

DIATICS Vaka Dokümanı

Hayvanat Bahçesi Projesi

500'e 500'lük bir alanda yaşayan 30 koyun (15 erkek,15 dişi), 10 inek (5 erkek,5 dişi), 10 tavuk,10 kurt (5 dişi,5 erkek) 10 horoz, 8 aslan (4 erkek, 4 dişi) ve 1 avcı bulunmaktadır.

Hayvanlardan;

koyun 2 birim,
kurt 3 birim,
inek 2 birim,
tavuk 1 birim,
horoz 1 birim,
aslan 4 birim,

avcı 1 birim rasgele şekilde hareket etmektedir ancak alanın dışına çıkamamaktadır.

kurt kendisine 4 birim yakınındaki koyun, tavuk, horoz'u avlayabiliyor.

aslan kendisine 5 birim yakınlıktaki inek, koyun'u avlayabiliyor.

avcı da kendisine 8 birim yakınlıktaki hayvanlardan herhangi birisini avlayabiliyor.

aynı cins farklı cinsiyetteki hayvanlar birbirine 3 birim yakınlaştığı zaman random cinsiyetli ve aynı cins bir hayvan meydana gelmektedir.

1000 birim hareket sonunda hayvanların sayısının bulunduğu bir console application yazılması beklenmektedir.

Projenin Amacı

İsteğe uygun olarak hayvanat bahçesindeki hayvanları simüle eden bir yazılım yazılacak. Tekrardan isteklere göre bazı kurallar göz önünde bulundurulmuştur.

Problem Analizi

1. Her canlının kendine ait adım miktarı vardır ve birbirinden farklı olabilir.

2. **Tüm** canlılarda ortak bulunan özellikler vardır. (örn: hareket etme, üreme) ve birbirinden farklı oranda olabilir.
3. **Bazı** canlılarda ortak bulunan özellikler vardır (örn: avlanma) ve birbirinden farklı oranda olabilir.
4. Harita 500×500 istenilmektedir.
5. İstekler net bir şey istemektedir ve eğer istenildiği gibi yapılırsa yazılımın geliştirilmesi zorlaşacaktır. Bu nedenle genelleştirilecektir. (Clean architecture yapısı ile yazılacaktır.)

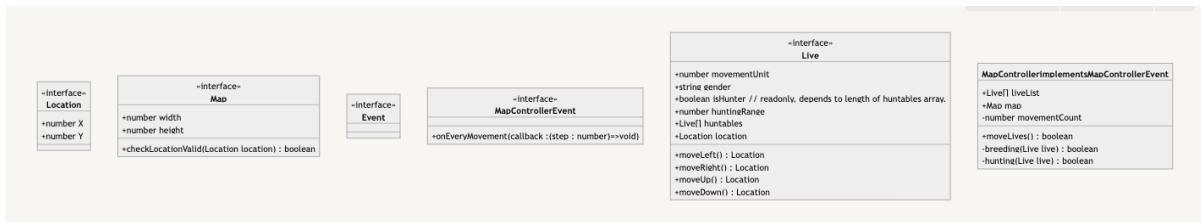
Genel Kurallar

1. Avlanabilen hayvanların belirli bir mesafesi vardır ve bu mesafeye giren canlılar avlanır ve sistemden çıkarılır.
2. Canlı her birimde farklı yönlere hareket edebilir.
3. Aynı cins canlılar belirli mesafede ürerler ve sisteme yeni bir canlı rastgele lokasyonda girer.

Varsayımlar

1. Bir canlı hareket ettikten sonra diğer tüm canlılar hareket etmeden tekrar harekete başlayamaz.
2. Hayvanlar her birimde istediği yönde hareket edebilir. (örn: kurt 3 birim hareket etme hakkı vardır ve yönleri birbirinden farklı olabilir: sağ, yukarı, sağ yada yukarı aşağı sol)

