



SAKARYA
ÜNİVERSİTESİ

T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

WILD LIFE

G201210090 - Mustafa Özcan

SAKARYA

Nisan, 2024

Programlama Dillerinin Prensipleri Dersi

Wild Life

Mustafa ÖZCAN

g201210090 1-C

Özet

Bu ödev, Mingw (gcc) kullanılarak C dilinde yazılması gereken bir programı içeriyor. Program, Veri.txt dosyasını okuyacak ve içindeki 1-99 aralığındaki tam sayıları analiz ederek canlıları temsil eden yapıları oluşturacak. Bu canlıların birbirleriyle etkileşimini belirli kurallara göre modelliyor ve bir rekabet senaryosu oluşturuyor. Temel olarak, ödevde şu adımlar izleniyor. Veri.txt dosyası okunuyor ve sayı aralıklarına göre ilgili canlı yapısı oluşturuluyor. Oluşturulan canlılar yapısı yine “Canlı” türünden bir matrise atanıyor. Matrisin tamamı canlılarla dolduktan sonra matris içerisinde bulunan canlılar matris indis sırasına göre canlı yapılarının ismi ekrana bastırılıyor. Bu adımdan sonra ise kullanıcı herhangi bir tuşa basarak süreci başlatıyor.

Süreç canlıların birbiriyle olan etkileşimi ile ilerliyor. Veriler.txt dosyasındaki sayı değerine göre oluşan canlılar arasında bir hiyerarşi bulunmakta olup bir canlı başka bir canlıyı yiyebiliyor ölen canlı matris içerisinde “X” ile temsil ediliyor. Her yeme-ölme durumunda sonra matris ekrana yazdırılıyor. Matris içerisinde her canlı birbiri ile karşılaştırılıyor ve en sona kalan canlı kazanan olarak belirleniyor kazanan canlının ismi ve matristeki konumu ekrana bastırılıyor.

1. YAZILIM İŞLEYİŞİ

Program çalışır çalışmaz “Veriler.txt” adlı dosyadan verileri okuyacaktır. Verilerin sayı değerlerine göre Canlılar oluşturur. Program “Canlı”, “Bitki”, “Böcek”, “Sinek”, “Pire” yapılarını içerir ve bunlar arasında kalıtım ilişkisi vardır. *Bitki*, *Böcek*, *Sinek*, *Pire* birer *Canlı*dır. Bunun yanı sıra *Sinek* ve *Pire* aynı zamanda *Böcektir*. Her canlının kendisine ait bir ismi ve sayı değeri vardır. Ayrıca tüm canlıların ekrana yazılması Canlı sınıfındaki “gorunum()” fonksiyon gösterici vasıtasıyla yapılır. Bu işlemi gerçekleştirmek için öncelikle her canlı kendi sınıfı içerisinde ekrana yazılma fonksiyonunu oluşturup bu fonksiyonu Canlı sınıfındaki “gorunum()” fonksiyon göstericisine bağlanması gerekmektedir.

Proje esnasında çeşitli algoritalar ve yoğun düşünsel süreçler geçirilmiştir. Bu kapsamda en çok zorlanılan adımlardan biri bellek hataları olmuştur. Canlı yapılarını oluştururken “heap” bellek bölgesinden yer açmamız gerektiğinden ötürü zaman zaman erişilmemesi gereken veya yanlış erişim hataları sebebi ile proje gerçekleştirme işlemi esnasında çok sayıda boş ekran alınmıştır. Karşılaşılan bu problem bellek bölgesi işlemlerinin yapıldığı

© 2024 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Kalıtım, Soyut sınıf, Nesneye Yönelimli Benzetim, Dosya işlemleri, Loglama, Bellek Tahsisi,

alanlara loglamalar yapılarak adete bir derleyici gibi satır satır kodu takip edilerek bellek hataları tespit edilmeye çalışılmış ve hatanın kaynağı bulunup düzeltilmiştir.

The screenshot shows a Windows command prompt window on the left and a C++ code editor on the right. The command prompt displays the output of a program, including a 4x5 grid of numbers and a series of log messages indicating game progress. The code editor shows the source code for the game, including a function to clear the screen and a function to check for a winner.

```

10 2 5 4 7
21 6 1 25 8
11 64 33 41 47
8 1 9 5 23

Sureci baslatmak icin herhangi bir tusa basin...
Surec basladi!
dosya satir sayisi :4
dosya col sayisi :5
karsilastir winner a atama yapiyor.
kazanini velirle
C: B: isimler farkli block giris
karsilastir winner a atama yapiyor.
kazanini velirle
C: B: isimler farkli block giris
karsilastir winner a atama yapiyor.
kazanini velirle
C: B: isimler farkli block giris
karsilastir winner a atama yapiyor.
kazanini velirle
C: B: isimler farkli block giris
karsilastir winner a atama yapiyor.
kazanini velirle
C: S: isimler farkli block giris
matrisi yadir.
makefile:13: recipe for target 'calistir' failed
mingw32-make: *** [calistir] Error -1073741819

C:\Users\asus\Desktop\PDPODEV2CALISMASI\Odev>

