

PAİNT UYGULAMASI

Murat Acar ve Batuhan Meşeci

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

murat_acar_1@outlook.com ve batuhanmsci@gmail.com

Özet

Proje için “graphics” kütüphanesini kullandık. Geometrik şekilleri çizdirmek için kullanıcının fareyle geometrik şeklin köşelerini belirtmesini istedik. Bu belirtme sırasında kullanıcının ne yaptığını unutmaması ihtimaline karşın kullanıcının kullanmakta olduğu aracı sahip olduğu güncel rengeyle sağ üst tarafa yazdırdık. Bu yazdırdığımız kısım proje boyunca aktif olan seçeneği ve rengi kullanıcıya devamlı olarak göstermektedir.

Geçişleri arayüz üzerindeki buton tıklamalarının konumlarını kullanarak sağladık.

Kalem seçeneği için hazır fonksiyon olan “lineto” fonksiyonunu kullandık. Bu fonksiyonla farenin konumunu sürekli ve hızlı bir şekilde aldırarak farenin bulunduğu konumlar arasına çizgi çekmesini sağladık. Bu sayede fare hızlı da hareket etse arada boşluk kalmasını engellemiş olduk.

Silgi seçeneği için farenin konumunu merkez alan beyaz daireleri sürekli ve hızlı olarak çizdirdik.

Renk seçimini belirlemek için global bir değişken tanımlayıp bu değişkenin değerini renk butonlarına tıklandıkça güncelledik.

Ekran temizle seçeneği ekledik. Grafik ekranının çizime ayırdığımız kısmının üzerine tam oturacak şekilde içi beyaza boyalı bir dikdörtgen çizdirdik.

Kaydetme seçeneği için yazdığımız fonksiyonu grafik ekranındaki her pixeli getpixel hazır fonksiyonu yardımıyla sırayla tarayarak her pixel renginin 16’lık tabandaki karşılığını bir .txt dosyasına “fputc” hazır fonksiyonunu kullanarak aktardık.

Kayıtlı dosyayı açma seçeneği için txt uzantılı kayıt dosyamızdan “fgetc” fonksiyonuyla 16’lık tabandaki

sayıların renk karşılıklarını grafik ekranına “putpixel” yardımıyla yerleştirdik.

1. Giriş

C programlama dilini kullanarak “Graphics” kütüphanesiyle bir grafik editörü yapımı.

2. Temel Bilgiler

Projede tercih ettiğimiz kütüphane C dilinde kullanılan eski bir kütüphane olan Graphics.h kütüphanesidir. Borland Software Corporation tarafından oluşturulmuştur. Oluşturduğumuz yazılımda çeşitli geometrik şekiller çizilebilir, silinebilir kaydedilebilir ve kaydedilmiş çizimler açılabilir.

3. Geliştirilen Mimari

Grafik editörü için ihtiyaç duyulan şekillerin butonlarını kullanımları için sol tarafa dayalı arayüze yerleştirdik. Kullanıcının bu butonlara basması durumunda seçim fonksiyonu devreye girer ve tıklanan konuma göre geçilmesi gereken fonksiyonun sayı karşılığını main de bulunan switch case yapısına döndürür. Değişim sonucu aktifleşen fonksiyon sağ üste yerleştirdiğimiz kullanıcı bildirim penceresinde gözükür.

Renk seçeneklerini de benzer şekilde arayüzün üst tarafına yerleştirdik. Bu renklere tıklanması durumunda renk değerini tutan global “clr” değişkeninin değeri güncellenir ve kullanıcı bildirim penceresinde görünür. Örneğin seçili güncel renk lacivert ise clr değişkeninin de 1 değeri saklanır ve bundan sonraki çizimlerde kullanılan çizimler lacivert renginde çizilir.

Aktif olan çizim fonksiyonlar kullanıcının çizim için ayrılmış yere tıklanmasını bekler. Tıklanan konumlar gerekli sayıya ulaştığında güncel renk değeriyle bu konumlar arasına istenen şekli çizdirir.

Kaydetme işlemi arayüzün kaydet butonuna basılmasıyla aktifleşir. Aktifleştiği an ve bittiği an arasında zaman farkı olduğundan güncel durumu kullanıcı bildirim penceresinden kullanıcıya gösterdik.

Grafik ekranındaki her pixelin renginin sayı değerinin 16lık tabandaki karşılığını belirlediğimiz konumda bulunan .txt uzantılı dosyaya aktardık. Dosya açma işlemi içinde aynı dosyadan karakterleri alıp grafik ekranına tek tek yerleştirdik. Bu işlem de yapılırken aynı şekilde kullanıcıya bildirim verdik. Eğer dosya açma işlemi yapılırken beklenen konumda dosya bulunmaz ise ekrana beyaz rengi bastırılır. Dosya açma işlemi kayıt işlemine bağımlıdır.

