

1. Ödevde Genel Olarak Ne Yaptım?

Ödevin videosu ve PDF dökümanı doğrultusunda önce mimariyi kurdum. Uygulamanın temel bileşenlerini aşağıdaki şekilde oluşturdum:

- **Shape (soyut sınıf):** Tüm şekillerin ortak özelliklerine sahip temel sınıf.
- **Triangle, Rectangle, Star:** Shape'den türetilen üç farklı şekil sınıfı. Star şeklinin aslında bir eşkenar dörtgen (diamond) gibi çizildiğini videolardan öğrendim.
- **Ekran (Screen):** 25 satır \times 80 sütun boyutunda bir ASCII buffer. Tüm çizimler buraya yapıldı.
- **LinkedList:** Şekilleri tutan *tek yönlü bağlı liste*.
- **DNode ve DoublyLinkedList:** Her bir düğüm içinde bir LinkedList barındıran *çift yönlü bağlı liste*.
- **App** sınıfı: Menü yönetimi, çizimler, hareket ettirme, silme, dosya okuma–kaydetme gibi tüm uygulama akışını yönetti.

İlk aşamada mimariyi kurup şekilleri ekrana basmayı başardım. Sonrasında soldaki liste panelinin çizimi, sağ taraftaki şekillerin düzgün hizalanması, rastgele şekil üretimi ve liste geçişlerinin düzgün çalışması gibi detayları geliştirdim.

2. En Çok Uğraştığım ve Zorlandığım Kısımlar

Bu ödevde en çok zorlandığım bölümler şunlardı:

a) shapeMenu ve listMenu akışları

Seçilen şekli silme, listede son şeklin kalması, listenin tamamen silinmesi ve menüler arası geri dönüşlerin doğru index ile yapılması beklediğimden daha karmaşık oldu.

Özellikle:

- **Son şekli silince düğümün silinmesi,**
- Son düğüm silinince **liste menüsüne düzgün dönme,**
- Silinen listeye ait pointer'ın tekrar kullanılmaması,

gibi konularda çok fazla hata aldım ve uzun süre debug yaptım.

b) Sınır ihlalleri (ekran dışına taşma)

25 \times 80 ekranı aşan şekillerde kontrol yapmak zorunda kaldım.

Özellikle Star (diamond) şeklinin yüksekliği $2*h-1$ olduğu için dikey taşma kontrolü detay gerektirdi.

c) Soldaki panelin doğru çizimi

6'lı sayfalandırma, ok işaretinin doğru hizalanması, kutucukların doğru ortalanması ve şekil sayısının ortalanması için epey ince ayar yaptım.

d) Çift yönlü bağlı liste yönetimi

Her liste içinde bir LinkedList, onun içinde Shape pointer'ları ve bunların delete edilmesi sırasında double-free, dangling pointer gibi problemlerle karşılaştım.

İlk denemelerde, silinen listenin ekranda görünmeye devam etmesi veya son şeklin silinememesi gibi hatalar oluştu.

e) Dosya okuma–kaydetme

Dosyadan şekilleri doğru formatta okuyup tekrar aynı formatta kaydetmek (T / R / S + x y h ch z) dikkat gerektiren bir kısımdı.

Özellikle dosya okuma sırasında hatalı satır olduğunda yapıyı temiz şekilde sıfırlamak için ekstra önlem aldım.

3. Öğrendiklerim

Bu ödev bana hem veri yapılarının pratikte nasıl kullanıldığını hem de C++ dilindeki nesne yönelimli yapının önemini tekrar göstermiş oldu. Özellikle:

- Soyut sınıf (abstract base class) kullanımını,
- Polimorfizmin gerçek bir projede nasıl işlediğini,
- Birden fazla dinamik listeyi yönetmenin zorluklarını,
- Pointer yönetiminin ne kadar dikkat gerektirdiğini,
- Grafiksel olmayan terminal ekranında bile mimari tasarımın kritik olduğunu,

çok somut şekilde deneyimledim.

Ayrıca kodun modüler olmasının, her sınıfın kendi görevinden sorumlu olmasının neden önemli olduğunu uygulama boyunca fark ettim.

4. Eksik Bıraktığım veya Tam Yapamadığım Kısımlar

Aşağıdaki kısımlar ödevi yetiştirme sürecinde biraz zorladığım, üzerinde daha çok zamanım olsa daha iyi optimize edebileceğim bölümlerdi:

- Soldaki panelde sayfalandırma tasarımını daha okunabilir yapabilirdim.
- Z değeri sıralamasını tüm şekiller için uygulayan global çizim fonksiyonunu sadece isteğe bağlı bıraktım, sürekli entegre etmedim.
- Bazı kısımlarda kod sadeleştirilmesi ve tekrar eden kodların fonksiyonlaştırılması yapılabilir.
- Dosya formatında ek bilgiler tutulabilir (örneğin şekil türleri için daha anlaşılır etiketler).
- Ekranı daha “güzel” gösterecek border’lar veya grid yapılabilirdi.

5. Genel Değerlendirme

Birçok yerde zorlandım, hata çözdüm, menüler arası geçişleri yeniden tasarladım, pointer sorunlarını giderdim ve uygulamanın hem stabil hem de videodaki örneğe benzer şekilde çalışmasını sağlamaya çalıştım.

Sonuç olarak, hem çok şey öğrendiğim hem de C++ proje mimarisi konusunda beni geliştiren bir ödev oldu.