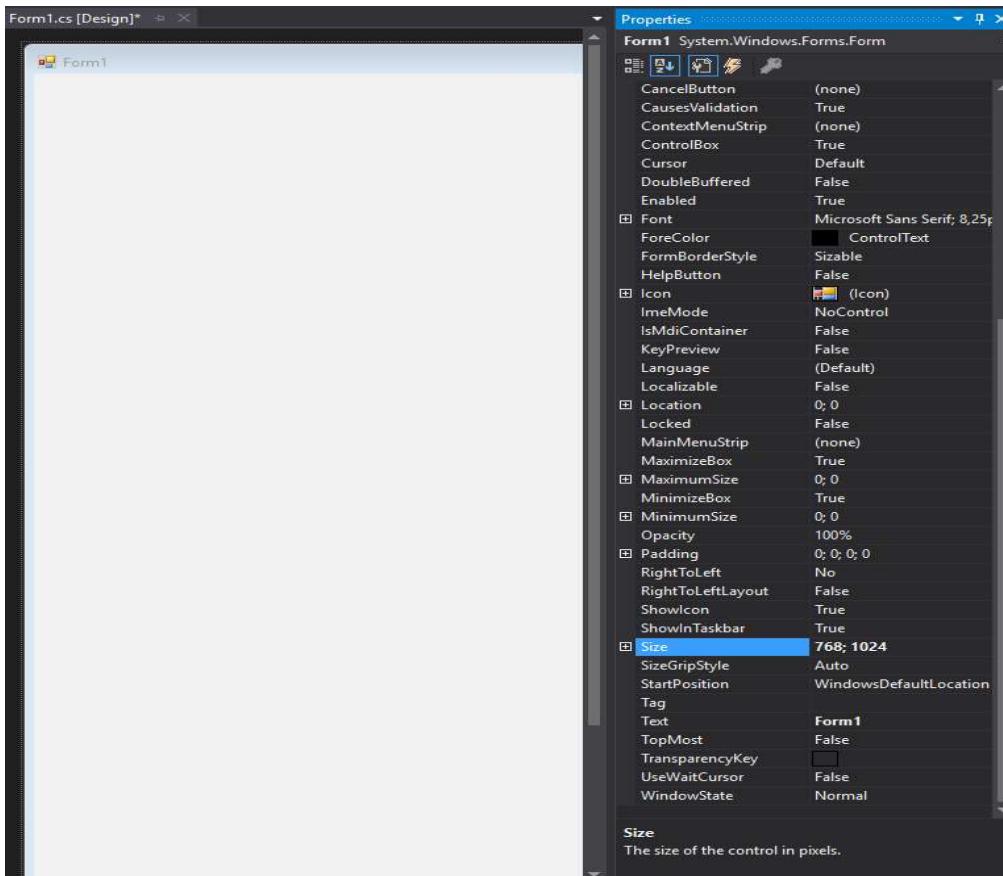
3. Elementler

- 1 Adet Form
- 6 Adet PictureBox
- 1 Adet Timer
- 1 Adet Panel
- 6 Adet Label
- 2 Adet Button

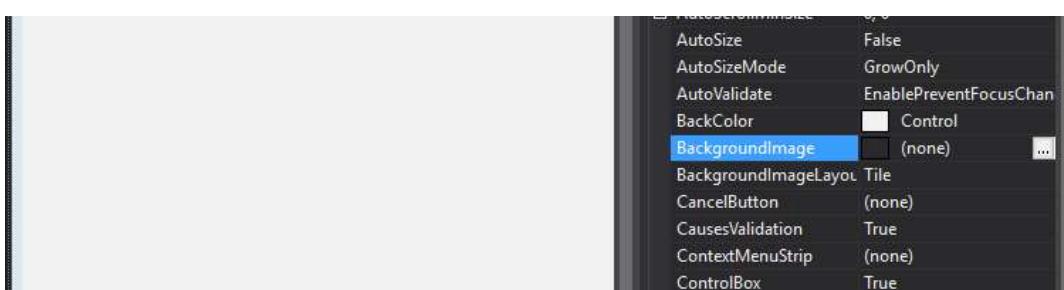
3.1 Form

- Form üzerinde sağ tıklayarak "Properties" aracı açılarak "Size" değeri **768; 1024** olarak ayarlanır.

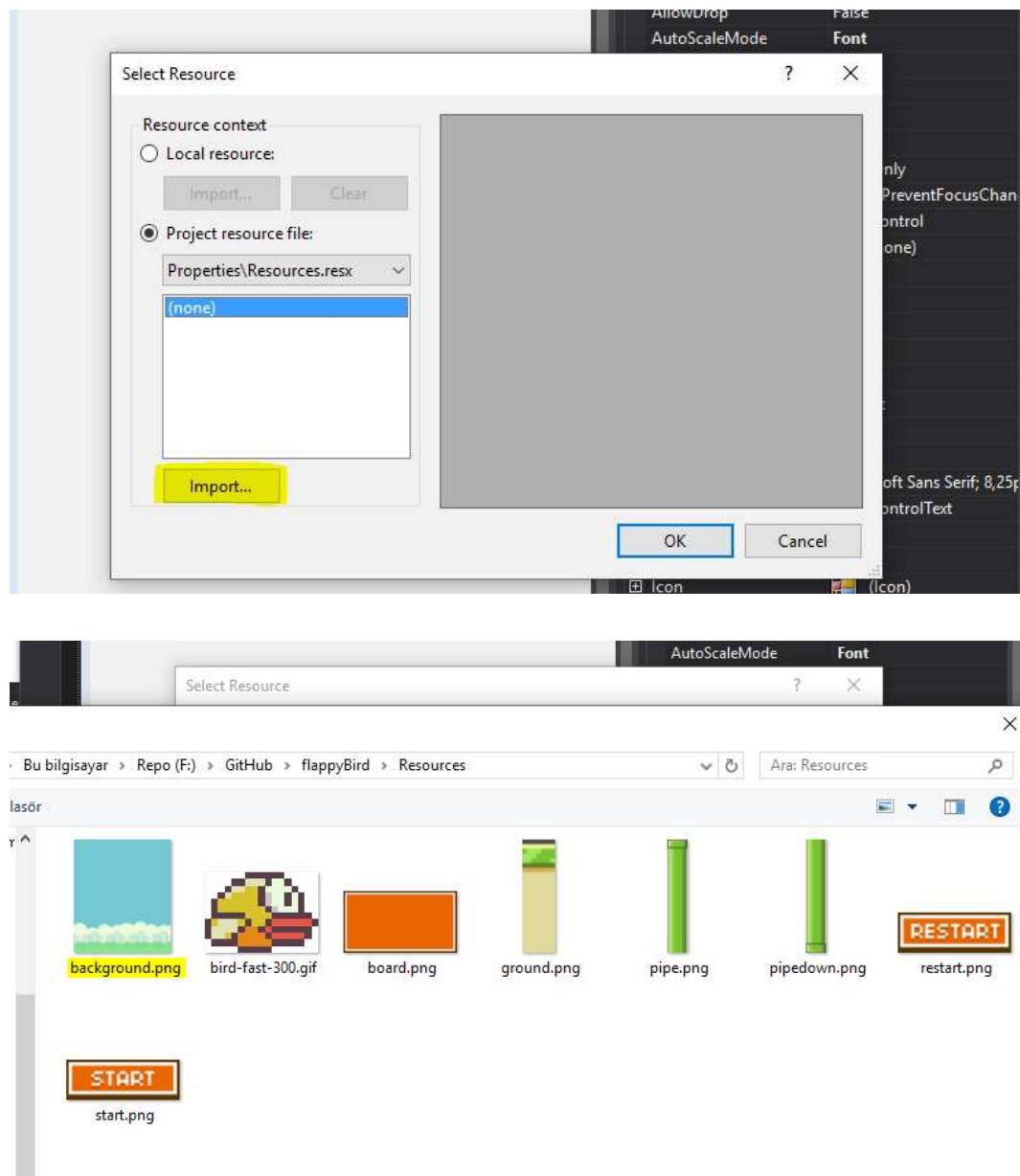




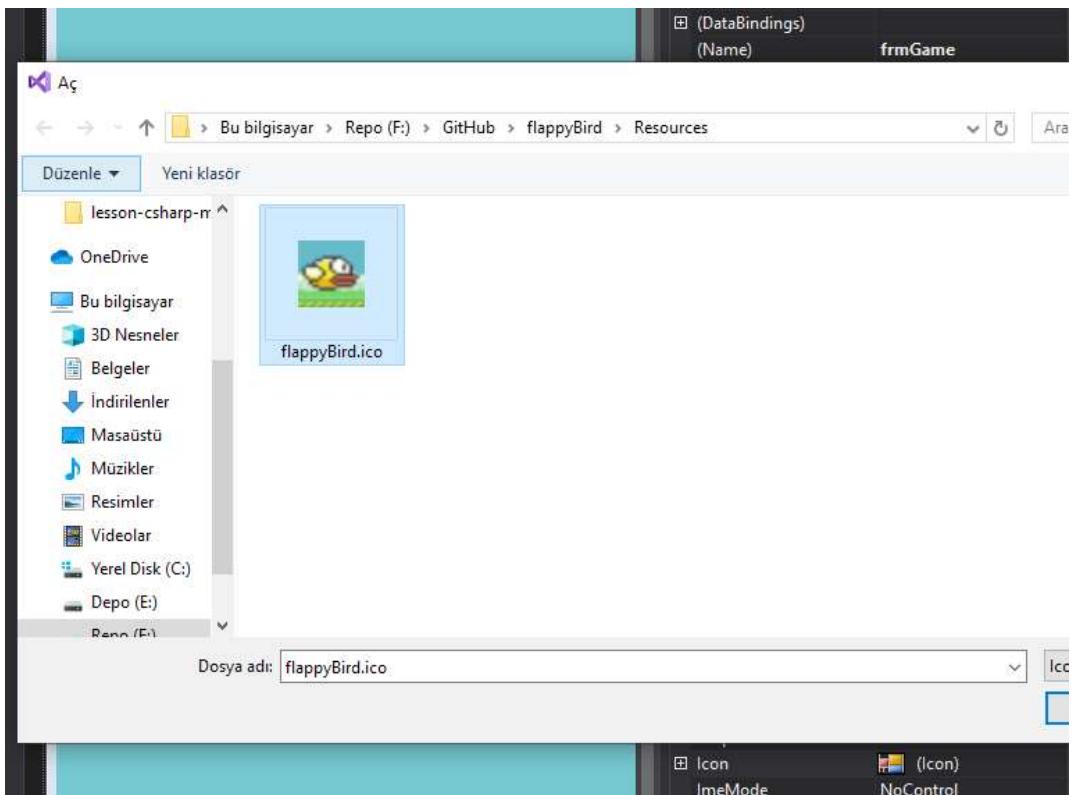
- "Properties" listesinde sırayla aşağıdaki değerler girilir.
 - Name:** frmGame
 - Text :** FlappyBird
 - MaximizeBox :** False *Oyun tam ekran yapılmaması için
 - KeyPreview:** True
- Form' un arkaplan resmi ayarlanır



Açılan ekranda arkaplan resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.

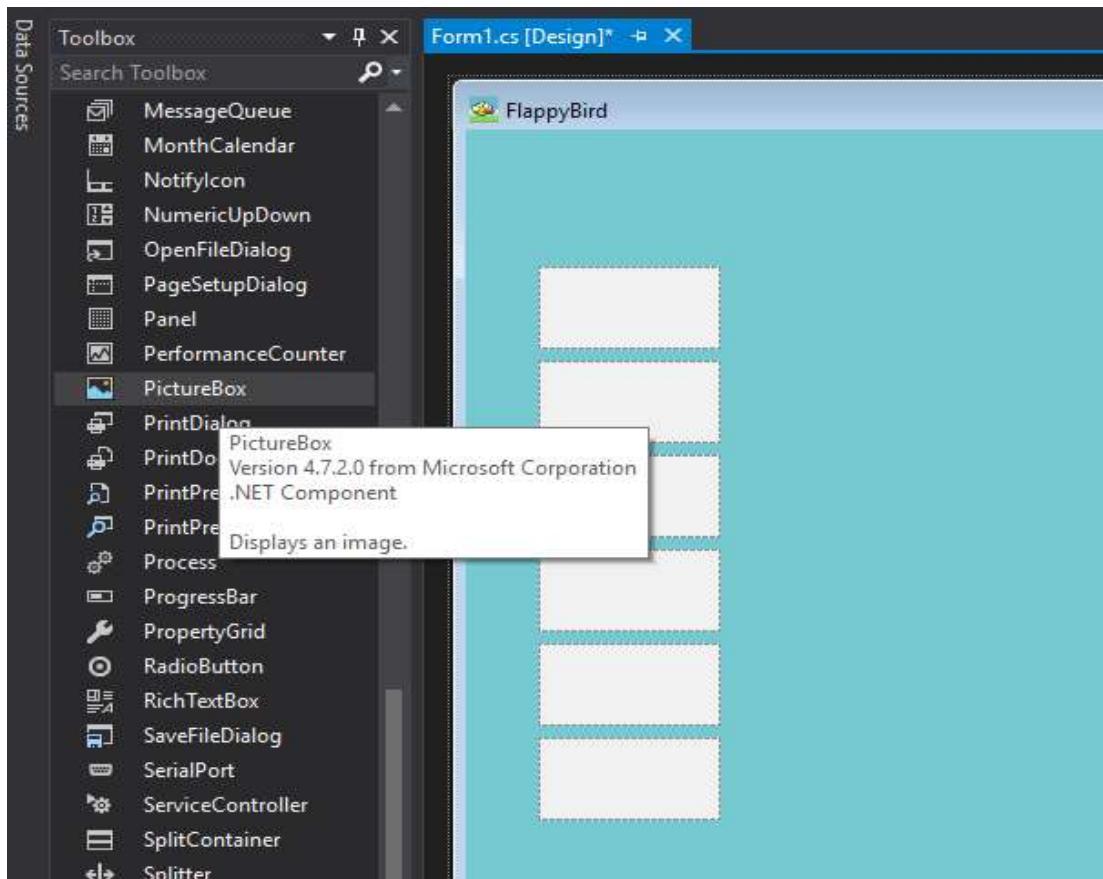


- Aynı şekilde "Icon" resmi ayarlanır.

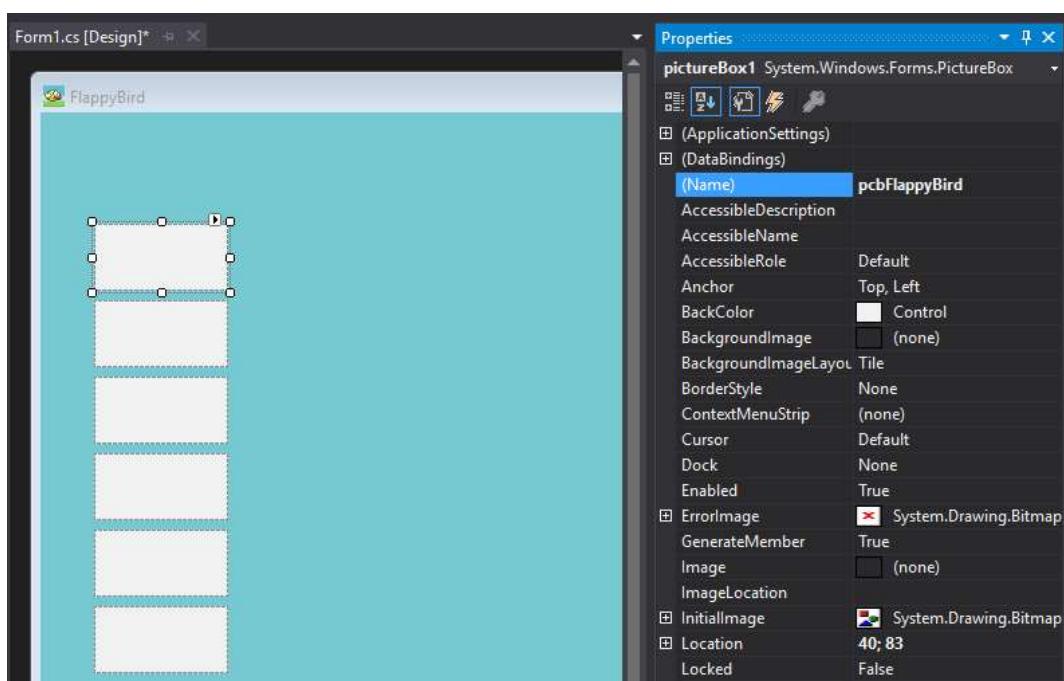


3.2 PictureBox

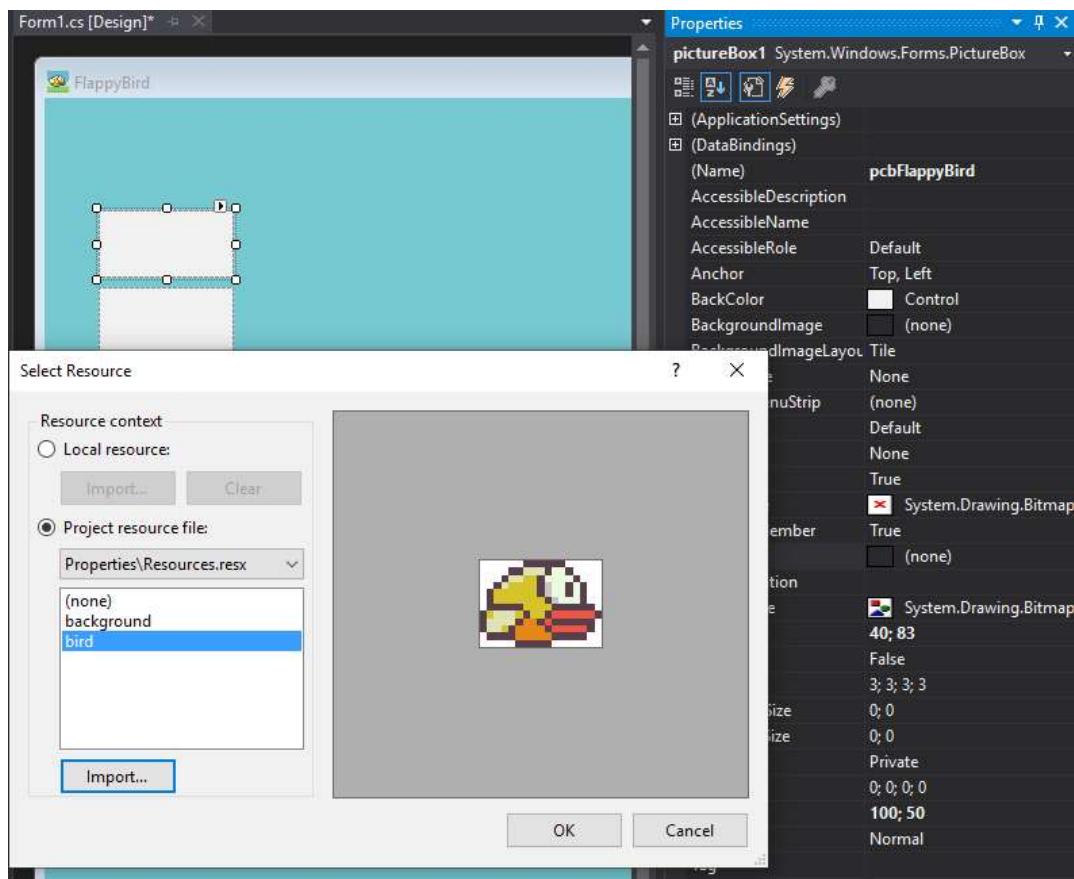
- "frmGame" üzerine 6 adet "PictureBox" Eklenir



- "PictureBox1" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbFlappyBird" olarak değiştirilir.

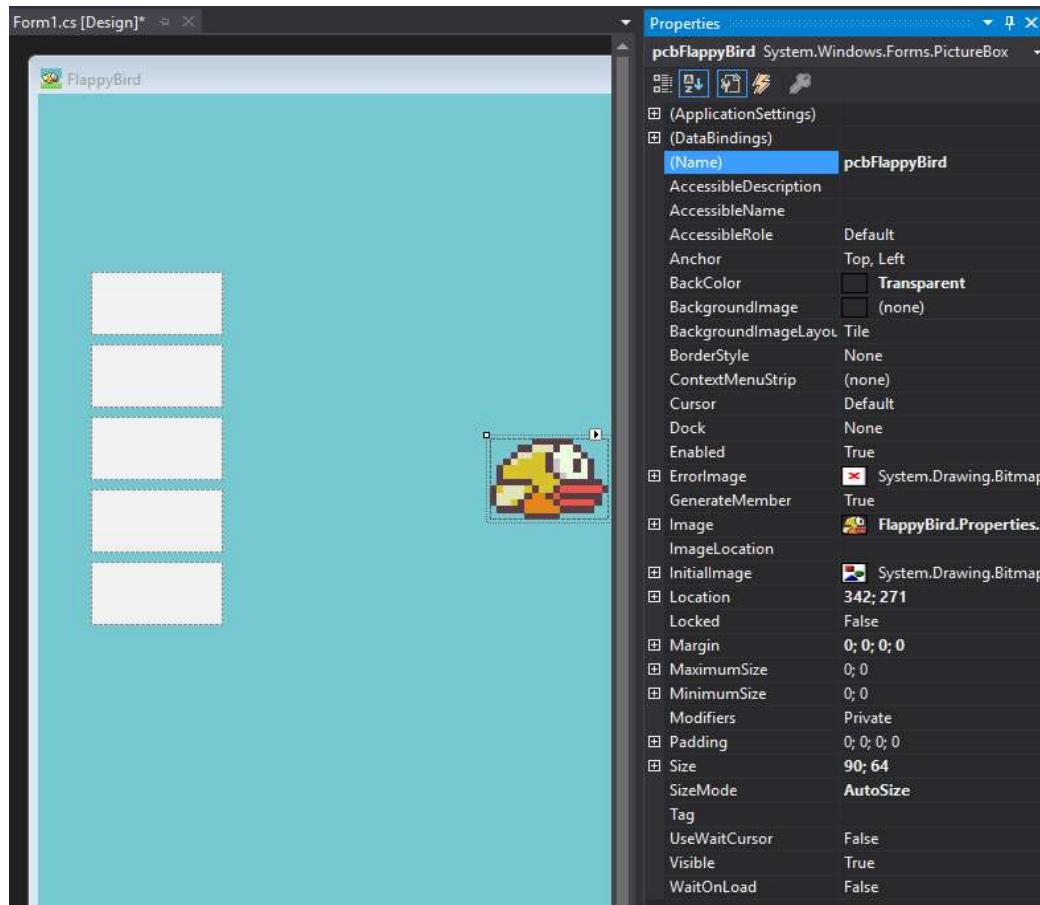


- "pcbFlappyBird" elementi için resim eklenir. "Properties" listesinden "Image" eklenir. Açılan ekranda "bird.gif" resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.

