|  |
| --- |
| C:\Users\de\Desktop\yazlab2_proje1\kocaeli_universitesi.jpg  **ÖĞRENCİ**  **AD SOYAD: UĞURCAN ERGÜN 090202003**    **HAZAL TÜRKMEN 090202036**  **BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ**  **DERS ADI:YAZILIM LAB2**  **PROJE NO:1**  **KONU:VERİLEN KAPLUMBAĞ ÇİZİM DİLİMİ İLE OPENGL KÜTÜPHANESİ ARACIĞLIYLA GRAFİK ÜZERİNDE GÖSTERMEK** |

*ÖNSÖZ*

*Verilen kaplumbağ çizim diliminin grafik olarak uygulaması yapılmıştır. Buton ve dışarıdan girilen komutlar sayesinde 10x10 luk ızgara üzerinde kaplumbağnın haraketleri görülmektedir.*

**1.GİRİŞ**

Yukarıda da verildiği gibi hazırlanan program dışarıdan verilen komutlara göre kaplumbağnın hareketlerinin grafik üzerinde gösterimini yapmaktadır.

**2.PROBELME ÇÖZÜM VE UYGULAMA**

**2.1Probleme çözüm yaklaşımı**

Amaç dışarıdan alınan komut bilgisi ile kamplumbağnın ızgara üzerinde çizim dilimini göstermek. Ayrıca yapılan hata kontrolleri ile kullanıcaya uyarı vermek. Hazrılanan bu programda komutlar yazılarak ve buton yardımı ile alınmaktadır. Hatalarda messagebox ile kullanıcıya görüntülenmektedir.

Kullanıcı 3 buton sayesinde kamplubağya hareket ettirebilir ve ayrıca dışarıdan textfield e kodu yazarakda çizimi görebilir. Kaydet butonu sayesinde de ekran görüntüsünü kayıt yapabilir.2 sınıf tasarlanmıştır.biri arayüz için diğeri ise arayüzdeki işlemleri gerçekleştiren fonksiyonlar içindir.matematik fonksiyonları ile koordinatlar hesaplanmıştır ve bu koordinatlar sayesinde çizim gerçekleştirilmiştir.

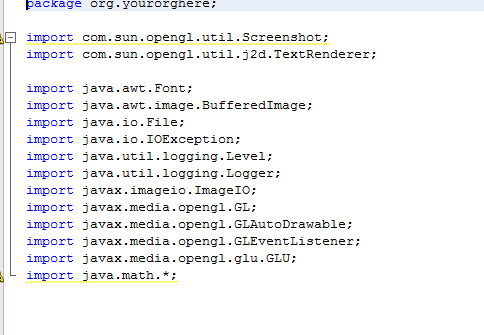
**2.1 Programın Çalıştırılması**

Yazılım netbeans ve eclipse üzerinde hazırlanmış ve denemeleri yapılmştır.2 classdan oluşmaktadır.SimpleGLPanel ana main sınıfını içermektedir.programın çalışması için derleyicilere opengl eklentisinin eklenmesi gerekmektedir. Kullanımı butonlar ve textfield üzerinden yapılmaktadır. Kaydet butonuda ekran görüntüsünün kaydedilmesi için kullanılmıştır.