Yazılım Mimarisi Genel Görünüm



```

classDiagram
    class Location{
        <<interface>>
        +number X
        +number Y
    }
    class Map{
        <<interface>>
        +number width
        +number height
        +checkLocationValid(Location location) boolean
    }
    class Event{
        <<interface>>
    }
    class MapControllerEvent{
        <<interface>>
        +onEveryMovement(callback : (step : number)=>void)
    }

    class Live{
        <<interface>>
        +number movementUnit
        +string gender
        +boolean isHunter // readonly, depends to length of h
        +number huntingRange
        +Live[] hunttables
        +Location location
        +moveLeft() Location
        +moveRight() Location
        +moveUp() Location
        +moveDown() Location
    }

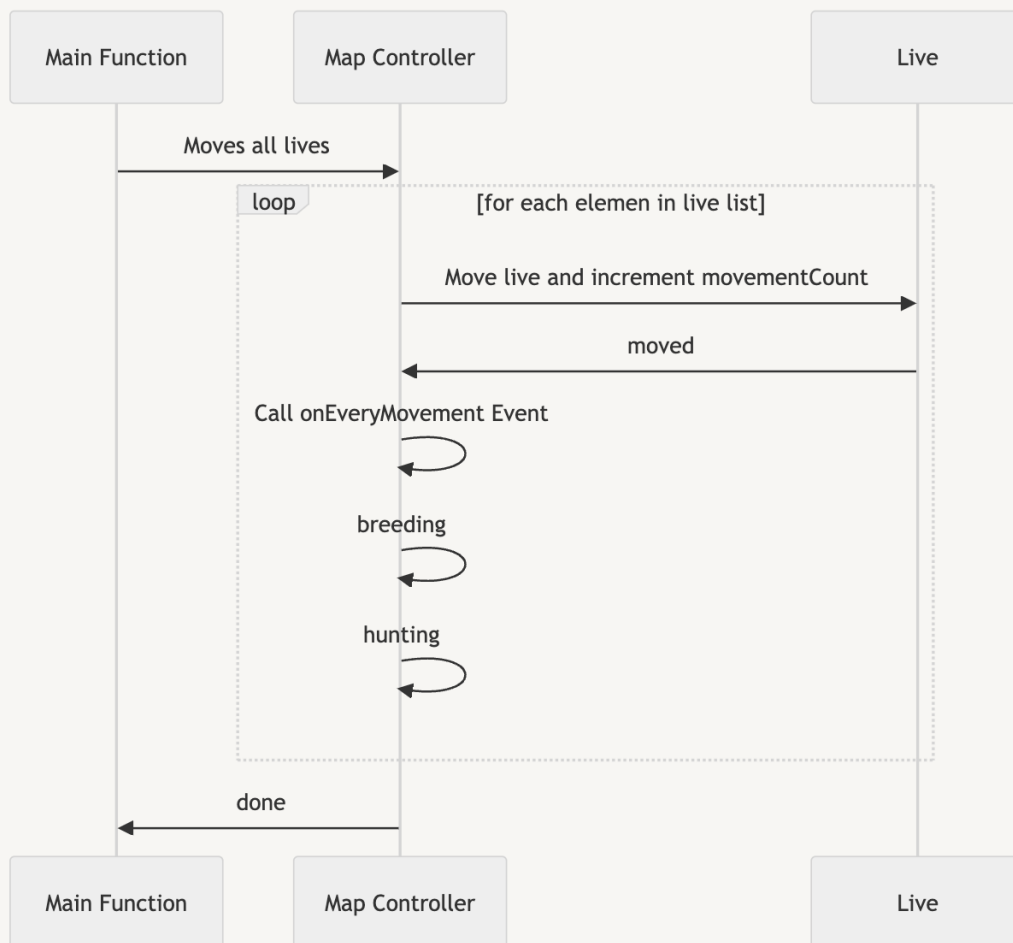
    class MapController implements MapControllerEvent{
        +Live[] liveList
    }

```

```

+Map map
  -number movementCount
  +moveLives() boolean
  -breeding(Live live) boolean
  -hunting(Live live) boolean
}

```



```

sequenceDiagram
    participant main as Main Function
    participant mc as Map Controller
    participant l as Live
    main ->> mc : Moves all lives
    loop for each elemen in live list

```

```
mc ->> l: Move live and increment movementCount
l ->> mc: moved
mc ->> mc: Call onEveryMovement Event
mc ->> mc: breeding
mc ->> mc: hunting
end
mc ->> main: done
```

Sürdürülebilirlik

Yazılım yazılırken geliştirilebilir şekilde clean architecture düşünce sistemiyle ve MVC dizaynı kullanılarak geliştirilmiştir. Böylelikle kolaylıkla istenilen boyutta yeni haritalar ve yeni canlılar sisteme dahil edilerek sistem daha da geliştirilebilir.