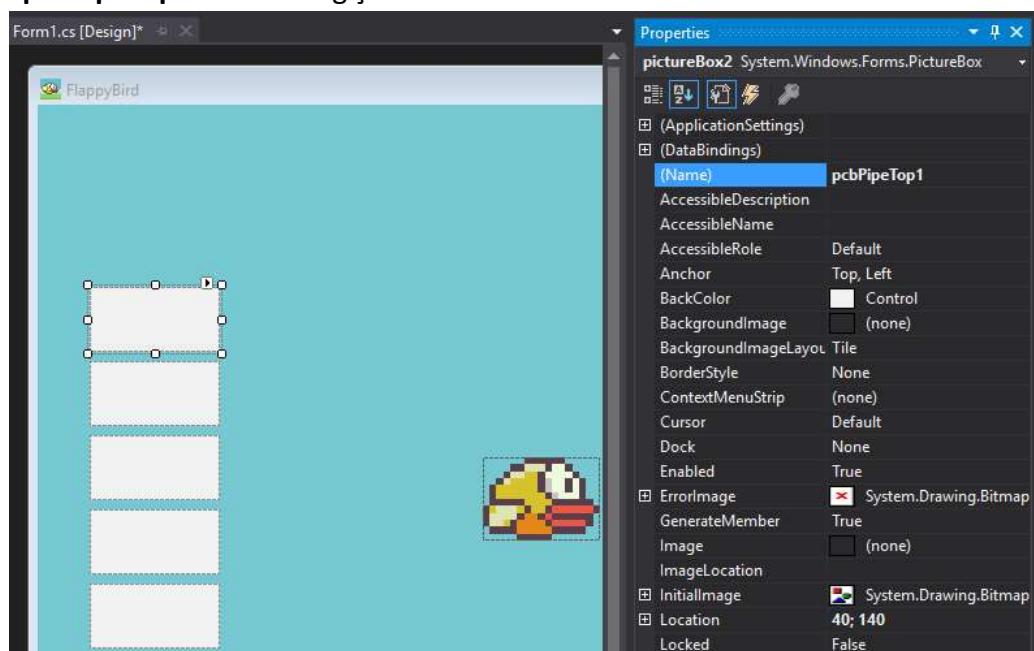


- "pcbFlappyBird" elementi için aşağıdaki properties değerleri verilir.

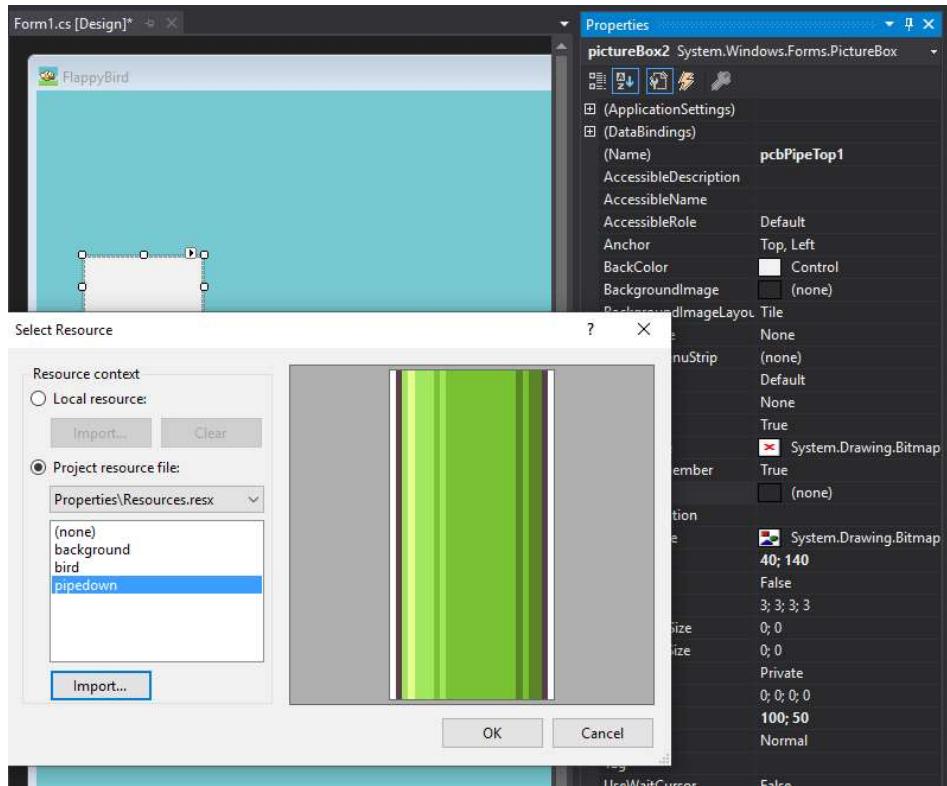
- BackColor:** Transparent
- Location:** 342; 271
- Margin:** 0;0;0;0
- SizeMode:** AutoSize



- "PictureBox2" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbPipeTop1" olarak değiştirilir.

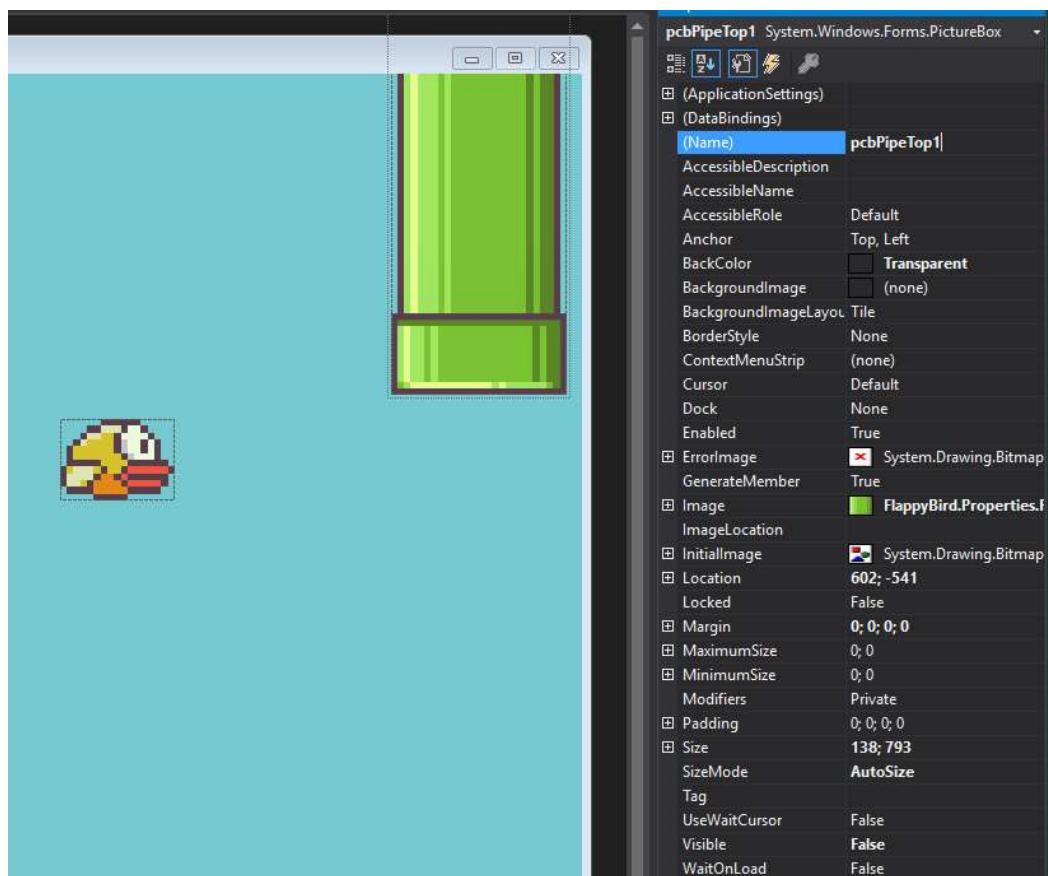


- "pcbPipeTop1" elementi için resim eklenir. "Properties" listesinden "Image" eklenir. Açılan ekranda "pipedown.png" resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.

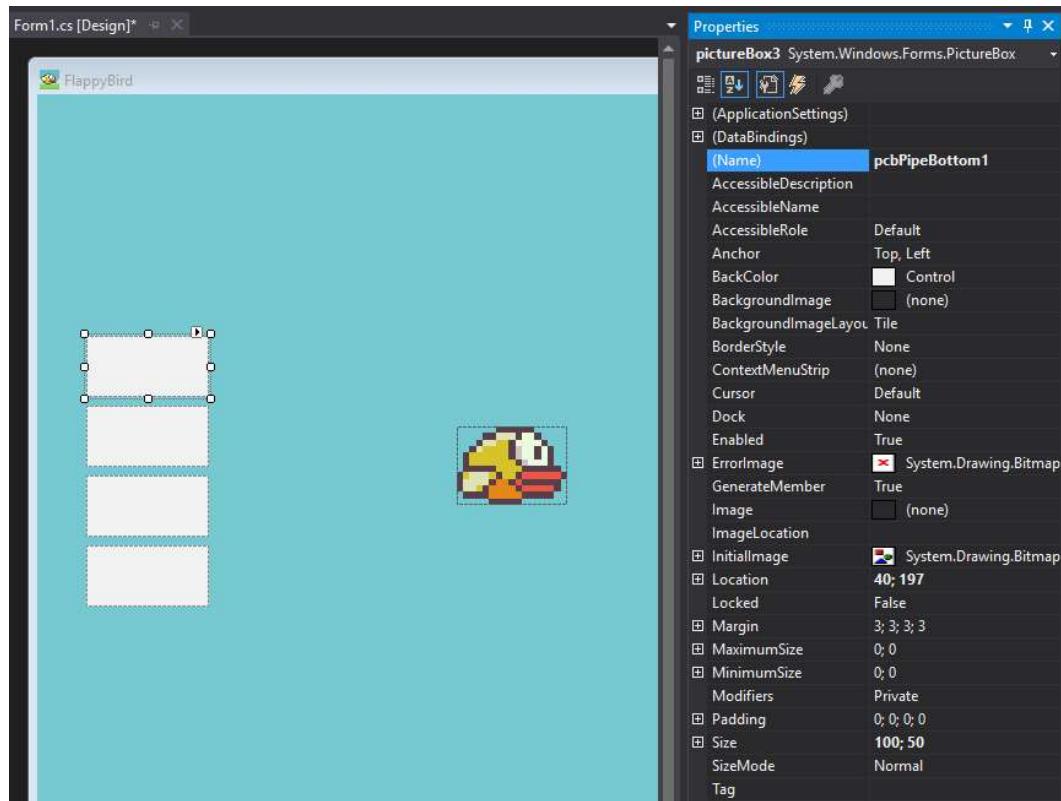


- "pcbPipeTop1" elementi için aşağıdaki properties değerleri verilir.

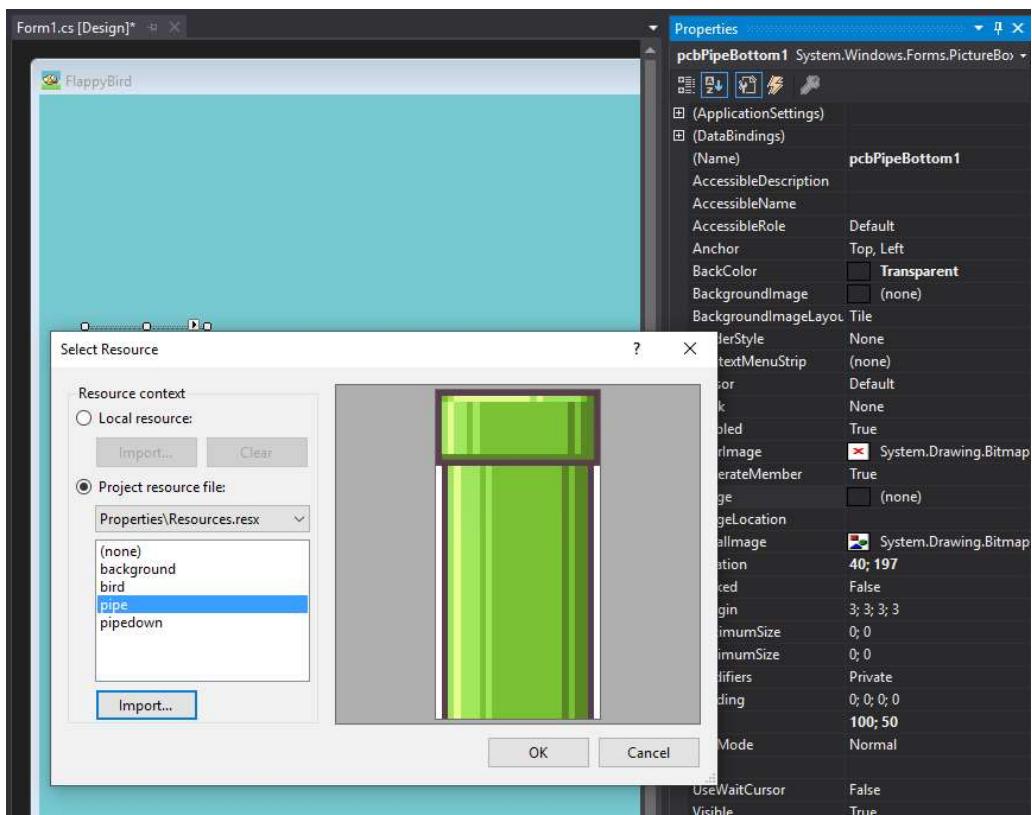
- BackColor:** Transparent
- Location:** 602; -541
- Margin:** 0;0;0;0
- SizeMode:** AutoSize
- Visible:** False



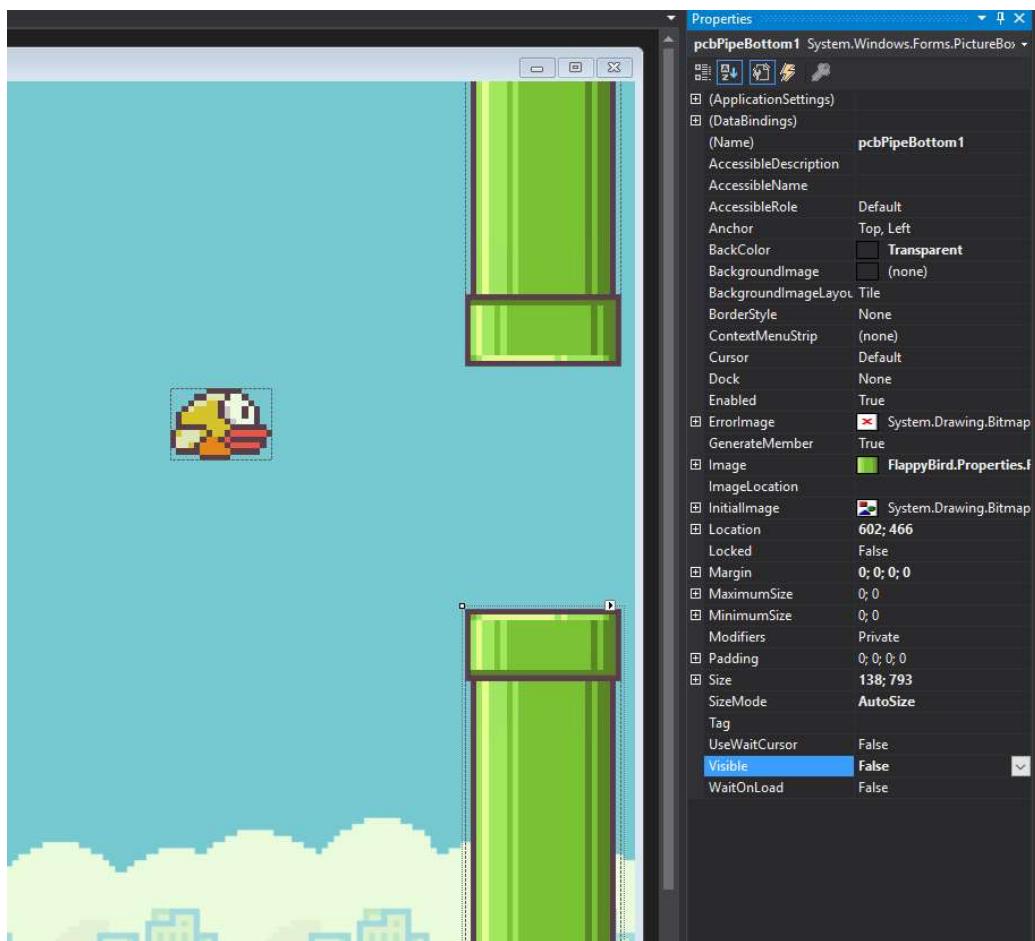
- "PictureBox3" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbPipeBottom1" olarak değiştirilir.



- "pcbPipeBottom1" elementi için resim eklenir. "Properties" listesinden "Image" eklenir. Açılan ekranda "pipe.png" resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.

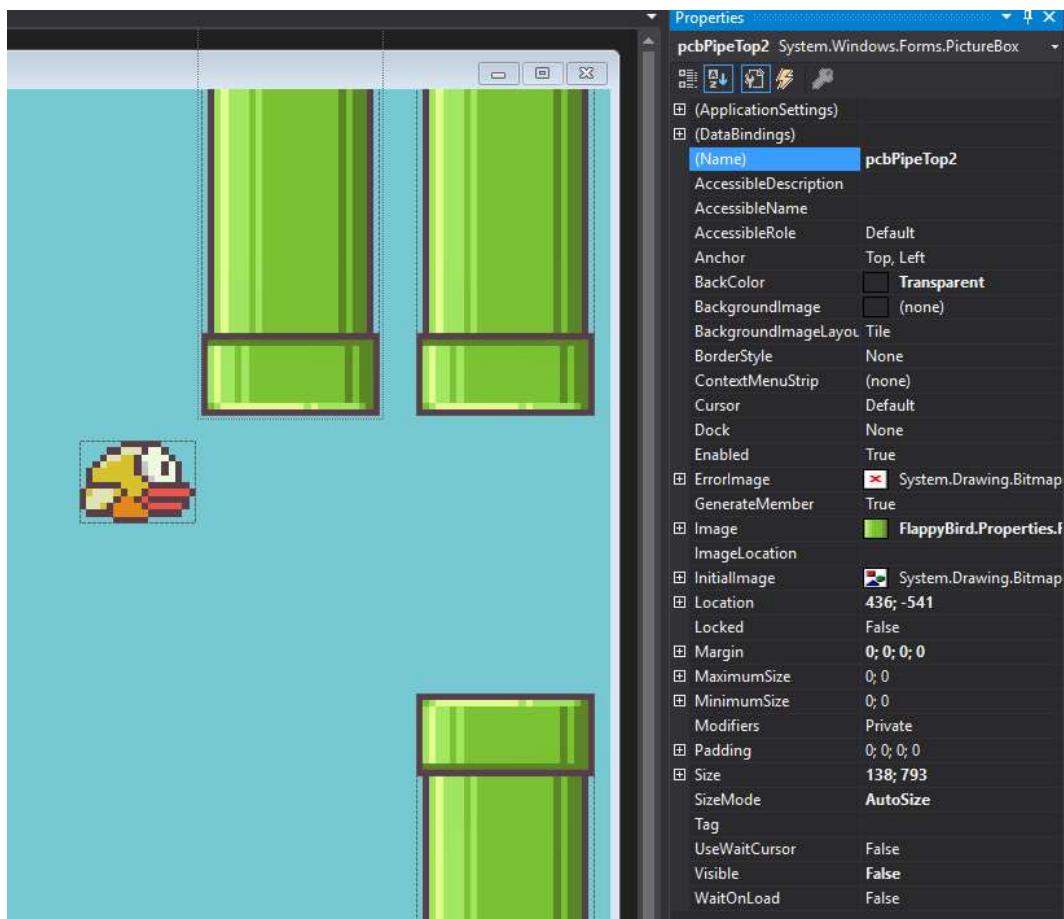


- "pcbPipeBottom1" elementi için aşağıdaki properties değerleri verilir.
 - BackColor:** Transparent
 - Location:** 602; 466
 - Margin:** 0;0;0;0
 - SizeMode:** AutoSize
 - Visible:** False



- "PictureBox4" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbPipeTop2" olarak değiştirilir.
- "pcbPipeTop2" elementi için daha önceden "Resources" listesine eklenen "pipedown.png" resmi seçilir.

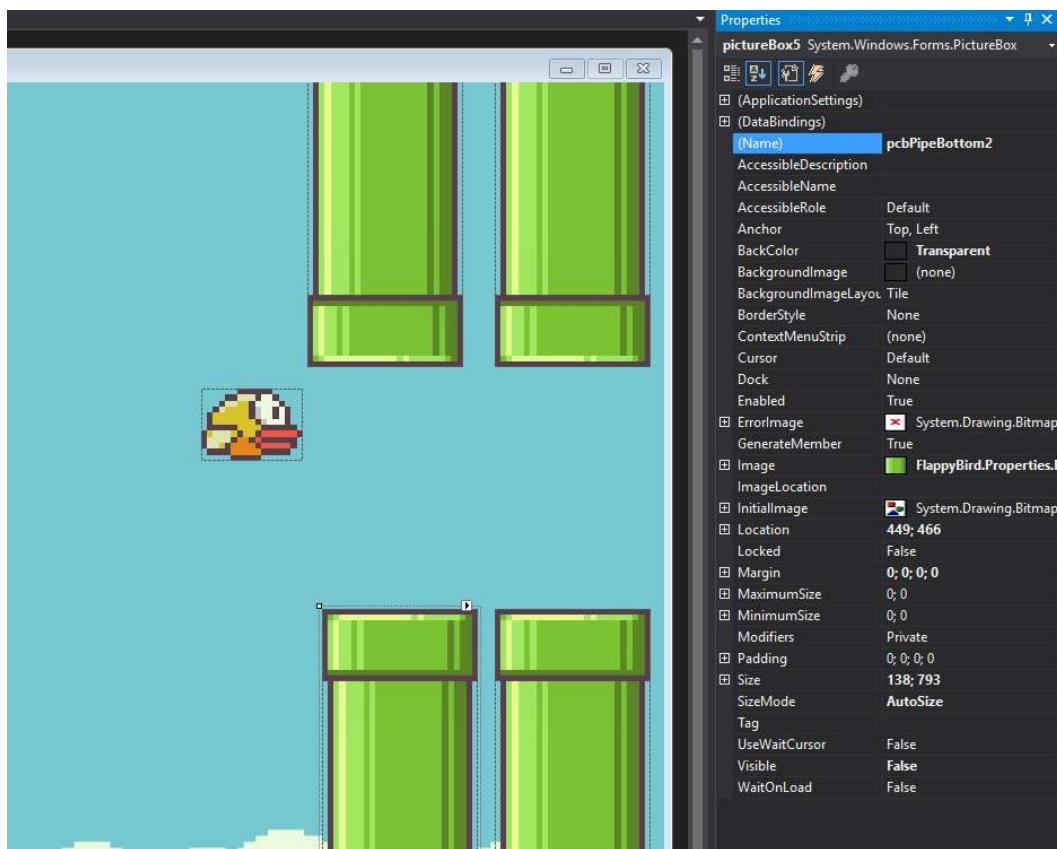
- "pcbPipeTop2" elementi için aşağıdaki properties değerleri verilir.
 - BackColor:** Transparent
 - Location:** 436; -541
 - Margin:** 0;0;0;0
 - SizeMode:** AutoSize
 - Visible:** False



- "PictureBox5" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbPipeBottom2" olarak değiştirilir.
- "pcbPipeBottom2" elementi için daha önceden "Resources" listesine eklenen "pipe.png" resmi seçilir.

