

**T.C.**

**GEBZE TEKNİK ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİL 395 PROJE I**

**GRUP 1**

**PROJE ÇALIŞANLARI:**

Ali Ozan TAÇALAN

Emrullah GENÇOĞLU

Ekber AZİZ

Serap UÇUŞ

Hasan BİLGİN

Şahin EĞİLMEZ

Oğuz Erdi ERKMEN

**DANIŞMAN:** Prof. Dr. Erkan ZERGEROĞLU

**İÇİNDEKİLER**

**1. GİRİŞ** 3

**1.1.** **Projenin Tanımı** 3

**1.2.** **Projenin Amacı** 3

**1.3.** **Proje Kapsamı ve Kriterleri** 3

**2. PROJE DİYAGRAMI** 4

**2.1 Donanım Modülü** 4

**2.1.1 Proje malzemeleri** 4

**2.1.2 Donanim Montajlama** 7

**2.1.3 Plaka Dengeleme Sisteminin Mekanik Tasarımı** 9

**2.2 Yazılım Modülü** 10

**2.2.1 Kullanılan Yöntem Ve Kütüphaneler** 11

**2.2.2 Genel Algoritma** 12

**2.3 Arayüz Modülü** 13

**2.3.1 Modellemede Kullanılan Program** 13

**2.3.2 Kullanıcı Ekranı** 13

**2.3.3 Modelleme** 13

**3. GÖREV DAĞILIMI** 15

**4. REFERANSLAR** 16

# **1. GİRİŞ**

# **Projenin Tanımı**

Bu proje zemini düz bir yüzey üzerinde topun dengede durmasını sağlayacak mekanik düzeneği ve bu hareketleri bilgisayar üzerinde 3 boyutlu gösterimini sağlamaktır. Tasarlanacak olan düzenek topun konumun kontrol edecek ve bu konuma göre yapması gereken hareketleri yapacaktır. Topun dengede durması için gerekli hesaplamalar topun anlık konumuna göre belirlenmiştir.

# **Projenin Amacı**

Düzenekte düzlem olarak kullanılacak dokunmatik ekran üzerinden anlık konum bilgileri alınacak ve topun dengede durması için dokunmatik ekranın bağlı olduğu düzlem gerekli yönlere doğru hareket etmesini sağlamaktır. Bu hareketleri sağlamak için bir Arduino Mega 2560 [1] ve 2 tane Servo motor kullanılmıştır. Servo motorlar düzlem hareketini sağlayacak Arduino ise motorlar için gerekli enerjiyi sağlayacaktır. Top, dokunmatik ekran üzerinde gerekli hareketleri yapıp düzlem üzerinden düşmemektedir. Bu hareketler eş zamanlı olarak Windows ve Linux platformlarında OpenSceneGraph [2] kullanılarak hazırlanan üç boyutlu modelleme ile etkileşime sahiptir.

# **Proje Kapsamı ve Kriterleri**

Proje kapsamında 38 mm ve 24 mm çapa sahip metal iki top ile test edilen dengeleyici düzlem donanımsal olarak birleştirilmiştir. Bu durumu gerçekleyebilme adına dokunmatik ekran üç boyutlu uzayda olası her yöne salınım hareketi yapabilerek topu dengede tutmaktadır. Bu kapsamda dengeleyici işlemleri yapan algoritma yazılmıştır. Algoritmalar topun hangi yöne hangi şiddette gitmesi gerektiğini hesaplayıp sonuçlar Arduino Mega 2560 ile bilgisayar ortamına iletilmektedir. Bilgisayar üzerinde düzenli olarak konum verileri alınıp aynı zamanda Arduino Mega 2560’a talep edilen komutlar iletilebilmektedir.

# **2. PROJE DİYAGRAMI**

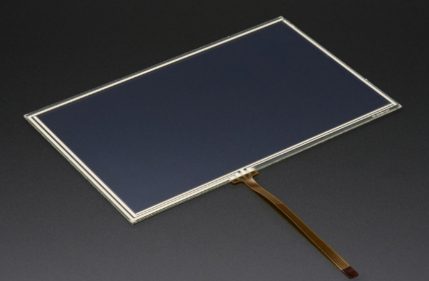
Aşağıda tüm sistemin blok şemasıdır.