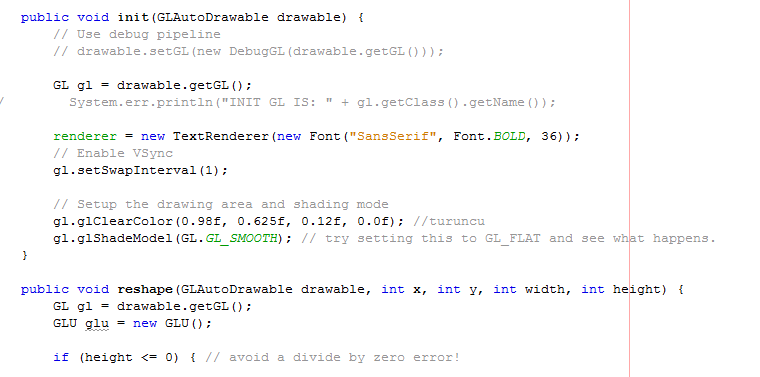
**2.2 Yazılım Mimarisi**

Proje 2 sınıftan oluşmaktadır.Glenderer.java ve SimpleGLPanel.java

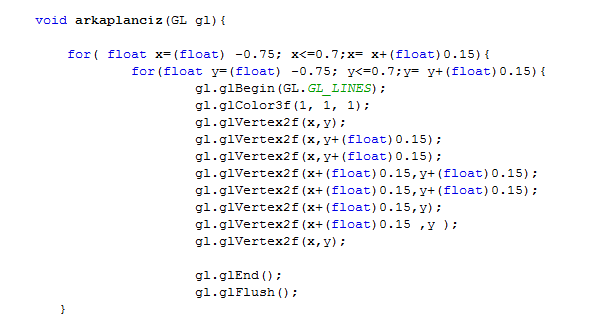
Glenderer.java sınıfı



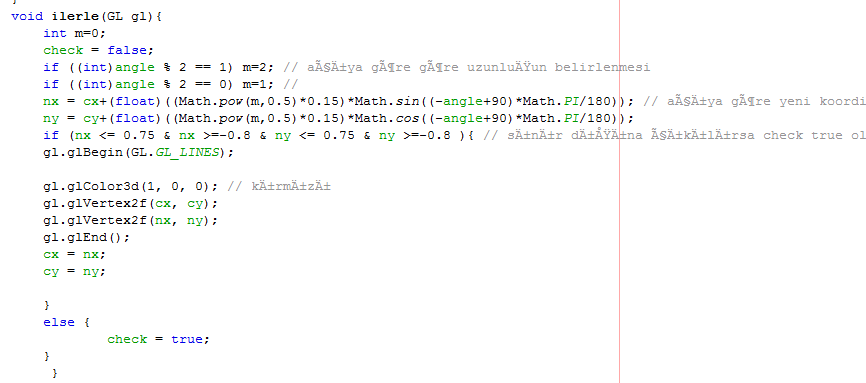
Yukarıda görüldüğü gibi opengl paketleri ve jarları derleyiciye yüklenmiştir.netbeans için opengl plugi indirildi ve eclipse içinde jarları eklendi.



Opengl çalışmaya başladığında ilk çalışan fonksiyonlardır. Programın ilk çalışmaya başladığı andaki görünümün düzenlenmesini sağlar.



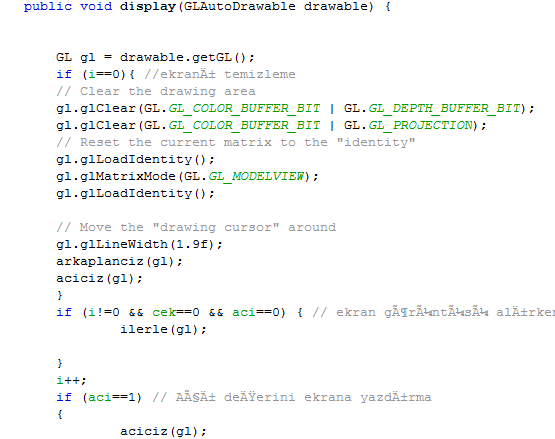
10x10 luk ızgara diliminin çizimi sağlanmaktadır. Opengl de çizimler begin ve end blokları arasında yapılır. Iç içe for döngüsü sayesinde 10x10 luk kareler oluşturulmaktadır.



Ilerle fonksiyonu yapılan açıyı hesaplar ve böylece koordinatları elde etmiş oluruz.sinüs ve cosinüs fonksiyonlar java radyanlı bir şekilde hesaplanmak zorundadır. Nx ve ny değerleri yeni elde etmiş olduğumuz koordinatlar cx ve cy değeleri ise ilk koordinatlarımızıdır..if kontorlu ilede çizimin dışarı taşıp taşmadığının kontrolunu yapmış oluyoruz.

Ilk başta derece olarak denenmiş fakat hatalı sonuçlar üretilmiştir. Bu sebebten dolayı radyan olarak hesaplanmıştır. Javanın math kütüphanesi double döndürdüğü iiçin float olarak tyepcasting yapılmıştır.

