****

**ATATÜRK ÜNİVERSİTESİ**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**Görsel Programlama-Güz**

**Yılan Oyunu Raporu**

Dr. Öğr. Üyesi Mete YAĞANOĞLU

Rüveyda Havva Sevinç

170707065

**Yılan Oyunu Mantığı ve Algoritmasını Nasıl Oluşturdum**

*Yılanın belirli bir başlangıcı olmak zorundadır, yönlendirmenin hangi uçtan yapılacağı ve yenilen yemden sonra boyutun büyümesi için ekleme yapılacak ucun ayırt edilmesi için. Yılanın başlangıç noktası ve yemin yerleştirileceği konum rastgele yerleştirilmelidir. Yılanın her parçasını pictureBox kullanarakta yapabileceğim gibi ben grafik ile çizdirmeyi tercih ettim. Yerleştirilen konumların aynı olması ihtimaline karşı konumlar kontrol edilmeli ve eğer aynı konumda bulunurlarsa yem rastgele olarak tekrar yerleştirilmelidir. Sonrasında kullanıcının yılanı hareket ettirebilmesi gereklidir. Bunu <<bool değişkeni>> veya <<enum sabiti>> ile yapabilmek mümkündür. Aynı şekilde yılanı hareket ettirmek için TİMER kullanılmalıdır.Timer ile yılanın hareket hızını değiştirmek mümkündür. Yılan hareket ettirildikçe yemi yiyip yemediği kontrol edilmelidir. Bunu << location.X / location.Y >> ile konumlarını kontrol edip veya rectangle kullanarak <<IntersectsWith>> ile yapabiliriz. Eğer konumlar aynı ise aynı boyutlarda oluşturduğumuz şekli kuyruğa ekliyoruz. Oyun sürekli bu şekilde dönerken oyunun bitme şartları ile oyunu kontrol ediyoruz. Yılan sınırlara değerse oyun bitirilmelidir. Bu yaptığım projede kullanıcının tercihine göre oyun kaybedildikten sonra tekrar oynayıp oynamak istemediğini soruyorum. Bunu MessageBoxButton yani butonlu mesaj kutusu ile yapılabilir. Seçime göre oyun sonlandırılır ya da tekrar edilir.*

*Projede belirtilen şartlara göre;*

*Yeni oyun başlangıcı seçildiyse kullanıcı oyuna başlarken textBox’a yazdığı isim ile tekrar kaydedilip oyun başlamalıdır. Bunu şu şekilde yapabiliriz, kullanıcının girdiği bütün textBox.text’leri bir dizi veya liste de tutup son elemanını geri döndürerek.*

*Aynı şekilde aynı hızda oyunun başlatılması için seçilen hızları değiştirip formlar arası kullanabileceğimiz bir değişkende tutup veri aktarımı yaparak.*

