

Marslı Gezgin Projesi

Projenin Özeti:

Robotik bir gezgin, Mars gezegeninin yüzeyine NASA tarafından konumlandırılmıştır. Dikdörtgen şeklinde olan bu yüzey, robotik gezginin üzerindeki kamera yardımıyla görüntülenmekte ve bu görüntüleri robotik gezgin dünyaya yollamaktadır.

Robotik gezginin pozisyonu ve konumu, x-y koordinatları ve pusulada yer alan yön harfleri ile belirtilmektedir (N,W,S,E). Mars yüzeyindeki bu yüzey, x ve y koordinatlarına göre bölünmüştür.

Robotik gezgin için 0,0,N pozisyonu, dikdörtgen yüzeyin sol alt köşesinde ve yönünün kuzeye dönük olduğunu göstermektedir. NASA robotik gezgini kontrol etmek için bir takım harf katarları göndermektedir. Gönderilebilen harfler ise 'L', 'R' ve 'M' dir. 'L' ve 'R' komutu, robotik gezgini ve kamerasını hiç hareket ettirmeden olduğu yerde 90 derece sola ya da sağa döndürmektedir. 'M' ise robotik gezgini bulunduğu yönde bir adım ilerletmektedir.

Robotik gezgin, (x,y) koordinatından (x,y+1) koordinatına konumlandığında kuzey (N) yönünde 1 adım gittiği kabul edilmektedir.

Kullanılan Teknolojiler:

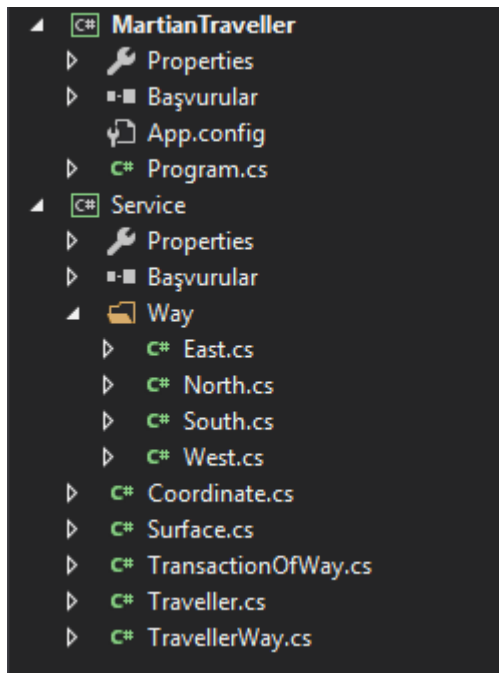
.Net

C#

Analiz:

Projede en zor kısım girilen değerlerin istenen sınırlarda olup olmadığının kontrolü ile ilerledikten sonra gerçek yerinin bulunması için kullanılan fonksiyonların algoritmaları oldu.

Projede 2 katmandan oluşan bir yapı kullanıldı. Böylece yapılmak istenen işlemleri bir serviste toplanarak program.cs içerisinde organik bir bağ kurulması ve spaghetti kod engellendi.



Program.cs çıktının alınacağı arayüz olarak kullanıldı. Coordinate sınıfı x ve y değerleri alan girilen koordinatların tutulduğu sınıf, Surface yeryüzü değerlerinin tutulduğu ve bu değerlerin tanımlanan değerler içinde olup olmadığının kontrol edildiği sınıftır.

```
namespace MartianTraveller
{
    class Program
    {
        static void Main(string[] args)
        {
            Coordinate minCoordinate = new Coordinate(0, 0);
            Coordinate maxCoordinate = new Coordinate(5, 4);

            Surface surface = new Surface(maxCoordinate, minCoordinate);
            Traveller traveller = new Traveller(new Coordinate(1, 2), "N", surface);
            traveller.MovementDirective("LMLMLMLMM");

            string sonuc = traveller.GetTravellerLocation();
            Console.WriteLine(sonuc);
            Console.ReadLine();

            Traveller traveller2 = new Traveller(new Coordinate(3, 3), "E", surface);
            traveller2.MovementDirective("MMRMMRMRRM");
            string sonuc2 = traveller2.GetTravellerLocation();

            Console.WriteLine(sonuc2);
            Console.ReadLine();
        }
    }
}
```

Traveller sınıfı gezginin tüm hareketlerinin kontrolünün yapıldığı fonksiyonlarının olduğu sınıftır.

```
private Coordinate _coordinate;
private string _compassdirection;
private Surface _surface;
private TravellerWay _travellerWay;

public Traveller(Coordinate coordinate, string compassdirection, Surface surface)
{
    _coordinate = coordinate;
    _compassdirection = compassdirection;
    _surface = surface;
    _travellerWay = TransactionOfWay.CreateTravellerDirection(compassdirection, this);
}

public Coordinate GetCoordinates()
{
    return _coordinate;
}

public void CoordinateSet(Coordinate coordinate)
{
    bool IsReal = _surface.IsAtPlane(coordinate);
    if (IsReal)
    {
        _coordinate = coordinate;
    }
    else
    {
        Console.WriteLine("Dikdörtgen yüzeyin bulunduğu x veya y koordinatından birini aşılıyor");
        Console.ReadLine();
    }
}

public void TravellerDirectionSet(TravellerWay travellerWay)
{
    this._travellerWay = travellerWay;
}

public string GetTravellerLocation()
{
    return "Gezginin gerçek konumu -> x : " + _coordinate.X + ", y : " + _coordinate.Y + ", Yön : " + _travellerWay.ToString();
}

public void MovementDirective(string talimat)
{
    for (int i = 0; i < talimat.Length; i++)
    {
        if (talimat[i] == 'L')
        {
            _travellerWay.TurnLeft();
        }
        if (talimat[i] == 'R')
        {
            _travellerWay.TurnRight();
        }
        if (talimat[i] == 'M')
        {
            _travellerWay.Move();
        }
    }
}
```

Son olarak yönlerin olduğu ve bu yönlerin kontrolleri yapıldığı gezginin hangi yönden hangi yöne ilerleyeceğinin belirlendiği fonksiyonlar da west, East, North ve South sınıflarında oluşturuldu.

```
public class South : TravellerWay
{
    public South(Traveller traveller) : base(traveller)
    {
    }

    public override void Move()
    {
        Coordinate coordinate = _traveller.GetCoordinates();
        coordinate = new Coordinate(coordinate.X, coordinate.Y-1);
        _traveller.CoordinateSet(coordinate);
    }

    public override string ToString()
    {
        return "S";
    }

    public override void TurnLeft()
    {
        _traveller.TravellerDirectionSet(new East(_traveller));
    }

    public override void TurnRight()
    {
        _traveller.TravellerDirectionSet(new West(_traveller));
    }
}
```