|  |  |  |
| --- | --- | --- |
|  |  |  |

**T.C.**

**SİVAS CUMHURİYET ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

|  |  |  |
| --- | --- | --- |
|  | **BİL3011 BİL. G. GRAFİK PROGRAMLAMA**  **DAMA TAHTASI**  **ÖDEV-1**  **17/10/2023**    **Öğrencinin Adı SOYADI** |  |

**Ekim-2023**

**SİVAS**

**1 . GİRİŞ**

**Bu rapor, bilgisayar grafik arayüzü kullanılarak 8x8 boyutunda 3 farklı dama tahtasını oluşturmak için algoritma geliştirme ve uygulama sürecini göstermektedir.**

**Dama tahtalarından ilki yalnızca çizgilerden oluşturulmuş iken ikincisi ilk örnekteki çizgilerin her biri farklı renkte ve kalınlıkta çizilecek ve üçüncü örnekteki dama tahtası ilk örnekteki karelerin siyah beyaz dolgularla olan halidir.**

**Bu ödevin amacı çizgileri çizmeye yarayan metodun yalnızca bir kere kullanıldığı kodu oluşturan algoritmanın tasarımını sağlamak, farklı renk ve kalınlıkların üretiminde basit ve estetik bir yaklaşım sağlamaktır. Üçüncü örnekteki grafikte ise önceki tahtaları çizen algoritmanın esnetilmesini sağlamaktır.**

**2. YÖNTEM**

Bu ödevin konusu itibariyle kullanılması planlanan teknoloji Java programlama dilinin built-in kütüphanesi olan Swing API’dir. Bahsi geçen kütüphane ile problemimizde bahsi geçen rastgele renk ve rastgele kalınlığı belirli aralıklarla değiştirebilir ve fazla zahmet gerektirmeden basitçe bir UI ve bu UI içine de tuval yerleştirebiliriz. Kodlanma aşaması ve kodun ifade edilebilirliği açısından da Swing bu ödev için uygun bir araç olacaktır.

Swing API çalışma mantığı itibariyle bir çeşit thread ile başlar ve basit bir pencere oluşturur. Bu pencereye birden fazla tuval ekleyebilir, belirli şekilleri anime edebilir, penceredeki eventleri tespit edebilir ve dinleyebilir.

Swing API bize çizgi çekmek konusunda kolaylık sağlar:

graphics.drawLine(x1, y1, x2, y2);

Örneğin yukarıdaki metot 2D bir düzlem üzerinde iki noktanın koordinatlarını vererek düz bir çizgi çizmemize yarar.

graphics.fillRect(x1, y1, width, height);

Yukarıdaki metot ise aynı düzlemde verilen bir noktanın doğusu yönünde width kadar genişlikte ve güneyi yönünde height kadar yükseklikte içi dolu bir dikdörtgen çizer.

graphics.setStroke(new BasicStroke(width));

İkinci örnekteki dama tahtasında çizgilere kalınlık atayacak olan metot bu şekilde içerisine BasicStroke nesnesinden bir örnekle tabiri caizse elimizdeki kalemin kalınlığını değiştirir.

graphics.setColor(new Color(r, g, b));

setStroke metodunun aksine kalınlığı değil, elimizdeki kalemin rengini değiştirir. İçerisine Color nesnesinden RGB formatında bir değer ile renk alır ve kalemimizin rengini buna ayarlar.

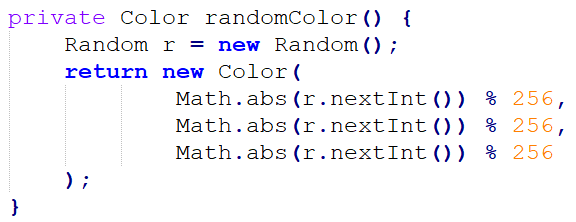
Problemin çözümünde kullanılacak başlıca araçlar bunlardır. İlk dama tahtası örneği için yalnızca basit çizgiler çekilecektir. 8x8 ölçülerinde bir tahtanın sahip olduğu yekpare düz çizgi sayısı en az 18’dir. drawLine komutunun tek seferde kullanılması arzulandığından 18 tekrarlı bir for loop çalıştırılacaktır; ancak bu komuta göndereceğimiz iki noktanın koordinatları için başlangıç değerleri atanacaktır ve her çizgiden sonra bu değerler güncellenecektir. Bu algoritmada dikkat edilmesi gereken husus şudur: soldan sağa paralel çizgiler çekildikten sonra koordinatların baştan tekrar kalibre edilmesi ve döngünün ikinci yarısında bu sefer dikey paralel çizgiler çekmesinin sağlanmasıdır.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıda görüldüğü üzere döngüdeki i değişkeni 8’e geldiğinde dikey çizgilerin tamamlandığını anlıyoruz ve koordinatları başlangıç pozisyonuna çekiyoruz.