***Form 1’in algoritmasını nasıl oluşturdum?***

*1.Adım ) Başla*

*2.Adım ) Eğer ClickButton1 ise MessageBox a yazdır*

*3.Adım) Eğer ClickButton2 ise TextBox ın Oku ve Kaydet*

*4.Adım) Eğer B tuşuna basılırsa TextBox’ ı kontrol et*

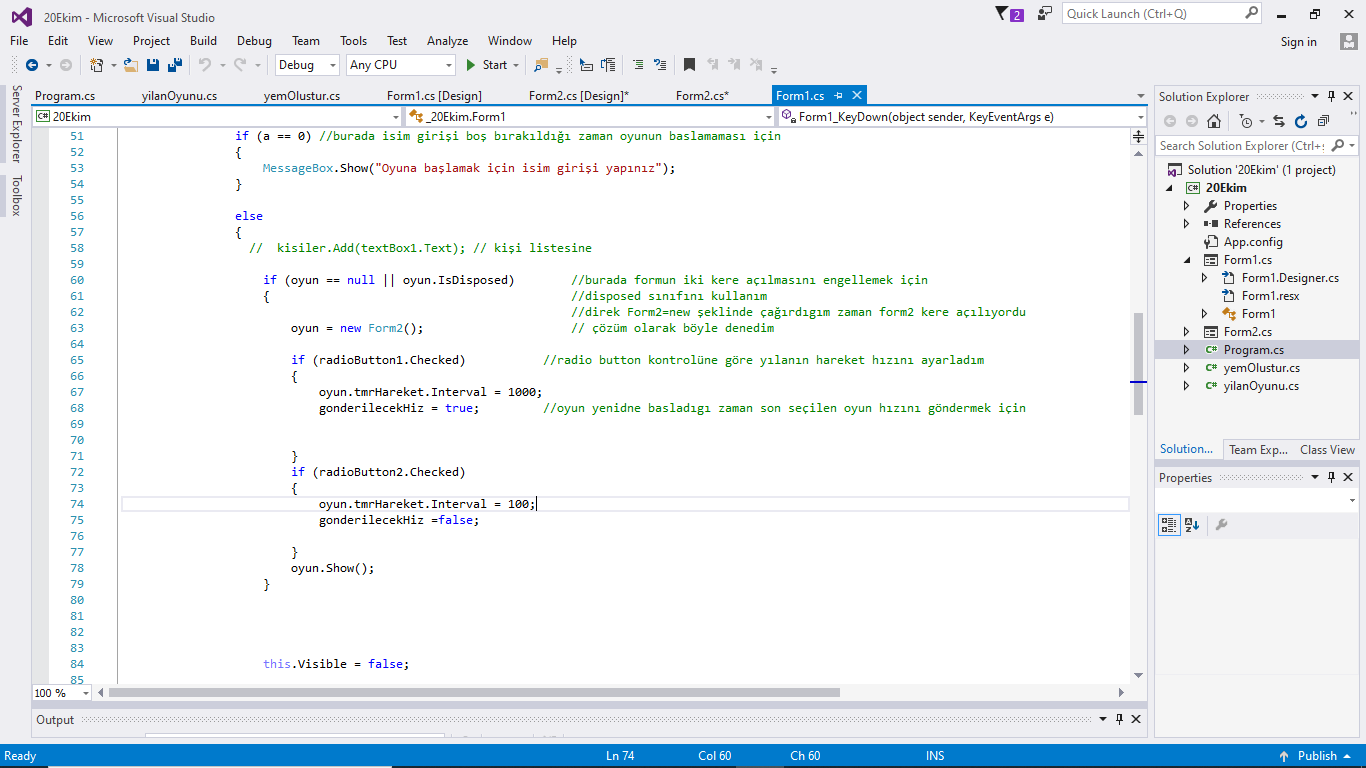
*5.Adım) Eğer TextBox’ta veri girişi olduysa Form2 yi aç*

