

## Hayvanat Bahçesi Simülasyonu Dökümantasyonu

### Proje Amacı

Bu simülasyonda farklı hayvan türleri ve bir avcı, 2D bir alanda rastgele hareket ediyor. Kurt ve aslanlar belli hayvanları avlıyor, yakın mesafedeki aynı tür hayvanlar üreyebiliyor. Simülasyon boyunca tüm olaylar loglanıyor ve hayatta kalanlar takip ediliyor.

### Kullanılan Teknolojiler

**Yazılım Dili :**Python

**Kütüphaneler :** Random, Math, Logging

**IDE:** VsCode

### Sınıf Yapısı

- Alive: Ortak özellikleri taşıyan base sınıf. (x, y, hareket, id vs.)
- Animal: Alive'dan türeyen hayvan sınıfı. Tür, cinsiyet, alive = True/False içeriyor.
- Hunter: Alive'dan türeyen avcı. hunt() fonksiyonuyla yakın hayvanları öldürüyor.

```

48 class Animal(Alive):
49     def __init__(self, animal_type, gender, move_unit):
50         super().__init__(move_unit)
51         self.animal_type = animal_type
52         self.gender = gender
53         self.alive = True
54
55 class Hunter(Animal):
56     def __init__(self, move_unit=1):
57         super().__init__(move_unit)
58         self.animal_type = "Avcı"
59     def hunt(self, animals):
60         for prey in animals:
61             if prey.alive and self.distance(pre) <= 8:
62                 prey.alive = False
63                 logger.info(f"Avcı {self.gender} {self.id}, {prey.animal_type} {prey.gender} {prey.id}'i avladı.")
64
65 class Alive:
66     def __init__(self, move_unit):
67         self.x = random.randint(0, WIDTH)
68         self.y = random.randint(0, HEIGHT)
69         self.move_unit = move_unit
70         self.id = id(self)
71     def move(self):
72         angle = random.uniform(0, 2 * math.pi)
73         dx = int(self.move_unit * math.cos(angle))
74         dy = int(self.move_unit * math.sin(angle))
75         old_x, old_y = self.x, self.y
76         self.x = min(max(self.x + dx, 0), WIDTH)
77         self.y = min(max(self.y + dy, 0), HEIGHT)
78         return old_x, old_y, self.x, self.y
79     def distance(self, other):
80         return math.hypot(self.x - other.x, self.y - other.y)

```

## Simülasyon Akışı

Her turda şu olaylar oluyor:

1. Tüm canlılar rastgele hareket eder.
2. Kurt ve Aslan kendi kurallarına göre av yapar.
3. Avcı av yapar.
4. Avlananlar loga yazılır
5. Koşullar uygunsa üreme olur
6. Üreyenler loga yazılır
7. Her tur sonunda hayatta kalanlar ve loga yazılır.

INFO | Kurt (Erkek) 2568570016016, Koyun (Dişi) 2568570240656'i avladı.

### Hayatta kalanlar:

```
Koyun (Erkek): 17 ,  
Koyun (Dişi): 18 ,  
İnek (Erkek): 5 ,  
İnek (Dişi): 5 ,  
Tavuk (Dişi): 5 ,  
Horoz (Erkek): 5 ,  
Kurt (Erkek): 5 ,  
Kurt (Dişi): 5 ,  
Aslan (Erkek): 4 ,  
Aslan (Dişi): 3
```

### Kodlama Detayları

- Kod tamamen nesne tabanlı yazıldı.
- random ile hareket ve doğumlar rastgele yapıldı.
- logging modülüyle her şey loga yazıldı.
- Hareket, av, üreme hepsi adım adım kontrol ediliyor.

### Çıktılar ve Sonuç

- Simülasyon sonunda hangi hayvandan kaç tane kaldığı yazdırılıyor.
- Log dosyasında tüm hareketler ve olaylar kayıtlı.
- Her adımda neler olduğu adım adım takip edilebiliyor.

```
PS C:\Users\ASUS\OneDrive\Masaüstü\DiatricVaka> & C:/Users/ASUS/AppData/Local/Microsoft/windowsApps/python3.11.exe c:/Users/ASUS/OneDrive/Masaüstü/DiatricVaka/hayvanat_bahcesi.py  
Koyun (Erkek): 12  
Koyun (Dişi): 10  
İnek (Erkek): 4  
İnek (Dişi): 3  
Tavuk (Dişi): 4  
Horoz (Erkek): 3  
Kurt (Erkek): 6  
Kurt (Dişi): 7  
Aslan (Erkek): 4
```