

MANGALA OYUNU

Mangala oyun nedir ? Nasıl Oynanır ?

Mangala oyunu Dünya çapında onlarca farklı versiyonu olan ve çok uzun yıllardır oynanan bir oyundur. Ben Türk Mankalasını denilen Mangala versiyonunu baz aldım ve oyunu Türk Mankalasının kurallarına göre tasarladım, kodunu yazdım.

Türk Mankalasını (Mangala) 12 küçük kuyu ve hazine adı verilen 2 büyük kuyudan oluşan oyun zemininde 48 taş ile iki oyuncu arasında oynanır.

Dört temel kuralı vardır.

1. Oyunun ilerleme yönü saat yönündedir. Yani soldan sağa doğru ilerler. Hamle yapan oyuncu elindeki son taşı hazinesine bırakırsa bir oyun hakkı daha kazanır.
2. Hamle yapan oyuncu elindeki son taşı rakibin kuyularından birine bırakırsa ve bu kuyudaki taş sayısı çift sayı olursa, bu kuyudaki tüm taşları kendi hazinesine katar. Sıra rakibe geçer.
3. Hamle yapan oyuncu son taşı kendi boş kuyusuna bırakırsa bu kuyunun karşısında bulunan rakibin kuyusundaki taşları ve kendi kuyusundaki taşları hazinesine katar. Sıra rakibe geçer.
4. Kuyularındaki taşları biten ilk oyuncu, rakibinin kuyularındaki taşları da kendi kuyusuna katar ve oyunu kazanır.

Mangala oyunu modeli



Şekil 1

Şekil 1 deki oyun modelinde görülen alt bölümdeki butonlar Oyuncunun kuyuları ve sağdaki textArea onun hazinesini simgelerken, üst kısımdaki butonlar Rakip Oyuncunun kuyularıdır ve soldaki textArea ise onun hazinesini simgeler.

Yapılanlar

Öncelikle butonlarla kuyuları ve hazineleri textArea ile temsil ettiğim bir JFrame oluşturdum. Ardından mesajlaşma objesi olarak iki boyutlu dizi kullanmaya karar verdim(pit[][]). Oyun başında bu diziyi diziDoldur() metodu ile 0,4,4,4,4,4,4 / 4,4,4,4,4,4,0 şeklinde doldurdum. Rakibe mesaj gönderirken bu diziyi çaprazladığım bir metot yazdım(diziCevir()).

Ardından butonları actionCommand ile isimlendirerek her butonun actionPerformed eventinde oluşacak senaryoları belirlemek için düzenli bir şekilde switch-case yapısı oluşturdum. Bu yapıda bir butona tıklandığında ilgili case de öncelikle sıranın kimde olduğunu kontrol edip ardından sıra oyuncuda ise tıklanan butona göre buton-dizi ilişkisine göre tasarladığım gamePlay() metoduna aktarılır.

gamePlay() metodunda dizideki değer karşılığını movePit() adlı oyunun asıl kurallarının işlendiği metoda iletilir. movePit() metodunda tüm kurallara göre taşlar dağıtılır ve ardından dizi değerleri ve butonlarda görünen text değerleri güncellenir. Ardından dizi, diziCevir() metodu ile çaprazlanır ve rakibe bu çaprazlanmış olan yeni dizi (RivalsPit[]) gönderilir.

Rakip client, çaprazlamış bu yeni diziyi aldığı zaman kendi dizilerini ve butonlarını günceller. Ardından bu clientın hamle yapması beklenir. Oyun bu şekilde devam eder.

Buraya kadar anlattığım işlemler Mangala.java classında yapılıyor. Mesaj iletimi için ise bir server sürekli olarak arkada dinlemede çalışır haldedir. Oyun Clientlar arasında oynanır.

Oyun ilk açıldığı zaman bir giriş ekranı açılır. Burada kullanıcı oyun kurallarına erişebilir ve isterse oyun içi kullanıcı adını girerek oyuna giriş yapabilir. Kullanıcı adı girildikten sonra client adlandırılır, servera bağlanır ve client threadleri çalışmaya başlar. Ardından eğer hali hazırda bir client rakip bekliyorsa bu client ile eşleşir. Eğer rakip yoksa, rakip gelene kadar beklenir.

Clientlar eşlenirse eğer initial condition olarak Sclient.java da WhosTurn mesaj türünde mesaj gönderilir ve bu mesaja göre clientlara oyun sırasının kimde olduğu söylenmiş olur. Oyuncular her hamlelerinde eğer yeni bir hamle hakkı kazanmazlar ise sıra rakibe geçer. Bu şekilde oyunun sonuna kadar gelindiği zaman kazananın kim olduğu ve kaç puan ile kazandığı her iki kullanıcıya da gösterilir. Ardından her iki clientın socketleri kapatılır ve giriş sayfasına geri döndürülür. Kullanıcılar isterlerse tekrar oyuna giriş yapabilir ya da oyundan çıkabilirler.

Bir de oyunun yapımına verildiği ilk hafta başladım(githubdan görebilirsiniz.). Belirli aralıklarla eklediğim geliştirmeleri commitleyerek projeyi sonlandırdım.

Yapılmayanlar

- Butonlara numara yazmak yerine resim koymayı denedim ancak estetik bir görüntü oluşmadı ve hazinelerde 30-40 adet taş bulunabileceği için hazinelerde olası olarak 1 taşın olduğu resimden 50 taşın olduğu resme kadar png, jpeg dosyası oluşturmam gerekicekti. Bu tasarım hoşuma gitmediği için ve Samet Hoca'dan tasarımda direk sayıları kullanabilmeme dair onay aldığım için buton tasarımında resim kullanmamayı tercih ettim.
- Eşleşmiş bir client çiftinden oyun sırasında bir client çıkış yaparsa diğer client bunun farkında olmuyor. Bu kontrolü yapmadım.
- Server açıldıktan sonra örneğin bir çift client oyunu oynadıktan sonra çıkış yaptı ve yeni bir client oyuna giriş yaptığı zaman aslında yeni bir client açılmazsa kapatılan clientlardan rastgele bir tanesi sanki hala çıkış yapmamış gibi yeni açılmış client ile eşleşebiliyor. Bunun çözümü için sanırım açılan her clientın çıkışını yaparken build dosyalarını da silmem gerekiyordu.