***Grafik ile karenin parçalarını nasıl oluşturdum ve birleştirdim?***

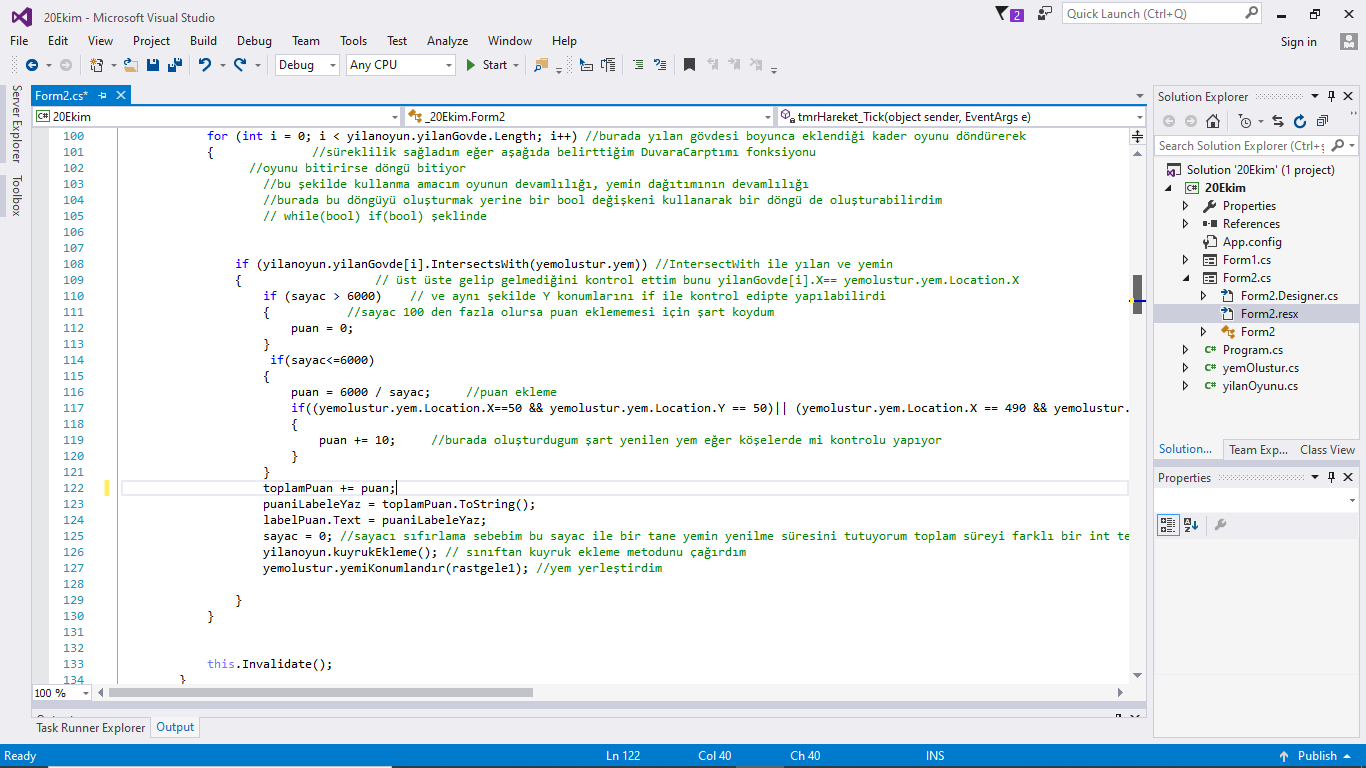
*Rectangle [] nesnesi oluşturup yılanı başlangıcı olarak belirledim. Random olarak belirlediğim konumları ve belirttiğim boyutlarda Drawing sınıfını isimalanına ekleyerek yılanın ilk başlangıcını oluşturdum. Sonrasında Rectange nesnesinin listesini oluşturup yemin eklendiği kareleri gövdesi olarak belirttim. Yemin yenilme şartına göre listeye ekleme yaparak yılanın büyümesini sağladım. Aynı şekilde Rectangle[] nesnesi ile bir kare oluşturup yem olarak tanımladım oyun devam ettikçe random konumlarla panelin içinde çizdirilmesi için yemOlusturma ve yilanOlusturma classlarını Form2’ye ekleyerek PanelPaint’te kullandım.*