İkinci dama tahtasında ilk algoritmaya ek olarak döngü her tamamlandığında, yani her çizgiden sonra çizginin rengini ve kalınlığını değiştireceğiz. Bu örnekteki rastgele renk üreteci için şu metot yazılmıştır:



Rastgele bir renk üretmek için her biri 0 256 arasında değer alan RGB formatındaki bir rengin R, G, B kanallarına Random sınıfından faydalanarak rastgele değerler alırız. Böylelikle rastgele bir renk üretmiş oluruz. Bu fonksiyonun yardımıyla ikinci algoritmaya eklenen şu satırlar bizim için rastgele kalınlıkta ve renkte çizgiler ile oluşturulmuş dama tahtasını gösterir.

metin, yazı tipi, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Üçüncü dama tahtasında çizgi değil kare bazlı ilerleyeceğiz. 8x8 bir dama tahtasında 64 kare bulunduğundan bu sefer algoritmamızdaki loop 64 kere dönecek ve soldan sağa 8 kare çizdikten sonra alt satıra geçecektir. Her kare çiziminden sonra kalemin rengi değiştirilecektir. Bu nedenle mod8’i 0’a eşit olan noktalar dönüm noktasıdır. Dönüm noktalarının özelliği şudur beyazla biten bir kare alt satırda yeniden beyazla başlayacağından dönüm noktalarında renk değişimi pass geçilecektir.

Siyah kalemi beyaza, beyaz kalemi siyaha çeviren bir adet switchBW(graphics); isimli bir metot oluşturulacaktır:

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

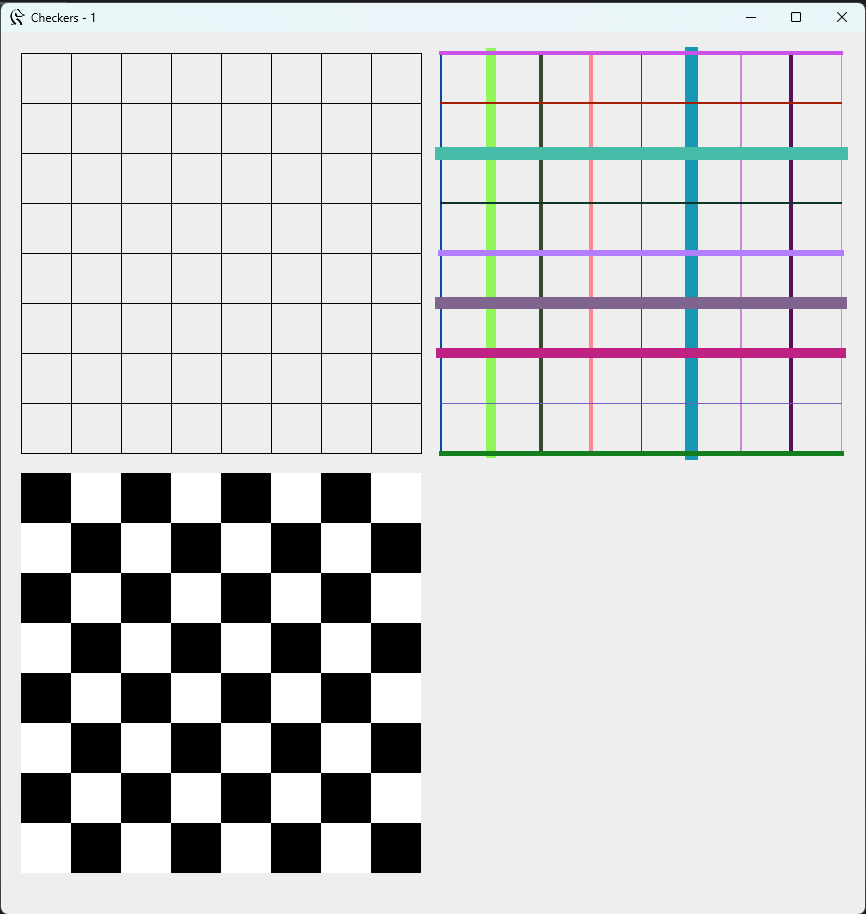
Bu metodun yardımıyla üçüncü dama tahtasının kodları aşağıdaki gibi olmalıdır:

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3. SONUÇ**

Bahsi geçen algoritmalar kullanılarak aşağıdaki görsel oluşmuştur. İkinci ve üçüncü dama tahtasının başlangıç koordinatları birinci damanın sağına e altına gelecek şekilde konumlandırılmış olduğundan tek tuvalde 3 çeşit grafiği de ekrana basmış olduk.



**KAYNAKLAR**

[1] https://zetcode.com/gfx/java2d/basicdrawing/

[2] https://www.javatpoint.com/java-swing