Q:\Documents\GitHub\395_Project1\report\assets\blok_şeması.png

## **2.1 Donanım Modülü**

### **2.1.1 Proje malzemeleri**

### .

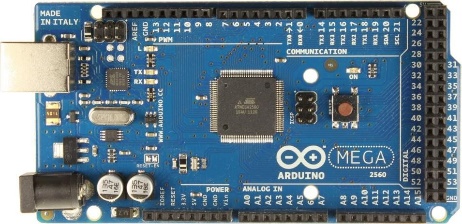
Dokunmatik ekran paneli içindeki dört adet telden aldığı potansiyel fark değerleri ile (x,y) biçiminde konum değerleri döndürmektedir 

***Şekil 2.1.1: Tel Dirençli Dokunmatik Ekran Paneli***



Panel üzerine koyulan topun dengeye gelmesi için paneli aşağı yukarı hareketlerle dengeleyen çubukların hareketini sağlanmıştır.

**Şekil 2.1.2: Servo motor**

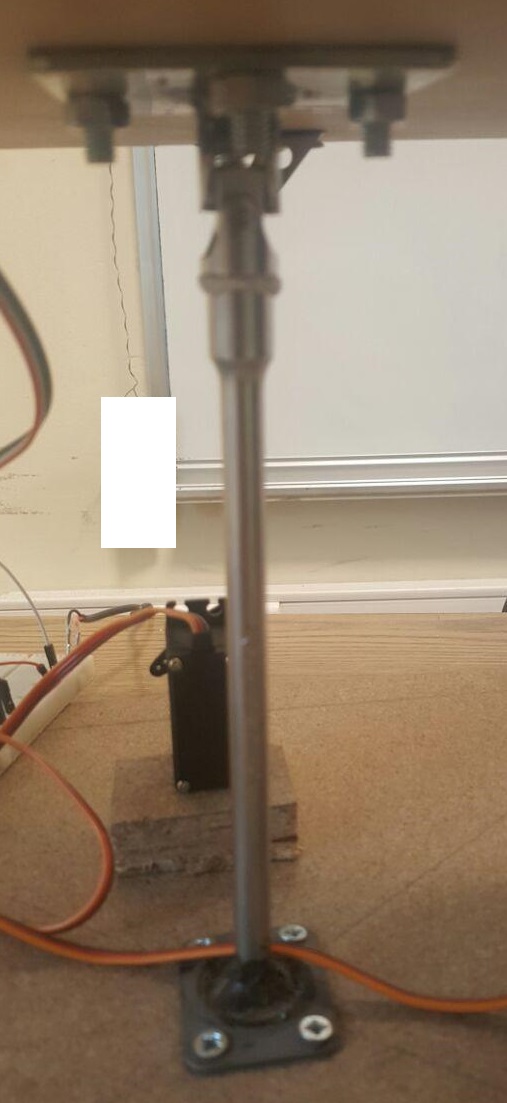
Arduino IDE yazı düzenleyicisi aracılığı ile Arduino’ya bootloader ile (karta yazılım yüklemeye yarayan kod parçası) program yüklemesi yapıldı.

***Şekil 2.1.3:Arduino Mega 2560***

### **Donanim Montajlama**

**Plaka ve Motorlar Montajlama adımları:**

**1**.**Adım**: Plakanın merkezine mafsalı sabitledik ve tabana tutturulmuş çubuğun üzerine monte edildi. Plaka, her yönden serbestçe hareket edebilmesi sağlandı.

****

**Şekil 2.2.2**

**2.Adım**: Motorlar, iki motorun her birine dik olacak şekilde tabana sabitledik.

**3**.**Adım**: Kolları motorlardan plakaya bağlayan eklemler yapıldı. Bu eklemler harekette iki derece serbestlik sağlandı. Bu yalnızca bir motor hareket ederken plakanın merkez etrafında dönme hareketinden kaçınmak için yapıldı.

