

# Programlama Laboratuvarı 1

## Proje 1

### I. ÖZET

Bu proje denizlerde doğal kaynak arama ve çıkarma operasyonları gerçekleştiren bir şirket için maksimum kâr elde edebilmek amacıyla arama bölgesini en doğru sayıda ve optimal boyutlara bölünmüş alanlara (parsellere) ayırmayı hedeflemektedir. Proje, bu hedef doğrultusunda bize verilecek olan URL adresinden koordinat verilerini çekip, kullanıcıdan alınan birim sondaj ve birim platform maliyetleri ile en kârlı olacak şekilde sondajın nasıl yapılabileceğini görsel olarak kullanıcıya gösterir.

### II. GİRİŞ

Proje, C programlama dili kullanılarak yazılmıştır. URL adresinden veri çekmek için curl.h kütüphanesi ve grafik penceresi açmak, şekil çizmek ve boyamak için graphics.h kütüphanesi kullanılmıştır.

Programın ana fonksiyonları şunlardır:

- Kapalı alanların (rezerv alanlarının) çizilmesi ve kapladığı alanların boyanması:

URL adresinden aldığımız koordinat bilgilerinden kullanıcının seçtiği satırdaki köşe noktalarıyla kapalı alanlar çizilir ve boyanır.

- Optimum durum için platform çizme:

Kullanıcıdan aldığımız sondaj maliyetine bağlı olarak en az sayıda platform kullanmaya çalışarak rezerv alanını kaplayan 1\*1, 2\*2, 4\*4, 8\*8 veya 16\*16 birim karelik platformlar oluşturulur.

- Maliyet ve kâr miktarlarının hesaplanması:

Oluşturulan platform sayısına bağlı olan platform maliyeti ve sondaj maliyetinin hesaplanmasıyla maliyet hesabı ve kâr miktarının hesaplanması ve kullanıcıya gösterilmesi.

### III. YÖNTEM

URL adresinden veri çekmek için libcurl kütüphanesinin içerisinde bulunan çeşitli hazır fonksiyonları kullandık. Öncelikle write\_callback fonksiyonu ile libcurl'e veri alım sürecinde dinamik bellek ayrılıyor ve veriyi doğru bir şekilde işlememizi sağlıyor. Main fonksiyonu libcurl'un temel adımlarını içeriyor. Bu adımlar; CURL nesnesinin başlatılması, URL'nin belirlenmesi, callback fonksiyonu ve first\_data dizisinin pointer'ının belirlenmesi, curl\_easy\_perform fonksiyonu ile veri çekilmesi gibi işlemleri gerçekleştiriyor. Kodun amacı, belirtilen URL'den alınan veriyi ekrana yazdırmaktır. Bu kod parçası bellek hatalarına karşı da çeşitli kontroller gerçekleştiriyor.

Kullanıcıdan alınan satır numarası kullanılarak üzerinde işlem yapılacak koordinat noktalarının bulunduğu satırın seçilmesini ve seçilen bu satırın first\_data isimli dizisinden allocated\_data isimli diziye aktarılmasını sağladık. Bu dizide string dizisi olarak bulunan verilerin içinden işlem yapmamıza yarayacak olan integer verilerin, while döngüsünü kullanarak integer bir dizi olan coordinates dizisine aktarılmasını sağladık. coordinates dizisinde bulunan sırayla her bir ikiliyi (x,y) ikilisi olarak kabul ettik. Bu döngü içerisinde coordinates dizisinin eleman sayısını da buldurup numberofCoordinates değişkenine atadık.

Bir for döngüsü içerisinde üzerinde işlem yapacağımız noktaların bulunduğu coordinates dizisindeki en küçük x, en küçük y, en büyük x ve en büyük y değerlerini buldurduk.

Kullanıcıdan birim sondaj ve birim platform maliyetlerinin değerlerini aldık. Birim sondaj maliyetinin 1 ile 10 arasında olması bir zorunluluk olduğundan kullanıcı bu aralık dışında bir değer girdiğinde tekrar birim sondaj maliyeti isteyebilmek için do-while döngüsü kullandık.