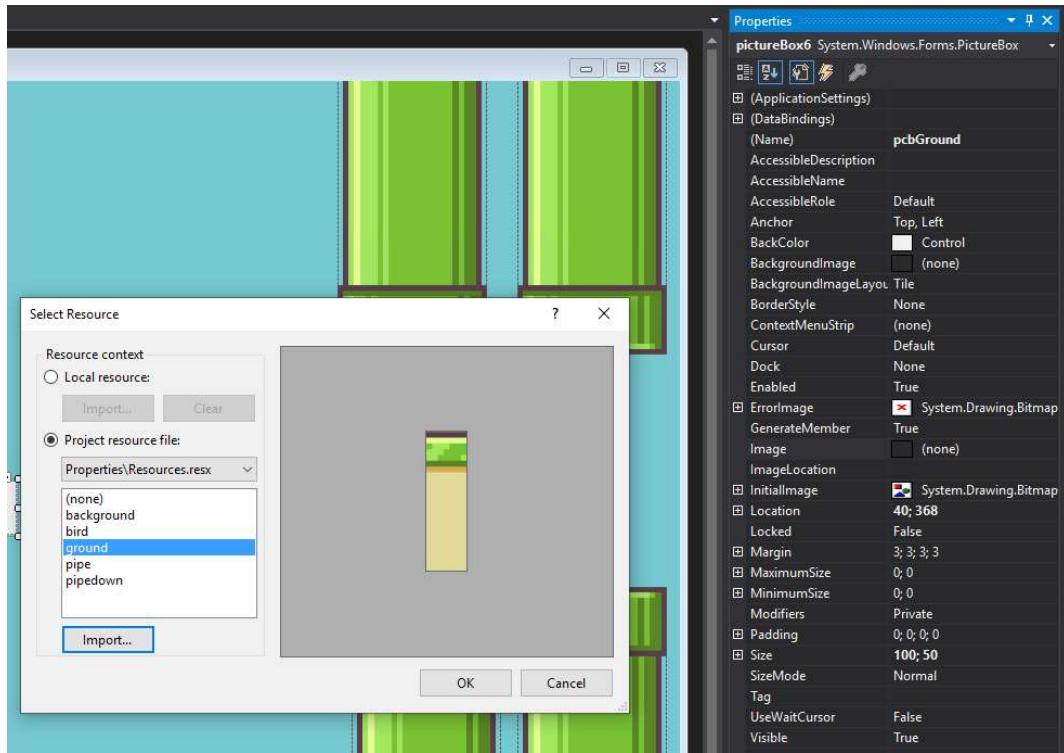
- "pcbPipeBottom2" elementi için aşağıdaki properties değerleri verilir.

- a. **BackColor**: Transparent
- b. **Location**: 449; 466
- c. **Margin**: 0;0;0;0
- d. **SizeMode**: AutoSize
- e. **Visible**: False

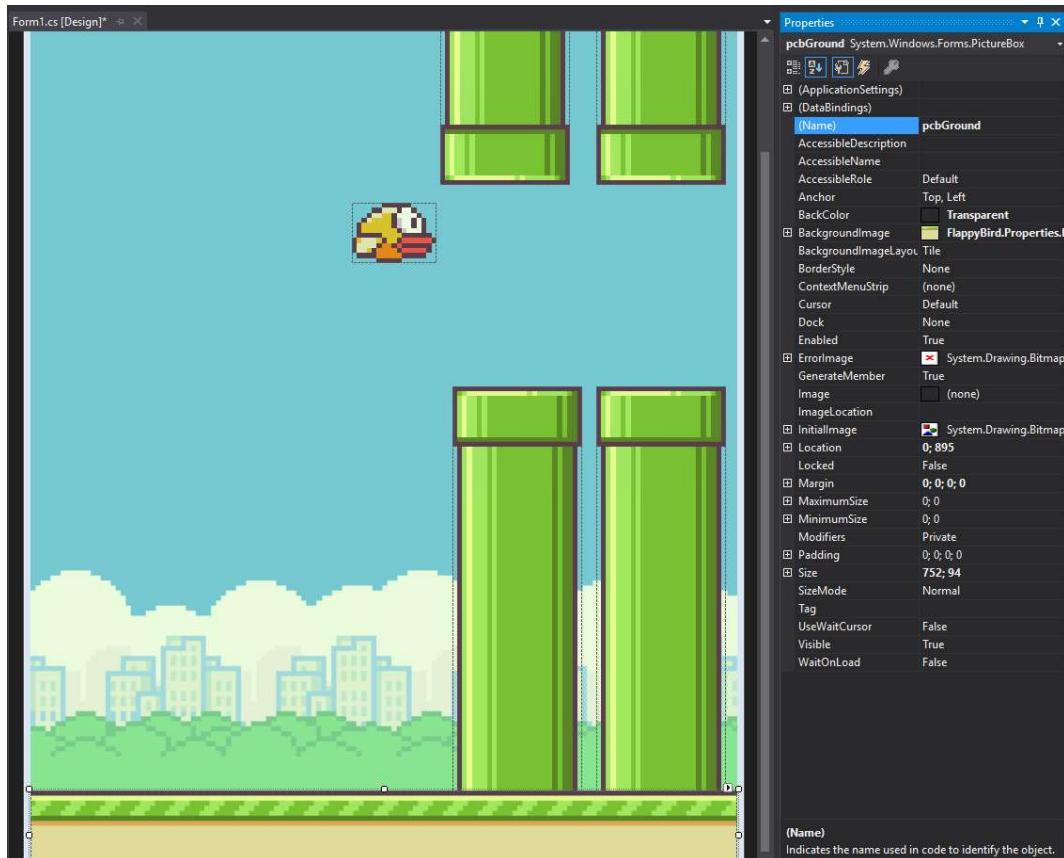


- "PictureBox6" elementi üzerinde sağ tıklayarak "Properties" listesinden "Name" değerini "pcbGround" olarak değiştirilir.

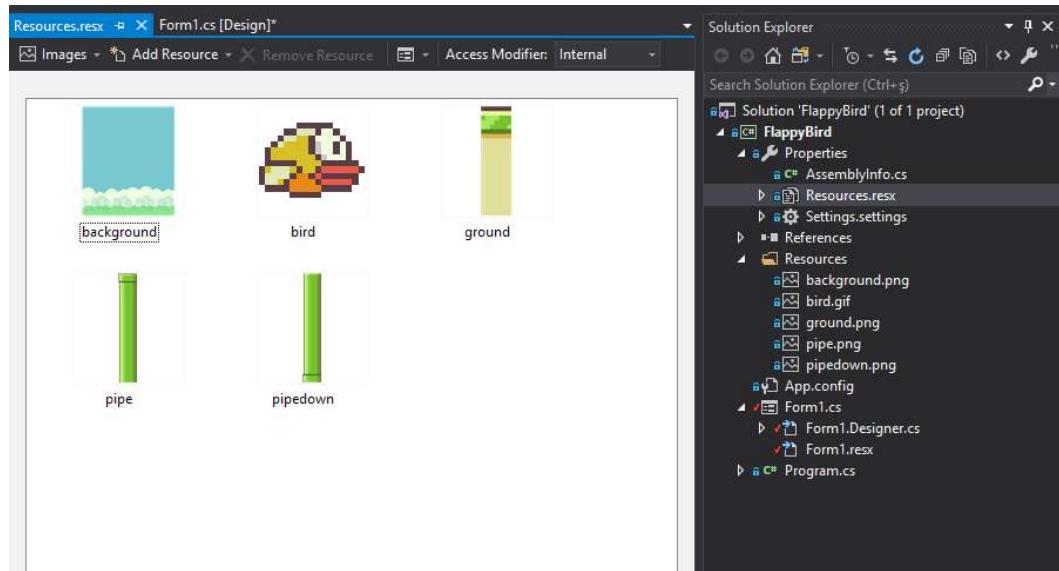
- "pcbGround" elementi için resim eklenir. "Properties" listesinden "BackgroundImage" eklenir. Açılan ekranda "ground.png" resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.



- "pcbGround" elementi için aşağıdaki properties değerleri verilir.
 - BackColor:** Transparent
 - Location:** 0; 895
 - Margin:** 0;0;0;0
 - Size:** 752; 94

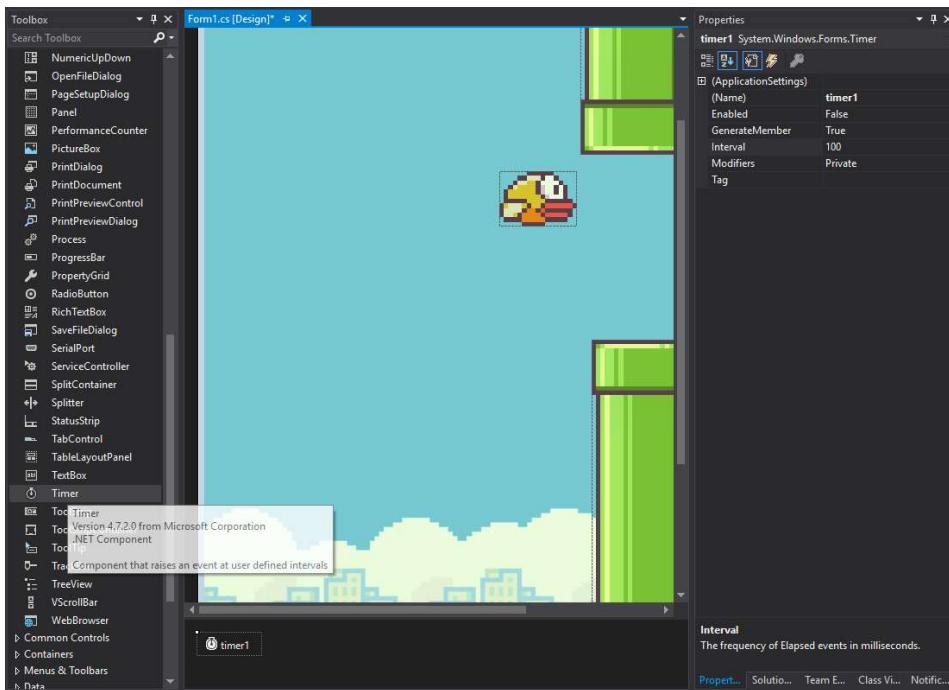


- PictureBox elementleri için eklenen resimler "Solution Explorer" listesinde "Resources" klasörü içerisinde yer alır. Resources içerisinde eklenmiş resimleri güncellemek ya da yeni resim eklemek için ayrıca "Properties" içerisindeki Resources.resx dosyasına çift tıklayarak açılan ekran kullanılabilir.

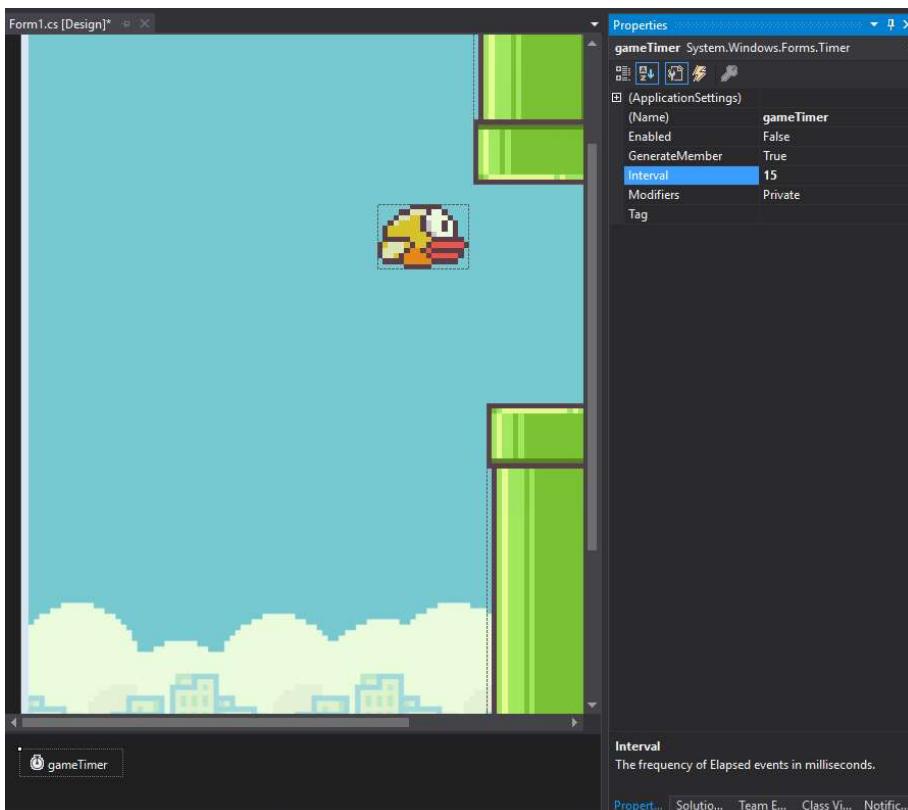


3.3 Timer

- "frmGame" üzerine 1 adet "Timer" Eklenir

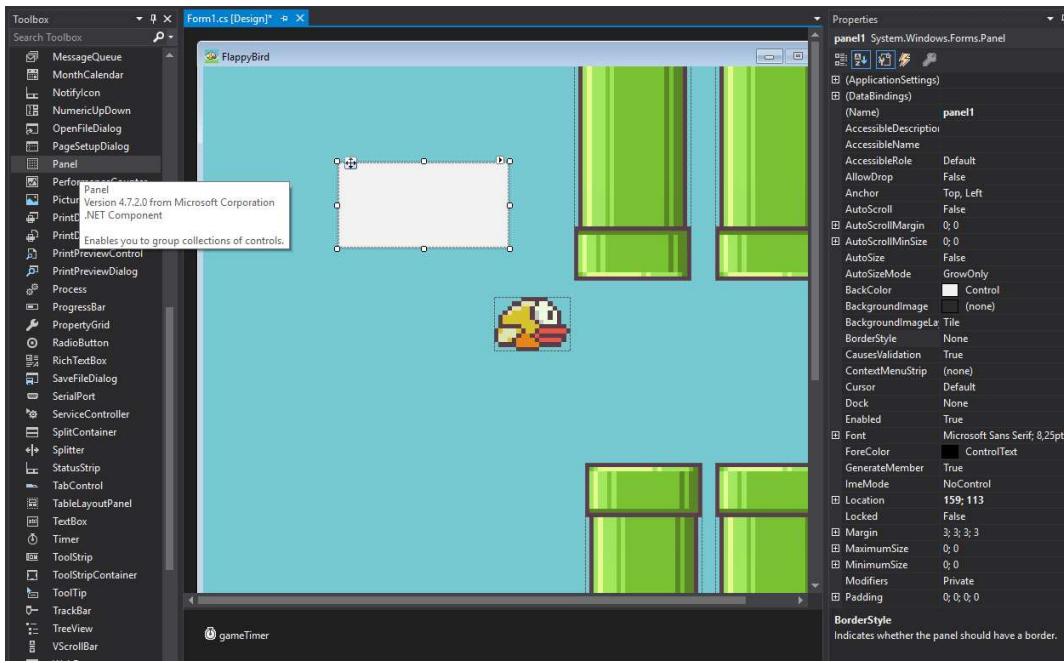


- "timer1" elementi için aşağıdaki properties değerleri verilir.
 - Name:** gameTimer
 - Enabled:** True
 - Interval:** 15



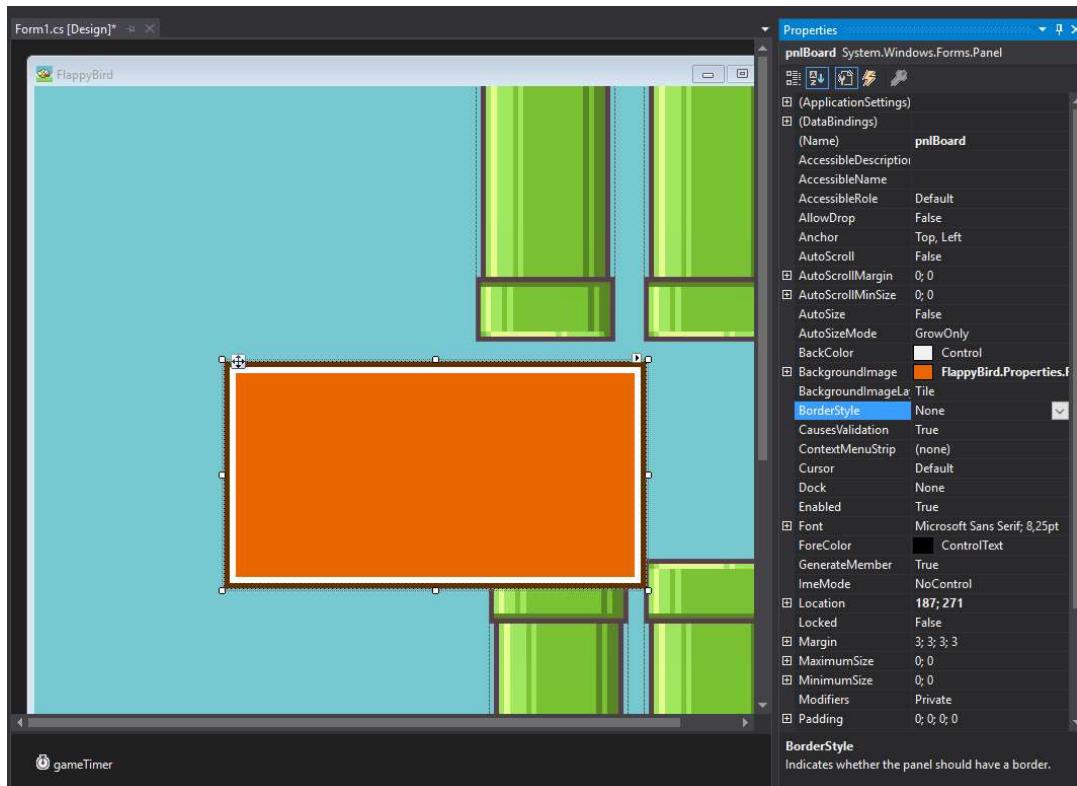
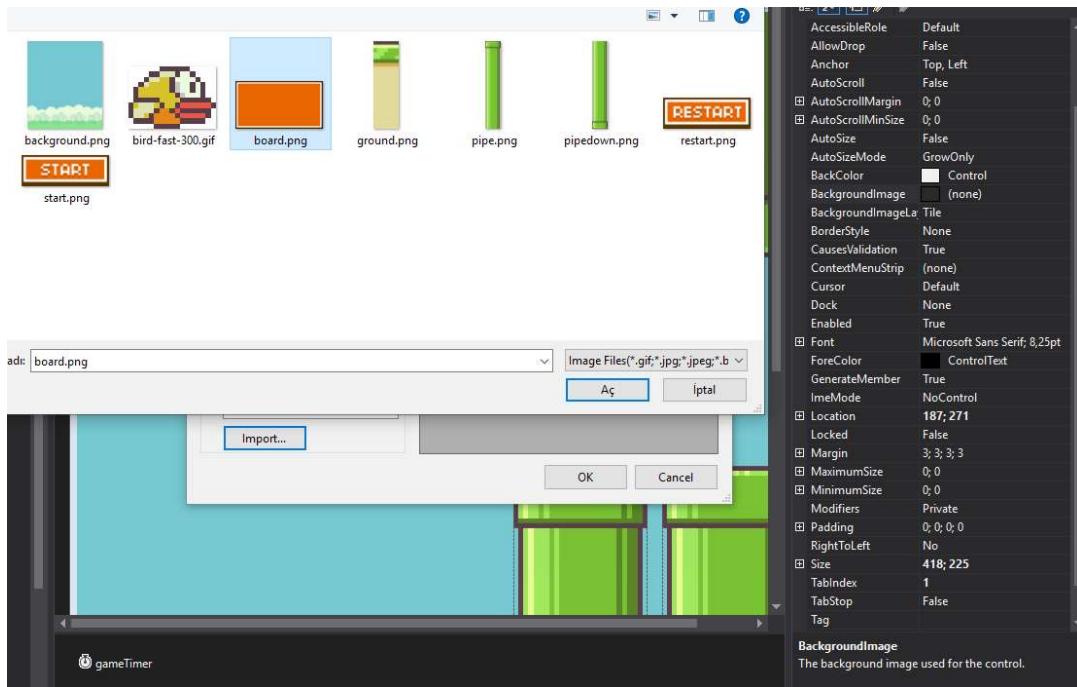
3.4 Panel

- "frmGame" üzerine 1 adet "Panel" Eklendir



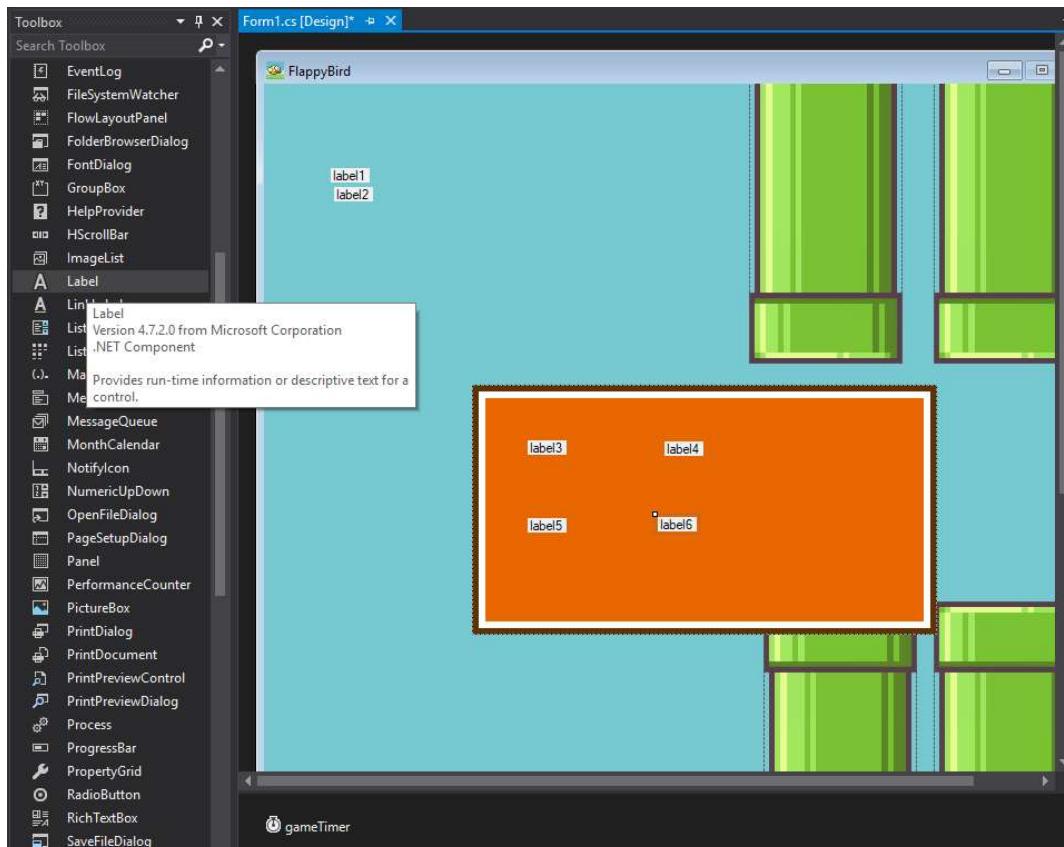
- "panel1" elementi için aşağıdaki properties değerleri verilir.
 - a. **Name:** pnlBoard
 - b. **Location:** 187; 271
 - c. **Size:** 418; 225
 - d. **Visible:** False

- "**pnlBoard**" elementi için resim eklenir. "Properties" listesinden "**BackgroundImage**" eklenir. Açılan ekranda "**board.png**" resmi "**Import**" butonu ile seçilerek "**Resources**" bölümüne eklenir.