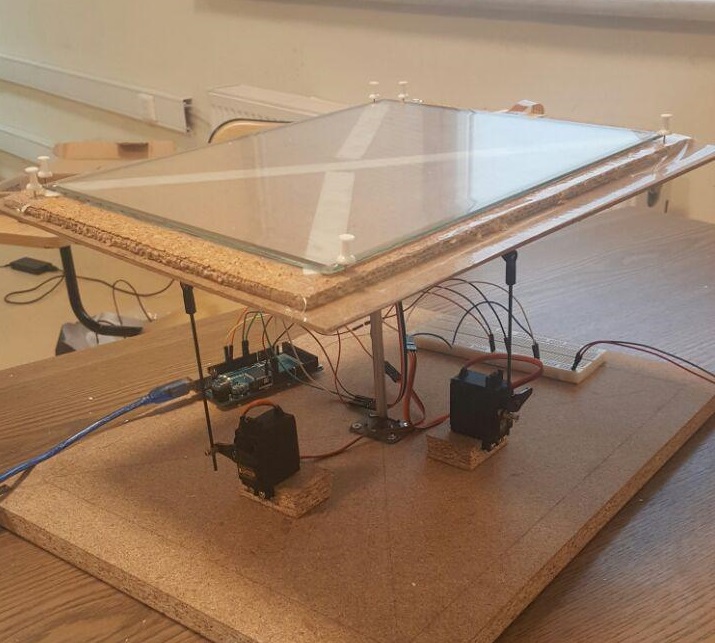
****

**Şekil 2.2.3**

**4**.**Adım**: Eklemler plakaya köşebentler kullanarak vidalanmıştır. Motorlar tabandaki vida merkezinden olduğu için eklemler plaka merkezinden uzaktadır.

**5**.**Adım**: Eklemler motorun koluna vida ile bağlandı.

**6.Adım**: Plakanın hareketi test edildi.



**Şekil 2.2.4**

## **2.2 Yazılım Modülü**

Yazılım Modülünde şu algoritmalar kullnılmıştır.

1. Mikrodenetleyici'de PID CONTROL'u kullanacağız,

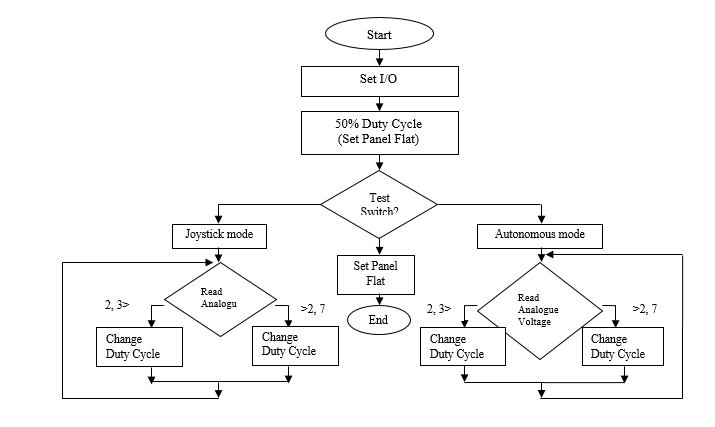
Two Joysticks for X-Y axis control

Servos control

CONTROLLER

Şekil : Dengeleme Bilyesinin Blok Diyagramı

Joystick Control: Bir tabanda dönen bir çubuktan oluşan bir giriş aygıtıdır. Analog değeri okunur ve plakanın dönmesi gereken bir açıyla döner.

****

Şekil : Dengeleme Bilyesinin Yazılım Akış Diyagramı

### **2.2.1 Kullanılan Yöntem Ve Kütüphaneler**

Projemizde touchscreenden gelen X ve Y koordinatlarını işleyerek servo motorlara göndereceğimiz açı değerlerini belirliyoruz. Bu açı değerlerini belirlerken otomatik kontrol denetiminde kullanılan PID denetleyicisi yöntemini kullanıyoruz. Bu işlemler için kullandığımız yöntem ve kütüphaneler aşağıda gösterilmiştir:

**PID :**  Kapalı bir döngüde denetim sistemi içinde kontrol eden organın (kontrolörün) görevi, ölçme üzerinden geri beslenen çıkış büyüklüğünü (hız, devir sayısı, pozisyon vb.), referans olarak verilen giriş büyüklüğü ile karşılaştırarak aradaki farkın yani hata değerinin miktarına kendi kontrol sisteminin etkisine uygun bir şekilde kumanda denetim sinyali üreterek çıkışa tekrar sunulması ile sistem kontrol edilir. Bu yöntemlerden PID denetleyicisinin açılımı ;

► P (Proportional) Oransal  
► I (Integral) İntegral  
► D (Derivative) Türevsel

Bir PID denetleyici çıkıştan geri besleme (feedback) ile gelen sinyali giriş (referans) sinyali ile karşılaştırır ve farkından bir hata oluşur. Bu oluşan hataya göre PID denetleyici hatayı en aza indirgemeye çalışarak bir etki yapar ve çıkışa gönderir. Bu şekilde hata en aza indirilene kadar çıkıştan girişe sürekli geri besleme ile hatalar belirlenir ve denetleyici etkisini çıkışa göndererek hata azaltılır

Burada önemli olan Kp, Ki ve Kd kazançlarının belirlenmesidir. Bu katsayıların uygun şartlarda sistemi kontrol etmesi için belirlenmesi gerekir. Bunu belirlemek kolay değildir ve net bir cevabı yoktur. En uygun ayar için en az ve sıfır hata, sistemde minimum aşım, kısa sürede hatayı giderme ve sistemde kararlılığı sağlama gibi kriterleri yerine getirecek Kp, Ki ve Kd kazançları seçilmelidir. Yani sistemin çıkışında en küçük hatayı en az zamanda ve en az salınım yapması gerekir. Bu projede de anlık ölçümler sonucu Kp , Ki, Kd değerleri belirlenmiştir.

