

Programlama Lab I Proje I

Devran Yılmaz

Bilgisayar Mühendisliği(İÖ)

Kocaeli Üniversitesi

İstanbul / Türkiye

220202011@kocaeli.edu.tr

Ecem Yalçın

Bilgisayar Mühendisliği(İÖ)

Kocaeli Üniversitesi

Düzce / Türkiye

220202040@kocaeli.edu.tr

ÖZET

Bu proje, C yazılım dili ve kütüphaneleri kullanılarak bir şirketin maksimum kâr elde edebilmesi amacıyla kullanıcıyla etkileşime girip şirket için olabilecek en optimum durumu analiz etmek için çalışmaktadır.

I. GİRİŞ

Bu proje DEV C++ idesi ve Graphics.h kütüphanesinden faydalanılarak C yazılım dilinde yazılmıştır. Bu projede misyonumuz C yazılım dilini kullanarak algoritma ve problem çözme yeteneğimizi kütüphane bilgimizle harmanlayarak geliştirmektir. Proje başlangıçta kullanıcıdan birtakım veriler istemektedir. Uygulama süreç esnasında kullanıcının verdiği verilerden faydalanarak isterleri karşılamaya çalışmaktadır.

II. YÖNTEM

İlk aşamada, edestek2.kocaeli.edu.tr adresinden bize verilen ve koordinatları içeren bir txt dosyasını C'de dosya okuma yöntemini kullanarak okuttuk. Adından Windows BGI (Borland Graphics Interface) kullanılarak grafiksel arayüz oluşturulup yeni bir pencere açtık. Bu adım, projenin görsel olarak sunumunu sağlar. Okunan koordinatlar, iki adet boş char dizisine aktarıldı. Bu diziler, verilerin işlenmesi için kullanıldı. Fonksiyon kullanılarak karakter dizilerinin uzunluğu hesaplandı. Bir sayaç tanımlanarak "for" döngüsü içinde dizilerin elemanları dolaşıldı ve dizinin sonuna ulaşılan kadar sayaç artırıldı. Sonunda sayaç fonksiyon sonucu olarak döndürülerek dizinin uzunluğu hesaplandı. Farklı bir fonksiyon içerisinde karakter dizileri içindeki koordinat değerlerinin kaç basamaklı olduğunu belirlemek için ASCII tablosu kullanıldı. Eğer ASCII değeri kırk sekizden büyük eşitse ve elli yediden küçük eşitse, bu aralıkta bir değerse döngümüz çalışır. İ sayacı döngü dışında tanımlanarak, fonksiyona parametre olarak gelen karakter içindeki sayının basamaklarını işlemek ve sonucu döndürmek için kullanıldı. Yeni bir fonksiyon açarak bir tam sayı dizisinin uzunluğu hesaplandı.

X ve Y koordinatlarının ayrılabilmesi için fonksiyonumuza mainde tanımladığımız dizileri parametre olarak yolladık. Bu algoritmanın çalışma temelinde yay ayraç diye tabir ettiğimiz '(' ifadesinin dizideki konumundan faydalanıp sayıları ayıklamaktır. Bu kısımda özellikle dikkat edilmesi gereken iki şey vardır: Birincisi ve en önemlisi, X eksenindeki sayılar her zaman yay ayraç'ın bir önündeki indekste var olacağının bilincinde olunmasıdır. İkinci dikkat edilmesi gereken husus ise X eksenindeki sayıların basamak sayısı Y eksenindeki sayıların elde edilmesinde hayati rol oynamasıdır. Belirttiğimiz gibi X değeri her zaman yay ayraç'ın önündeki indekste konumlanır. Fakat X 'in basamak sayısı değiştiği zaman bundan Y'de etkilenir. Bu sebeple daha önceden yazdığımız algoritmadan faydalanılarak X 'in basamak sayısı kadar char dizisini ayrıca ileri iteliriz.

Bu aşamadan sonra for döngülerinden faydalanılarak ayıklanmış sayı değerlerimizi geçici char dizilerine aktarıyoruz. String.h kütüphanesindeki fonksiyonlardan faydalanılarak geçici char dizilerinin içerisindeki sayıları gerçek int değerlerine aktarıyoruz. Aktarılan değerler iki ayrı integer dizi içerisinde tutulur. Bu algoritmamız da böylece tamamlanmış olur.