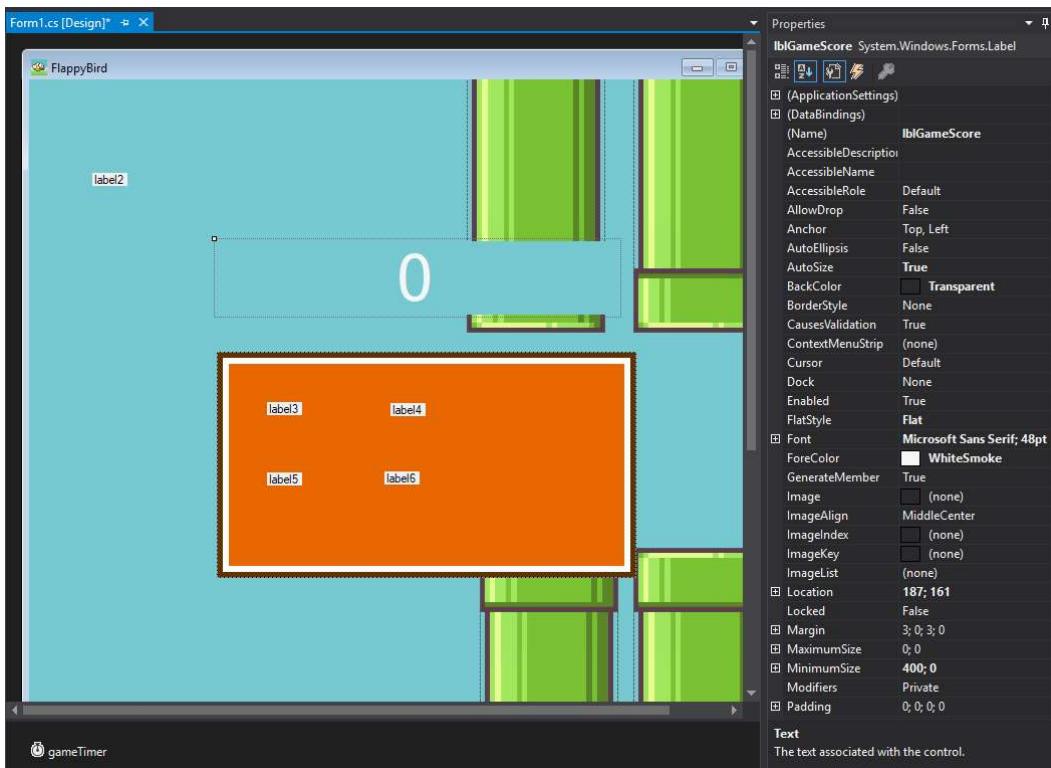


3.5 Label

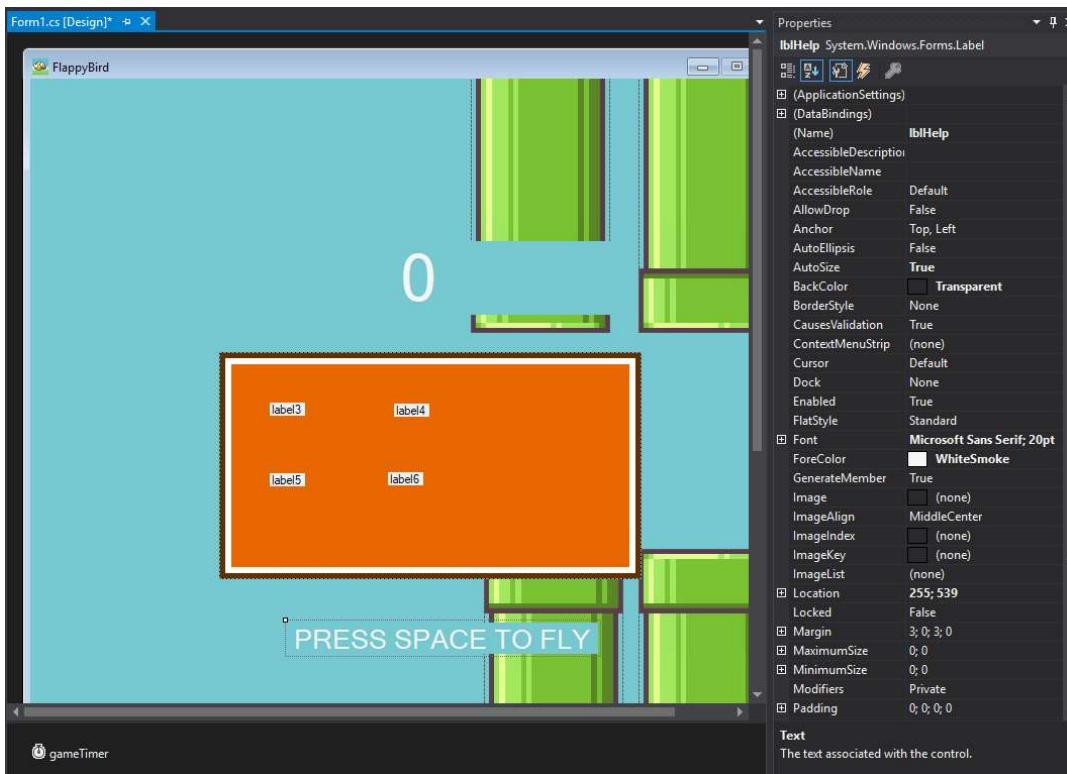
- "frmGame" üzerine 2 adet label ve "pnlBoard" üzerine de 4 adet label elementi eklenir



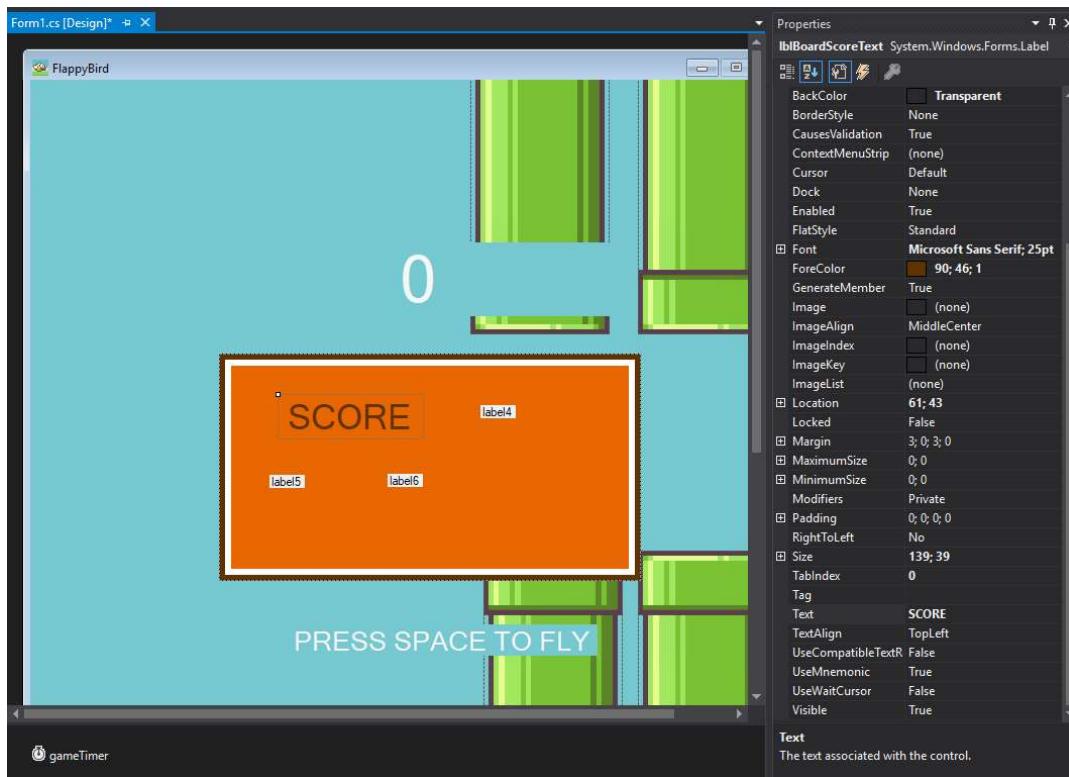
- "Label1" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblGameScore
 - AutoSize:** True
 - BackColor:** Web > Transparent
 - FlatStyle:** Flat
 - Font > Size :** 48
 - ForeColor:** Web > WhiteSmoke
 - Location:** 187;161
 - MinimumSize:** 400;0
 - Size:** 400;73
 - Text :** 0
 - TextAlign:** MiddleCenter



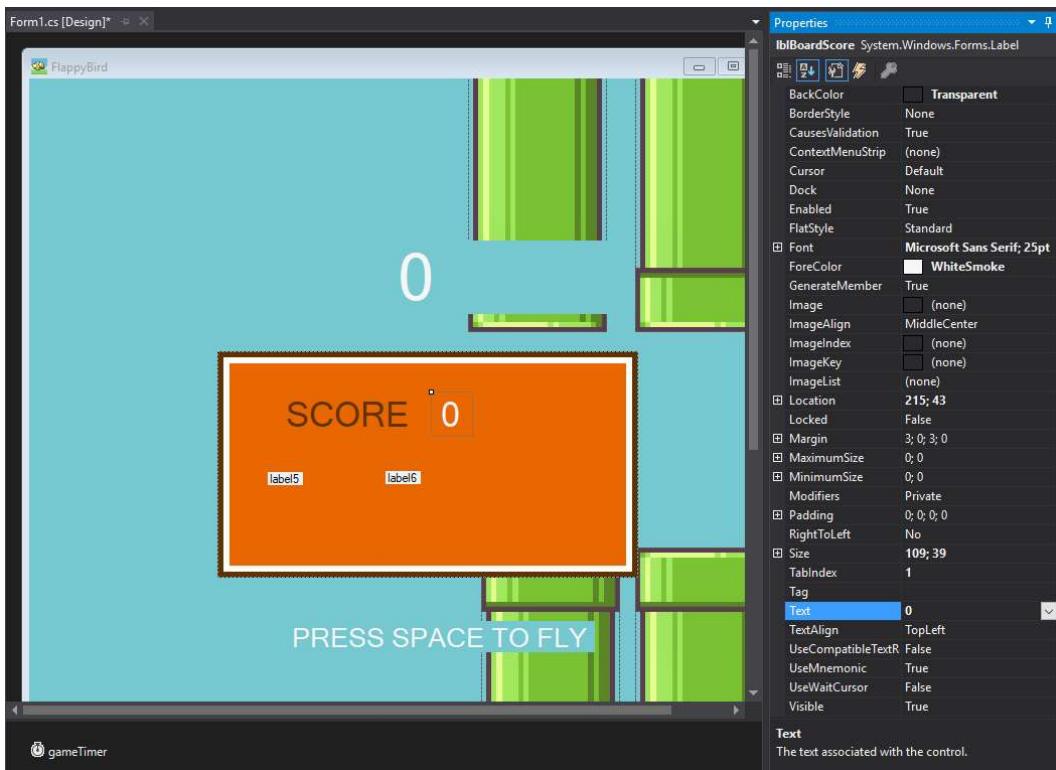
- "Label2" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblHelp
 - BackColor:** Web > Transparent
 - Font > Size :** 20
 - ForeColor:** Web > WhiteSmoke
 - Location:** 255;539
 - Text :** PRESS SPACE TO FLY



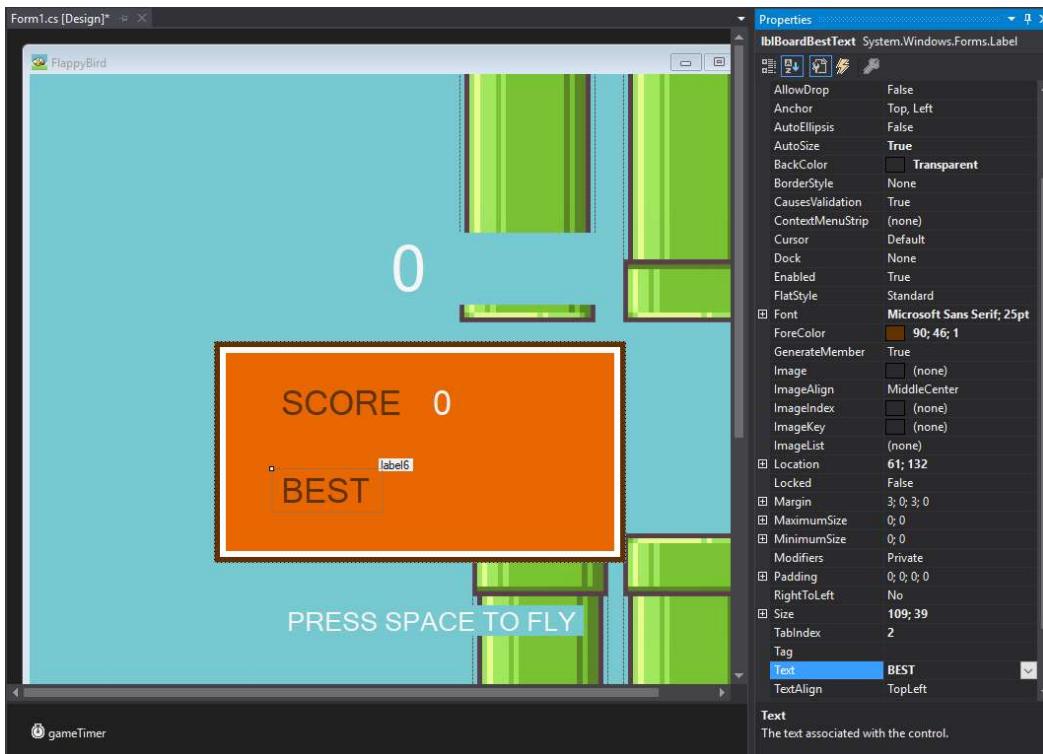
- "Label3" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblBoardScoreText
 - BackColor:** Web > Transparent
 - Font > Size :** 25
 - ForeColor:** 90; 46; 1
 - Location:** 61; 43
 - Text :** SCORE



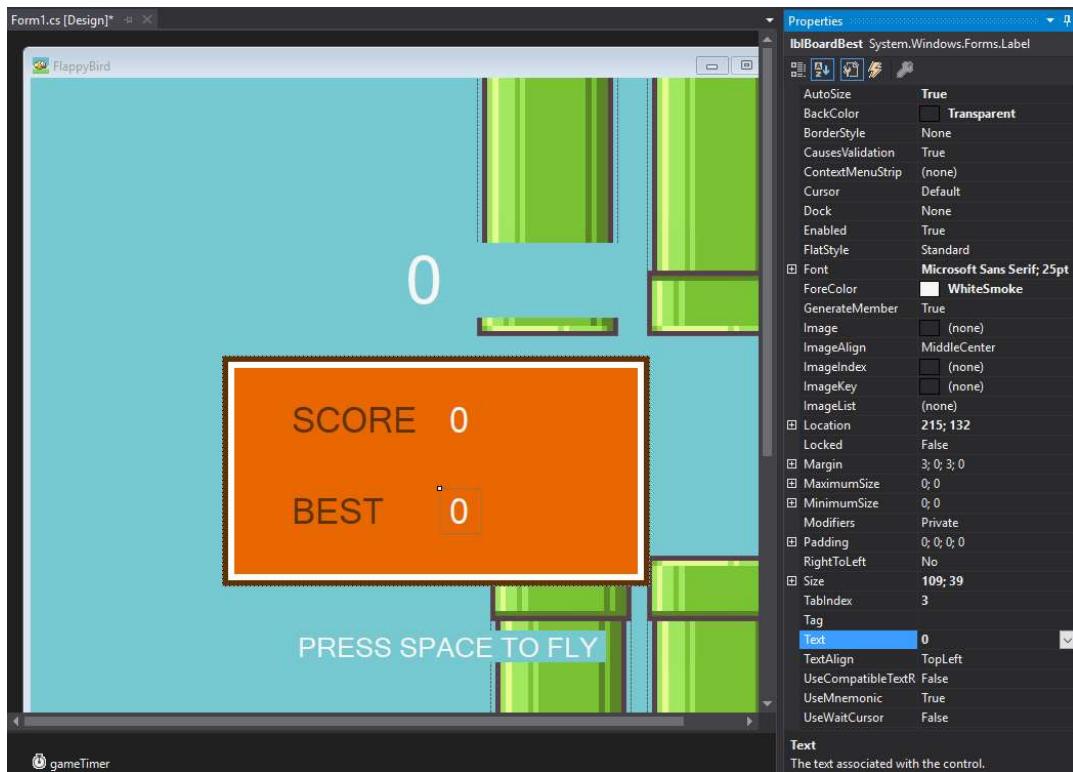
- "Label4" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblBoardScore
 - BackColor:** Web > Transparent
 - Font > Size :** 25
 - ForeColor:** Web > WhiteSmoke
 - Location:** 215; 43
 - Text :** 0



- "Label5" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblBoardBestText
 - BackColor:** Web > Transparent
 - Font > Size :** 25
 - ForeColor:** 90; 46; 1
 - Location:** 61; 132
 - Text :** BEST

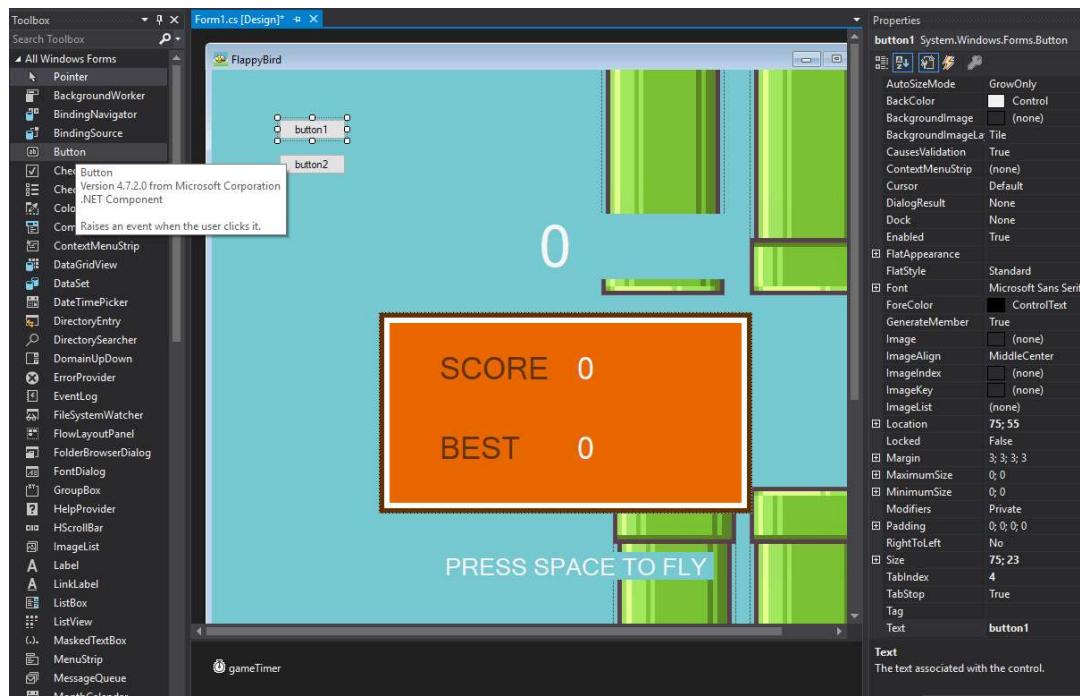


- "Label6" elementine sağ tıklayarak sırayla aşağıdaki properties değerleri verilir.
 - Name:** lblBoardBest
 - BackColor:** Web > Transparent
 - Font > Size :** 25
 - ForeColor:** Web > WhiteSmoke
 - Location:** 173; 132
 - Text :** 0



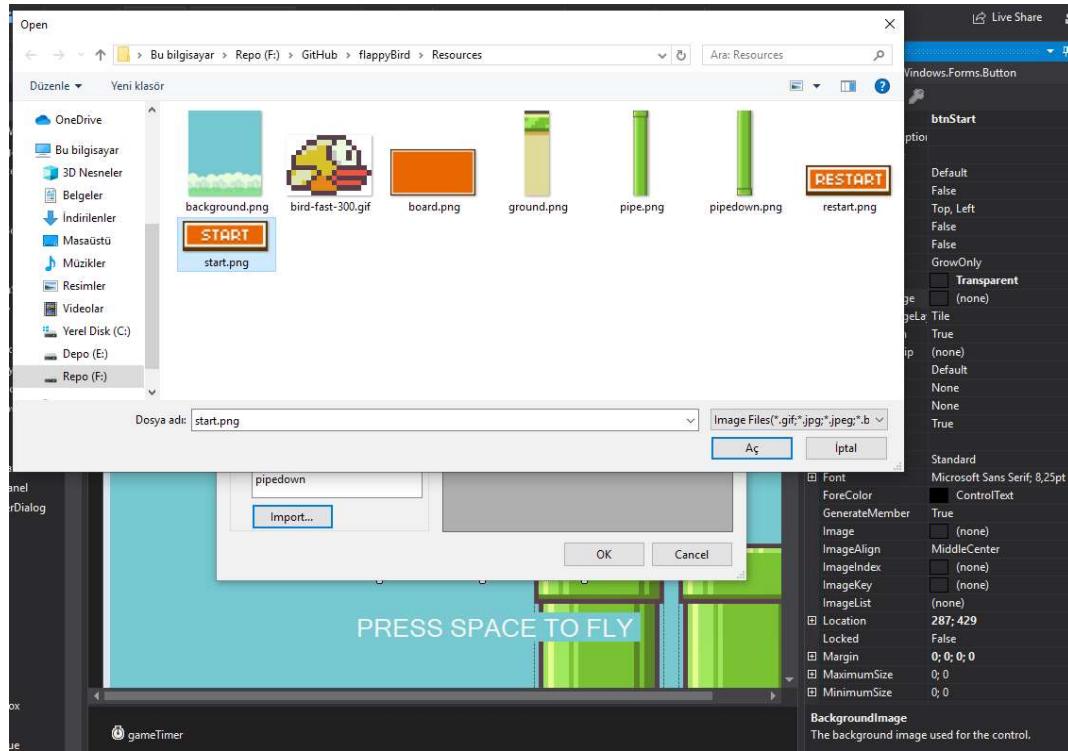
3.6 Button

- "frmGame" üzerine 2 adet button elementi eklenir



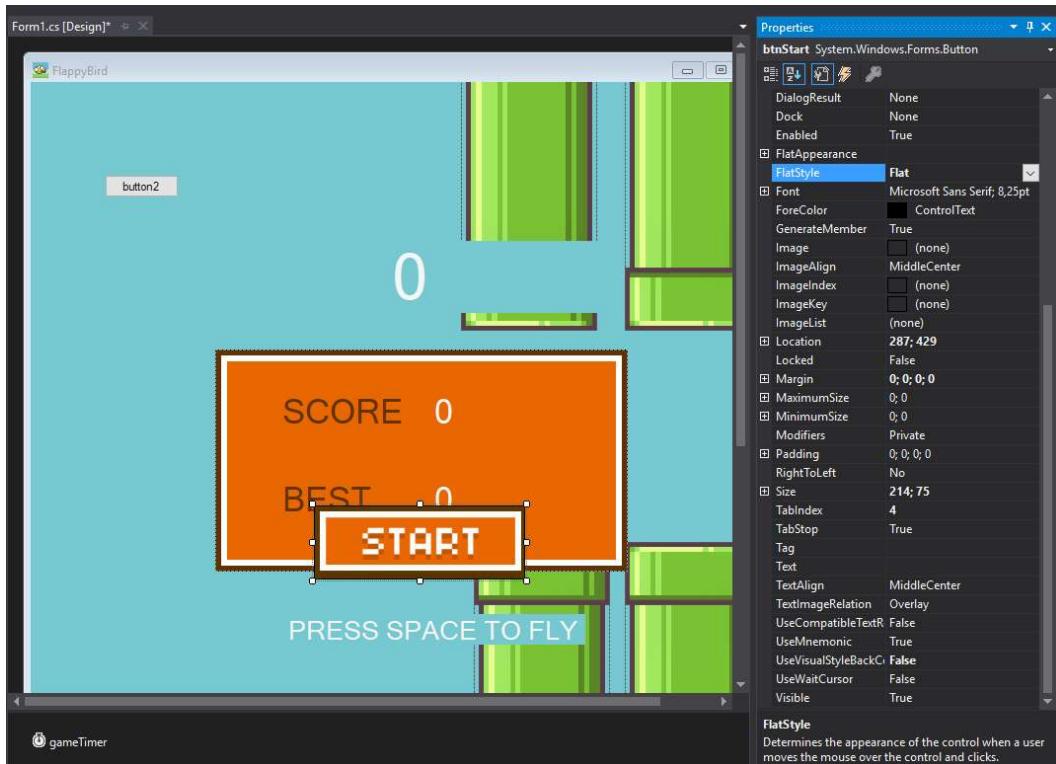
- "button1" elementi için aşağıdaki properties değerleri verilir.
 - Name:** btnStart
 - FlatStyle:** Flat
 - Location:** 287; 429
 - Margin:** 0;0;0;0
 - Size:** 214; 75
 - Text:** (Boş)

- "btnStart" elementi için resim eklenir. "Properties" listesinden "BackgroundImage" eklenir. Açılan ekranda "start.png" resmi "Import" butonu ile seçilerek "Resources" bölümüne eklenir.