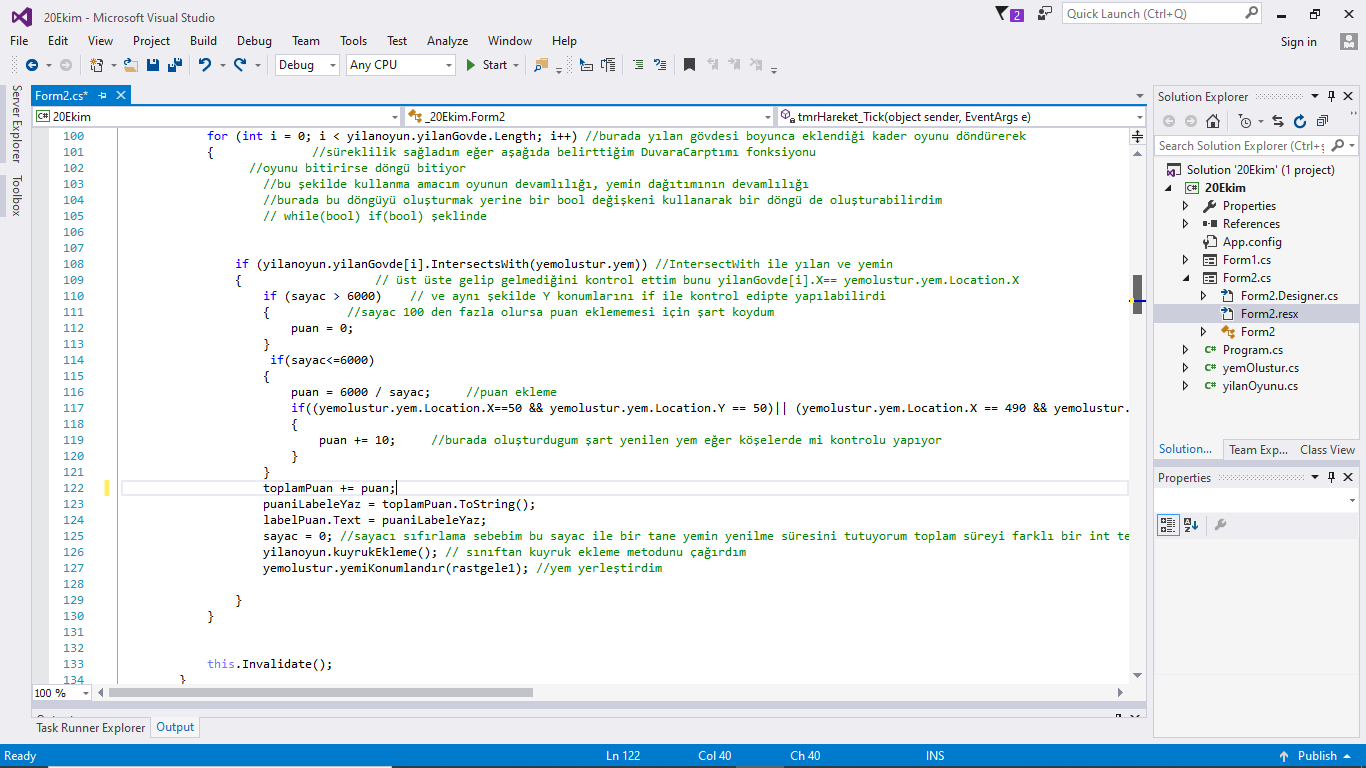
***Projede olması gereken şartları nasıl sağladım? Kod örnekleri ve Ekran çıktıları***

* **2 Farklı seviye de hız ayarlama**: *RadioButtonlar ile kontrolünü sağlayarak kullanıcıya öncelikle oyunun zorluk seviyesini belirlettim. Sonrasında <<if>> yapısı kullanarak RadioButtonların işaretli olup olmadığını kontrol ettim. Seçilen butona göre kodda <<tmrHareket>> olarak adlandırdığım Timer’ın Interval değerlerini değiştirerek iki seçenek için farklı değer atadım. Timer’ın değerini değiştirmek için Özellik kısmındaki Enabled=false; olmalıdır.[Kodumun oyun.tmrHareket.Interval şeklinde olmasının sebebi Timer’ın farklı bir formda olması ve o formdaki Timer’ı kullanıyor olmam]*