Graphics.h kütüphanesinden yararlanarak bir grafik penceresi başlattık. Bu pencerenin içerisinde yaptığımız ilk işlem grafik penceresini birim karelere ayırmak oldu. Çünkü kullanıcıya gösterdiğimiz şekillerin bulunduğu konumların daha net anlaşılmasını istedik. Bu kısımda 10\*10'luk birim kareler oluşturmamızın nedeni bir sonraki adımda 1\*1'lik birim karelerle ve rezerv alanlarını gerçek değerler ile

çizdirdiğimizde boyutlarının fazla küçük gözüktüğünü fark etmemizdi. Bunu engelleyebilmek için şekil çizdirip boyatırken elimizdeki koordinat bilgilerini ve arka planda gözüken birim kareleri çarpı 10 oranında büyüttük.

`draw_paint_areaofShapes` fonksiyonu alınan koordinat bilgileriyle rezerv alanlarını çizen, boyayan ve alanını bulan bir fonksiyondur. Bu kısımda rezerv alanlarını renkli bir çizgiyle gösterebilmek için `setcolor()` fonksiyonunu kullandık. Başlangıçta `coordinates` dizisini her bir rezerv alanı için farklı dizilere bölmediğimizden dolayı bir for döngüsü içerisinde başlangıçtaki (x,y) ikilisine eşit başka bir ikili var mı diye kontrol ettirdik. Eğer böyle bir ikili varsa `second` adıyla tanımladığımız ve başlangıçta sıfır olan integer değere bu ikiliden sonraki ikilinin x değerinin indexini saklattık. Eğer eşit ikili bulunamazsa belirtilen ikililer arasında şeklin kenar çizgisini çektirdik. Bu işlemi de `line()` fonksiyonuyla sağlattırdık. Aynı for döngüsünün içerisindeki ikinci ve üçüncü koşullarda, `second` ve `third` değişkenlerimiz sıfır değil ise yani `coordinates` dizimiz birden fazla rezerv alanının koordinat bilgilerine sahipse aynı işlemi bu seferde diziyi `second` indexinden başlatıp devam ettirdik.

Çizgi çektikten sonra, yine çizgi çekmek ile aynı mantıkta eğer ikililer birbirine eşit değil ise bu ikilileri kullanarak rezerv alanı hesaplattık. Alan hesaplatmasında Gauss Shoelace formülü diye bilinen bir formül uyguladık. Bu formül temelde köşe noktalarının koordinatları bilinen kapalı şekillerinin alanlarının bulunmasında kullanılıyor. Shoelace formülünü kullanırken bulduğumuz sonucun mutlak değerini aldırarak için `math.h` kütüphanesini kullandık.

Gauss Shoelace formülü:

$$\frac{1}{2} \left| \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \right|$$
$$\frac{1}{2} |x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 + \dots - \dots + x_n y_1 - x_1 y_n|$$

`draw_paint_areaofShapes` fonksiyonunun içerisinde `array_allocation` isimli ikinci bir fonksiyon çağırısı yaptık. Bu fonksiyon `coordinates` isimli diziyi en fazla üç şekil olacak şekilde düşünülerek (eğer varsa) üç ayrı diziye aktarıyor. Bu fonksiyon içerisinde bir, iki veya üç rezerv alanının olması ihtimaline karşı yine `second` ve `third` değişkenleri kullanılarak kontrol yapılıyor. Sonrasında `memcpy()` fonksiyonu kullanılarak `shape1`, `shape2` ve `shape3` dizilerine kopyalama işlemi yapılıyor. Bu fonksiyonun yazılmasındaki amacımız `fillpoly()` fonksiyonunu doğru kullanmayı sağlayabilmektir.

`array_allocation` fonksiyonundan sonra ayrılan dizilerin yine kaç rezerv alanına sahip olduğu kontrol ediliyor. Duruma göre içleri boyanacak alanların koordinatlarının bulunduğu dizi `multiplied10()` fonksiyonuna gönderiliyor. Dizinin her elemanı burada 10 ile çarpılarak yeni bir diziye aktarılıyor. Yeni dizi ile oluşan rezerv alanlarının içleri `setfillstyle()` ve `fillpoly()` fonksiyonları kullanılarak boyanıyor.