Ayıklanan koordinatları tutan diziler birden fazla şekli ayıklamak amacıyla yazılan fonksiyona parametre olarak gönderilir. Fonksiyon içerisine üçer adet yeni değişken tanımlanır sebebi, üç farklı şeklin çizdirilebilmesi ihtimalinden kaynaklıdır. Dizilerin eleman sayılarının takibi yapılarak "for" döngüsü açılır. Buradaki önemli husus başlangıç indeksini hesaba katmaktır. İlk indeks son indekse eşit olana kadar döngü devam eder.

Dolayısıyla şekiller kenar sayısının bir fazlası kadar döner. Açtığımız koşullarla beraber hata olasılığı en aza indirilmeye çalışılır. İlk şeklin uzunluğu toplam uzunluktan çıkartılarak diğer şekillerin bulunmasında yardımcı olarak kullanılır. Bulunan bu değer, aynı algoritma kullanılarak tekrar kontrol ettirilir ve şekiller varsa ayıklanır. Ayıklanan şeklin uzunluk sayacı saklanır ve eğer varsa üçüncü şekil için hesaplamalarda toplam uzunluğun dışında bırakılarak üçüncü şeklin bulunmasına katkı sağlar. Aynı işlemler eğer varsa üçüncü şekil için de uygulanır böylelikle tüm şekillerin çizdirilmesi ve aynı zamanda alanlarının hesaplanması sağlanır. Çizim

tamamlandıktan sonra hangi satırdaki şeklin çiziminin yapılacağı da başarıyla tamamlanmış olur.

III. SONUÇ

Alan bulma işlemi sonrasında şeklin parsellenebilmesi için şeklin kestiği en dış birim karelerin koordinatları bulunur. Bu yöntem içinse yine aynı şekilde eğim ile taşma oranı yönteminden yararlanılır. Daha sonrasında ise şeklin içerişi boyatılarak sonraki aşamaya geçilir.

Kullanıcı inputları için yeni fonksiyona geçilir. Bu fonksiyonda kullanıcıdan sondaj ve platform maliyetleri istenir. Alınan değerler, daha sonraki fonksiyonlarda kullanılır.

Parselleme işlemi içinse aynı X veya Y koordinatlarında bulunan değerler değişken içerisine alınır. Daha sonra bu değerler kullanılarak şeklin parsellenmesi için bulunan değerler birleştirilir. Son aşamada ise parselleme tamamlanır ve karelerin boyutlarına göre sondaj ve platform maliyetleri hesaplanır. Toplam kâr zarar miktarı kullanıcının bilgisine sunulur program görevini yerine getirir.

IV.PSEUDO KODLAR & DENEYSEL SONUÇLAR

I. Giriş

```
1-)Verilen URL'deki koordinatlar system curl ile metin belgesine yazdır.
2-)Metin belgesinden değerler iki ayrı char dizisi içerisine aktar.
3-)initwindow ile sekme aç.
4-)DisplayBackground fonksiyonu ile arka plandaki çizgiler çizdir.
5-)Kullanıcıdan hangi satır çizdirilecekse o satır while döngüsü içerisinde iste.
6-)Alınan değerin kontrolü yapılır, alınan değere göre satırlar yolla.
7-)SeperateArray fonksiyonu içerisine satırın array'i, array'in uzunluğu ve daha sonra kullanmak üzere X ve Y koordinatlarını tutan diziler parametre olarak yolla.
8-)Jstrcpy fonksiyonu ile gelen array yeni bir diziyi al.
9-)for döngüsü içerisinde satırın uzunluğu kadar dönlülür. Sayıları ayırmak için 2 adet char dizisi tutulur. Eğer array[i] paranteze eşit olursa if'in içerisine gir.
10-)If bloğu içerisinde parantezden sonraki sayıyı almak için CheckForInt fonksiyonuna atılır. CheckForInt parametre olarak kelimelerin dizisini ve başlangıç indexini parametre olarak al. Geriye sayının basamak sayısını döner.
11-)lenght2 int değişkeni içerisinde basamak sayısı tut.
12-)sscanf fonksiyonu ile char içerisindeki sayı int diziyi atılır. Aynı işlemler Y için de yapılır. Fakat Y değeri bulunurken ek olarak X değerinin basamak sayısı kadar gelen Array ötelenir. Örn (13,12) (5,12) X değişkeni her zaman '(' + 1 indextir fakat Y'de ek olarak X'in basamak sayısı da ekler.
```

