

## ÖNEMLİ NOTLAR

- Ödev C++ / OpenGL kütüphaneleri kullanılarak yazılmıştır.
- Programdaki tüm aşamalar bu rapordaki sıra ile yapılmalıdır. (Draw1 aşamasından sonra polygon noktaları değiştirilemez. Draw 3 fonksiyonu run time çalışmaz.)
- Aynı anda iki polygon çizim yöntemi kullanılamaz.
- Polygon çizimi için 3 yöntem mevcuttur.
- Mouse ile çizim yönteminde menüdeki “save text” çalışmaktadır.
- Program ekranı üzerinde sağ tık ile ulaşılabilecek tek menü bulunmaktadır.
- Program draw1(polygon+q\_noktası), draw2 ve draw3 olarak aşamalı çalışır.

### Polygon çizimi için;

**İlk yöntem,** direkt olarak polygon köşelerini ekran üzerinde mouse ile tıklayarak belirlemedir. Soruda belirtilmemesine rağmen hızlı test yapılabilmesi için eklenmiştir. Bu yöntemde polygon noktaları güncel olarak program tarafından belirlenen nokta sırasıyla tıklandıkça (run time olarak) çizilir. Son noktada belirlendikten sonra ilk nokta ile birleştirmek için menüden “line loop” seçeneğine tıklanır. Bu polygon kullanıcı tarafından çizildiğinden dolayı overlap durumu program tarafından kontrol edilmez. Ardından tıklanacak nokta q noktasını belirler. q noktası her tıklamada güncellenir.

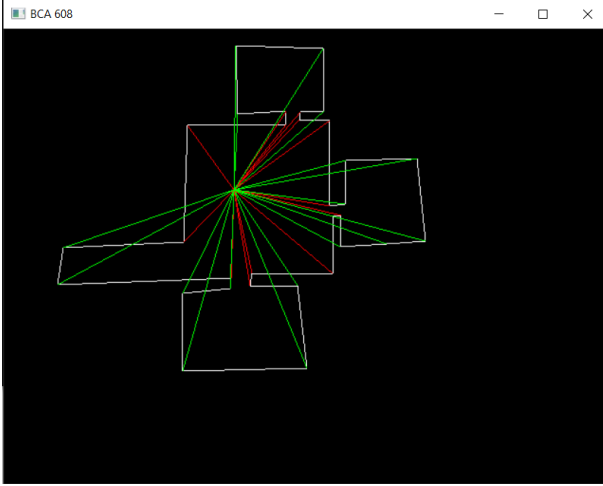
**İkinci yöntem,** menüden “load text” seçeneğine tıklanarak text dosyasından polygon noktalarını ve q noktasını okumaktır. Bu yöntemde yine program polygon çizgilerinin overlap durumunu kontrol etmez. Okunacak text dosyası aynı dizinde olmalı ve adı example.txt şeklinde olmalıdır. Bunun sebebi program konsol c++ üzerinden yazılmıştır file browser yoktur. Dizin sadece konsol üzerinden girilmesi uzun olacağından bu şekilde yapılmıştır.

**Üçüncü yöntem,** menüden “random polygon” seçeneği ile rastgele bir polygon çizdirmedir. Polygon köşe sayısı 4 den az olamaz. Bu aşamadan sonra yine kullanıcı mouse ile q noktasını seçmelidir. q noktası ilk yöntemdeki gibi yeni bir yere tıklandıkça güncellenir.

Bu üç yöntemden biriyle polygon çizdirme ve q noktası belirlenir ve menüdeki draw2 fonksiyonu çalıştırılır. Menüden draw2 seçeneğini aktif hale getirilmelidir. Bu aşamada q noktası için run time olarak tekrar draw2 hesaplaması yapılır.

Draw2; hesaplaması q noktasından tüm polygon köşelerine giden doğrular çizer. Bu doğru verileri programda ray\_lines değişkeninde saklanır. Ray\_lines değişkeninde bulunan çizgiler eğer polygon çizgileriyle köşe dışında polygon çizgisi üzerinde kesişiyorsa bu çizgiler yeşil ile çizilir ve draw3 fonksiyonunda diziden atılır.