**MAP** : Map fonksiyonu temel olarak belirli bir değer aralığında olan tam sayı değerlerini başka bir değer aralığına çevirir. Projede, touchscreen’den gelen input değerlerindeki gürültüyü azaltmak için, aralık daha küçük bir aralığa çekildi. Input aralıkları, her iki eksen için 0 – 1024 iken map yöntemi ile bu değerleri X ekseni için 0 - 400, Y ekseni için 0 – 300 aralığına çekildi.

**TOUCHSCREEN.H KÜTÜPHANESİ :** Touchscreen’den gelen X ve Y koordinatları getPoint() ile alındı. Touchscreen’in üzerinde top yoksa pressure değeri 0 olur ve bu sayede sistem denge konumuna dönebilir. Ancak bu projede top pressure değeri yaratmadığından bu değer kullanılmadan sistemin X ve Y değerlerinin mantıklı olup olmadığına bakılarak sistem dengeye getirilmiştir.

**AVERAGE (ORTALAMA ALMA) :** Servo motorlardan gelen açı değerlerinin çok fazla gürültü oluşturup sistemi salınım hareketine zorladığı için son 5 açı değerinin ortalaması alınarak sisteme hareket verilmektedir.

### **2.2.2 Genel Algoritma**

* Touchscreen’in üzerinde top yok iken motorlar 90’ar derecede olup sistem denge konumundadır.
* Touchscreen üzerine top konulduğunda X ve Y koordinatları Arduino’ya gitmektedir.
* Yazılım modülüne bu koordinatlar (0 – 1024) aralığında gelmektedir. Bu koordinatlar map fonksiyonu kullanarak 400\*300 çözünürlüğe oranlanmaktadır.
* Varsayılan olarak topun dengelenmesini istediğimiz noktası orta noktasıdır. Bu yüzden Setpoint noktası (200,150) olarak alınır. Bu ileri versiyonlarda istenilen herhangi bir nokta seçilebilecektir.
* PID denetleyicisi deneysel olarak bulunan Kp, Ki, Kd değerlerine bakarak topun mevcut konumunu alıp topun hızını ve ivmesini hesap ederek servo motorların yapacakları dönüş açısını hesaplar.
* PID denetleyicisinden gelen değerler servo motorlara gönderilir ve sistemin hareketi sağlanır.
* Topun el ile itilme hızı veya topun başlangıç konumuna göre system birkaç salınım yaptıktan sonra denge noktasında(Setpoint) dengelenir.

## **2.3 Arayüz Modülü**

### **2.3.1 Modellemede Kullanılan Program**

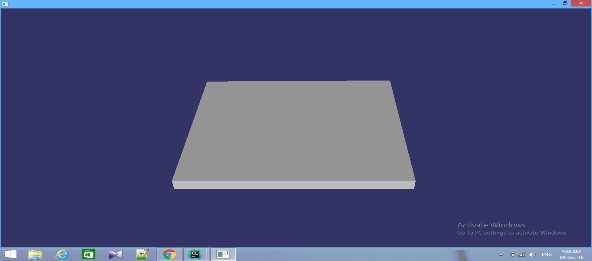
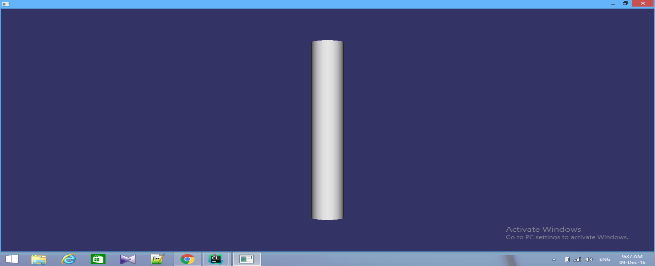
OpenSceneGraphbu projede arayüz modülünün ana taşıyıcısıdır. Şöyle ki OpenSceneGraphkütüphaneleri kullanılarak GUI ve 3B modelleme yapılabilmektedir. OpenSceneGraphkaynağının temeli OpenGl’nin üzerinde bulunarak geliştiriciyi, düşük seviyeli grafik çağırımlarını uygulama ve uyumlaştırma gibi işlemlerden kurtaran nesneye yönelik bir framework ile desteklemektedir. Projede OpenSceneGraphsorunsuz bir şekilde entegre oldu.