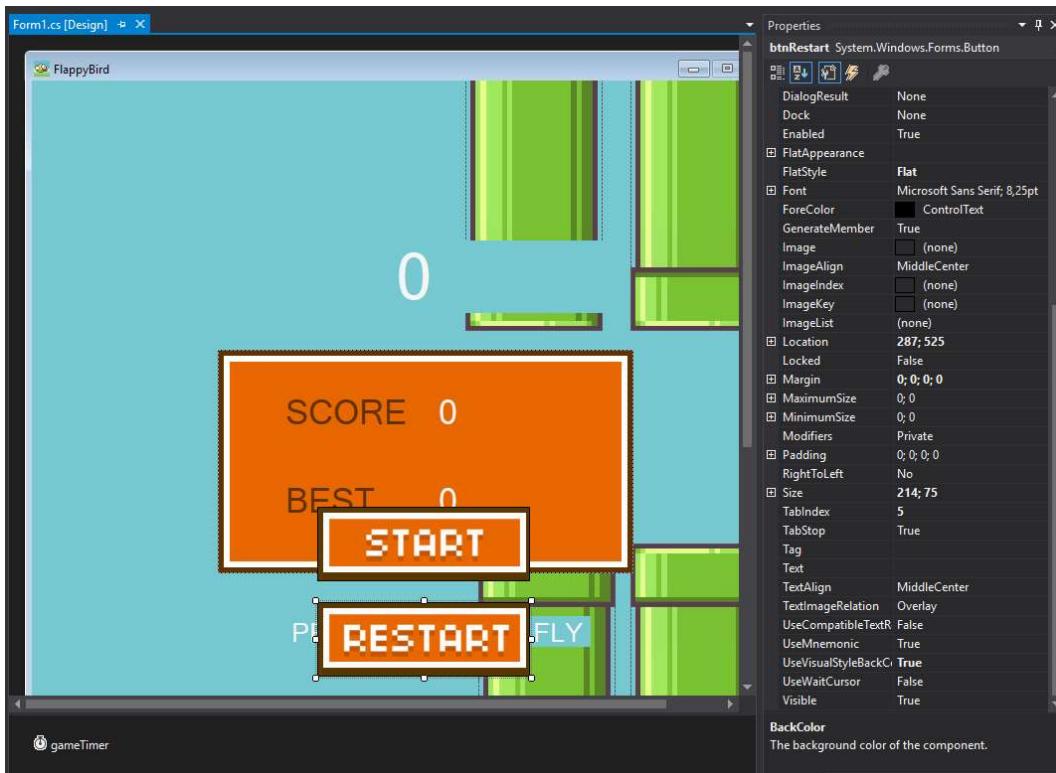
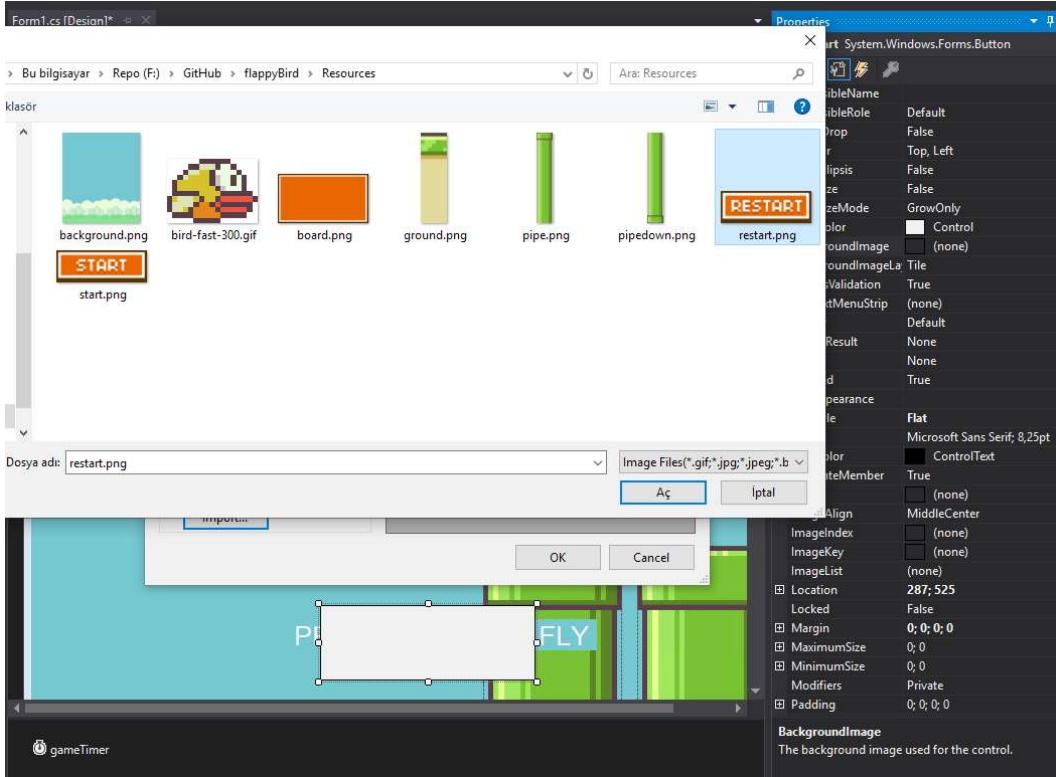


- "button2" elementi için aşağıdaki properties değerleri verilir.

- Name:** btnRestart
- FlatStyle:** Flat
- Location:** 287; 525
- Margin:** 0;0;0;0
- Size:** 214; 75
- Text:** (Boş)

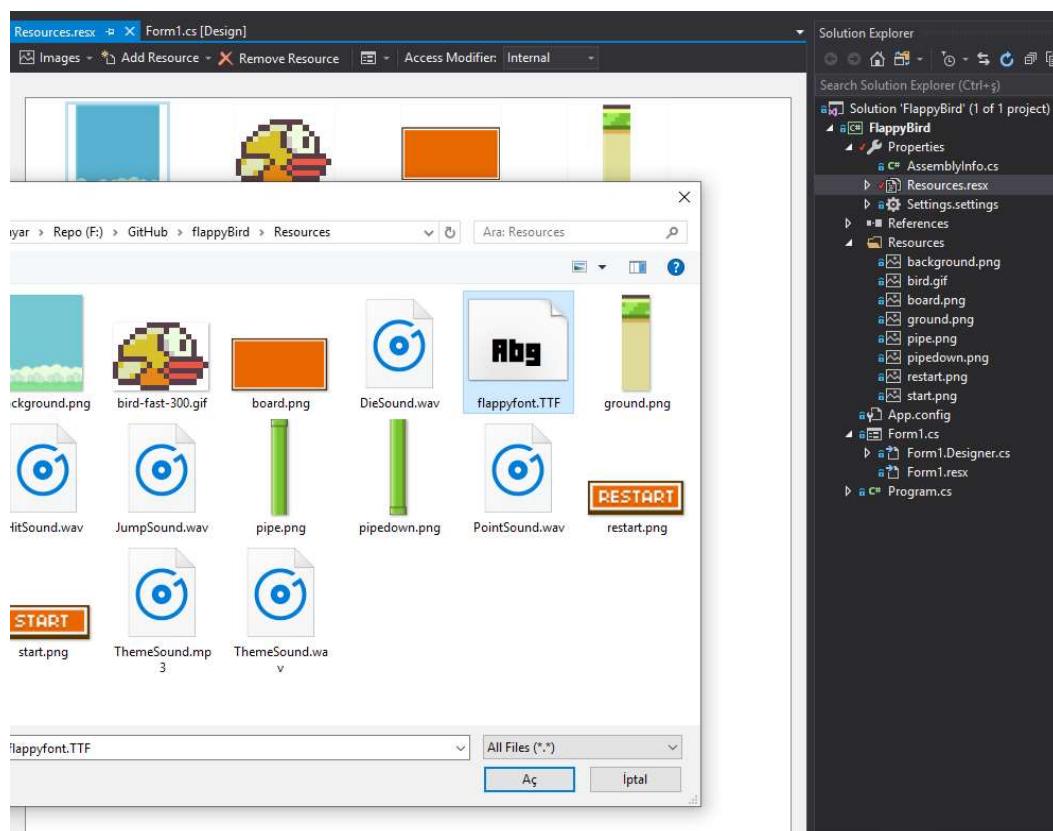
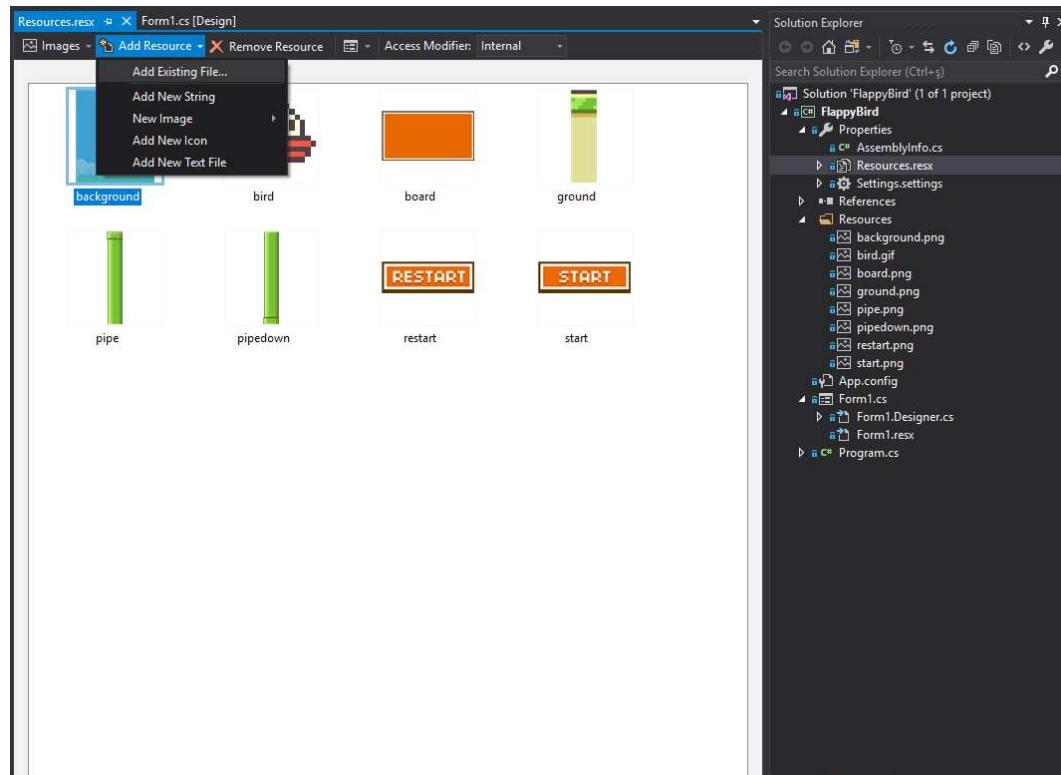


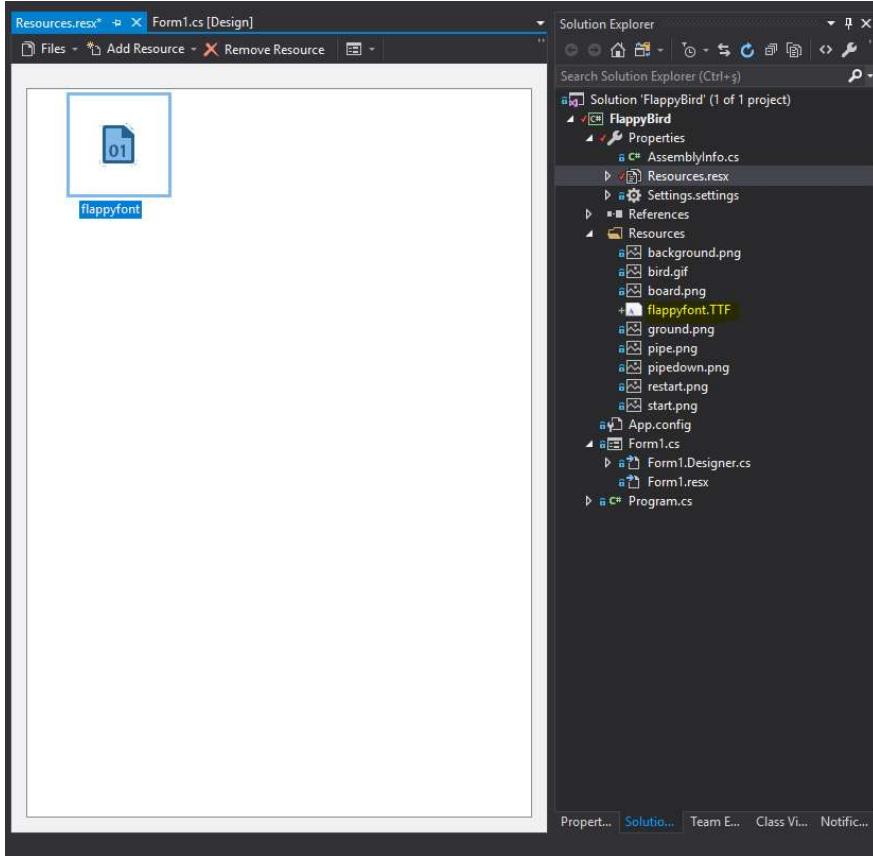
- "**btnRestart**" elementi için resim eklenir. "Properties" listesinden "**BackgroundImage**" eklenir. Açılan ekranda "restart.png" resmi "**Import**" butonu ile seçilerek "**Resources**" bölümüne eklenir.



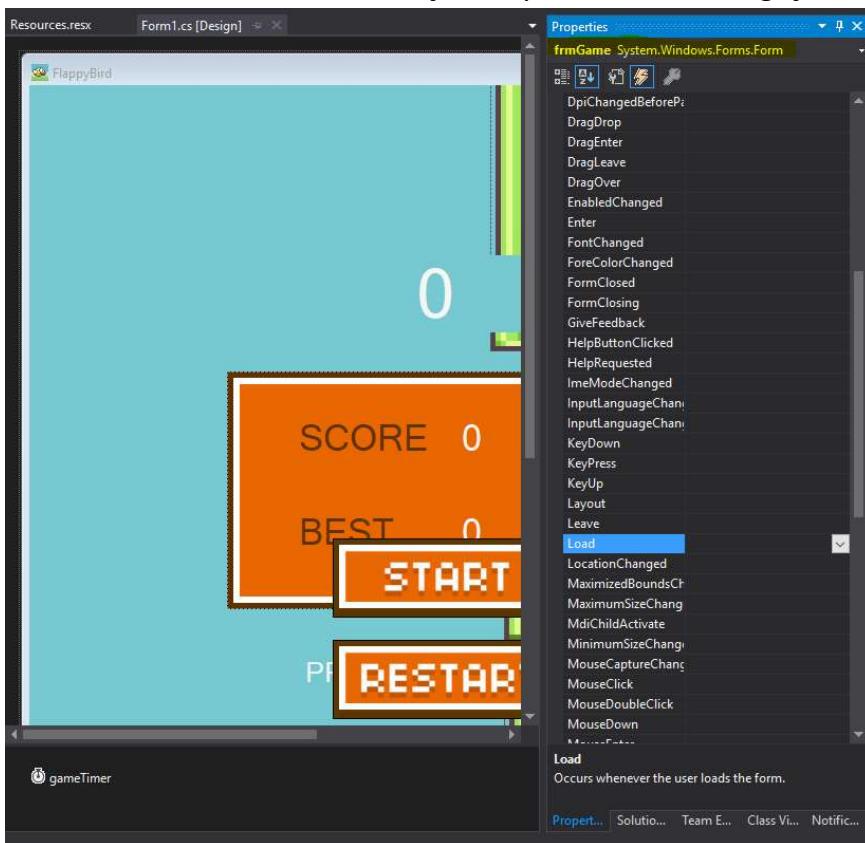
3.7 CustomFont

- Oyun için kullanılan label elementleri için dışarıdan font yüklemesi yapılır. FlappyBird fontunu projeye eklemek için
 - "Solution Explorer" listesinden "Properties" içindeki "Resources.resx" dosyasına çift tıklanarak ".TTF" uzantılı font dosyası eklenir.





- Form Tasarımında "frmGame" properties ekranından "Events" listesi açılır. Listede tanımlı olan "Load" methoduna çift tıkayarak kod kısmına geçilir.



- Label elementlerin "Font" özelliğinde özel yazı tipi koleksiyonunu kullanabilmek için PrivateFontCollection sınıfından FontFamily koleksiyonu oluşturulur. Bu koleksiyon public olarak oluşturulur. Ayrıca özel yazı tipinin oyun çalışmaya başladığı an çalıştığı sürece RAM'de kalmasını sağlamak amacıyla Marshal sınıfına başvurulur.

```


4 references
public partial class frmGame : Form
{
    /*özel yazı tipi koleksiyonu için PrivateFontCollection sınıfından FontFamily koleksiyonu eklenir.*/
    public PrivateFontCollection pfc = new PrivateFontCollection();

    reference
    public frmGame()
    {
        InitializeComponent();
    }

    reference
    private void FrmGame_Load(object sender, EventArgs e)
    {
        /*Resources eklenmiş olan özel yazı tipini oyun çalışmaya başladığı an çalıştığı sürece RAM de tutulması sağlanır.*/
        int fontLength = Properties.Resources.flappyfont.Length;
        byte[] fontdata = Properties.Resources.flappyfont;
        IntPtr data = Marshal.AllocCoTaskMem(fontLength);
        Marshal.Copy(fontdata, 0, data, fontLength);
        pfc.AddMemoryFont(data, fontLength);
        Marshal.FreeCoTaskMem(data);

    }
}


```

Özel fontun Label nesnelerinde kullanılması sağlanır.

```


public partial class frmGame : Form
{
    /*özel yazı tipi koleksiyonu için PrivateFontCollection sınıfından FontFamily koleksiyonu eklenir.*/
    public PrivateFontCollection pfc = new PrivateFontCollection();

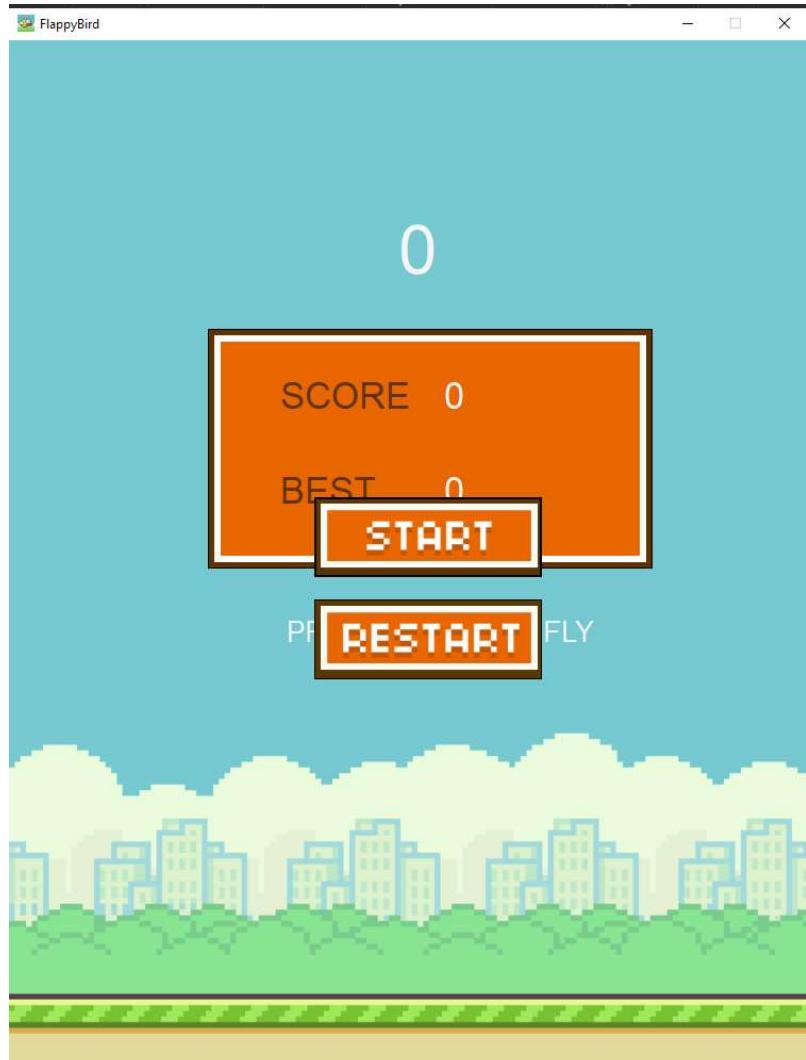
    reference
    public frmGame()
    {
        InitializeComponent();
    }

    reference
    private void FrmGame_Load(object sender, EventArgs e)
    {
        /*Resources eklenmiş olan özel yazı tipini oyun çalışmaya başladığı an çalıştığı sürece RAM de tutulması sağlanır.*/
        int fontLength = Properties.Resources.flappyfont.Length;
        byte[] fontdata = Properties.Resources.flappyfont;
        IntPtr data = Marshal.AllocCoTaskMem(fontLength);
        Marshal.Copy(fontdata, 0, data, fontLength);
        pfc.AddMemoryFont(data, fontLength);
        Marshal.FreeCoTaskMem(data);

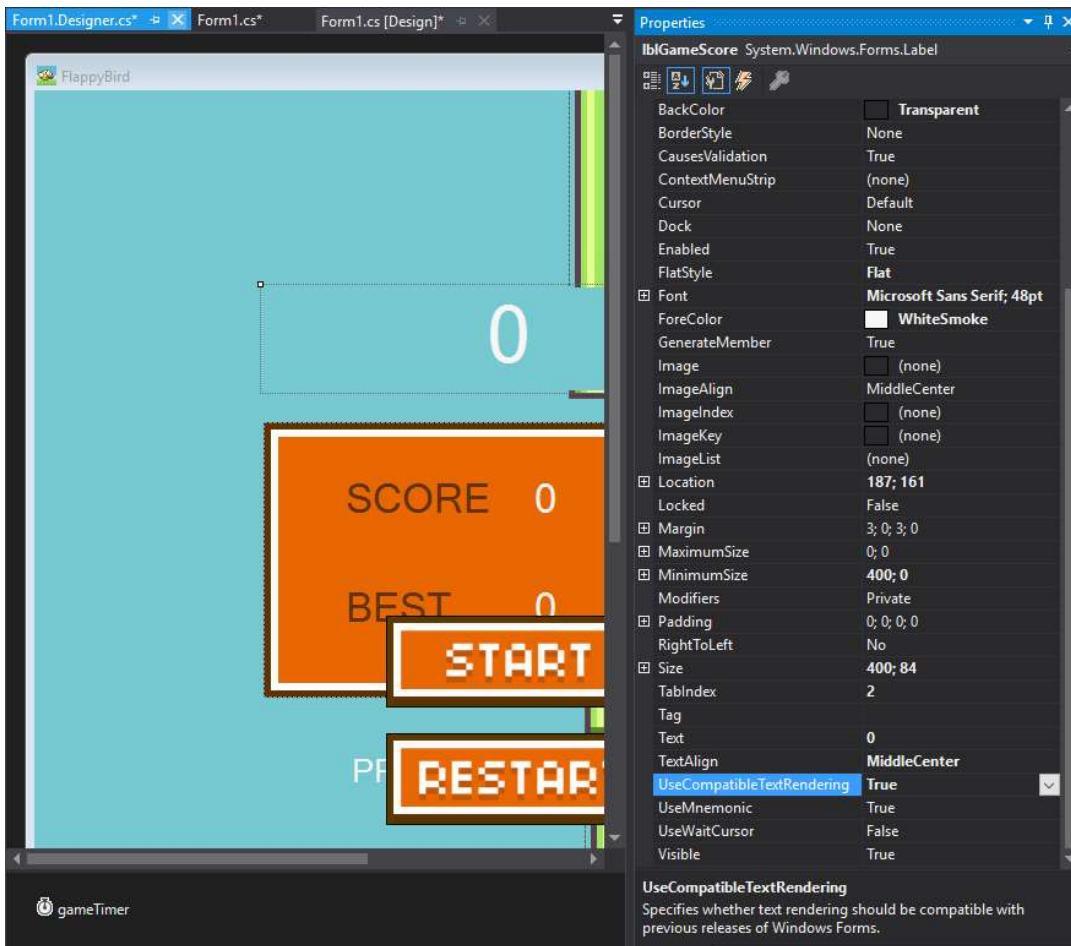
        /*frmGame Üzerine eklenen label elementlerinin font ayarına yüklenen PrivateFontCollection parametresi atanır */
        lblGameScore.Font = new Font(pfc.Families[0], lblGameScore.Font.Size);
        lblHelp.Font = new Font(pfc.Families[0], lblHelp.Font.Size);
        lblBoardScore.Font = new Font(pfc.Families[0], lblBoardScore.Font.Size);
        lblBoardScoreText.Font = new Font(pfc.Families[0], lblBoardScoreText.Font.Size);
        lblBoardBest.Font = new Font(pfc.Families[0], lblBoardBest.Font.Size);
        lblBoardBestText.Font = new Font(pfc.Families[0], lblBoardBestText.Font.Size);
    }
}