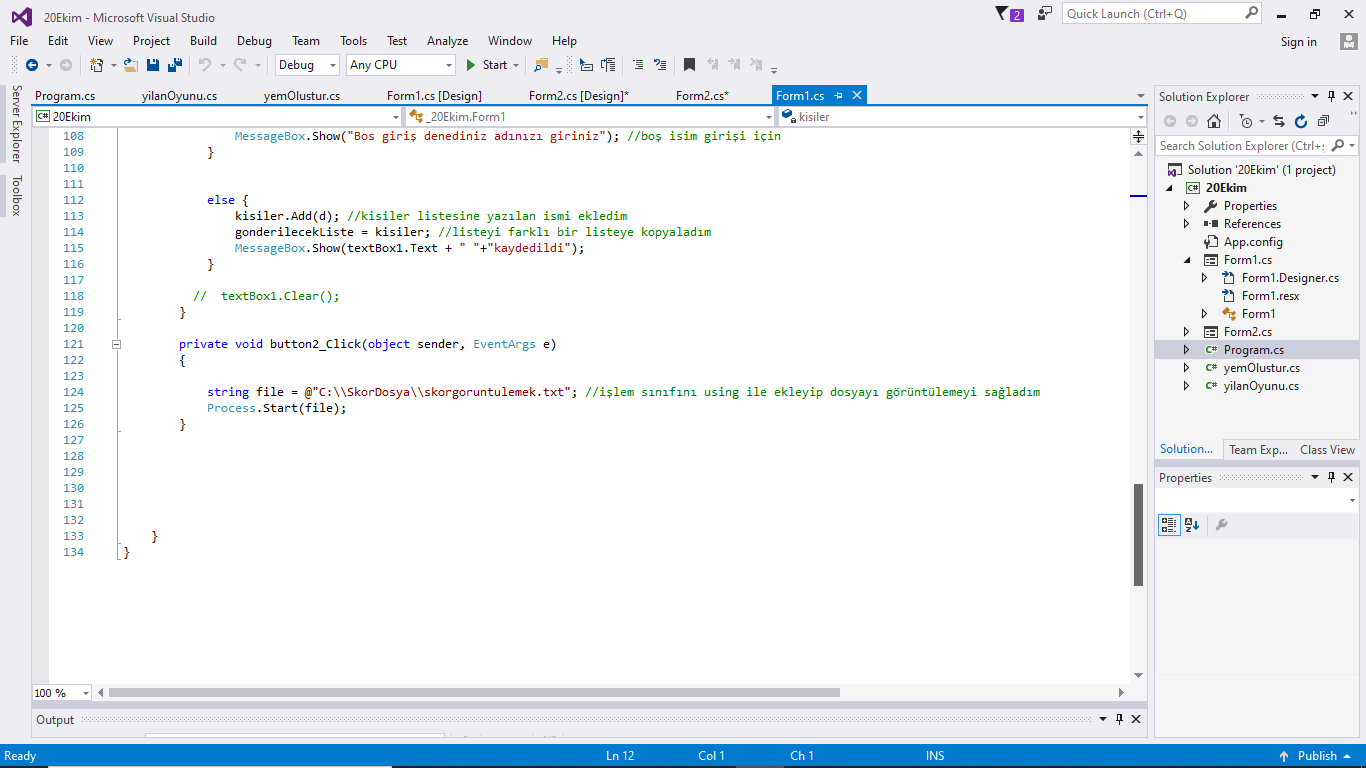
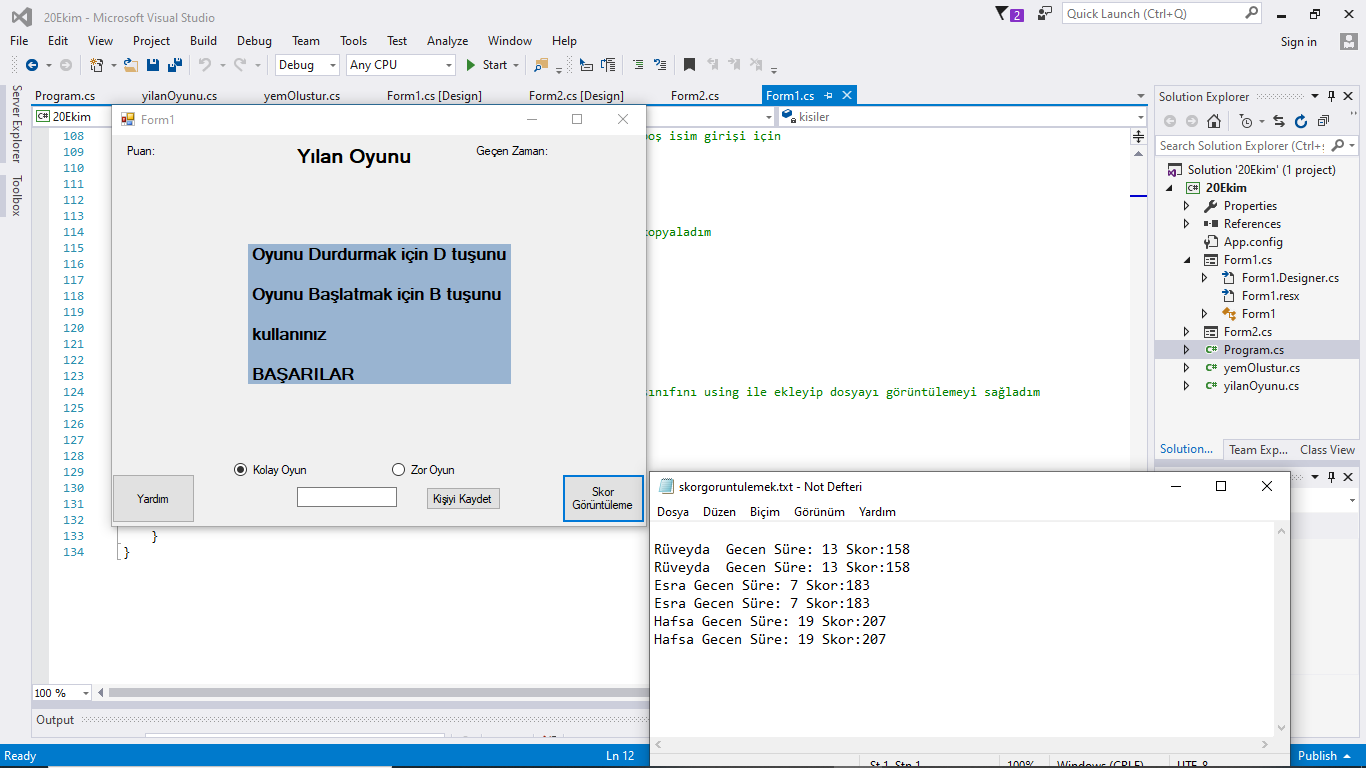