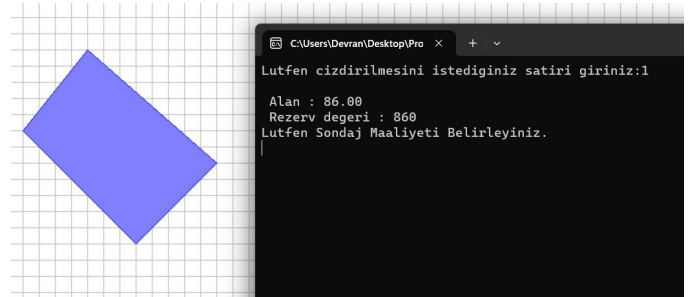
II. Gelişme

```
1-)DisplayShape fonksiyonuna Giriş kısmında ayırdığın ayrı X ve Y dizilerini ve şeklin uzunluğunu yolla.
2-)Uzunluk kadar for içerisinde dön.
Eğer X ve Y dizilerinin i, indeksinde 0, elemama denk gelirse ve aynı zamanda i!=0 ise ve il-lenght-1 ise if içerisine gir.
3-)Toplam şekil uzunluğundan i değerini çıkartıp int bir değere atama yap. Bu atamayı varsa 2. şekil için kullan ve onun uzunluk sayacını arttır.
4-)3. şekil kontrolü için toplam uzunluk - 2. şeklin uzunluğu - 1 işlemini yapıp count3 değişkenine ata.
5-)for içerisinde l=0'dan başlayarak l < lenght-count2-count3 kadar dönlüp her seferinde l yi bir defa arttırarak 1. şeklin koordinatlarını newarray ve newarray arraylerinde tut.
6-)Aynı işlemi 2. şekil için de yap. count3 'ün 1 den büyük olup olmadığı kontrolünü sağla ve 3. şeklin var olup olmadığını kontrol et.
7-)CalcArea fonksiyonunu çağır. Parametre olarak X , Y arrayini ve uzunluğu parametre olarak yolla.
8-)CalcArea fonksiyonu içerisinde gelen şeklin X ve Y arraylerinin MAX ve MIN değerlerini bul.
9-)for içerisinde i=0 olacak şekilde lenght-1'e kadar l'yi 1 arttır. Gelen koordinatlardan yararlanarak ardışık köşelerin arasında kalan üçgen alanlarını hesap et.
10-)Üçgen alanlarının mutlaklığı olarak area değişkenine ata. Ardışık köşelerin koordinatları arasındaki farkı kullanarak tanjant değeri hesapla.
11-)Ardışık Y'ların farkı 0'a eşit değilse if bloğu içerisine gir.
12-)if içerisinde tanjant'ın durum kontrolünü yap. Eğer tanjant 1'e eşit veya tanjant ondalıklı sayıya denk değilse üçgen alanlarından ardışık X'lerin farkının yarısını tanjant ile çarparak çıkar.
13-)Eğer tanjant 1'den küçükse aynı işlemi yap. Bu sefer 1 fazladan çıkar. Eğer tanjant 1'den büyükse aynı işlemi yap. Bu sefer 2 fazladan çıkar.
14-)if içi 2 tane for tanımla. i ve k değerlerini 0'dan başlatıp 100'e kadar her seferinde 1 arttır. i nın en küçük ve en büyük X değerleri arasında olup olmadığını ve k nın en küçük ve en büyük Y değerleri arasında olup olmadığını kontrol et. Eğer arasında ise totalPoints değişkenini 1 arttır.
15-)totalArea değişkenine totalPoints ile area değişkenlerinin farkını ata. printf fonksiyonu içerisinde totalArea'yı ve altındaki rezerv değerini totalArea*10 olacak şekilde yazdır.
16-)Yukarıdaki eğim işlemlerini tekrarla. Kare dışındaki noktaları xArray ve yArray olacak şekilde iki ayrı diziyi aktar.
17-)Bu yeni dizilerin maksimum ve minimum X,Y değerlerini bulup değişkenlere ata.
18-)Her adımda yeni bir değişken tanımla.
Bu değişken içerisinde şeklin uzunluğu kadar döner ek eş noktadaki X ve Y değerlerini ardışık olarak ata.
19-)Bu değeri kullanarak şeklin içerişini boyat.
```