```

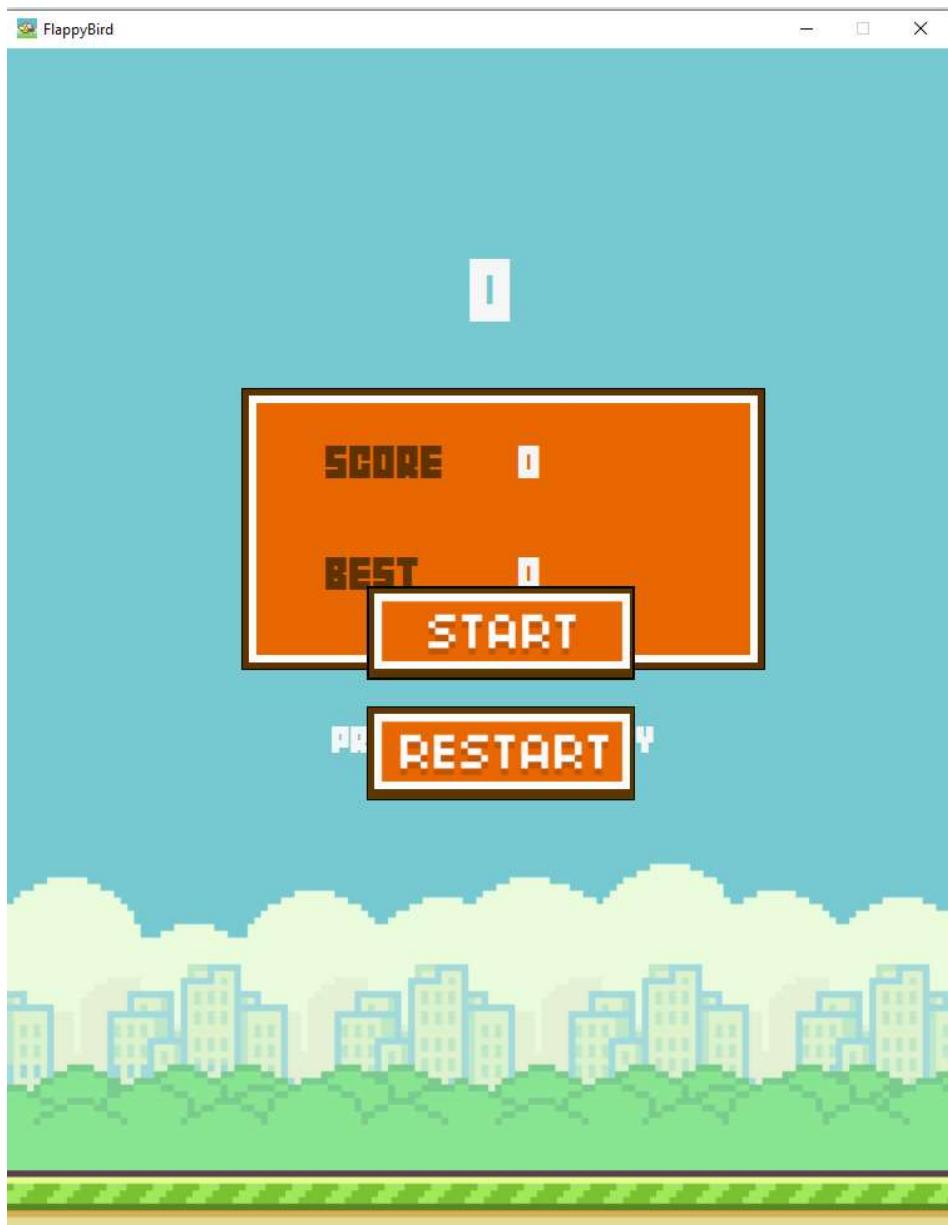
- Program çalıştırıldığında Label elementlerinde font değişimi bu aşamada görüntülenmeyecektir.



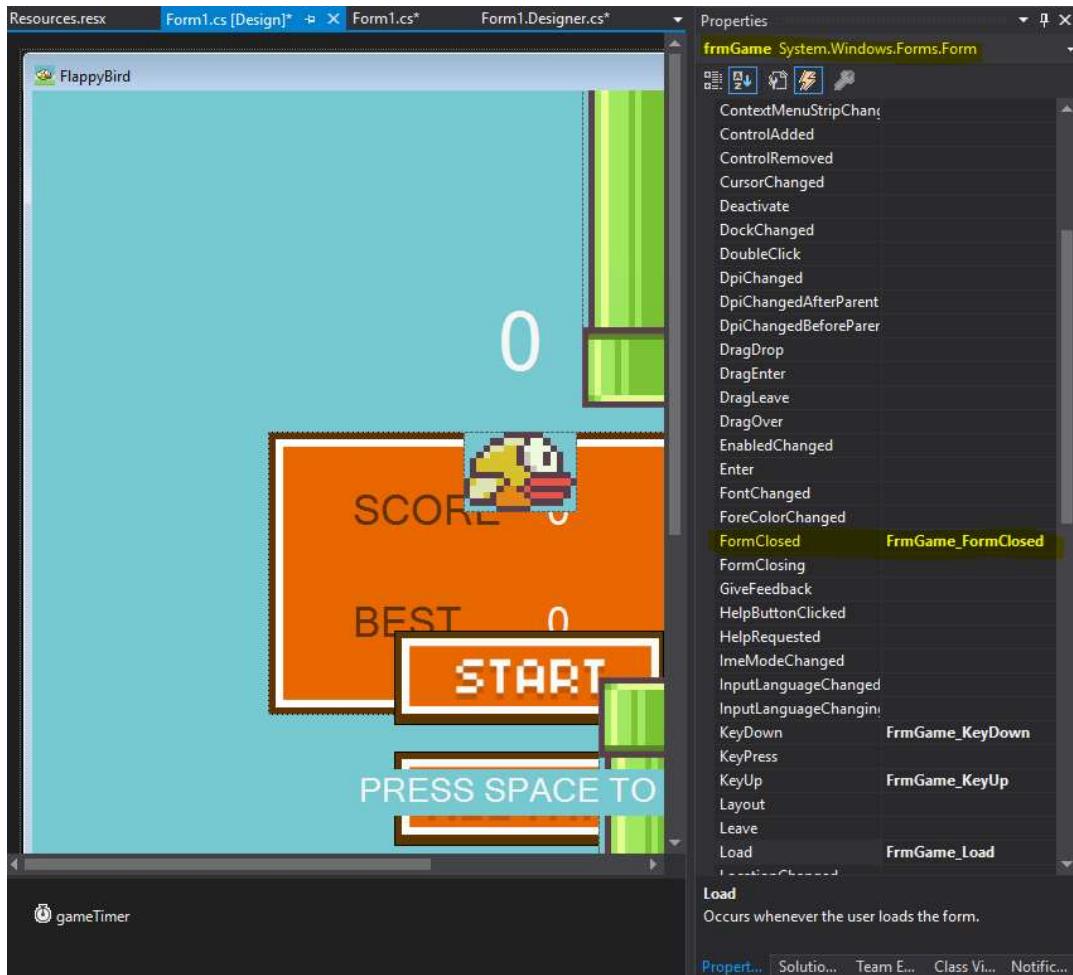
- Label elementinde özel font ailesini aktif olmasını sağlamak için "UseCompatibleTextRendering" değerini "True" olarak ayarlanması gereklidir. frmGame üzerinde eklenen 6 adet label için properties değeri True olarak ayarlanır.



- Label elementlerinin düzenlenmesinden sonra uygulama çalıştırıldığında fontların özel font ile değiştiği görülür.



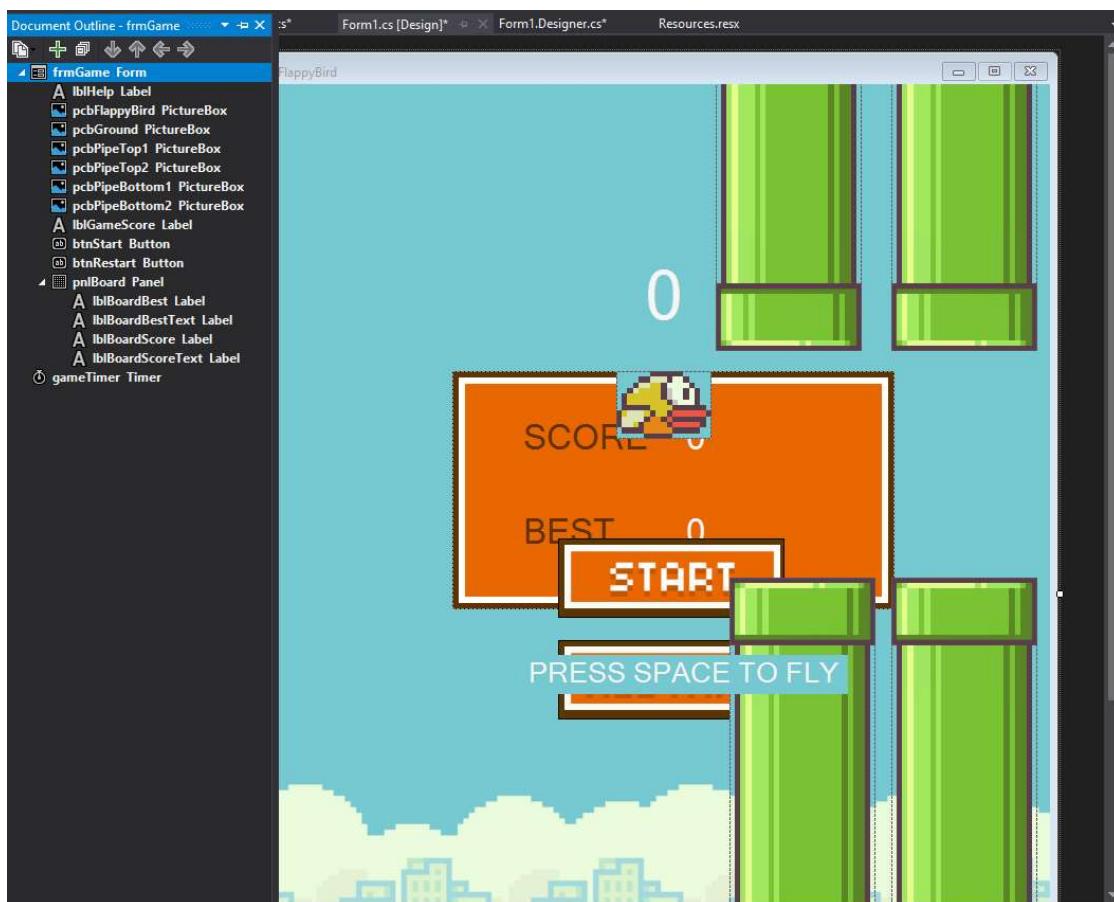
- Form nesnesi kapatıldığında *FontCollection* tarafından kullanılan kaynakları "Dispose" methodu ile serbest bırakır. Bu methodu kullanmak için frmGame event listesinde "FormClosed" eventine çift tıklayarak "**Form1_FormClosed**" methodunun oluşması sağlanır.



```
1 reference
private void FrmGame_FormClosed(object sender, FormClosedEventArgs e)
{
    pfc.Dispose();
}
```

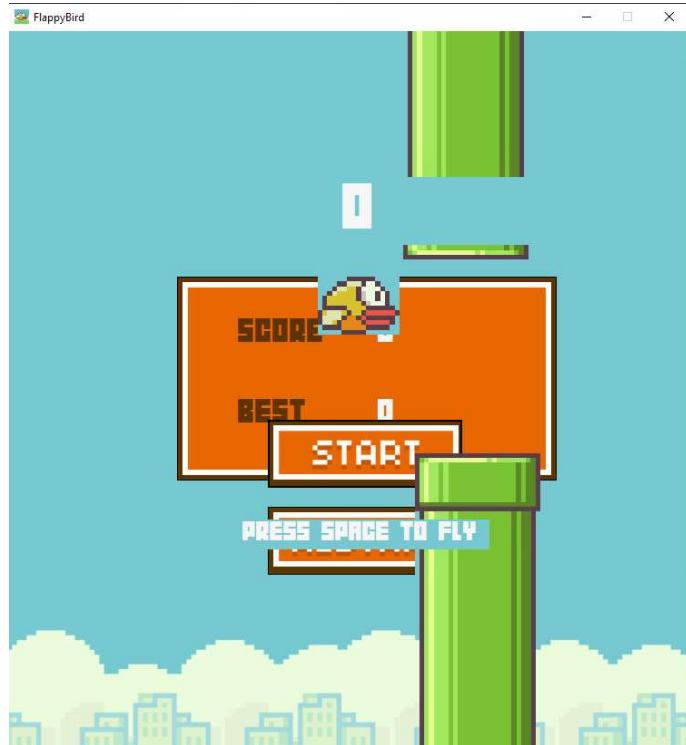
3.8 Hiyarşi

- frmGame üzerine eklenen elementlerin birbirlerinin önünde ya da arkasında olması durumları oyun sürecinde önemli bir durum oluşturmaktadır. Elementlerin sıralamasını yapmak için "Ctrl + Alt + T" tuş kombinasyonu ile "Document Outline" Tool'u sol bölümde açılır. Sıralama aşağıdaki gibi yapılmalıdır.

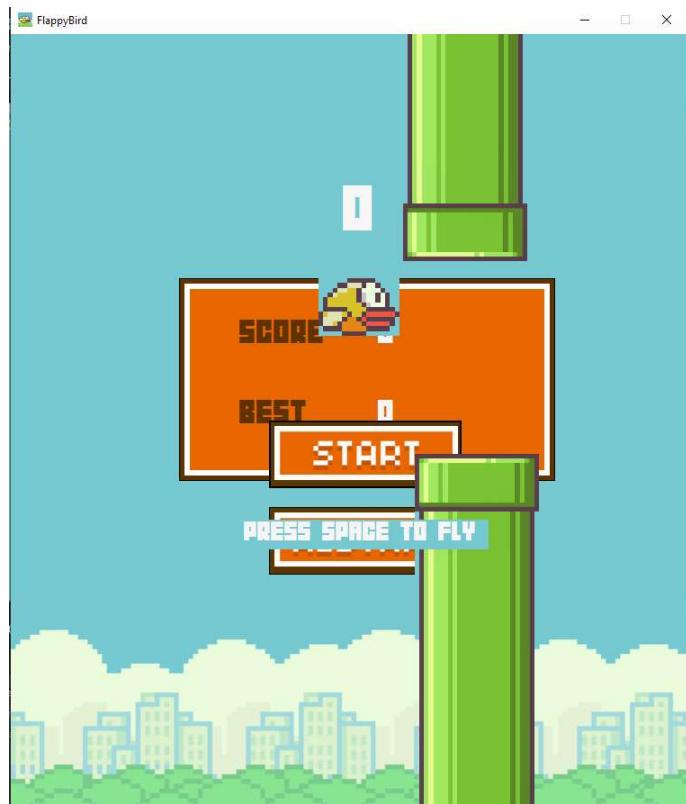


- a. pcbFlappyBird elementi oyunun en önde olmalıdır.
- b. pcbGround elementi pcbPipeBottom1 ve pcbPipeBottom2 nin önünde olmalıdır. Bu şekilde hareket işlemlerinde yerin arka tarafından çıkış görüntüsü oluşturulur
- c. lblGameScore elementi pcbPipeTop1 ve pcbPipeTop2 elementlerinin arkasında olmalıdır. Windows forms uygulamalarında transparent BackColor'a sahip elementlerin hareketli elementler ile üst üste gelmesinde grafiksel olarak uyum bulunmamaktadır. Skor puanını boruların arkasına almak görsel olarak problemi çözmektedir. Aşağıda iki durum görülebilir.

d. lblGameScore pcbPipeTop1 önünde



e. lblGameScore pcbPipeTop1 arkasında



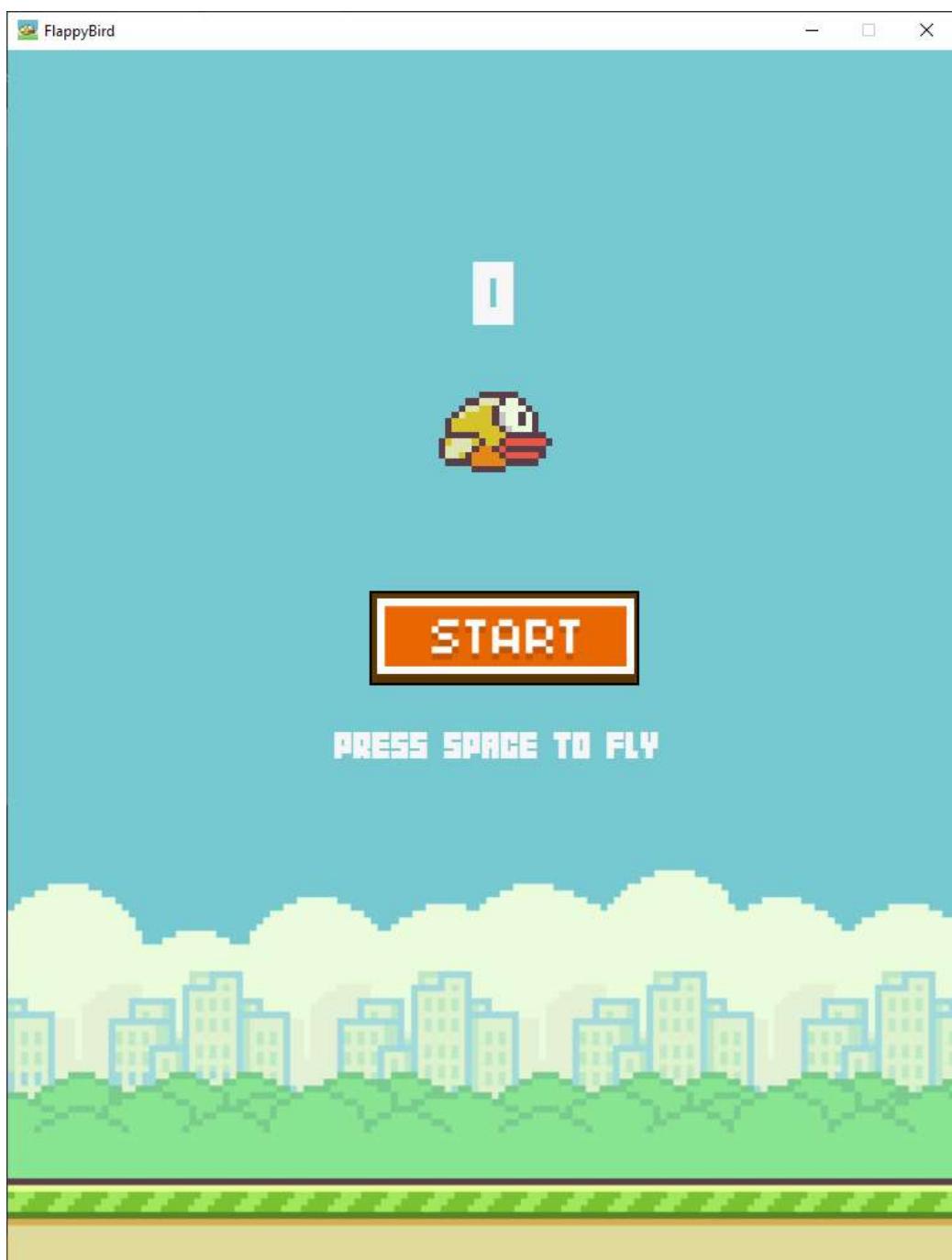
3.9 Gizlilik

- Oyun başlangıç ve bitiş süresince kullanılacak elementlerin "Visible" değerinde değişiklik yapılması gerekmektedir.
- frmGame üzerindeki elementlerin "Properties" listesinde "Visible" değeri aşağıdaki tablo gibi ayarlanır

Element	Visible
lblGameScore	True
pcbFlappyBird	True
btnStart	True
lblHelp	True
pnlBoard	False
btnRestart	False
pcbPipeTop1	False
pcbPipeBottom1	False
pcbPipeTop2	False
pcbPipeBottom2	False

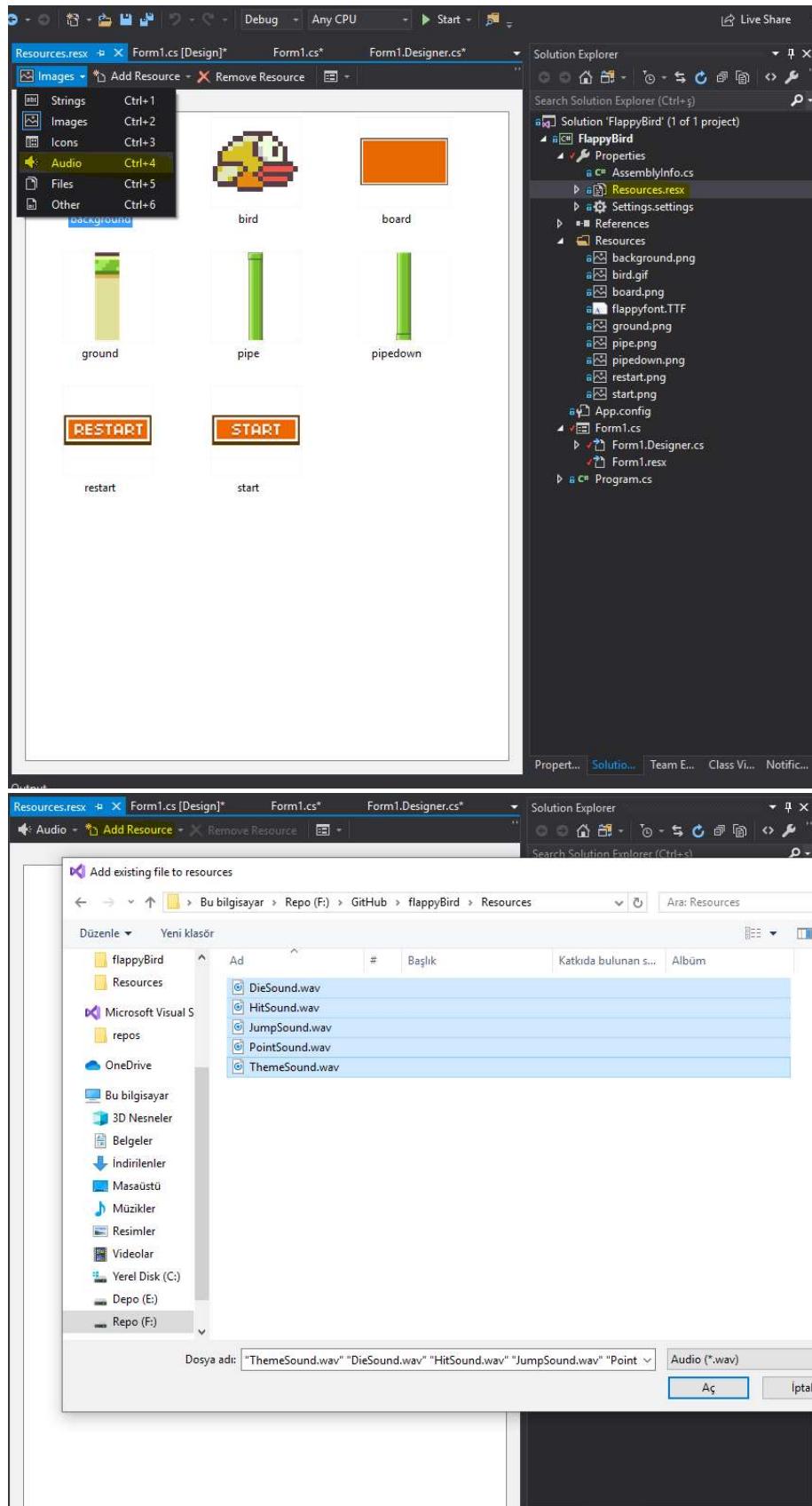
- Oyun sürecinde elementlerin gizlilik/görünürlük durumları;
 - a. **lblGameScore**: Oyun ilk yüklendiğinde ve oyun başlatıldığında, oyun sürecince ve oyun bitiminde görünür durumda kalır.
 - b. **pcbFlappyBird**: Oyun ilk yüklendiğinde ve oyun başlatıldığında ve oyun sürecince görünür durumda kalır. Oyun bitiminde gizli ayarlanır.
 - c. **pcbPipeTop1, pcbPipeBottom1, pcbPipeTop2, pcbPipeBottom2** : Oyun ilk yüklenliğinde gizli olarak ayarlanır. Oyun süresince sol pozisyonda hareket ettirilmesi gerektiğinden oyun başladıkta sonra görünür yapılır. Oyun bitiminde gizli ayarlanır.
 - d. **pnlBoard**: Oyun ilk yüklenliğinde ve oyun sürecinde gizli olarak ayarlanır. Oyun bitiminde görünür yapılır
 - e. **lblHelp**: Oyun ilk yüklenliğinde görünür ayarlanır. Oyun başlatıldığından gizlenir. Oyun sürecince gizli ayarlanır.
 - f. **btnStart**: Oyun ilk yüklendiğinde görünür ayarlanır. Oyun başlatıldığından gizlenir. Oyun sürecince ve oyun bitiminde gizli ayarlanır.
 - g. **btnRestart**: Oyun ilk yüklendiğinde, oyun başlatıldığından, oyun süresince gizli ayarlanır. Oyun bitiminde görünür yapılır

- Düzenlemeler sonunda oyun ilk açılış görüntüsü aşağıdaki gibi olmalıdır.