* **Puanlama yapılması ve köşe noktalara geldiği zaman ekstra puan verilmesi:** *Timer kullanarak program çalıştığı sürece oyunun durdurulması şartı hariç zamanı hesapladım. Süre hesaplamayı saniye olarak arttırdım. Şart koyarak 60 salise=1 sn, 60sn=1 dk olarak şart koyarak arttırdığım salise arttıkça, labele yazdırdım. Bir tane sayacı her yem yenildiği sürece sıfırlayarak puan hesaplamasında kullandım.Timer’ın eventlerinde Tick fonksiyonu altında sayaçları arttırma işlemlerini sağladım. İki farklı değişkende sayacı arttırdım. Puan için kullandığım sayaca öncelikle doğru puanlama yapabilmek için <<if>>şartı ile sayacın 6000’den büyük olup olmadığını kontrol ettim.1 saniye=60 ise 100sn =6000 olduğu için ben saniye değil, salise üzerinden hesaplattım. Sayaçın 6000’den büyük olduğu durumlarda puan değişkenine sıfır değerini atadım. Küçük olduğu blokta ise 6000/sayac hesaplaması ile değer atadım. Döngü devam ettikçe bütün puanları topladım. Ve ekstra puan için sayacın 6000’den küçük olduğunu kontrol eden <<if>> bloğu içerisine yemin konumunu kontrol edip panelin köşelerinde olup olmadığını kontrol ettim. Eğer şart sağlanırsa 10 puan ekleme işlemini yaptım. Değişkenleri float olarak atadım.Floatı label’e yazdırmaya çalışırsam hata alacağım için floatı string veri türüne çevip labele yazdırdım Ve bunu stringe dönüştürürken şu kod satılarını kullanarak yaptım.*