Editörün silme işleminin rahat yapılabilmesi için temizle isimli fazladan bir fonksiyon kodladık. Bu fonksiyon çizim için ayrılmış bölgeyi sıfırlar.

Proje için harcanan saatler:

Batuhan Meşeci : Tahmini 40 saat

Murat Acar: Tahmini 25 saat

4. Sonuçlar

Arayüz: Arayüzü dükdörtgen üçgen vb birçok şeklin hazır fonksiyonlarını kullanarak [1][2][6] oluşturduk ve bu oluşturduğumuz arayüzü kontrol etmesi rahat olması için iki fonksiyona böldük. Bu fonksiyonlar renkutucukları ve butonlar. Butonlar fonksiyonunda istenilen geometrik şekilleri , dosya işlemlerini ve silgiyi barındırıyor. Butonlar arayüzü istenilen seçeneğin anlaşılabilir olması için çizimlerden oluşuyor. Renk kutucukları ise içi dolu çemberlerden oluşuyor. Programın işleyişinde herhangi bir fonksiyonla arayüz çizilemez.

Seçim: Fonksiyonlar arası geçişi sağlamak için gerekli olan bütün fonksiyonları değer döndürecek şekilde kodladık. Eğer fonksiyonda tıklanan yer arayüz ise tıklanan konum seçim fonksiyonuna gönderilir. Seçim fonksiyonunda arayüzün tıklanan bölgesine göre aktifleşmesi gereken fonksiyon seçilir. Bu seçim değeri main fonksiyona sec değişkenine atanır ve bu sayede herhangi fonksiyonlar arası geçiş sağlanmış olur. Bu geçiş sağ üste yerleştirdiğimiz kullanıcı bildirim penceresinden görülenebilir.

Renk Seçimi: Renk değişimini fonksiyonlardan bağımsız tutmak için clr adlı global bir değişken kullandık. Bu değişken güncel renk değerinin sayısal karşılığını içinde bulundurmakta. Renk kutucuklarından bir kutucuğa tıklanıldığı zaman fonksiyon olan renk çalışır ve güncel renk değerini yani clr değişkenini günceller. Bu sayede renk değişimi sağlanmış olur. Bu değişim sağ üste yerleştirdiğimiz kullanıcı bildirim penceresinden görünebilir.

Çizgi : Cizgi fonksiyonunu kullanabilmek için graphics.h kütüphanesinde hazır olarak bulunan line fonksiyonunu temel aldık. Bu fonksiyon için gerekli olan konum değerlerini almak için farenin tıklanıldığı zamanki iki konumunu istedik. [5] Farenin konumunu sürekli olarak alırken her alımdan sonra tıklı halde olup olmadığını kontrol ettik. Farenin her sol tuşuna tıklanıldığında konum değerlerini line fonksiyonunun gerekli yerine atadık. Her 2 konum alışımda tıklanan yerler arasına çizgiyi çizdirdik.[1] Çizginin arayüzü çizmesini engellemek için tıklanan konumun arayüzde olduğu durumlarda arayüzdeki işlevini yerine getirmesini sağladık.

Dikdörtgen: Seçim fonksiyonundan gelen geri dönüş değeri dikdörtgen fonksiyonunu çağırmak için kullanılan değere eşitse çalışmaya başlayan dikdörtgen fonksiyonunun çalışımında dikdörtgen çizdirmek için hazır bulunan rectangel fonksiyonunu kullandık[2]. Çizgi fonksiyonuna benzer olarak farenin konumunu devamlı olarak alırken tıklı olup olmadığını kontrol ettik. Tıklandığı andaki fare konumlarını rectangel fonksiyonunun konum gerektiren yerlerine atadık. Eğer tıklanan konum arayüzün bir parçasıysa dikdörtgeni çizdirme fonksiyonunu sonlandırıp , arayüzün tıklanan konumundaki işlevini çalıştırmasını sağladık.