3.10 Ses Dosyaları

- Oyun sırasındaki olaylar için ses dosyalarını kullanmak üzere projenin "Resources" kayınağına eklenir.



4. Kodlama

- Oyun için kullanılacak parametreler Form1.cs içine tanımlanır.
 - a. bool jumping = false
 - b. int pipeSpeed = 5
 - c. int gravity = 5
 - d. int inscore = 0

The screenshot shows the Microsoft Visual Studio interface. In the Solution Explorer on the right, the project 'FlappyBird' is selected, displaying its structure: Properties, AssemblyInfo.cs, Resources.resx, Settings.settings, References, Resources, App.config, Form1.cs, Form1.Designer.cs, Form1.resx, and Program.cs. The Form1.cs file is open in the main editor window, showing C# code for a Windows Form application. The code defines a class named Form1 that contains variables for jumping, pipe speed, gravity, and score, along with a constructor that initializes the component.

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace FlappyBird
12 {
13     public partial class Form1 : Form
14     {
15         bool jumping = false;
16         int pipeSpeed = 5;
17         int gravity = 5;
18         int inscore = 0;
19
20         public Form1()
21         {
22             InitializeComponent();
23         }
24     }
25 }
26
```

- Oyun içinde bilgilendirme yapılacak olan labeller için Text değerleri ayarlanır.
 - endText1.Text = "Oyun Bitti!";
 - endText2.Text = "Kazandığınız Puan: " + inscore;
 - gameDesigner.Text = "Game Designed By Hamide";
 - endText1.Visible = false;
 - endText2.Visible = false;
 - gameDesigner.Visible = false;

The screenshot shows the Visual Studio IDE interface. On the left, the code editor displays `Form1.cs` with the following content:

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11  namespace FlappyBird
12  {
13      public partial class Form1 : Form
14      {
15          bool jumping = false;
16          int pipeSpeed = 5;
17          int gravity = 5;
18          int inscore = 0;
19
20          public Form1()
21          {
22              InitializeComponent();
23
24              endText1.Text = "Oyun Bitti!";
25              endText2.Text = "Kazandığınız Puan: " + inscore;
26              gameDesigner.Text = "Game Designed By Hamide";
27              endText1.Visible = false;
28              endText2.Visible = false;
29              gameDesigner.Visible = false;
30          }
31
32
33      }
34  }
```

On the right, the Solution Explorer shows the project structure:

- Solution 'FlappyBird' (1 of 1 proj)
 - FlappyBird
 - Properties
 - AssemblyInfo.cs
 - Resources.resx
 - Settings.settings
 - References
 - Resources
 - App.config
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Program.cs

- Oyun için 4 adet method oluşturulur.
 - a. Timer
 - b. KeyDown
 - c. KeyUp
 - d. GameEnd

4.1 Global Değişkenler

- Oyun sırasında methodlar arası kullanılmak üzere tanımlanan global değişkenler aşağıdaki gibidir.

- **birdTopCount:** Space tuşu (Key Up/Key Down) ile değer alarak time methodu ile flappyBird nesnesini yukarı / aşağı taşınmasını sağlamak için kullanılır.

```
public int birdTopCount = 0;
```

- **gameScore:** Bir oyun başlayıp bitme sürecinde her bir pipeden geçişde alınan 1 puanı tutmak için kullanılır

```
public int gameScore = 0;
```

- **gameBestScore :** Oyundan çıkış yapılmadığı sürece her yeni oyunda en yüksek elde edilen puanı tutmak için kullanılır

```
public int gameBestScore = 0;
```

```
4 references
public partial class frmGame : Form
{
    public PrivateFontCollection pfc = new PrivateFontCollection();

    public int birdTopCount = 0;

    public int gameScore = 0;

    public int gameBestScore = 0;

    1 reference
    public frmGame()
    {
        InitializeComponent();
    }

    2 references
```

4.2 PipesReset

Form üzerine yerleştirilen pipe nesnelerinin her oyun başlatıldığında pozisyonlarının sabitlenmesi gerekmektedir. Bu amaçla aşağıdaki method Form1.cs içerisinde eklenir.

```
2 references
private void PipesReset()
{
    pcbPipeTop1.Location = new System.Drawing.Point(602, -535);
    pcbPipeBottom1.Location = new System.Drawing.Point(602, 500);
    pcbPipeTop2.Location = new System.Drawing.Point(1000, -535);
    pcbPipeBottom2.Location = new System.Drawing.Point(1000, 500);
    pcbFlappyBird.Location = new System.Drawing.Point(345, 271);

    pcbPipeTop1.Visible = true;
    pcbPipeBottom1.Visible = true;
    pcbPipeTop2.Visible = true;
    pcbPipeBottom2.Visible = true;
    pcbFlappyBird.Visible = true;
}
```

4.3 PlaySound

Oyun sırasında oluşan durumlarda ses çalma methodları oluşturulur

- **PlaySound:** "pcbFlappyBird" nesnesinin oyun sırasında durumları için kullanılan methoddur.

```
5 references
private void PlaySound(string soundName)
{
    SoundPlayer sound = new SoundPlayer();

    switch (soundName)
    {
        case "jump":
            sound.Stream = Properties.Resources.JumpSound;
            break;
        case "point":
            sound.Stream = Properties.Resources.PointSound;
            break;
        case "die":
            sound.Stream = Properties.Resources.DieSound;
            break;
        case "hit":
            sound.Stream = Properties.Resources.HitSound;
            break;
    }

    sound.Play();
}
```

- **PlayThemeSound:** Oyun başlangıç ve bitiş ekranında tema müzik dosyasının çalınmasını sağlayan methodur.

```
2 references
private void PlayThemeSound()
{
    SoundPlayer sound = new SoundPlayer(Properties.Resources.ThemeSound);
    sound.PlayLooping();
}
```

4.4 FormLoad

- frmGame load methodunda custom font düzenlemeleri dışında oyun tema müziğinin çalışması sağlanır.

```
2 references
private void FrmGame_Load(object sender, EventArgs e)
{
    int fontLength = Properties.Resources.flappyfont.Length;
    byte[] fontdata = Properties.Resources.flappyfont;
    IntPtr data = Marshal.AllocCoTaskMem(fontLength);
    Marshal.Copy(fontdata, 0, data, fontLength);
    pfc.AddMemoryFont(data, fontLength);
    Marshal.FreeCoTaskMem(data);

    lblGameScore.Font = new Font(pfc.Families[0], lblGameScore.Font.Size);
    lblHelp.Font = new Font(pfc.Families[0], lblHelp.Font.Size);
    lblBoardScore.Font = new Font(pfc.Families[0], lblBoardScore.Font.Size);
    lblBoardScoreText.Font = new Font(pfc.Families[0], lblBoardScoreText.Font.Size);
    lblBoardBest.Font = new Font(pfc.Families[0], lblBoardBest.Font.Size);
    lblBoardBestText.Font = new Font(pfc.Families[0], lblBoardBestText.Font.Size);

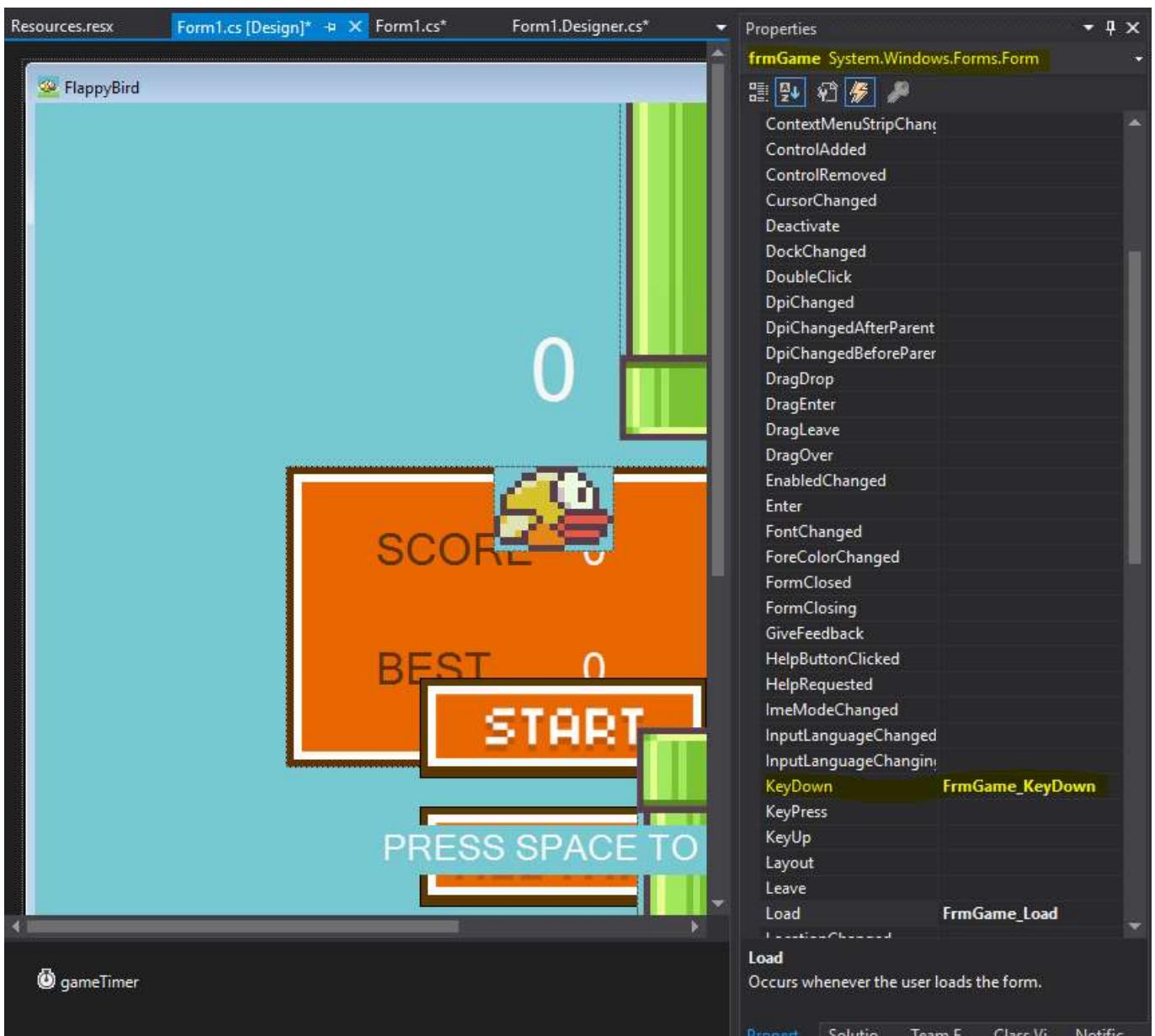
    PlayThemeSound();
}
```

4.5 From KeyDown

- "frmGame" üzerinde sağ tıklayarak formun Properties listesi açılır.



- Listedeki Events butonuna tıklayarak listeden "KeyDown" bulunarak çift tıklanarak "FrmGame_KeyDown" methodunun olması sağlanır



- Form üzerinde "Space" tuşuna basıldığında "pcbFlappyBird" PictureBox elementinin yukarı hareket etmesini sağlamak için global değişkenlerden "**birdTopCount**" değerinin 15 birim azaltılması sağlanır. Her bir yukarı hareket işleminde "jump" sesi çalınır. Timer Tick methodunda pcbFlappyBird nesnesinin top pozisyonu **birdTopCount** değeri kadar arttırılmaktadır. Negatif değere sahip olunduğuunda pcbFlappyBird nesnesinin top pozisyonu azalır ve yukarı doğru hareketi sağlanmış olur.

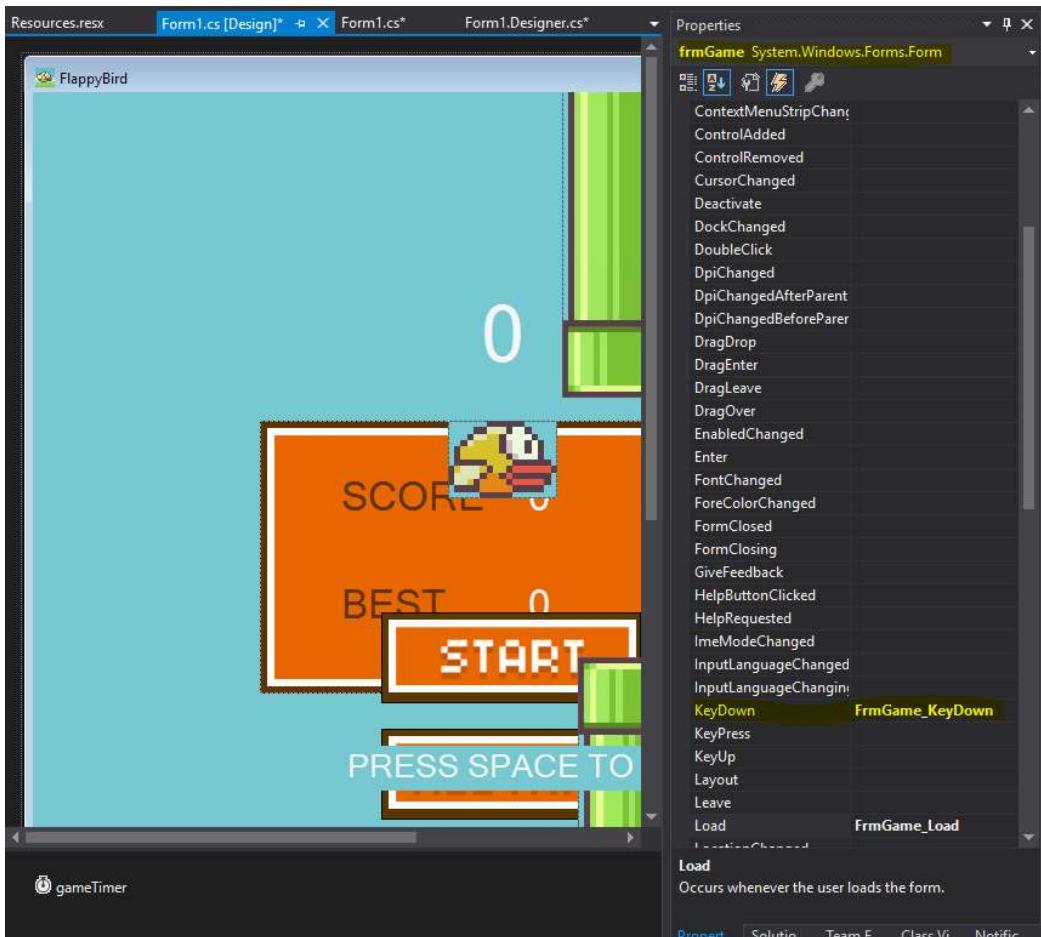
```
1 reference
private void FrmGame_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Space)
    {
        birdTopCount = -15;
        PlaySound("jump");
    }
}
```

4.6 Form KeyUp

- "frmGame" üzerinde sağ tıklayarak formun Properties listesi açılır.



- Listedeki Events butonuna tıklayarak listeden "KeyUp" bulunarak çift tıklanarak "FrmGame_KeyUp" methodunun oluşması sağlanır



- Form üzerinde "Space" tuşuna basılmadığı anlarda "pcbFlappyBird" PictureBox elementinin aşağı hareket etmesini sağlamak için global değişkenlerden "**birdTopCount**" değerinin 15 birim atanması sağlanır. Timer Tick methodunda pcbFlappyBird nesnesinin top pozisyonu **birdTopCount** değeri kadar arttırılmaktadır. Pozitif sabit değere sahip olunduğunda pcbFlappyBird nesnesinin top pozisyonu artar ve aşağı doğru hareketi sağlanmış olur.

```
1 reference
private void FrmGame_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Space)
    {
        birdTopCount = 15;
    }
}
```

4.7 StartGame

Oyunun başlaması için nesnelerin ve değişkenlerin ayarlanması yapılır.

- Oyun puanı yeniden başlatılması için gameScore global değişkeninin değeri 0 olarak atanır.

GameScore = 0;

- Oyun sırasında herbir pipe nesnesi geçişinde kazanılan 1 puan oyuncuya gösterimi sağlanır

lblGameScore.Text = "0";

- Oyun başlaması için flappyBird ve pipe nesnelerinin pozisyonları ve gizlilikleri düzenlenir.

PipesReset();

- Oyun başladıkten sonra btnStart ve lblHelp nesneleri gizlenir.

btnStart.Visible = false;

lblHelp.Visible = false;

- Oyun başladıkten sonra btnStart ve pnlBoard nesneleri gizlenir.

btnStart.Visible = false;

pnlBoard.Visible = false;

- Oyun başlaması için flappyBird ve pipe nesnelerinin pozisyonları ve gizlilikleri düzenlenir.

PipesReset();

- Oyun sırasında herbir pipe nesnesi geçişinde kazanılan 1 puan oyuncuya gösterimi sağlanır

lblGameScore.Text = "0";

- Eğer, bir önceki sonlanan oyun sırasında pcbFlappyBird nesnesini hareket ettirmek için arttırılmış / azaltılmış birdTopCount değeri var ise yeniden 0 olarak atanır.

birdTopCount = 0

- Oyun algoritmasının devreye alınması sağlanır.

gameTimer.Start();

```
3 references
private void StartGame()
{
    gameScore = 0;

    lblGameScore.Text = "0";

    PipesReset();

    btnStart.Visible = false;
    lblHelp.Visible = false;

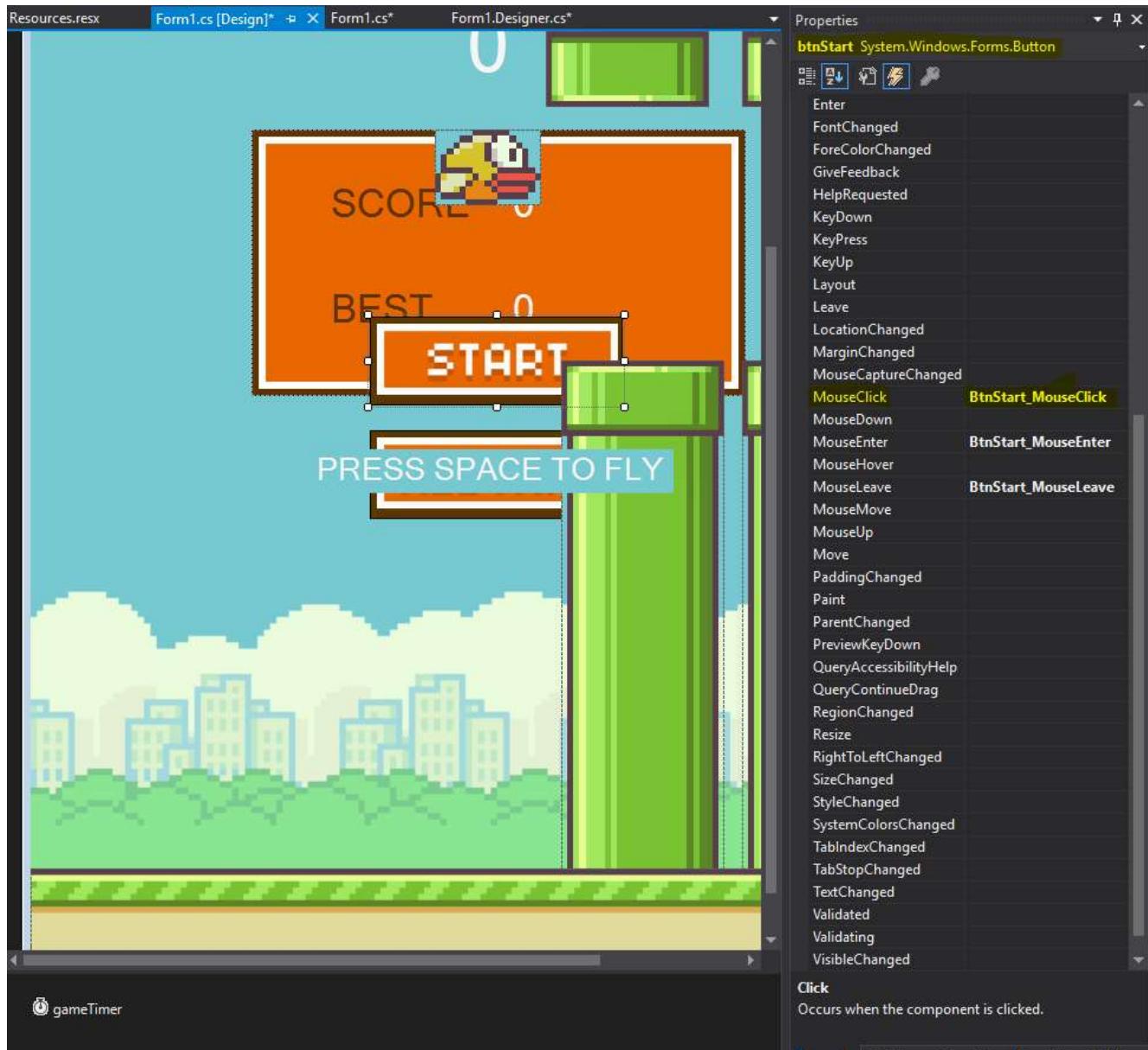
    btnRestart.Visible = false;
    pnlBoard.Visible = false;

    birdTopCount = 0;

    gameTimer.Start();
}
```