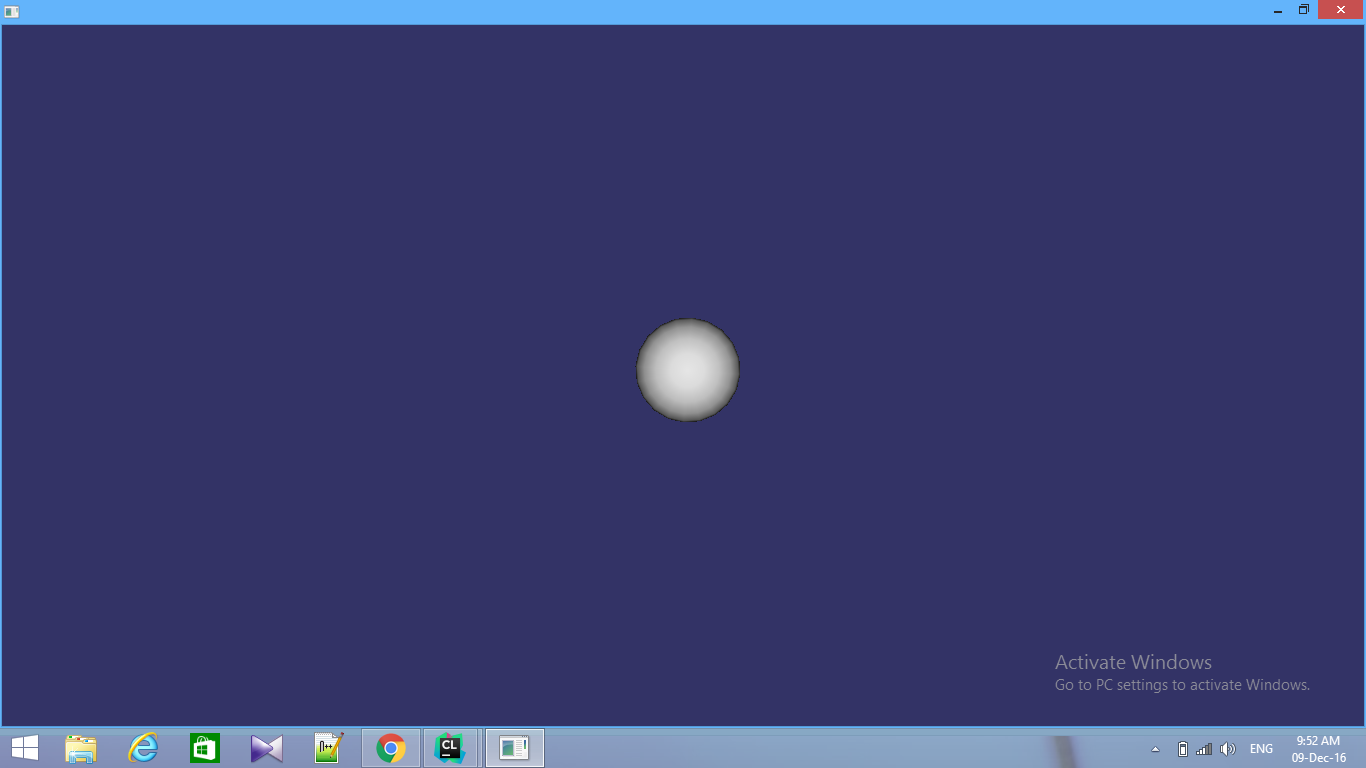