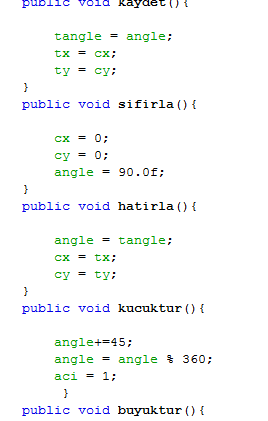
Glcolor3d fonksiyonunun içinde değişkenler sayesinde çizimimizi kırmızı renginde gercekleştiriyoruz.çizim yapıldıktan sonra arda arda çizimleri gerçekleştirmek amacıyla yeni çizimleri ilk koordinatlara atamasını yapıyoruz



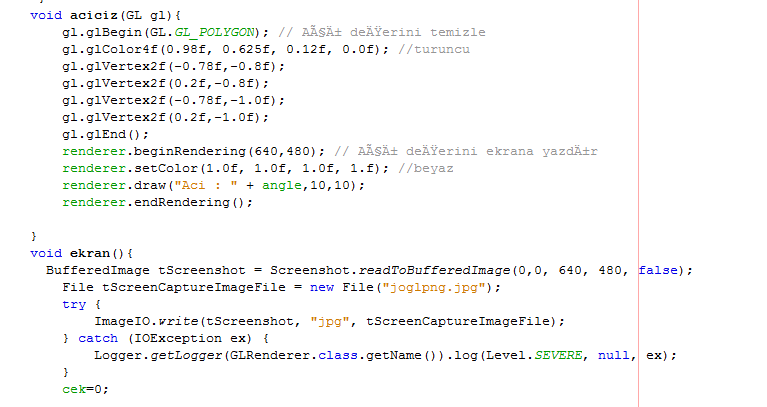
Display metotu çizim metotları çağırmak için kullanılır. Bu fonksiyonda yazılan ilk komutlar rengin sıfırlanması var mode un kaydedilmesidir. Ekrana bir kere 10x10 luk ızgaranın çizdirilmesi için kontrol ve resim kaydı için kontroller yapılır. Ilerle ile kaplumbağnın hareketleri ekrana çizdirilmiş olunur.

Opengl de çizimler display metodundan çağrılır. Bundan dolayı butonların içinde display fonksiyonunu çağırmada sorun yaşanmıştır. Çağrımı panel.display şeklinde yapılarak sorun giderilmiştir.sıfırla metodu projenin daha modüler olması için yazılmıştır ve her yeni bir çizim yapılacağı zaman koordinatlar sıfırlanmaktadır.

Aci cizimini yapmak içinde aci değişkeni kullanılmıştır.



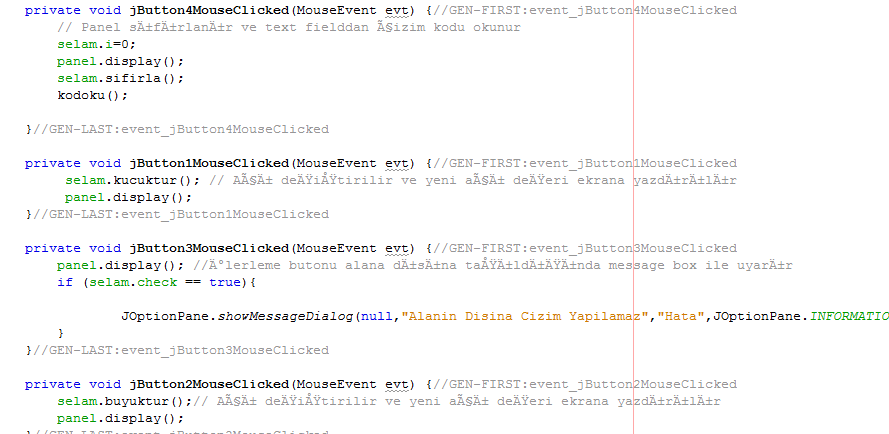
Kaydet ve hatırla fonksiyonları [ ] komutu içindir.[ girildikten sonra koordinatlar ve açı kaydedilir ] yapıldıktan sonra o konuma gidilir. Kucuktur ve buyuktur fonksiyonlar açı hesaplamasında kullanılmıştır.



Aciciz ekrana hesaplanan acinin görüntüsünü göstermek için kullanılmıştır.

Ekran fonksiyonuda ekran görüntüsünün jpeg formatında kaydedilmesini sağlar. Bunun için opengl in screenshot sınıfını kullanılmıştır.ekranımızın boyutuna göre değişkenleri belirterek resmimizi projenin bulunduğu dizine kaydediyoruz.