Çember : Çember fonksiyonunun hazır fonksiyonu yarıçap alıyordu. Tıklanan konumları diğer fonksiyonlarla aynı yöntemle aldık. Bu alınan konumlar arasındaki uzaklığı 2 nokta arası uzaklık formülü sayesinde hesaplayıp gerekli yere atadık. Bu sayede kullanıcıdan alınan 2 konum arasına tam gelecek şekilde bir çember çizilebilir oldu. [3] Çizilen çemberin yan kısımları belli durumlarda arayüzün üstüne gelebildiğinden dolayı çember çiziminden sonra arayüzü tekrar çizdirdik. Bu sayede arayüzün üstü çizilse bile kullanıcı tarafından farkedilemez hale geldi. Her çember çiziminden sonra arayüz tekrar çizildiğinden çizim aşamasında kullanıcı arayüzün yok olduğunu birkaç milisaniyelikliğine görebiliyor. Ve tekrar çizimden sonra çizgilere dikkatli bakıldığında bir kalınlık farklı gözlenebiliyor. Çemberin konumlarının alınması sırasında konumlardan biri arayüzün bir parçasının olduğu konumdaysa çember işlemi iptal edildiğinden kaynaklı herhangi çizim yaşanmaz.

Üçgen: Üçgen fonksiyonunun hazır kullandığımız kütüphane hazır fonksiyonu olmadığından line fonksiyonunu baz alarak kendimiz oluşturduk. Kullanıcıdan diğer seçeneklerde kullanılan aynı yöntemle 3 konum alıp bu konumların her birinin arasına çizgi çizdirdik. Diğerleri gibi üçgen fonksiyonunda da konum olarak arayüzde bir nokta kullanmak mümkün değildir. Tıklanan fonksiyona geçer.

Fırça: Fırça fonksiyonu oluşturabilmek için devamlı olarak fare konumu almamız gerekiyordu.[5] Tıklıyken çizmek yaygın bir kullanım olsa da touchpad lerde çizmenin zorlayıcı olduğunu düşündük. Bu yüzden ilk tıklamadan sonra fırçanın son boyduğu konumdan farenin günzel konumuna lineto fonksiyonuyla çizgi çekmesini sağladık Çizgi çektiği için arada boşluk kalma durumu engellenmiş oldu ancak bir başlangıç konumu lazımdı. Başlangıç konumu belirtmek için ise farenin sadece ilk tıklamasında

çalışmak üzere moveto fonksiyonunu kullandık. Boyama durumunun duraklatılması için farenin sağ tuşuna basmayı uygun gördük. Çizime kaldığı yerden devam etmesi için kullanıcının tekrar sol tuşa basması yeterlidir.

Fırça devamlı olarak konum alıp boyadığı için bu süreçte arayüze girebilir ancak arayüze girdiği yerdeki fonksiyona geçiş yapılacağı için arayüzün üzerini boyayamaz. Bu durum arayüze tıklanmadan fonksiyon geçişine sebebiyet veriyor. Aynı şekilde renk değişimi de yaşanmakta.

Silgi: Silgi seçeneğinin çalışabilmesi için arkaplana boyamasını en kolay çözüm olarak gördük. Bu boyama işlemi için fırçada kullandığımız gibi devamlı olarak farenin konumunu aldık ve bu konumu merkez alan bir çember çizdirdik.[7] Bu çemberin içini boyayarak istediğimiz daireyi elde etmiş olduk. Ancak dairenin çizim hızı farenin hareket hızından düşük olduğu için hızlı hareket sonucu silinen kısımlar arasında boşluk oluşabiliyor.

Kullanıcının tüm ekranı silmeyi istemesi durumunda silgiyle vakit kaybetmemesi için temizleme fonksiyonu oluşturduk.

Temizle :Bu fonksiyon sayfanın çizim için ayrılmış beyaz kısmını sıfırlamak yerine aynı kısmın üstüne sınırları aynı olan içi dolu beyaz dikdörtgen çizdirip temizlenmiş hissi uyandırıyor. Sadece çizim alanını tekrar boyadığından arayüze yada çizimlere herhangi etkisi olmuyor.

Kaydet: Ekrandaki çizimleri kaydedebilmek için grafik ekranındaki her pixelin rengini bir dosyaya kaydetmeye karar verdik. Renklerin hali hazırda sayısal karşılıkları olduğundan hepsini getpixel [8] fonksiyonuyla alıp txt uzantılı dosyamıza putpixel [9] fonksiyonuyla yazdırdık. 2 basamaklı sayılarla uğraşmak istemediğimizden ve 16 farklı renk var olduğundan bütün numaraları 16lık tabandaki karşılıklarıyla kayıt ettik. Örneğin alınan pixelin rengi beyazsa getpixelden geri dönen değer 15dir ve bu değer txt dosyasına F olarak kaydedilir. Bu kayıt işlemi biraz zaman aldığından başlarken ve biterken sağ üste yerleştirdiğimiz kullanıcı bildirim penceresinden bildiri verdik.