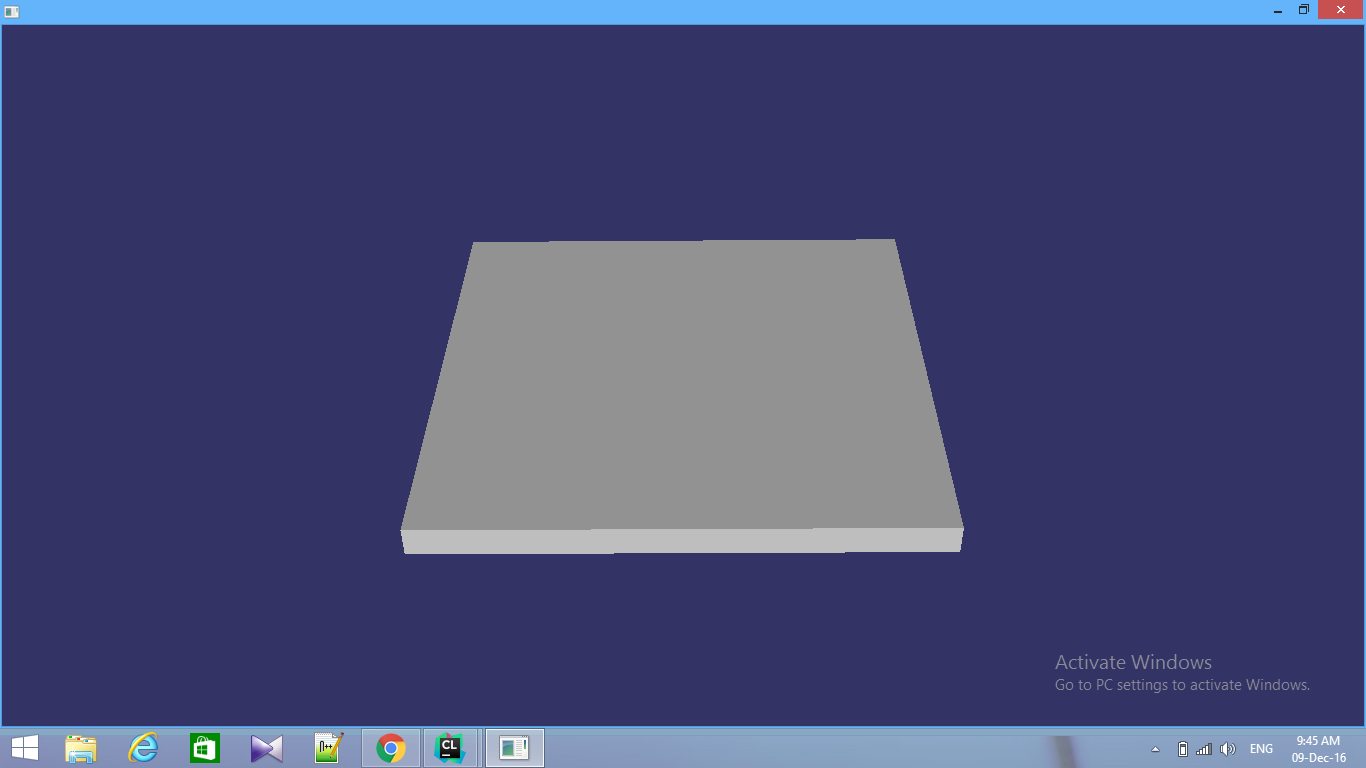
### **2.3.2 Kullanıcı Ekranı**