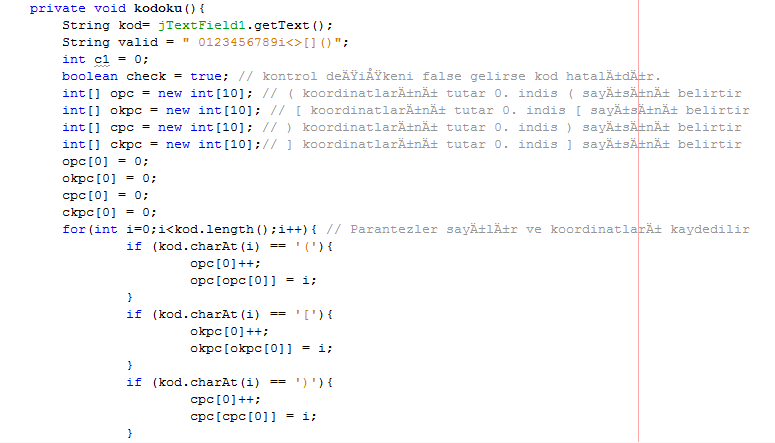
SimpleGLPanel.java



Git butonu için paneli sıfırlıyoruz bu yüzden bunu kontrol eden I değişkenini 0 a eşitliyoruz daha sonra panel.displayi çağırdıktan sonra sonra bir baska çizim yapmak çin koordinatları sıfırlıyoruz ve kodoku fonksiyonunu çağrıyoruz.

Butonlara yazdığımız fonksiyonları yerleştiriyoruz. Bunun için sınıfımızı oluşuturuyoruz ve bu sınıfa ait fonksiyonları butonlar içinde çağrıyoruz.çizimi göstermek içinde display metodunu çağryoruz.

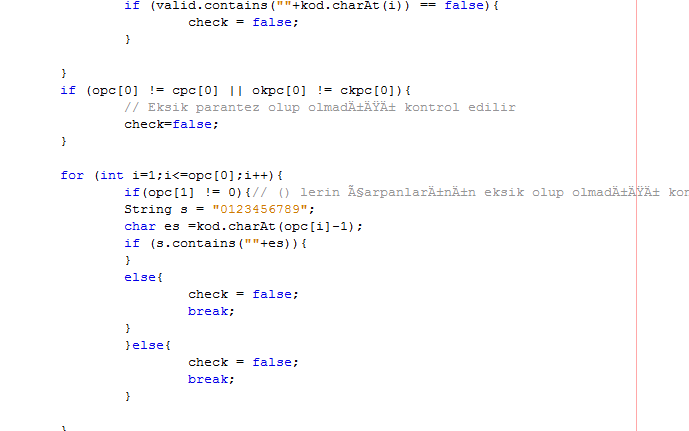
Eğer alanın dışında bir çizim yapılırsa message box ile kullanıcıya hata mesajı verliyor.



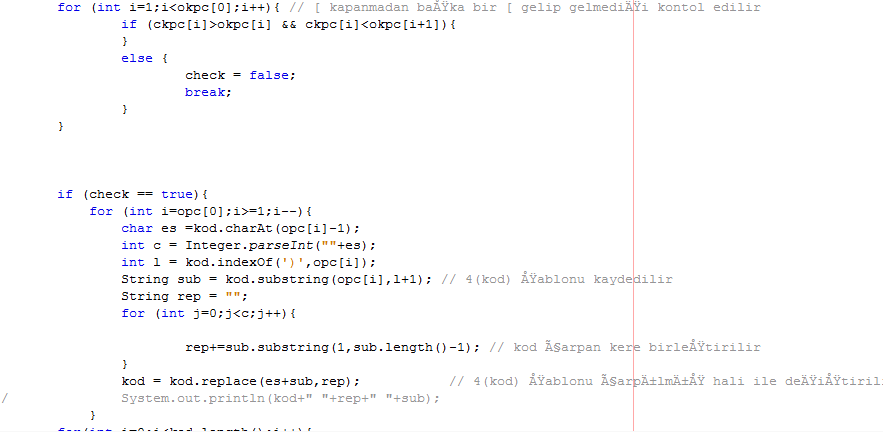
Kodoku fonksiyonu parantez kontrollerinin yapılmasını sağlar.textfielden kod okunur.eğer kontrol değişkeni hatalı gelirse kod hatalıdır.

