

— Methodlara dağıtım bakış

- parametre almaları
- geri dönüş değerleri
- erişim yetkileri
- parametrelerini genelleştirme kapsamlarını arttırmak

— Özel Listeler (Enum)

- ↳ Hedef kullanılır
- ↳ teknik olarak tanımlama

— Yeni türler ve objeleri ile çalışmak

- ↳ Rect
- ↳ IntersectWith } Diğer türleri nasıl kullanabiliriz?

— "Görünür" "Property" lar ile çalışmak

- Kaynak kodun okunurluğunu arttırmak/organize etmek
- ↳ Region kullanmak.

— Hiyerarka dışındaki resimleri göstermek

- ↳ Resources.resx

— Staffa Bilinçli: Bakmaya Çalıştık.



1.TUR

```
private void Hareket()
{
    AsaIdus();
    OyuncuyaCarmaKontrolu();
    UssuricikCarmaKontrolu();
}

private void OyuncuyaCarmaKontrolu()
{
    Rect rectPara = SinirlariniBul(this);
    Rect rectOyuncu = SinirlariniBul(mainWindow.oyuncu);
    if (rectPara.IntersectsWith(rectOyuncu))
    {
        this.Dispose();
    }
}

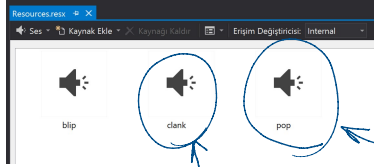
private Rect SinirlariniBul(FrameworkElement element)
{
    return new Rect(Canvas.GetLeft(element), Canvas.GetTop(element), element.ActualWidth, element.ActualHeight);
}
```

2.TUR

```
public enum OlmeTurleri //Olme türleri
{
    YakalanarakOldu, EceliyleOldu
}

private OlmeTurleri mefta;
public OlmeTurleri Mefta

private void OyuncuyaCarmaKontrolu()
{
    Rect rectPara = SinirlariniBul(this);
}
```



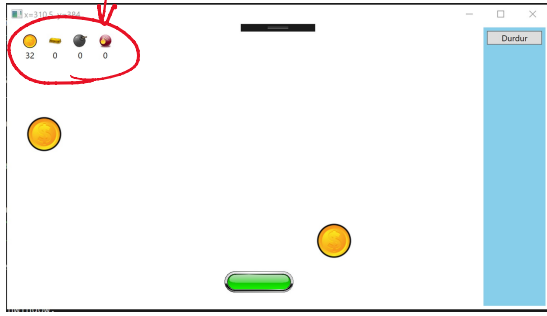
```
private OlmeTurleri mefta;
public OlmeTurleri Mefta
{
    get { return mefta; }
    set
    {
        mefta = value;
        switch (mefta)
        {
            case OlmeTurleri.YakalanarakOldu:
                new SoundPlayer(Properties.Resources.pop).Play();
                break;
            case OlmeTurleri.EceliyleOldu:
                new SoundPlayer(Properties.Resources.clank).Play();
                break;
            default:
                break;
        }
        this.Dispose();
    }
}
```

```
private void OyuncuyaCarmaKontrolu()
{
    Rect rectPara = SinirlariniBul(this);
    Rect rectOyuncu = SinirlariniBul(mainWindow.oyuncu);
    if (rectPara.Intersects(rectOyuncu))
    {
        //this.Dispose(); //yakalanarak öldü
        this.Mefta = OlmeTurleri.YakalanarakOldu;
    }
}
```

```
private void DisariCikmaKontrolu()
{
    //para oyun alanının dışına çıktıysa
    if (Canvas.GetTop(this) > mainWindow.cOyunAlani.ActualHeight)
    {
        //this.Dispose(); //eceliyle öldü
        this.Mefta = OlmeTurleri.EceliyleOldu;
    }
}
```