Draw3; hesaplaması herhangi bir polygonun kesmediği, köşelere ulaşabilen ray line çizgilerinin nerede tamamlandığını hesaplar. Ray\_line devamında başka bir noktada polygon line ile overlap olup olmadığı kontrol edilir. Köşeden devam eden ışın eğer başka bir noktada polygon ile kesişiyorsa ters kesişme kontrolü yapılır. Ters kesişme kontrolü ray\_line doğrultusunu kontrol eder. Eğer kesişen nokta q noktasına ray\_line ucundan daha yakınsa ray\_line ters yönde bir polygonu kesmiştir. Ters kesişme durumu yoksa ve yeni bir polygon çizgisi ile kesişmişse o ray\_line uç noktası son kestiği polygon noktası olarak güncellenir. Draw3 ile çizilen ray\_line uzantıları mavi çizgi ile gösterilir.



BEYAZ ÇİZGİLER

polgon\_lines

KIRMIZI ÇİZGİLER

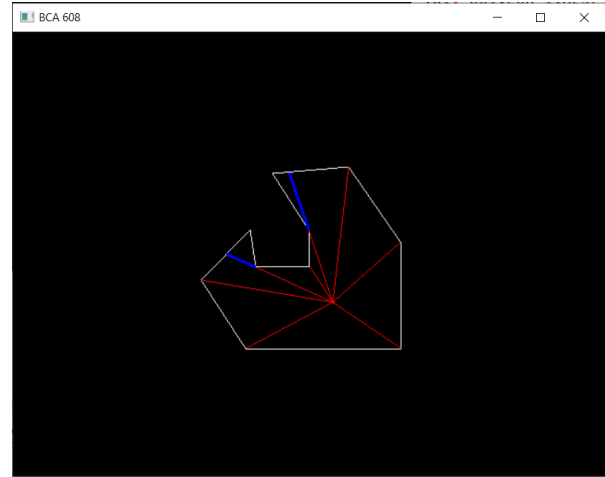
sadece polygon köşelerini kesen ray\_lines

YEŞİL ÇİZGİLER

polygon çizgilerini kesen ray\_lines

MAVİ ÇİZGİLER

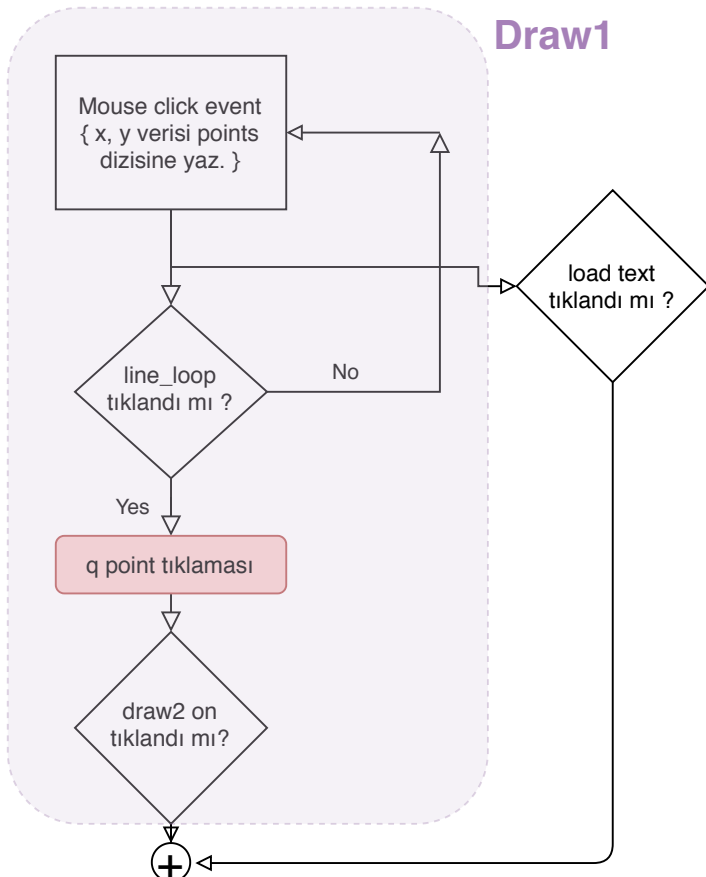
köşeden sonra başka bir polygon çizgisini kesen ray\_lines



\* Program random polygon çiziminde overlap durumunu kontrol etmemektedir.

\* Arka arkaya olan polygon çizimlerinde doğru çalışmamaktadır.

## Draw1



## Açıklamalar

1.Koşul belirten yapılarda çıkan aşağı ve yukarı yönlü oklar koşulun gerçekleştiği durumunda gidilen işlemi gösterir.

2.Sağ veya sol yönler ise gerçekleşmediği durumda gidilen işlemi gösterir.

3.Koşul belirtilen şekillerde "tıklandı mı ?" şeklinde olan yazılar sağ tık menüsünde bulunan seçeneklere değinir.

## Draw2