Kullanıcı ekranında, topun klavye üzerinde konumunu x ve y koordinatları şeklinde veren bizim belirlediğimiz klavye tuşları ile hareketi sağlamaktayız. Yine bu x ve y koordinatlarını kullanarak sistemin 3B bir simülasyonu gösterilecektir. Bu yapılırken ekranın çipe verdiği x ve y koordinatları çipten alınarak simülasyon da topun hareketi gerçekleşecektir.

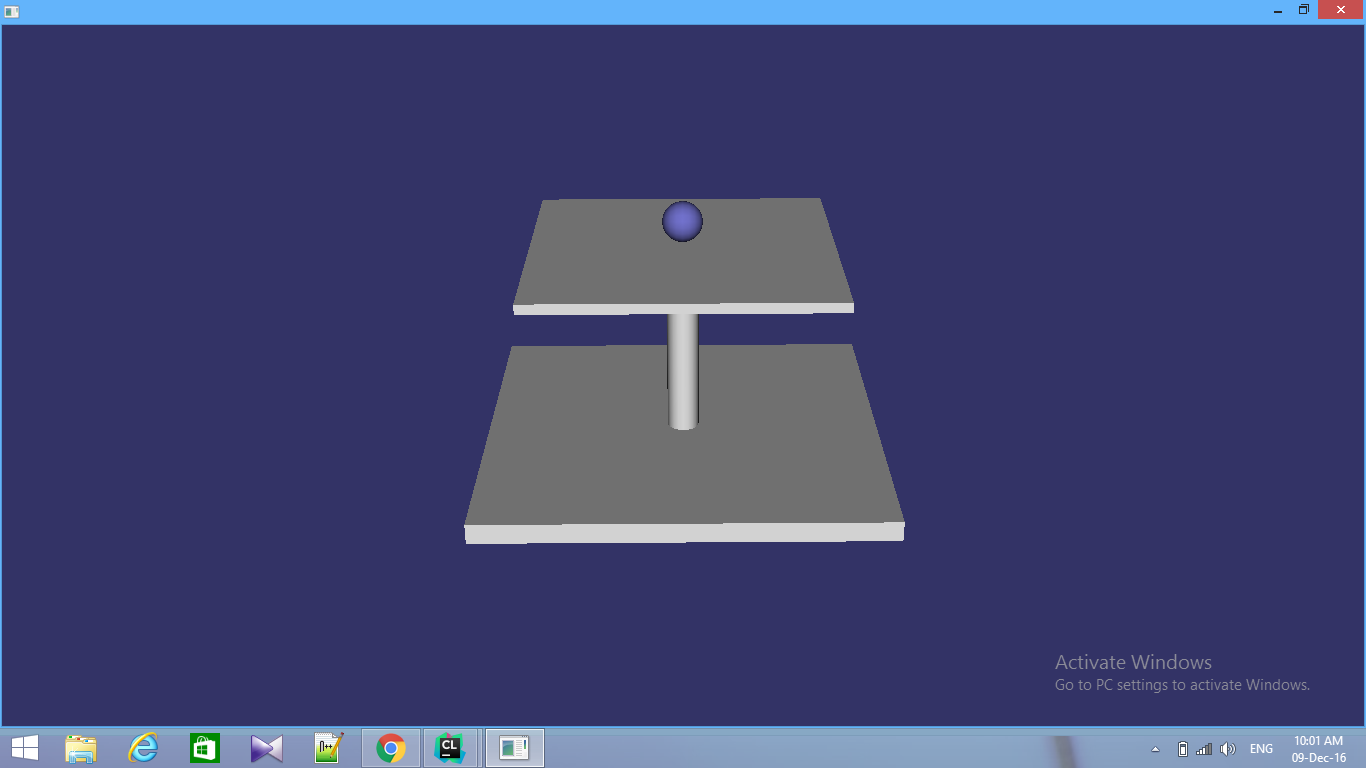
### **2.3.3 Modelleme**

Projemiz için gerekli olan objeleri oluşturduk. Bunlar ise top ekran direk ve tahtadan oluşmaktadır.





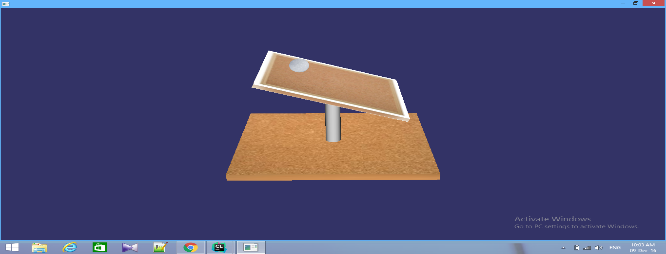
Yaptığımız objeleri birbiri ile entegre olmasını sağladık.

