



SAKARYA ÜNİVERSİTESİ

AHMAD SAFLO

G211210558

VERİ YAPILARI 2. ÖĞRETİM “B” GRUBU

1. ÖDEV RAPORU

Ödevin en önemli kısmı, birbirinin içine geçmiş iki farklı bağlı liste yapısı. Bunu bir tren katarı gibi düşündüm, ana yapı olarak çift yönlü bir bağlı liste kullandım. Kullanıcı arayüzünde yukarı ve aşağı yönde gezindiğimiz menü aslında bu listenin düğümleri. Ancak bu düğümlerin her biri, içinde başka bir veri yapısı olan tek yönlü bağlı listenin başlangıç adresini tutuyor.

İçerideki tek yönlü listelerde ise projenin asıl elemanları olan şekil nesnelerini sakladım. Burada nesneye dayalı programlamanın en önemli özelliklerinden biri olan çok biçimlilikten (polimorfizm) yararlandım. "Sekil" isminde soyut bir temel sınıf oluşturdum ve Dikdörtgen, Üçgen, Yıldız gibi sınıfları bu sınıfından türettim. Böylece listenin içinde farklı türden şekiller olsa bile hepsini aynı pointer türü üzerinden yönetebildim. Görselleştirme kısmında ise terminalde sıkça yaşanan ekran titremesi sorununu çözmek için çift tamponlama (double buffering) teknğini kullandım. Şekilleri doğrudan ekrana yazdırma yerine, önce arka planda bir matrise çizdiriyorum. Burada projenin en kritik algoritmalarından biri olan Z-index yani derinlik kontrolü devreye giriyor. Eğer iki şekil üst üste gelirse, program o pikseldeki derinlik değerlerine bakıyor ve hangisi daha üstteyse (Z değeri büyüğse) onu çiziyor. Bu sayede karmaşık şekil grupları bile ekranda doğru katman sırasıyla görünüyor.

Programın işleyişini de kullanıcı dostu olacak şekilde tasarladım. Program ilk açıldığında verileri rastgele mi üreteceğini yoksa var olan bir dosyadan mı okuyacağını soruyor. Rastgele seçiliirse sistem otomatik olarak listeleri ve şekilleri oluşturuyor. Kullanıcı "Mod 1" dediğim liste modunda W ve S tuşlarıyla ana düğümler arasında gezebiliyor. Eğer bir düğümün içindeki şekillere müdahale etmek isterse F tuşuna basarak "Mod 2"ye, yani şekil düzenleme moduna geçiyor. Burada da şekilleri hareket ettirebiliyor, silebiliyor veya değiştirebiliyor.

Ödev videolarında istediği üzere veriler kalıcı olabiliyor. Program kapatılırken arka planda çalışan bir fonksiyon, o anki tüm liste yapısını "veri.txt" adında bir dosyaya kaydediyor. Kayıt yaparken her şeklin türünü belirtmek için satır başlarına R (Dikdörtgen), T (Üçgen) gibi harfler koydum. Program tekrar açılıp dosyadan oku seçeneği seçildiğinde bu harflere bakarak doğru nesneyi yeniden oluşturuyor ve listeyi, programı kapattığımız andaki haliyle birebir tekrar inşa ediyor. Kodları daha düzenli tutmak için ödev pdfinde anlatıldığı gibi src, include ve lib gibi klasörlere ayırdım ve derleme işlemini kolaylaştırmak için bir Makefile hazırladım. Terminale sadece "mingw32-make" yazıldığında tüm proje otomatik olarak derlenip çalışmaya başlıyor.

