



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

Koloni Savaşı

B181210076 - Muhammet Kemal Güvenç

SAKARYA

Mayıs, 2023

Programlama Dillerinin Prensipleri Dersi

Koloni Savaşı

Muhammet Kemal Güvenç

B18B181210076 – I/C

Özet

İlk önce kullanıcıdan kolonilerin popülasyon bilgileri parametre olarak alınır. Alınan bu bilgiler doğrultusunda bütün koloniler oluşturulur. Koloniler oluşturulduktan sonra her tur hepsi birbiriyle savaştırılır. Bu savaşlar sonucu ölen koloniler yok sayılarak devam edilir. Ayrıca her turda koloniler savaştıktan sonra yenilenirler. Yenilenmekten kasıt şudur: Her koloninin kendi üretim taktiği doğrultusunda yemek üretir. Popülasyonları yüzde 20 büyür. Ayrıca popülasyonun 2 katı kadar yemek tüketirler. Bu şekilde koloniler mücadele eder. Mücadele belirli bir tur sonra biter ve sadece birisi kazanır. Bu programın en önemli özelliklerinden biri her ne kadar aynı parametreler verilse de farklı sonuçlar üretmesidir. Çünkü koloniler oluşturulurken her biri için rastgele üretim ve savaş taktiği verilir. Ayrıca iki koloninin beraber kalma ihtimali dahilinde değildir. Eğer her şeyleri aynı ise rastgele biri mücadeleyi kazanır. Bu rastgelelikler programın her defasında farklı sonuçlar üretmesine sebebiyet verir.

© 2023 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Rastgelelik, Koloni, Mücadele, Popülasyon

1. GELİŞTİRİLEN YAZILIM

1. Uretim Yapısı: İçinde 2 adet fonksiyon bulundurulur. Bunlar şunlardır:

- a. Uretim UretimOlustur(): Uretim yapısı için nesne oluşturur.
- b. void UretimYokEt(AUretim const this): Uretim yapısından oluşturulmuş değişkeni yok eder.

2. AUretim ve BUretim yapıları: Her iki yapıda da 3'er fonksiyon bulunmaktadır. Her iki yapıda kalıtım benzetmesiyle Uretim yapısından kalıtım almaktadır. İçlerindeki fonksiyonlar şunlardır:

- a. AUretim AUretimOlustur(): AUretim yapısı için nesne oluşturur.
- b. void AUretimYokEt(AUretim const this): AUretim yapısından oluşturulmuş değişkeni yok eder.
- c. int AUret(AUretim const this, int populasyon, int yemek): Bu üretim taktiğinde koloninin popülasyon ve yemek değerleri birbirinden çıkarılır. Bu değer 10'u geçmemesi için 10'la modu alınır.
- d. BUretim BUretimOlustur(): BUretim yapısı için nesne oluşturur.
- e. void BUretimYokEt(BUretim const this): BUretim yapısından oluşturulmuş değişkeni yok eder.
- f. int BUret(BUretim const this, int populasyon, int yemek): Bu üretim taktiğinde koloninin popülasyon ve yemek değerleri birbiriyle toplanır. Bu değer 10'u geçmemesi için 10'la modu alınır. int değerinin taşmasından korunmak için negatif değer -1 ile çarpılır.

3. Taktik Yapısı: İçinde 2 adet fonksiyon bulundurulur. Bunlar şunlardır:

- a. Taktik TaktikOlustur(): Taktik yapısı için nesne oluşturur.
- b. void TaktikYokEt(): Taktik yapısından oluşturulmuş değişkeni yok eder.

4. ATaktik ve BTaktik yapıları: Her iki yapıda da 3'er fonksiyon bulunmaktadır. Her iki yapıda kalıtım benzetmesiyle Taktik yapısından kalıtım almaktadır. İçlerindeki fonksiyonlar şunlardır:

- a. ATaktik ATaktikOlustur(): ATaktik yapısı için nesne oluşturur.
- b. void ATaktikYokEt(ATaktik const this): ATaktik yapısından oluşturulmuş değişkeni yok eder.
- c. int ASavas(ATaktik const this, int populasyon, int yemek): Bu savaş taktiğinde koloninin popülasyon ve yemek değerleri çarpılır. Bu değer 1000'i geçmemesi için 1000'le modu alınır. int değerinin taşmasından korunmak için negatif değer -1 ile çarpılır.
- d. BTaktik BTaktikOlustur(): BTaktik yapısı için nesne oluşturur.
- e. void BTaktikYokEt(BTaktik const this): BTaktik yapısından oluşturulmuş değişkeni yok eder.

f. `int BSavas(BTaktik const this, int populasyon, int yemek):` B Bu savaş taktiğinde koloninin popülasyon ve yemek değerlerinin kareleri alınıp toplanır. Bu değerin 1000'i geçmemesi için 1000'le modu alınır. `int` değerinin taşmasından korunmak için negatif değer -1 ile çarpılır.

5. Koloni Yapısı: Koloni ile ilgili bilgiler tutmakla beraber içinde 6 farklı fonksiyon barındırmaktadır. Ayrıca bu 6 fonksiyona yardımcı birçok fonksiyon bulunmaktadır. Bahsedilen 6 fonksiyon şunlardır:

a) `Koloni KoloniOlustur(int populasyon, char simge):` Koloni yapısı için nesne oluşturur.

b) `void Kazan(Koloni this, int kazanilanYemek):` Eğer koloni bir savaşı kazanırsa parametre olarak aldığı değeri yemek miktarına ekler.

c) `int Kaybet(Koloni const this, int savasPuaniFarki):` Eğer koloni bir savaşı kaybederse parametre olarak aldığı değeri oranın yemek ve popülasyon kaybeder. Popülasyonun kaybettiği yemek miktarını diğer kolonin de bilebilmesi için bu değeri döndürür.

d) `void Yenilenme(Koloni const this, int uretilenYemekMiktari):` Koloninin savaştan sonra yapması gereken işleri yapar. Koloni üretim yapar; popülasyonu artar; yemek tüketir.

e) `bool YasiyorMu(Koloni const this):` Koloninin yaşayıp yaşamadığını döndürür.

f) `void KoloniYokEt(Koloni const this):` Koloni yapısından oluşturulmuş değişkeni yok eder.

6. Oyun Yapısı: Oyun ile ilgili bilgiler tutmakla beraber içinde 3 farklı fonksiyon barındırmaktadır. Ayrıca bu 3 fonksiyona yardımcı birçok fonksiyon bulunmaktadır. Bahsedilen 3 fonksiyon şunlardır:

a) `Oyun OyunOlustur(char* veri):` Oyun yapısı için nesne oluşturur.

b) `void Baslat(Oyun const this):` Oyunu başlatır. Her turda koloniler ile ilgili bilgileri ekrana çıkartır. Onları savaşa sokar ve koloniler tur sonunda onlara yapmaları gerekeni yaptırır.

c) `void OyunYokEt(Oyun const this):` Oyun yapısından oluşturulmuş değişkeni yok eder.

2. ÇIKTILAR

Bu programa ister aynı ister farklı girdiler verilse hepsinde çıktı farklı olacaktır. Bunun sebebi projedeki birçok şeyin rastgele değişkenler aracılığı ile yapılmasıdır.

3. SONUÇ

Bu yazılım ile farklı popülasyonlara ait kolonilerin nasıl mücadele ettiğini simüle ederek görebilirsiniz.