```

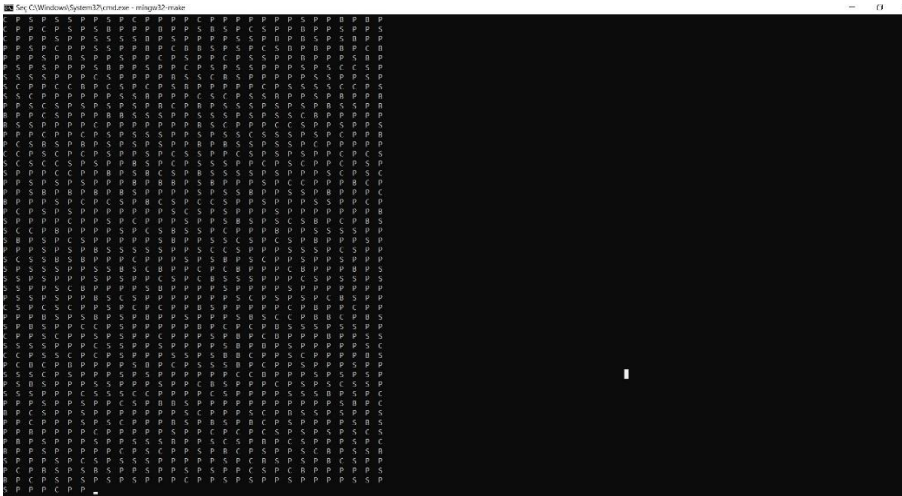
```

9 #include "canli.h"
10
11 //ekrani temizle
12
13 void clearScreen() { ...
16 int _satir_sayisi(const char *dosyaAdi) { ...
37 int _sutun_sayisi(const char *dosya_adi) { ...
66
67 /* void karsilastir_ve_degistir(Canli matris[_sutun_sayisi("Veriler.txt")], int satir, int sutun) {
145
146 Canli karsilastir(Canli winner, Canli current, int satir, int sutun, Canli matris[_sutun_sayisi("Verile
147
148 if (current != NULL && winner != NULL) {
149 // Kazananı belirle
150 printf("kazanini velirle\n");
151 printf("%s: ", winner->isim);
152 printf("%s: ", current->isim);
153 if (winner->isim != current->isim) { // isimler farklıysa, kazananı belirle
154 printf("isimler farkli block giris\n");
155
156 if (winner->isim == "B") {
157 if (current->isim == "S" || current->isim == "P") {
158 current->isim = "X";
159 return winner;
160 printf("1. cikis\n");
161 }

```

Şekil 1. Loglama işlemleri

Ödev bellek sızıntı test etmek amacıyla büyük verilerle test edilmiş ve problem yaşanmamıştır yalnızca veri boyutuna bağlı olarak ekrana yazdırma işlemleri zaman almaktadır.



Şekil 2. 100 Satır 30 Sütun sayılı veri dosyası

ÇIKTILAR

Ödev dosyasında yer alan “Veriler.txt” için aşağıdaki çıktı alınmaktadır :

Veriler.txt :

© 2024 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Kalıtım, Soyut sınıf, Nesneye Yönelimli Benzetim, Dosya işlemleri, Loglama, Bellek Tahsisi,

10 2 5 4 7
21 6 1 25
8 11 64 33 41
47 8 1 9 5 23

C B B B B
S B B S B
C P S S S
B B B B S

Sureci baslatmak icin herhangi bir tusa basin...
Surec basladi!
dosya satir sayisi :4
dosya col sayisi :5

C X B B B
S B B S B
C P S S S
B B B B S

C X X B B
S B B S B
C P S S S
B B B B S

C X X X B
S B B S B
C P S S S
B B B B S

C X X X X
S B B S B
C P S S S
B B B B S

.....
.....
.....

:X X X X X
X X X X X
X X X X X
X X B X X
Kazanan: B
Kazananin konumu: (3,2)

3.SONUÇ

Ödev icrası sırasında çeşitli algoritmalar geliştirilmiş,Nesneye Yönelik Benzetim, Kalıtım, Dosya işlemleri, Heap Bellek Bölgesi Tahsis etme,Memory Leak,Matris işlemleri konularıyla ilgili çalışmalar yapılmış olup buna ek olarak loglama ve hata tespit işlemleri ile hatalar giderilip düzeltilmiştir.

© 2024 Sakarya Üniversitesi.

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

Anahtar Kelimeler:Kalıtım,Soyut sınıf,Nesneye Yönelimli Benzetim,Dosya işlemleri,Loglama,Bellek Tahsisi,
