**Hangman**

Bu çalışmada Visual Studio’yu kullanarak C# dilinde bir Console Application ile Adam Asmaca oyununu geliştiriyor olacağız.

**Adım 1**

“Utility” adında bir static class tanımlanmalı.

static class Utility

Fields

* static string[] Renders
  + Burada arayüzdeki her frame Array’in bir elemanı olarak tanımlanmalı.

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ║ " + '\n' +

@" ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 1

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 2

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 3

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" |\ ║ " + '\n' +

@" ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 4

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 5

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" \ ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

// 6

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

* static string[] DeathAnimation
  + Burada oyun kaybedilince oluşacak animasyonun frameleri array olarak tanımlanmalı.

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ███ ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o> ║ " + '\n' +

@" /| ║ " + '\n' +

@" >\ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" <o ║ " + '\n' +

@" |\ ║ " + '\n' +

@" /< ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o> ║ " + '\n' +

@" /| ║ " + '\n' +

@" >\ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o> ║ " + '\n' +

@" /| ║ " + '\n' +

@" >\ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" <o ║ " + '\n' +

@" |\ ║ " + '\n' +

@" /< ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" <o ║ " + '\n' +

@" |\ ║ " + '\n' +

@" /< ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" <o ║ " + '\n' +

@" |\ ║ " + '\n' +

@" /< ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" /|\ ║ " + '\n' +

@" / \ ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" o ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" | ║ " + '\n' +

@" | ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" / ║ " + '\n' +

@" \ ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ' ║ " + '\n' +

@" ║ " + '\n' +

@" |\_\_ ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" \\_\_ ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_\_\_\_ ║ " + '\n' +

@" ══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ' ║ " + '\n' +

@" . ║ " + '\n' +

@" \_\_ ║ " + '\n' +

@" /══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" \_ ' ║ " + '\n' +

@" \_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_ ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ' ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_ ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ' ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_ ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" . ║ " + '\n' +

@" ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ║ " + '\n' +

@" ' ║ " + '\n' +

@" \_\_/══════╩═══",

//

@" ╔═══╗ " + '\n' +

@" | ║ " + '\n' +

@" O ║ " + '\n' +

@" ║ " + '\n' +

@" ║ " + '\n' +

@" \_ ║ " + '\n' +

@" \_\_/══════╩═══",

Methods

* static T ChooseRandom<T>(T[] values)
  + Bu metotta gönderilen array içinden rastgele bir item seçilebilmeli.

**İpucu: Rastgele sayı belirleme**

Random random = new Random();

random.Next(10);

**Adım 2**

“IGameService” adında bir interface tanımlanmalı.

interface IGameService

Methods

* void Results ()

**Adım 3**

“HangmanService” adında bir class tanımlanmalı ve “IGame” interface’inden türetilmeli.

class HangmanService : IGameService

Fields

* string RandomWord
* char[] RandomWordChars
* int IncorrectGuesses

Constructors

* HangmanService()
  + RandomWord değişkeni rastgele bir kelime ile doldurulmalı.
  + RandomWordChars değişkeni RandomWord değişkeninin karakter sayısı kadar “\_” ile doldurulmalı. Örnek: RandomWord “mehmet” ise “\_ \_ \_ \_ \_ \_” olmalı.
  + IncorrectGuesses değişkeni “0” olarak atanmalı.

Methods

* void GetRandomWord()
  + Bu metotta Utility classındaki ChooseRandom metotu ile belirlenen kelime Array’inden rastgele bir eleman seçilmeli.
* void Render(string @string)
  + Bu metot ile adam asmaca şekillerini console a yazdıracağız. Utility classındaki Renders ve DeathAnimation Arraylerinin itemlarını bu metotun içine vereceğiz ve bu metot aynı şekilde console a veridğimi şekli yazdıracak.

**İpucu:** Console’daki cursor pozisyonu aşağıdaki gibi belirlenir.

Console.SetCursorPosition(x, y++)

* void RenderGameState()
  + Bu metot tahmin ettiğimiz her harf için çalışacak ve harfi yanlış veya doğru girme durumumuza göre oyunun görselini bir sonraki aşamaya getirecek.
  + Ayrıca cursor ekranda görünmeyecek şekilde ayarlama yapılıp, cursor positionı da ayarlanmalı.
* bool CheckContinue()
  + Bu metotta hatalı girdiğimiz harf sayısı kelimenin sayısından büyük mü ya da kelimeyi tamamlamış mıyız kontrolü yapılarak oyuna devam edilip edilmeyeceğinin kararı verilecek.
* void Results()
  + Bu metotta oyun bittiği andaki durum ekrana yazdırılacak. Eğer oyunu kaybettiysek DeathAnimation Array’inin itemları ekrana sırasıyla yansıtılacak.

**İpucu :** Animasyon işlemi için ekranı bir süre dondurmak gerekli. Bu işlem için aşağıdaki kod kullanılabilir.

Thread.Sleep(150);

**Adım 7**

“Program.cs” altındaki “Main” metodunda programı geliştirmeye bu adımda başlanacak.

* Oyun tamamlandığında kullanıcıdan sürekli oyuna devam edilip edilmeyeceği bilgisi istenecek.
* Her oyun başladığında HangmanService classından bir obje yaratılacak.
* CheckContinue metotu ile oyunun devam edip etmediği kontrol edilecek ve oyun devam ettiği sürece kullanıcıdan harf alınacak.

**İpucu :** ConsoleKey key = Console.ReadKey(true).Key;

* Kullanıcıdan alınan değer harf değilse işlem yapılmamalı tekrar harf alınmaya devam etmeli.

**İpucu :** if (key >= ConsoleKey.A && key <= ConsoleKey.Z)

Girilen harf alınırken char guess = (char)(key - ConsoleKey.A + 'a'); şeklinde alınmalı.

* Oyun sonuçlanana kadar harf alınmaya devam edilmeli, oyun sonuçlanınca Results metotu ile sonuç ekrana yazdırılmalı.
* Sonuç yazdırıldıktan sonra tekrar oyuna devam edilip edilmeyeceği sorulmalı.