4.8 btnStart

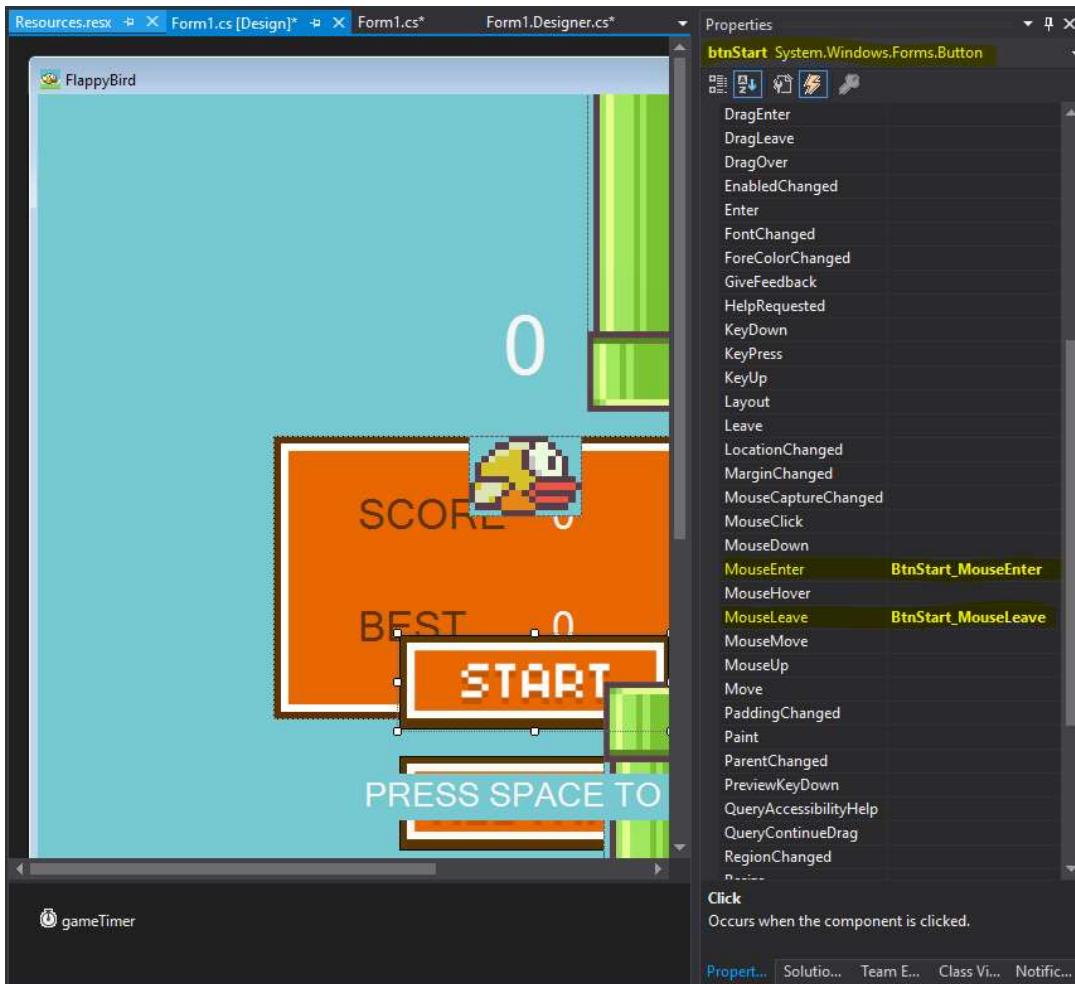
- "btnStart" elementinin events listesinden "MouseClick" eventine çift tıklayarak "BtnStart_MouseClick" methodunun oluşması sağlanır.



- BtnStart_MouseClick methodu ile oyunun başlaması için gerekli olan işlemlerin yapılması için EndGame methodu çağrılır.

```
private void BtnStart_MouseClick(object sender, MouseEventArgs e)
{
    StartGame();
}
```

- "btnStart" butonunda oyuncu için fare ile üzerine gelindiğinde animasyon yapılarak oyun için hareketlilik katılabilir.



- "BtnStart_MouseEnter" event methodu ile btnStart üst (top) pozisyonu 2 pixel arttırılması sağlanarak aşağı doğru hareket etmesi sağlanır.

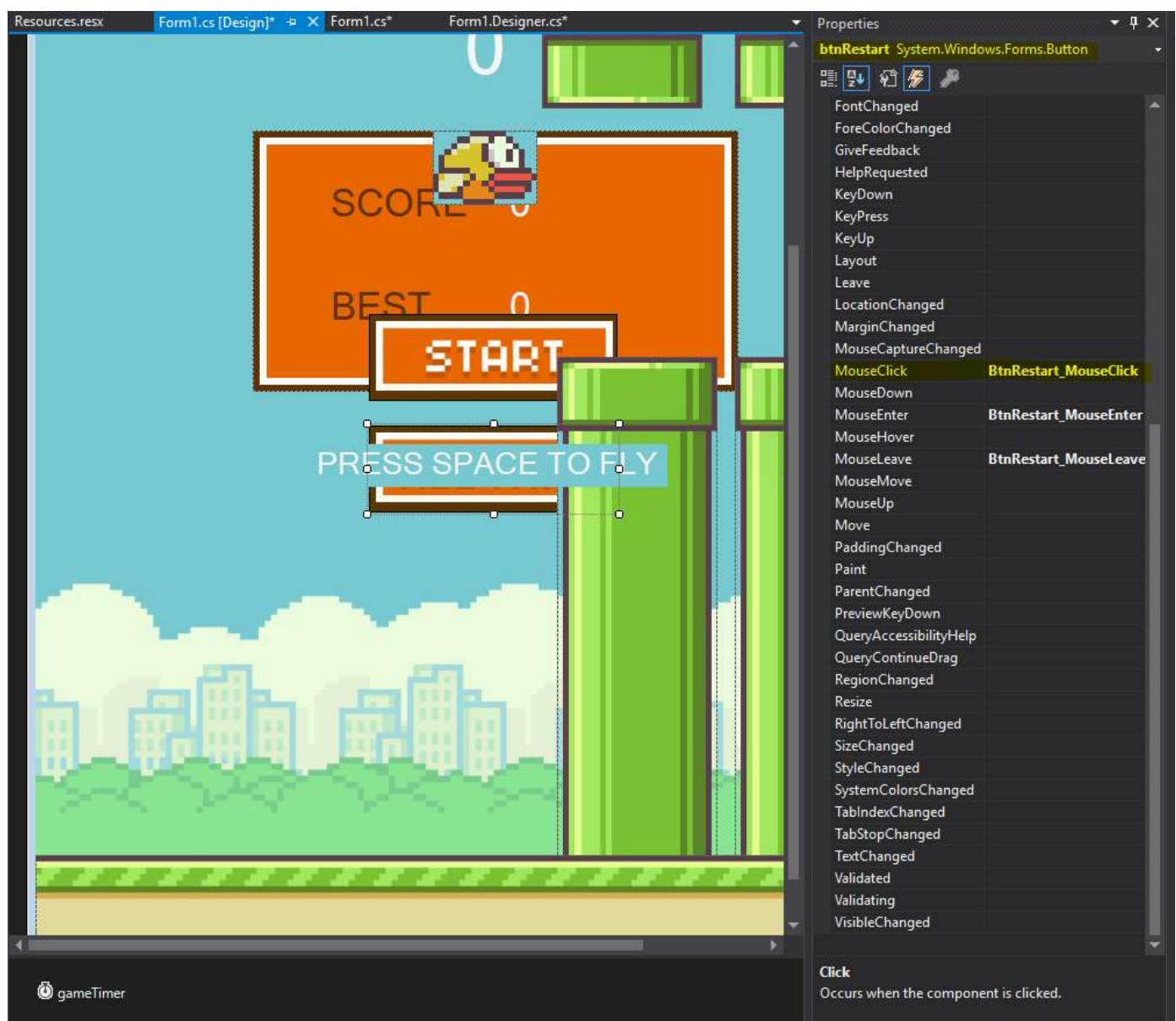
```
1 reference
private void BtnStart_MouseEnter(object sender, EventArgs e)
{
    btnStart.Top += 2;
}
```

- "BtnStart_MouseLeave" event methodu ile btnStart üst (top) pozisyonu 2 pixel azaltılması sağlanarak yukarı doğru hareket etmesi sağlanır.

```
1 reference
private void BtnStart_MouseLeave(object sender, EventArgs e)
{
    btnStart.Top -= 2;
}
```

4.9 btnRestart

- "btnRestart" elementinin events listesinden "Click" eventine çift tıklayarak "BtnRestart_Click" methodunun oluşması sağlanır.

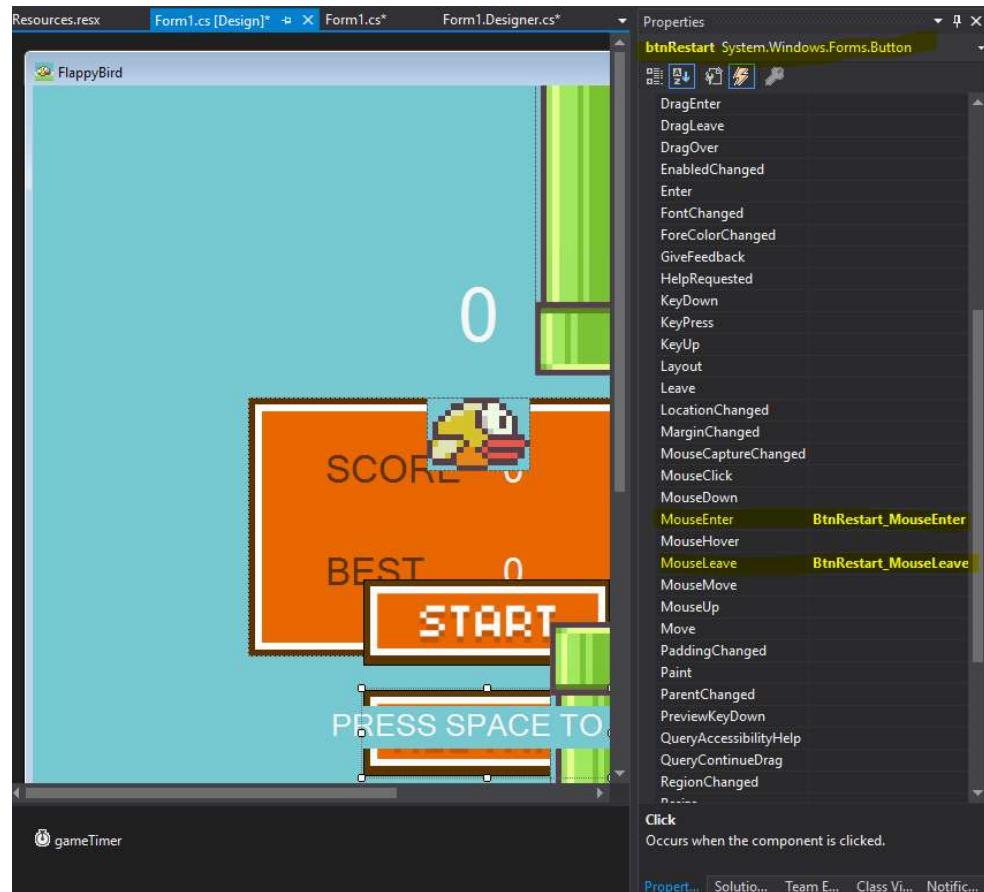


- BtnRestart_Click methodu ile oyunun yeniden başlaması için gerekli olan işlemlerin yapılması için EndGame methodu çağrılır.
- "btnRestart" butonunda oyuncu için fare ile üzerine gelindiğinde animasyon yapılarak oyun için hareketlilik katılabilir.

```

    1 reference
    private void BtnRestart_MouseClick(object sender, MouseEventArgs e)
    {
        | StartGame();
    }

```



- "BtnRestart_MouseEnter" event methodu ile btnStart üst (top) pozisyonu 2 pixel arttırılması sağlanarak aşağı doğru hareket etmesi sağlanır.

```

    1 reference
    private void BtnRestart_MouseEnter(object sender, EventArgs e)
    {
        btnRestart.Top += 2;
    }

```

- "BtnRestart_MouseLeave" event methodu ile btnStart üst (top) pozisyonu 2 pixel azaltılması sağlanarak yukarı doğru hareket etmesi sağlanır.

```
lreference
private void BtnRestart_MouseLeave(object sender, EventArgs e)
{
    btnRestart.Top -= 2;
}
```

4.10 EndGame

Oyun sonlanması durumunda Oyun sonu ve Oyun tekrarı işlemlerinin yapılması sağlanır.

- Form1.cs içerisinde "EndGame" methodu oluşturulur. Oyun bitiminde TimerTick methodu durdurulur. Ardından oyuncuya oyun bittiğini ve kazandığı puanın bilgisinin verildiği "pnlBoard" nesnesinin gösterimi sağlanır.

Oyun bitişinde pipe nesneleri ekrandan gizlenmelidir. Oyun bittiğine dair ses dosyası çalınır ve 10 sn sonra tema müziği başlatılır. Bu şekilde oyuncunun oyun bittiğine dair akıcılığı sağlanır.

```
5 references
private void EndGame()
{
    PlaySound("hit");

    lblGameScore.Text = "Game Over";

    gameTimer.Stop();

    System.Threading.Thread.Sleep(1000);

    if (gameScore > gameBestScore)
    {
        gameBestScore = gameScore;
    }

    PlaySound("die");

    pcbPipeTop1.Visible = false;
    pcbPipeBottom1.Visible = false;
    pcbPipeTop2.Visible = false;
    pcbPipeBottom2.Visible = false;
    btnRestart.Visible = true;
    pcbFlappyBird.Visible = false;

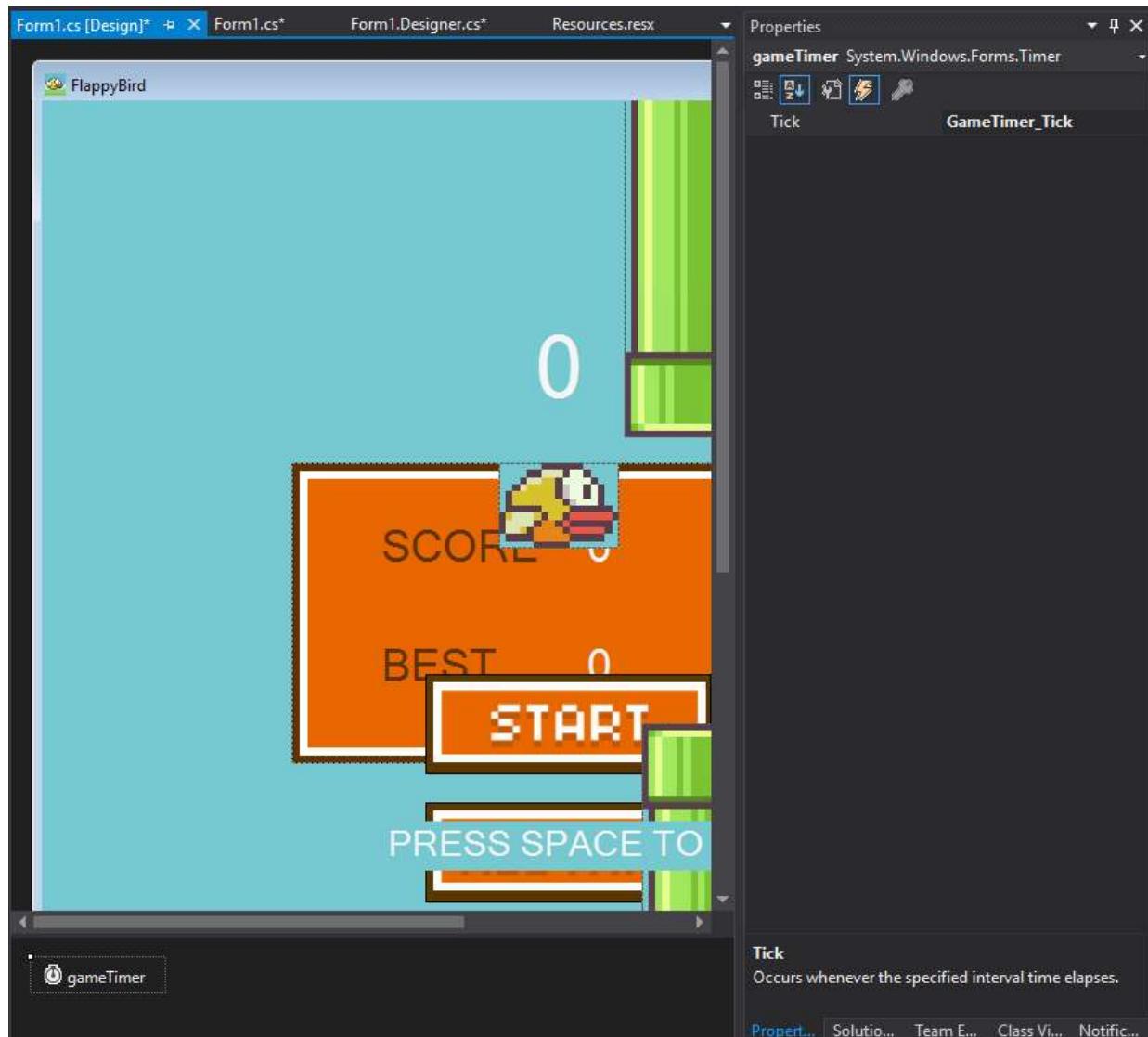
    lblBoardScore.Text = gameScore.ToString();
    lblBoardBest.Text = gameBestScore.ToString();

    pnlBoard.Visible = true;

    PlayThemeSound();
}
```

4.11 TimerTick

- Form Design ekranındaki "gameTimer" elementine çift tıklayarak ilgili methodun oluşturulması sağlanır.



- gameTimer her çalıştığında "pcbFlappyBird" nesnesinin "pcbPipeTop1", "pcbPipeTop2", "pcbPipeBottom1", "pcbPipeBottom2" ve "pcbGround" nesneleri ile aynı pozisyonda olup olmadığı kontrol edilir. Aynı pozisyonlarda tespit edilir ise oyun sonlanır.

```

1 reference
private void GameTimer_Tick(object sender, EventArgs e)
{
    if ((pcbPipeTop1.Bottom + birdTopCount) >= pcbGround.Bounds.Top)
    {
        EndGame();
    }
    else if (pcbFlappyBird.Bounds.IntersectsWith(pcbPipeBottom1.Bounds))
    {
        EndGame();
    }
    else if (pcbFlappyBird.Bounds.IntersectsWith(pcbPipeTop1.Bounds))
    {
        EndGame();
    }
    else if (pcbFlappyBird.Bounds.IntersectsWith(pcbPipeBottom2.Bounds))
    {
        EndGame();
    }
    else if (pcbFlappyBird.Bounds.IntersectsWith(pcbPipeTop2.Bounds))
    {
        EndGame();
    }
    else
    {
    }
}

```

- Nesnelerin pozisyon kontrollerinde oyun sonlanması yaşanmadı ise;
 - Pipe nesnelerinden en az 1'inden geçmiş olduğu kabul edilir. Pipe nesnesinden geçme durumunda 1 puan kazanılır. "Point" ses kaydı çalınır.

```

if (pcbFlappyBird.Left >= pcbPipeTop1.Right && pcbFlappyBird.Left - 10 <= pcbPipeTop1.Right)
{
    gameScore++;
    lblGameScore.Text = gameScore.ToString();
    PlaySound("point");
}
else if (pcbFlappyBird.Left >= pcbPipeTop2.Right && pcbFlappyBird.Left - 10 <= pcbPipeTop2.Right)
{
    gameScore++;
    lblGameScore.Text = gameScore.ToString();
    PlaySound("point");
}

```

- "pcbFlappyBird" nesnesinin üst (top) pozisyonu "space" tuşu ile değeri atanan "birdTopCount" değeri kadar arttırılır. Bu şekilde yer çekimine doğru hareket sağlanır.

```

pcbFlappyBird.Top += birdTopCount;

```

- "pcbPipeTop1", "pcbPipeTop2", "pcbPipeBottom1" ve "pcbPipeBottom2" PictureBox elementlerinin sol (left) pozisyonları 10 birim kadar azaltılır. Bu şekilde yer çekimine yatay hareket sağlanır.

```
pcbPipeBottom1.Left -= 10;
pcbPipeTop1.Left -= 10;

pcbPipeBottom2.Left -= 10;
pcbPipeTop2.Left -= 10;
```

- "pcbPipeTop1", "pcbPipeTop2", "pcbPipeBottom1" ve "pcbPipeBottom2" yeni pozisyonları "frmGame" nesnesinin sol (left) pozisyonunu geçip geçmediği kontrol edilir. Bir pipenin form ekranından kaybolması için kendi genişliği kadar 0 değerini geçmiş olması kabul edilir.
- Pipe ekranдан kaybolmuş ise form elementinin genişliği kadar sol (left) değeri verilir. Bu şekilde ekranının sol hizasından çıkararak sağ hizasından tekrar ekranда görünmesi sağlanır.

```
Random rnd = new Random();

if (pcbPipeBottom1.Left < -pcbPipeBottom1.Width)
{
    pcbPipeBottom1.Left = this.Size.Width;
    pcbPipeTop1.Left = this.Size.Width;

    int rndTop = rnd.Next(500, 750);
    int fark = pcbPipeBottom1.Top - rndTop;
    pcbPipeBottom1.Top = rndTop;
    pcbPipeTop1.Top -= fark;
}
```

- Kaydırılan pipe nesneslerinden ikinci pipe nesnesinin birinci pipe nesnesinin bulunduğu pozisyon'a göre biraz daha geride ve her yenilenme işleminde değişik uzaklıklarda ekranда görüntülenmesi sağlanır.

```
if (pcbPipeBottom2.Left < -pcbPipeBottom2.Width)
{
    int rndDistance = rnd.Next(450, 590);
    pcbPipeBottom2.Left = pcbPipeBottom1.Left + rndDistance;
    pcbPipeTop2.Left = pcbPipeBottom1.Left + rndDistance;

    int rndTop = rnd.Next(500, 750);
    int fark = pcbPipeBottom2.Top - rndTop;
    pcbPipeBottom2.Top = rndTop;
    pcbPipeTop2.Top -= fark;
}
```

- Her kaydırma işleminde pipe nesnelerinin yükseklikleri değişiklik göstermelidir. Bu şekilde oyundaki engellerin değişimi sağlanarak oyuna akıcılık kazandırılmış olur.
 - pipe nesnesi yüksekliği

```
int rndTop = rnd.Next(400, 750);
int fark = pcbPipeBottom1.Top - rndTop;
pcbPipeBottom1.Top = rndTop;
pcbPipeTop1.Top -= fark;
```

- *pipe nesnesi yüksekliği*

```
int rndTop = rnd.Next(400, 750);
int fark = pcbPipeBottom2.Top - rndTop;
pcbPipeBottom2.Top = rndTop;
pcbPipeTop2.Top -= fark;
```