Dosya: Bu fonksiyonda kaydet fonksiyonuna benzer mantıkla çalışmakta. Yine iki tane for döngüsü içinde bir switch case yapısı mevcut ama fark olarak switch in içinde dosyadan bir karakter okumaya yarayan "fgetc" fonksiyonu bulunur. Case lerde ise graphics.h kütüphanesinde hazır olarak bulunan "putpixel"

fonksiyonu ve verilen pixel koordinatına belitilen rengi sıralı olarak koyulduktan sonra [9] grafik ekranı çizilmiş olur. Dosya açma işlemi de kayıt işlemi gibi zaman aldığından aynı şekilde sağ üst taraftaki pencereye bildirimler yerleştirdik. Açılmaya çalışılan dosya bilgisayarda yok ise ekranı beyaza boyar.

Kaynakça

[1] <http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html>

[2] <https://www.cs.colorado.edu/~main/bgi/doc/rectangle.html>

[3] <https://www.cs.colorado.edu/~main/bgi/doc/circle.html>

[4] <http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html>

[5] <http://forum.ceviz.net/t/graphics-h-ve-fare.85742/>

[6] <https://www.cs.colorado.edu/~main/bgi/doc/index.html>

[7] <https://www.cs.colorado.edu/~main/bgi/doc/fillellipse.html>

[8] <https://www.cs.colorado.edu/~main/bgi/doc/getpixel.html>

[9] <https://www.cs.colorado.edu/~main/bgi/doc/putpixel.html>

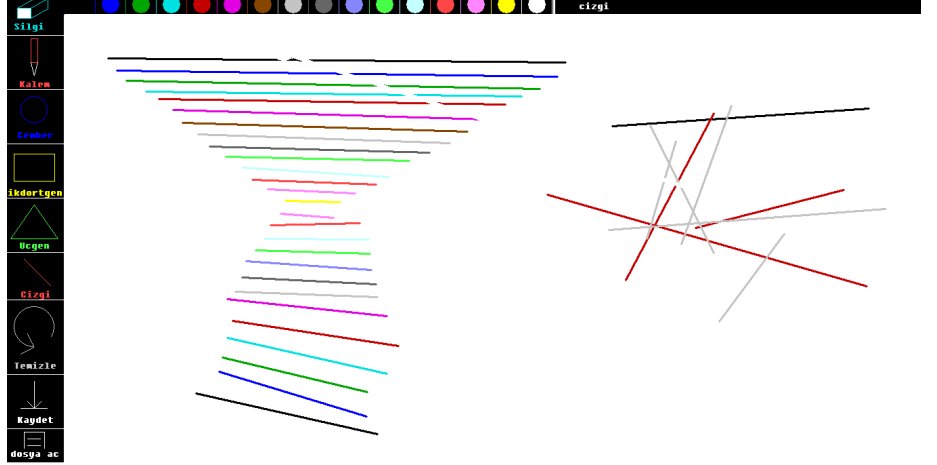
[10] <https://tr.scribd.com/doc/44093291/Graphics-in-C-Language>

[11] <http://www.techcrashcourse.com/2015/08/c-graphics-programming-tutorial.html>

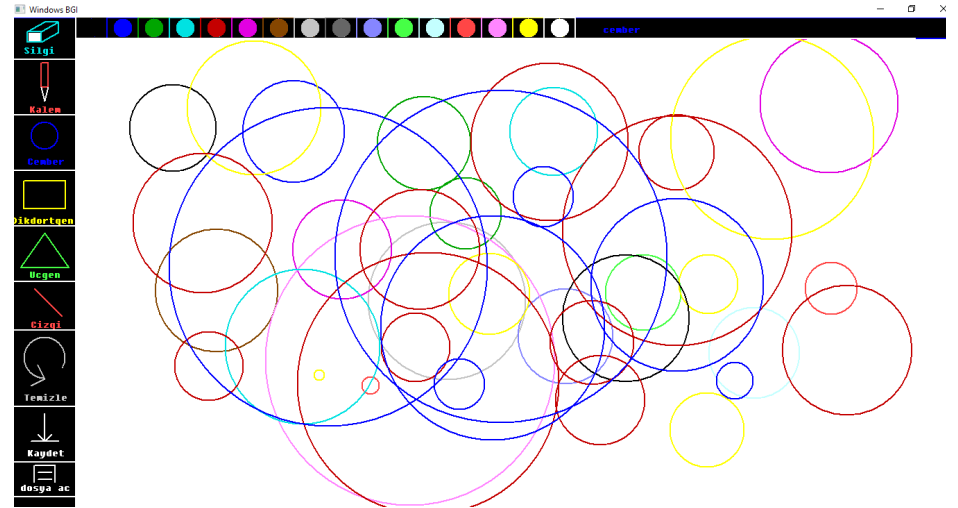
[12] codershunttest.blogspot.com.tr/2013/07/drawing-symbols-lines-polygons-shapes.html

