



**SAKARYA**  
ÜNİVERSİTESİ

# Canlılar Arası Savaş Simülasyonu

2. Ödev – C

2023-2024

Younes Rahebi | B221210588

**PROGRAMLAMA DİLLERİNİN PRENSİPLERİ**

Dr.Öğr.Üyesi HÜSEYİN DEMİRCİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## ➤ Proje Özeti

Proje, bir dosyadan alınan sayıları okuyarak, bu sayıları temsil eden canlı organizmaları oluşturur. Bu organizmaların türlerine göre, belirlenen kurallara göre birbirleriyle etkileşimlerini simüle eder. Sonunda, hayatta kalan son canlının konumunu ve türünü belirler.

## ➤ Yaptıklarımız

- **Yapı Tanımları:** Proje başlangıcında, bitki, böcek, sinek, pire gibi canlı organizmaları temsil eden yapılar tanımlandı. Her bir organizma, bir üst yapı olan CANLI yapısından türetilerek özelliklerini (satır, sütun, değer) alır.
- **Görünüm Fonksiyonları:** Her canlı türü için görünüm işlevleri belirlendi. Bu işlevler, ekrana canlının sembolünü yazdırmak için kullanılır.
- **Canlı Türünü Belirleme Fonksiyonu:** Verilen bir değere göre, o değeri temsil eden canlı organizmayı oluşturan fonksiyon tasarlandı. Bu fonksiyon, değere göre hangi türde bir canlının oluşturulacağını belirler.
- **Kazanan Canlıyı Belirleme Fonksiyonu:** İki canlının birbirini yiyip yiyemeyeceğini belirleyen fonksiyon yazıldı. Bu fonksiyon, organizmaların türlerine ve değerlerine göre birbirlerini yiyip yiyemeyeceklerini kontrol eder.
- **Dosyadan Okuma ve Bağlı Liste Oluşturma:** Verilen bir dosyadan sayıları okuyarak, bu sayıları temsil eden canlı organizmaları oluşturan fonksiyonlar geliştirildi. Bu fonksiyonlar, dosyadan alınan verileri okuyarak, her bir organizmayı oluşturur ve bu organizmaları bir bağlı liste içinde saklar.
- **Ekrana Yazdırma Fonksiyonu:** Oluşturulan bağlı listeyi ekrana yazdıran fonksiyon tasarlandı. Bu fonksiyon, her bir organizmanın sembolünü ekrana basarak habitatın görünümünü sağlar.
- **Canlı Savaşı Fonksiyonu:** Oluşturulan canlıların birbiriyle etkileşimlerini simüle eden fonksiyon geliştirildi. Bu fonksiyon, birbirleriyle etkileşime giren canlıları belirleyerek, kazanan canlıyı belirler ve kaybeden canlıyı listeden çıkarır.

```
C Canli.h  X
include > C Canli.h > ...
1  #ifndef CANLI_H
2  #define CANLI_H
3
4  struct CANLI {
5      char* (*gorunum)();
6      int row;
7      int column;
8      int value;
9  };
10
11 struct CANLI* canliTurunuBelirle(int deger, int row, int column);
12 int kazananCanli(struct CANLI* canli1, struct CANLI* canli2);
13
14 #endif
```

## ➤ Eksik Bırakılanlar ve Zorluklar

- **Bellek Yönetimi ve Hafıza Sızıntıları:** Projenin geliştirilme sürecinde, özellikle canlı organizmaların bellek yönetimi kısmında bazı zorluklar yaşandı. Özellikle canlıTurunuBelirle fonksiyonunda bellek tahsisi ve serbest bırakma işlemlerinin doğru şekilde yönetilmesi önemliydi.
- **Algoritmik Zorluklar:** Canlı organizmalar arasındaki etkileşimleri simüle etmek için algoritmik olarak karmaşık bir süreç gerekiyordu. Özellikle kazanan canlının belirlenmesi ve bağlı listeden çıkarılması gibi işlemler, doğru ve verimli bir şekilde gerçekleştirilmesi gereken algoritmik zorluklar içeriyordu.

## ➤ Öğrenim Deneyimi

Bu proje, Nesne Yönelimli Programlama (OOP) prensiplerini derinlemesine anlamamıza ve uygulamamıza olanak sağladı. Özellikle, Kalıtım ve Soyut Sınıf Benzetimi gibi yaklaşımları kullanarak, C programlama dilinde yapıların ve fonksiyonların nasıl tasarlanacağına dair kapsamlı bir anlayış geliştirdik.

Bu projede, dosya işleme, bellek yönetimi ve veri yapıları gibi kritik programlama kavramlarını uygulamak için mükemmel bir fırsat bulduk. Nesne Yönelimli Benzetim sayesinde, gerçek hayattan alınan senaryoları programlamaya dönüştürerek, soyut kavramları somut örneklerle anlama şansına sahip olduk.

İlk Durum

```
C:\Windows\System32\cmd.exe - mingw32-make
Microsoft Windows [Version 10.0.19044.4291]
(c) Microsoft Corporation. All rights reserved.

D:\Dars\New Project>mingw32-make
gcc -I ./include/ -o ./lib/Canli.o -c ./src/Canli.c
gcc -I ./include/ -o ./lib/Bitki.o -c ./src/Bitki.c
gcc -I ./include/ -o ./lib/Bocek.o -c ./src/Bocek.c
gcc -I ./include/ -o ./lib/Pire.o -c ./src/Pire.c
gcc -I ./include/ -o ./lib/Sinek.o -c ./src/Sinek.c
gcc -I ./include/ -o ./lib/Habitat.o -c ./src/Habitat.c
gcc -I ./include/ -o ./bin/Test ./lib/Canli.o ./lib/Bitki.o ./lib/Bocek.o ./lib/Pire.o ./lib/Sinek.o ./lib/Habitat.o ./src/main.c
./bin/Test
C B B B B
S B B S B
C P S S S
B B B B S

Savasi Baslatmak Icin Enter'a Basiniz....
```

Son Durum

```
C:\Windows\System32\cmd.exe
X X X X X
X X X X X
X X X X X
X X B X X
Kazanan: B : (3, 2)
```