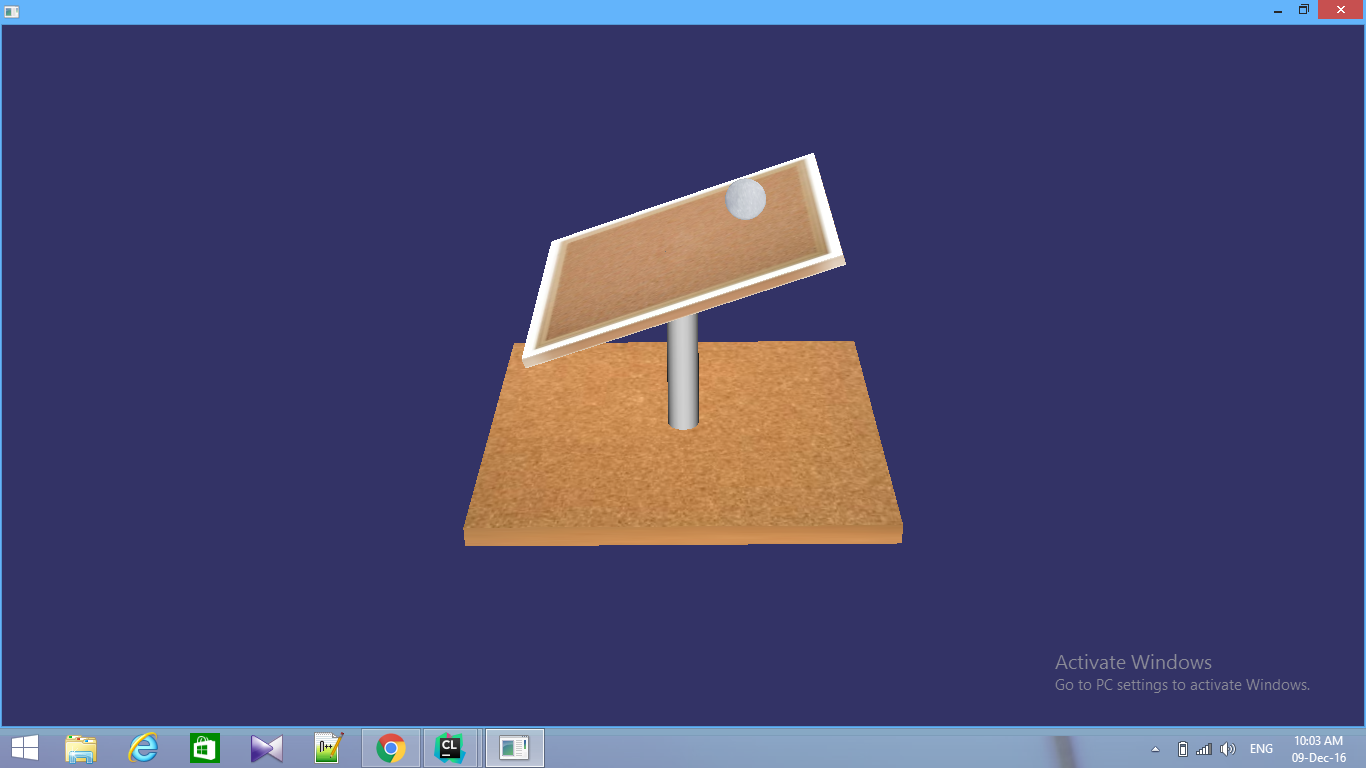


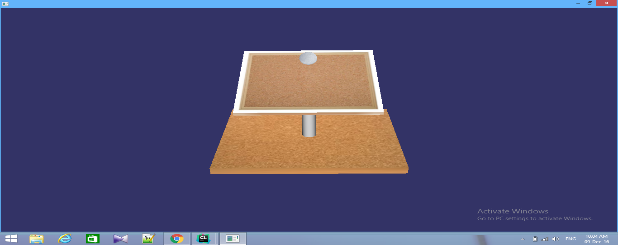
Texture Mapping (Doku Kaplama): Doku kaplama 2D bir grafiği, 3D bir cisme uygulamaktır. Bu sayede biz oluşturulduğumuz cisimlere texture mapping ile gerçekçi görüntüler ve dokular kazandırtık.

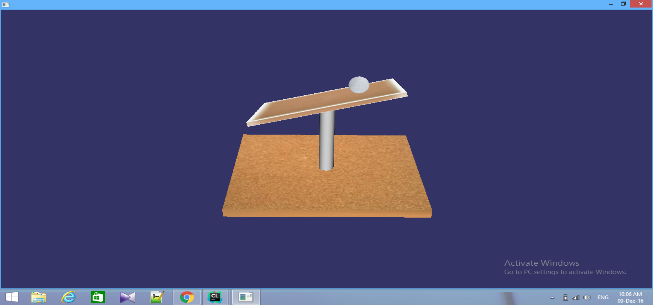


Oluşturduğumuz objelerin hareketlerini sağladık. Alttaki tahta ve direk hareket etme kabiliyeti yok. Top ve masa bizim klavye üzerinden girdiğimiz tuşlar ile hareket etmektedir. Bu hareketi matris çarpım işlemleri ile sağladık. İleriki aşamada donanımdan aldığımız inputlar ile hareket sağlanacaktır.









Işığın yansımasını cismin yüzeyine hangi açıyla geldiğini belirleyerek düzenlemesini ve kamera ayarlamasını yaptık.

# **3. GÖREV DAĞILIMI**

|  |  |
| --- | --- |
| Ali Ozan Taçalan | Donanım,Yazılım |
| Emrullah Gençoğlu | Donanım, Yazılım |
| Ekber Aziz | Modelleme, Yazılım |
| Serap Uçuş | Donanım , Modelleme |
| Hasan Bilgin | Donanım, Modelleme |
| Şahin Eğilmez | Modelleme, Yazılım |
| Oğuz Erdi Erkmen | Donanım, Yazılım |

# **4. REFERANSLAR**

[1] *Arduino Mega 2560*, <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

[2] *OpenSceneGraph*, <http://www.openscenegraph.org/>

[3] MathWorks: R2011a Documenj tation, http://www.mathworks.com/help/index.html.