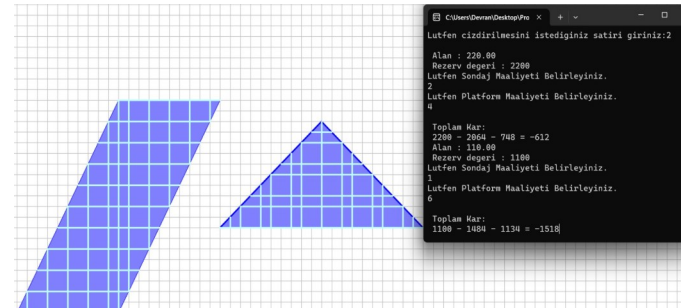
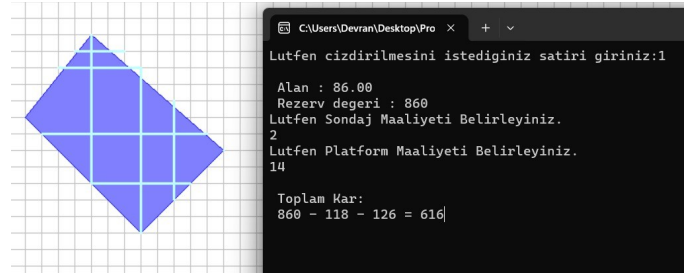
III. Sonuç

```
1-) Platform ve sondaj maliyeti için değişken oluştur.Bu değişkenlerin adreslerini DrillCostcalc fonksiyonuna gönder.
2-) kullanıcıdan sondaj ve platform maliyetini iste. Bu değerleri parametre olarak gelen değişkenlere ata.
3-) DisplayDrillArea fonksiyonuna daha önce bulduğun xArray,yArray, platform ve sondaj maliyeti, toplam alan ve xArray,yArray'ın maksimum ve minimum noktalarını parametre olarak gönder.
4-) i=0 ve k=0 olmak üzere iç içe for kullanarak lenght kadar dön.
Eğer arrayX[i] == arrayX[k] veya arrayY[i] == arrayY[k] il=k ise count değişkenini 2 arttır.
5-) bulduğun count değişkenini displayX ve displayY 'nin eleman sayısı olarak ata.
6-) i=0 ve k=0 olmak üzere iç içe for kullanarak lenght kadar dön.
Eğer arrayX[i] == arrayX[k] veya arrayY[i] == arrayY[k] ve il=k ise displayX ve displayY dizilerine bulunan X ve Y değerleri, X. index + 1 ve Y. index + 1 değerlerini displayX ve displayY dizilerine atayıp index'i 2 arttır.
7-) drillCostTotal ve platformCostTotal değişkeni ata. for içerisinde count değeri kadar dönlüp i'yi platform değeri kadar arttır. Eğer displayY[i] == displayY[i+1] veya displayX[i] == displayX[i+1] denkliliğini sağlarsan if bloğuna gir.
8-) displayY[i] - displayY[i+1] != 0 ise val değişkenini bu fark'a ata.
9-) displayX[i] - displayX[i+1] != 0 ise val değişkenini bu fark'a ata.
10-) val < 0 ise val = -val olarak ata.
11-) drillCostTotal değişkenini val*drill kadar arttır.
12-) platformCostTotal değişkenini platform kadar arttır.
13-) printf fonksiyonunda toplam kâr miktarını yazdır.
```

Kullanıcıdan satır istenip girdiği satırdaki şekil çizilir ve alanı hakkında bilgi verilir:



Kullanıcıdan sondaj ve platform maliyetlerini girmesi istenir. Kullanıcının girdiği değerlere göre parselleme işlemleri yapılır:



KAYNAKLAR

- [1] <https://www.geeksforgeeks.org/fillpoly-function-c/>
- [2] <https://www.geeksforgeeks.org/draw-line-c-graphics/>
https://www.tutorialspoint.com/c_standard_library/c_function_sscanf.htm
- [3] <https://www.youtube.com/watch?v=7CRf4nZ5CRw&list=PLD5D5Hj95BCFid63gy2ViVBXJZQXuwl2R>
- [4] https://notpast.com/c_programlama/C-Programlama-Dilinde-Dosya-Islemleri-71.html