* **Skor görüntüleme kısmında kişi ve skorun tutulması:** *Kişinin textBox’a yazdırılmasını form1’de gerçekleştirdim. Öncelikle* ***textBox’ın farklı bir formda kullanılması için özelliklerinden Modifiers özelliği <<public>> olmalıdır.*** *Yoksa erişilebilirlik özelliğini kullanamayız. Kişi,skor ve puan’ın dosyaya yazdırılma işlemini Form2’ de gerçekleştirdim. Form1’de girilen textBox’u statick olarak tanımladığım bir değişkende tutarak, Form2’den bu veriye erişerek Form2’de IO sınıfını ekleyerek Dosya yazdırma işlemlerini gerçekleştirdim. Sonrasında görüntüle kısmını oluşturmak için Diagnostics sınıfını isim alanına ekleyerek Procces metodundan yararlanarak açmak istediğim dosyayı görüntülettim.*



* **B ve D tuşları ile oyunun durdurulması ve başlatılması:** *Key eventleri ile yapılabilir. Aslında bu eventlerin kullanımı örnek projede kullandığımdan farklıdır. KeyEventsArgs parametresi bize hangi tuşa basıldığını gösterir. Ben proje de kodlarımı KeyDown eventi içerisinde yazdım. <<e.KeyCode>> ile hangi tuşa basıldığını belirttim. Ve bunları şart blokları içerisinde kodladım.*
* **Oyun yeniden başlatıldığı zaman kullanıcının aynı isimler oyuna başlaması ve tekrar skorlarının kaydedilmesi:** *Oyunun bitimini kontrol eden fonksiyon içerisine MessageBoxButton ile YesNo butonu yerleştirdim. Seçilen butonu kontol ederek oyunun devam edilip edilmediğini kontrol ettim. Oyunun yeniden başlatılması seçeneğinde Form1’den veri aktarımı ile değerlerini kullanıp ve aynı şekilde kullandığım verilerin son değerlerine yani Listedeki son kullanıcı ve seçilen oyun zorluğunu fonksiyonda değişkenlere ekledim. Sonrasında bu hız değerini Form2yi tekrardan çağırırken Interval değerini şarta göre ekledim. Yeni oyun başlarken Form1’den çektiğim veriyi yeni oyunun textBox’ı olarak ekledim bu şekilde son kişiyi tutup tekrar yazdırabildim.*

**Oyunu tek formda nasıl yapabilirdim ? Neden yapmadım?**

*Form1’i oluştururken içinde olan labeli oyun başlatırken visible özelliğini false olarak değiştiren bir şart koyarak panel içini boşaltıp sonrasında PanelPaint eventi kullanarak grafikleri (şekilleri) çizdirme işlemi yaptırabilirdim. Aynı şekilde buttonları da false yapıp oyunun bittiğini belirten blok içerisinde tekrar visible özelliğini true olarak değiştirip tekrar görünür olmasını sağlayabilirdim. Bu şekilde yaparken kodlarımda karışıklık yaşayıp yapamama ihtimalimin artacağını düşündüm. Bu yüzden iki form kullanıp formlarda. show ve .close metodlarını kullandım. Ve hatta form2’de de kodlarımda karışıklık yaşamamak için class(sınıf) kullanıp form içinde classlardan nesneler üreterek kullandım. Bu şekilde yazılabilirliğini, okunabilirliğini zorlaştırmış ve kod satır sayısını arttırdığımı farkındayım fakat yapmama sebebim okunabilirliğinden ziyade çalışır bir program teslim etmekti.*