3.TUR

```
<UniformGrid Canvas.Left="10" Canvas.Top="10" Width="146" Rows="2" >
<Image Source="Resimler/para.png" Height="20"/>
<Image Source="Resimler/kulce.png" Height="20"/>
<Image Source="Resimler/siyahBomba.png" Height="20"/>
<Image Source="Resimler/kirmiziBomba.png" Height="20"/>
<TextBlock x:Name="tbPara" Text="0" HorizontalAlignment="Center" VerticalAlignment="Center"/>
<TextBlock x:Name="tbKulce" Text="0" HorizontalAlignment="Center" VerticalAlignment="Center"/>
<TextBlock x:Name="tbSiyahBomba" Text="0" HorizontalAlignment="Center" VerticalAlignment="Center"/>
<TextBlock x:Name="tbKirmiziBomba" Text="0" HorizontalAlignment="Center" VerticalAlignment="Center"/>
</UniformGrid>
```



```
private OlmeTurleri mefta;
public OlmeTurleri Mefta
{
    get { return mefta; }
    set
    {
        mefta = value;
        switch (mefta)
        {
            case OlmeTurleri.YakalanarakOldu:
                new SoundPlayer(YakalanarakOlmeSesi).Play();
                YakalananSayisiYaz();
                break;
            case OlmeTurleri.EceliyleOldu:
                new SoundPlayer(EceliyleOlmeSesi).Play();
                break;
            default:
                break;
        }
        this.Dispose();
    }
}
```

```
private void YakalananSayisiYaz()
{
    YakalananSayisi += this.Degeri;
    mainWindow.tbPara.Text = YakalananSayisi.ToString();
}
```

```

1  Owing
2
3  namespace Parakap
4  {
5      class Para : Image, IDisposable
6      {
7          public enum OlmeTurleri
8          {
9              YakalanarakOldu, EceliyleOldu
10         }
11
12         private OlmeTurleri mefta;
13
14         public OlmeTurleri Mefta
15         {
16             get { return mefta; }
17             set
18             {
19                 mefta = value;
20                 switch (mefta)
21                 {
22                     case OlmeTurleri.YakalanarakOldu:
23                         new SoundPlayer(Properties.Resources.pop).Play();
24                         YakalandiSayisiHizmet();
25                         break;
26                     case OlmeTurleri.EceliyleOldu:
27                         new SoundPlayer(Properties.Resources.clank).Play();
28                         default:
29                             break;
30                 }
31                 this.Dispose();
32             }
33         }
34
35         private void YakalandiSayisiHizmet()
36         {
37             YakalananiSayisi += Degeri;
38             mainWindow.tPara.Text = YakalananiSayisi.ToString();
39         }
40
41         [Serializable]
42         public Para(MainWindow mainWindow) //Yapici method, new Para()
43         {
44             Nesneye Has Ozellikler Belirleniyor
45             this.Source = new BitmapImage(new Uri("Resimler/para.png", UriKind.Relative));
46             this.Degeri = 1;
47             this.DusmeHizli = 5;
48             this.YakalanarakOlduSesi = Properties.Resources.pop;
49             this.EceliyleOlduSesi = Properties.Resources.clank;
50         }
51     }
52 }

```

Programın çalışmasına bir bir etkisi yok. Okunuyor etkiler.

```

#region Genel Değişkenler
System.Timers.Timer timer = new System.Timers.Timer(10);
//DispatcherTimer timer = new DispatcherTimer();
static Random random = new Random(); //??? ++++
private readonly MainWindow mainWindow;
#endregion

```

```

#region Özellikler
static public int YakalananiSayisi { get; set; }
public int Degeri { get; set; }
public int DusmeHizli { get; set; }
public UnmanagedMemoryStream YakalanarakOlduSesi { get; set; }
public UnmanagedMemoryStream EceliyleOlduSesi { get; set; }
#endregion

```

```

#region Nesneye Has Özellikler Belirleniyor
this.Source = new BitmapImage(new Uri("Resimler/para.png", UriKind.Relative));
this.Degeri = 1;
this.DusmeHizli = 5;
this.YakalanarakOlduSesi = Properties.Resources.pop;
this.EceliyleOlduSesi = Properties.Resources.clank;
#endregion

```

⚠ Diğer nesnelere (Deperi Para, Kulce, Bomba...) bu deperiler depericecek

```

#region Konumu Ayarlanıyor
this.Height = 50;
this.Width = 50;
Canvas.SetTop(this, 0);
Canvas.SetLeft(this, random.Next((int)(mainwindow.coyunAlani.ActualWidth - this.Width)));
#endregion

```

```

#region Timer Yapılandırılıyor
//timer.Tick += Timer_Tick;
//timer.Interval = TimeSpan.FromSeconds(0.01);
//timer.Start();
timer.Elapsed += Timer_Elapsed;
timer.Start();
#endregion

```