Opc dizisi ( komutunun koordinatlarını tıtar ve 0. Indisi ( komutunun kaçtane olduğunu belirtmek için kullanılır. Okpc dizisi ) komutunun koordinatlarını tıtar ve 0. Indisi ) komutunun kaçtane olduğunu belirtmek için kullanılır. cpc dizisi [ komutunun koordinatlarını tıtar ve 0. Indisi [ komutunun kaçtane olduğunu belirtmek için kullanılır. ckpc dizisi ] komutunun koordinatlarını tıtar ve 0. Indisi ] komutunun kaçtane olduğunu belirtmek için kullanılır.

Eğer girilen kodda komutlar varsa o kumta ait dizinin 0.elamanı arttırılı ve böylece kodun içinde kaçtane komutun olduğu sayılmış olunur. Sayılan komutun kodun kaçıncı sırasında oluğu bilgisi diziye eklenir.



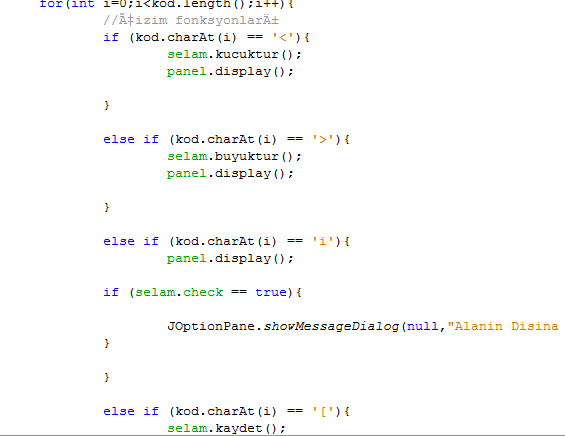
Parantezlerin kapanıp kapanmadığı kontorlu ve () parantezin önündekinin sayı olup olmadığının kontorlu yapılır.



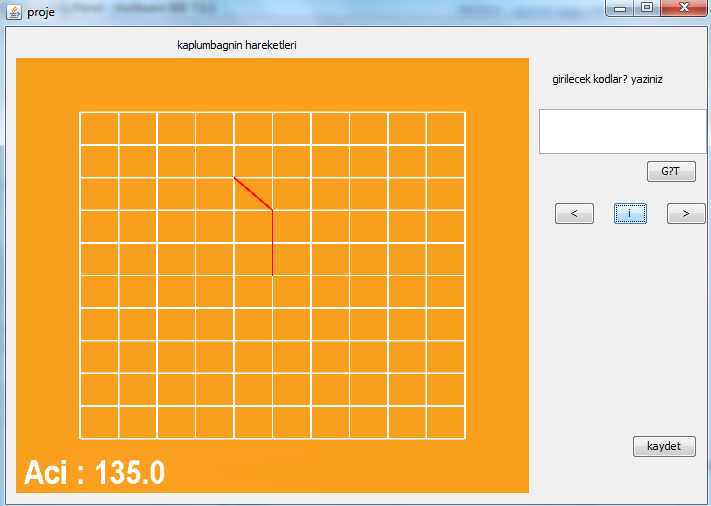
[ kapanmadan baska [ komutunun gelip gelmediğinin kontorlu yapılır.

Bütün kontroller doğru ise check true olur ve tekrarlamalı kod çalıştırılır. Bunun için opc dizisinden ( komutunun yeri belirlenir ve ( komutunun önündeki sayı kez sub string komutu sayesinde string olarak komut ard arda birleştirilir.Önündeki sayı string olduğundan dolayı parse işlemi yapılır ve string integer a dönüştürülür. Orjinal kod oluşturulan yeni kod ile değiştirilir ve bu komut işletilir.

Kodun sonuna kadr for döngüsü ile gidilier ve if kontorlleri yapılarak sınıfa ait olan fonksiyonlar çağrılır.



**2.3 Yazılımın çalıştırılması**

****

**2.3 KAYNAKÇA**

1. <http://stackoverflow.com/>

2. <http://www.opengl.org/documentation/>

3. <http://www.belgeler.org/howto/opengl-giris.html>