Kullanıcı Kataloğu

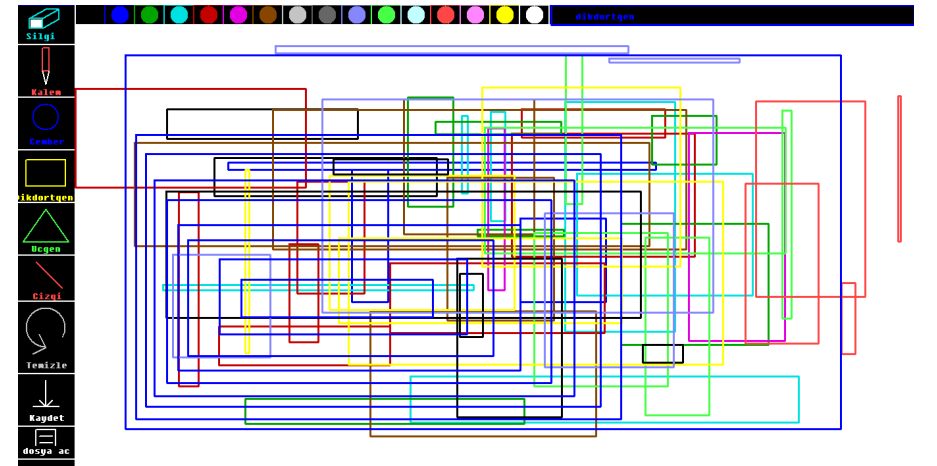
Çizgi: Çizgi butonunun üstüne tıkladıktan sonra çizim için ayrılmış olan beyaz bölümden çizmek istediğiniz konuma tıklayınız. 2. tıklamanızda çizginiz daha önce seçili olan renkle çizdirilecektir.



Çember : Çember yazılı butona tıkladıktan sonra çemberin çizilmesini istediğiniz aralığı tıklayarak belirtiniz.

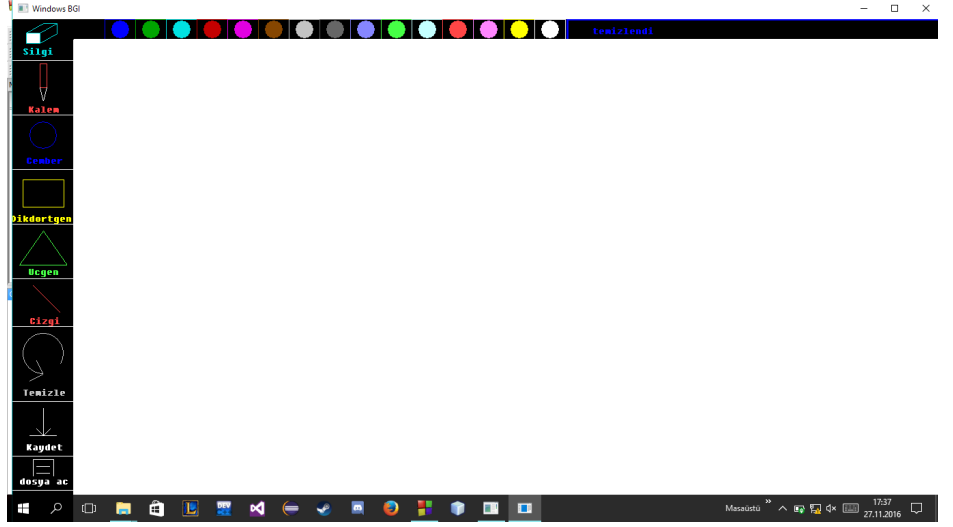


Dikdörtgen : Dikdörtgen yazılı butona tıkladıktan sonra dikdörtgenin köşelerinin bulunmasını istediğiniz yerlere tıklayınız



Renk Seçimi : Renk değiştirmek için üst taraftaki renk kutucuklarını kullanınız. Güncel rengi görmek için sağ üstteki bildirim penceresine bakabilirsiniz.

Ekrana Temizleme: Ekranı sıfırlamak için sadece temizle butonuna basabilirsiniz. Bildirim penceresinden temizlendiğine dair uyarı alacaksınız.



Üçgen : Üçgen şeklini çizmek için üçgen butonuna tıkladıktan sonra çizilmesini istediğiniz yere tıklayarak üçgeni çizdirebilirsiniz. Çizim alanına 3 kere tıklanmadıkça üçgen çizilmez.

