

# YAZ-307 Programlama Teoriği ve Pratiği 3 (C++)

## FİNAL PROJE RAPORU

**Proje adı:** Blackjack

**Amacı:** Kullanıcıya Klasik Blackjack'den farklı ve daha eğlenceli bir Blackjack oyunu sunmak

**Proje alanı:** Oyun/ Eğlence

**İşletim Sistemi:** Windows

**Proje Ekibi:**

Cemal BALABAN - 190701143

Enes Burak ÇOMRUK - 190701098

Mehmet MISIR - 190701094

Ali KIRLANGIÇ - 170701006

Muhammed Faruk ÇELİK – 180701023

**Github Linki:** [https://github.com/cmlblbn/black\\_jack\\_cpp](https://github.com/cmlblbn/black_jack_cpp)

**Görev Dağılımı:**

**Planlama, Sunum, Rapor, Veritabanı ve Bağlantıları:**

- Muhammed Faruk Çelik

**Front-End :**

- Ali Kırlangıç
- Enes Burak Çomruk
- Mehmet Mısır

**Back-End :**

- Cemal Balaban

**Proje geliştirilirken kullanılan diğer kütüphaneler :**

- QString
- QWidget
- QLabel
- QTimer
- QTime
- Stdlib.h
- QDebug

## **Proje boyunca kullandığımız teknolojiler :**

Projenin Yazıldığı Dil : C++

Kullandığımız Arayüz : QT

Kullandığımız Veritabanı: QTSQL

## **Proje Dosyaları:**

- Deck.h ve Deck.cpp
- Card.h ve Card.cpp
- Leaderboard.cpp ve leaderboard.h
- Leaderboard.ui
- main.cpp
- Mainwindow.h ve mainwindow.ui
- Playedform.ui
- Playedgame.h ve playedgame.cpp
- Playedgame.ui
- Pod.h ve pod.cpp

## **Arayüz:**

- Oyunumuz 3 ekrandan; Giriş ekranı, Oyun ekranı ve Puan Tablosu ekranı oluşmaktadır.
- Oyun Ekranımız 2 parçadan oluşur. Gösterge Yapısı ve İşlem Butonları.
- İşlem Butonları ile podlarımıza kart ekleyebilir, oyunu başlatabilir veya durdurabilirsiniz.
- Gösterge Yapısında ise Blackjack yaptıkça artan score sayacımız, Oyun başlangıcından oyun sonuna kadar geçen süre, podlara eklediğimiz kartların toplamı ve elimizdeki kart sayısı.

## **Proje geliştirilirken kullanılan yazılım geliştirme yöntemleri:**

Projemizdeki istekler ve gereklilikler önceden belli olduğu için ve proje yapım aşamasında ciddi değişikliklere gerek olmayacağı için "Geleneksel Yöntemler"i kullandık.

Bu yöntemde "Waterfall" modelini uyguladık. Yani doğrusal bir şekilde ilerledik.

Analiz → tasarım → kodlama → test → yayılma

Önce proje için gereksinimleri hazırladık ve analiz aşamasını yaptık. Burada oluşturduğumuz dokümantasyona bağlı kaldık. Analizden sonra tasarımı belirledik ve tüm takım arkadaşlarımızla belirlediğimiz görev paylaşımlarıyla kodları yazdık. Waterfall modelinde test aşaması her aşama sonrasında kullanıldığı için biz de her parça kod bitiminde ve kodların entegrasyonu sonrasında sorun yaşamamak için test yaptık. Böylece sorunlar büyümeden oluşan sorunları giderdik.

### **Proje süresince karşılaşılan zorluklar ve bu zorlukları aşmak için yapılanlar:**

- Zamanı ve Planlamayı iyi yönetemedik bu yüzden son haftaya bıraktık projemizi ve bu da bizi inanılmaz zorladı. Birçok gün 13-14 saat kod yazıp hızlıca bitirdik.
- Windows kullanmanın zorluğunu yaşadık.
  - Database ile QT arasındaki bağlantıyı kurmakta çok zorlandık. Farklı database complierları denedik. PostgraySQL'in windows üzerinde çalışmaması bu sorunu çözemedik en son sorunu qtsql complierı ile sorunu çözdük.
  - 3.parti kütüphanelerinde yaşanan uyumluluk sorunu uzun süreler internet üzerinden araştırmalar yaptık alternatif çözümler aradık.
- Grup arkadaşlarımızın bazıları github kullanmayı yeni deneyimlediği için github branchlerinde sorun yaşadık. Bu sorunu çözmeyi pek başaramadık. Bu yüzden arka arkaya branchler yüklemek yerine hep tek seferde yükledik.
- QT'nin yetersiz kaynaklarının olması, internet üzerinde çok fazla bilginin olmaması da bizi çok zorladı. Denemeler ile sorunlarımızı sıkıştığımız yerleri çözmeye çalıştık

### **Sonuçlar:**

Kullanıcının keyifli bir şekilde vakit geçirebilmesi için basit bir arayüzle BlackJack (21) oyunu hazırladık ve sunduk