`draw_paint_areaofShapes` fonksiyon çağırısından sonra sondaj maliyetine bağlı olarak bir kontrol yapılıyor. Bu kontrolün amacı en kârlı sondajın nasıl yapılacağına karar vermektir. Temel mantığımız eğer birim sondaj maliyetimiz düşüğe bu durum daha geniş platformlar kurmamıza imkân sağlar.

Platform kurma mantığımız daha önce çizdirip boyadığımız rezerv alanlarındaki renkli piksellerden yola çıkarak platform kuracağımız yerin rezerv alanının üzerinde olup olmamasını kontrol etmekte. Platform kurmadan önce piksel sayımızı hesaplamak için `numberOfRedPixels()` fonksiyonunu kullandık. Bu fonksiyon sırayla en küçük x ve en küçük y değerinden başlayarak en büyük x ve en büyük y değerine kadar tüm pikselleri gezerek (Bu gezme işlemi bir for döngüsünde ve `getpixel()` fonksiyonuyla sağlıyoruz.) bizim istediğimiz renkteki pikselden kaç tane olduğunu bulup `counter` adlı değişkene atıyor ve değişkeni fonksiyondan geri döndürüyor. Geri döndürülen bu değeri `redpixels` adlı integer değişkene atadık.

Sondaj maliyetimizin 1 veya 2 olduğu durumlarda platform çizme işlemine 16\*16 birim karelik platformlardan başladık. Burada bir for döngüsü başlatarak çizeceğimiz platformun yüzde yüz kırmızı pikselle dolu olup olmadığına baktık. Döngümüz yüzde yüzden başlayıp sırayla yüzde on azalarak platformumuzun en az yüzde elli kırmızı piksele sahip olup olmadığına kontrol ediyor. Ama 16\*16 birim karelik platformun yüzde ellisinden az renkli piksel varsa platform boyutunu düşürerek çizebileceğimiz diğer 8\*8, 4\*4, 2\*2, 1\*1 birim karelik platformları kontrol ettik.

Daha sonra aynı işlemleri sondaj maliyeti 3 veya 4 iken 8\*8 birim karelik platformdan, sondaj maliyeti 5 veya 6 iken 4\*4 birim karelik platformdan, sondaj maliyeti 7 veya 8 iken 2\*2 birim karelik platformdan başlayarak kontrol edip eğer koşulu sağlamıyorsa birim kare boyutunu küçülterek devam ettik. Bu küçültme sırasında bir kere üzerine platform çizdiğimiz pikselin üzerine tekrar platform çizmemek için tekrar `getpixel()` fonksiyonunu kullanarak bir dahaki platform çizeceğimiz pikselin daha önce boyadığımız platformun rengine sahip olup olmadığına baktırdık.

Sondaj maliyeti 9 veya 10 ise direkt 1\*1 birim karelik platformlar çizdirdik ama bu platformları çizerken zararı en aza indirmek için rezerv alanının içinin en az yüzde sekseninin renkli pikselden oluşup oluşmadığını kontrol edip sonrasında platformu çizdirdik. Platform çizildikten sonra döngü içerisinde içi boyanıyor, platform sayısı artırılıyor ve bu platformun alanı sondaj alanına dahil ediliyor.

En son toplam platform maliyetini, toplam sondaj maliyetini, toplam maliyeti ve kâr miktarını matematiksel olarak hesaplayıp, bunlarla beraber rezerv değerini, toplam platform sayısını, toplam sondaj alanını kullanıcıya gösterdik.

## IV. DENEYSEL SONUÇLAR

Optimum platform sayısını bulabilmek için grafik penceresinin tamamını kontrol etmek yerine yalnızca rezerv alanının başladığı ve bittiği koordinat noktalarını kullanmak daha kârlı ve daha optimum bir sonuca ulaşmamız için önemlidir.

Oluşturulan platformların görsel olarak gösterilmesi kullanıcıya tam olarak ne yapıldığını daha net bir şekilde görme fırsatı sunduğu için daha iyi bir kullanıcı deneyimi sunması açısından önemlidir.

## V. SONUÇ

Projenin ilk amacı URL adresinden koordinat bilgilerini çekmektir. Bunun için curl.h kütüphanesinin bize sağladığı hazır fonksiyonlar kullanıldı.

Sonrasında çizilecek olan rezerv alanları için koşul ifadeleriyle kontrol sağlanarak çizim yapıldı ve bu çizimin ardından Shoelace formülü ile alan hesabı yapıldı. Bunun için direkt rezerv alanının üzerinde bulunan birim kareler sayılabilirdi ancak bizim için rezerv alanını küçük tutmak gerçek kâr miktarına daha yakın bir hesap yapabilmemizi sağladı.

Rezerv alanına platformlar kurulurken yine koşul ifadeleriyle kontrol yapıldı ve piksel renklerinden yararlanarak platformlar çizildi. Bu kısımda daha optimum bir sonuca ulaşmak mantık olarak mümkün ama bizim projemiz maalesef en optimum durumu hesaplayamıyor.

Sonuç olarak, projemiz başarıyla rezerv alanı çizimini ve boyamasını, platform kurulumunu ve kullanıcıya göstermemiz gereken hesaplanmış bilgileri kullanıcıya sunuyor. Projenin geliştirilebilecek yönleri olsa da başarıyla tamamlaması açısından önemli adımlar yapıldı.

<https://stackoverflow.com/questions/26991149/libcurl-c-save-information-array>

<https://www.geeksforgeeks.org/fillpoly-function-c/>

<https://www.tutorialspoint.com/fillpoly-function-in-c>

<https://www.programmingsimplified.com/c/graphics.h/setfillstyle>

<https://www.geeksforgeeks.org/getpixel-function-c/>

<https://medium.com/@sddkal/python-ile-%C3%A7okgen-alan-hesab%C4%B1-algoritmas%C4%B1-788308b666d2>

[https://www.mathsisfun.com/geometry/area-irregular-polygons.html?source=post\\_page-----788308b666d2-----](https://www.mathsisfun.com/geometry/area-irregular-polygons.html?source=post_page-----788308b666d2-----)

<http://kodegon.blogspot.com/2015/01/cde-grafik-fonksiyonlar.html>

<https://forum.donanimhaber.com/c-dili-ile-grafik-dersleri-graphics-h-quincy--89514478>

[https://www.youtube.com/watch?v=CHFyEnlMnxg&ab\\_channel=CodeWar](https://www.youtube.com/watch?v=CHFyEnlMnxg&ab_channel=CodeWar)

<https://www.youtube.com/playlist?list=PLD5D5Hj95BCFid63gy2VtVBXJZQXuw12R>

[https://www.youtube.com/watch?v=YgAo7oXeV0M&t=2s&ab\\_channel=EvilTester-SoftwareTesting](https://www.youtube.com/watch?v=YgAo7oXeV0M&t=2s&ab_channel=EvilTester-SoftwareTesting)

[https://www.youtube.com/watch?v=IZJSk8WYAQ0&ab\\_channel=Code4xU](https://www.youtube.com/watch?v=IZJSk8WYAQ0&ab_channel=Code4xU)

[https://www.youtube.com/watch?v=PfulTGqWr4M&t=800s&ab\\_channel=StudyReadEducate](https://www.youtube.com/watch?v=PfulTGqWr4M&t=800s&ab_channel=StudyReadEducate)

[https://www.youtube.com/watch?v=Bv3ErAbukbo&t=107s&ab\\_channel=HarshitSharma](https://www.youtube.com/watch?v=Bv3ErAbukbo&t=107s&ab_channel=HarshitSharma)

Programlamayı C ile Öğreniyorum / Palme Yayınevi

Projemizin kabataslak akış şeması aşağıda görülmektedir.

