



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

SİTE : <http://ballandplate.wixsite.com/grup1>

İÇİNDEKİLER

1. GİRİŞ	3
1.1. Projenin Tanımı	3
1.2. Projenin Amacı	3
1.3. Proje Kapsamı ve Kriterleri	3
2. EYLEM RAPORU	4
2.1 Donanım Modülü	4
2.1.1 Proje malzemeleri	4
2.1.2 Donanım Birleştirme	5
2.2 Yazılım Modülü	7
2.2.1 Modüller	7
2.2.2 Genel Algoritma	9
2.3 Arayüz Modülü	10
2.3.1 Gereksinimler	10
2.3.2 Modelleme	10
3.KURULUM VE ÇALIŞMA	12
4. SONUÇ	14
5. GÖREV DAĞILIMI	14
6. REFERANSLAR	15

1. GİRİŞ

1.1. 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.

1.2. Projenin Amacı

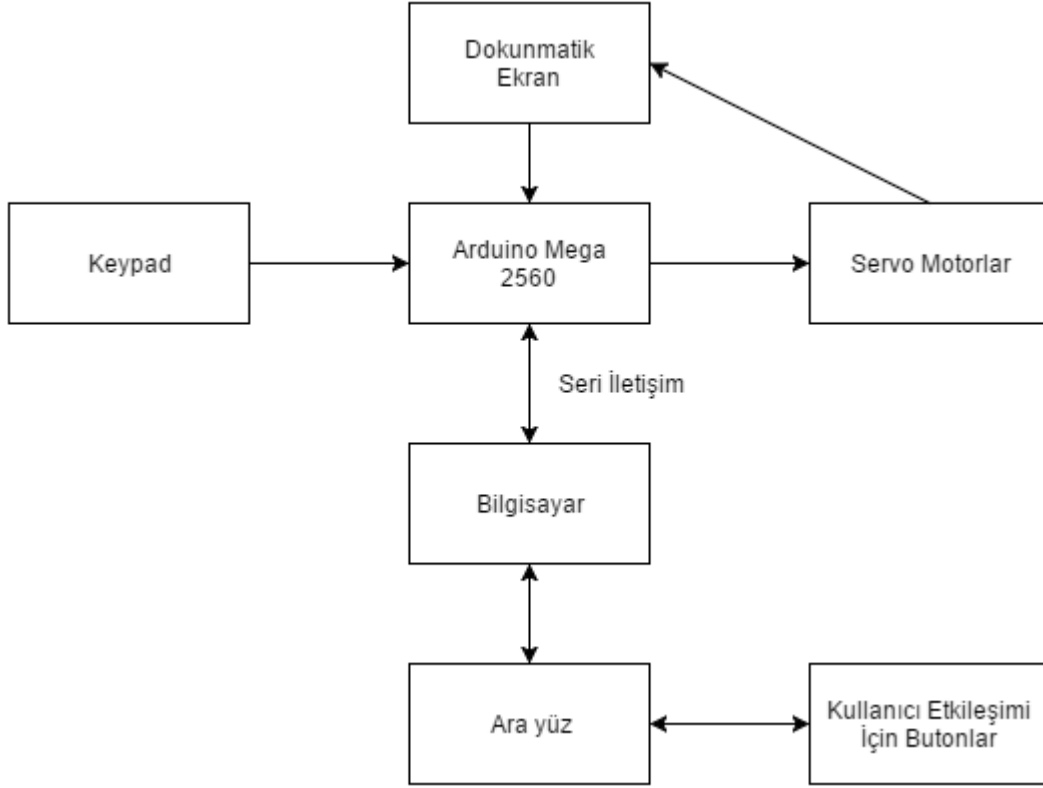
Düzenekte düzlem olarak kullanılacak dokunmatik ekran üzerinden anlık konum bilgileri alınacak ve topun dengede durması için dokunmatik ekranın bağlı olduğu düzlem gerekli yönler doğru hareket etmesini sağlamaktır. Bu hareketleri sağlamak için bir Arduino Mega 2560 [1] ve 2 tane Servo motor kullanılmıştır. 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.

1.3. 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 yapabilmek 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 iletebilmektedir.

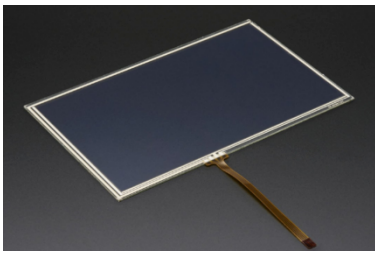
2. EYLEM RAPORU

Aşağıda tüm sistemin blok şemasıdır.



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 alınmaktadır.



Panel üzerine koyulan topun dengeye gelmesi için paneli aşağı yukarı hareketlerle dengeleyen Servo eklemlerine bağlı çubukların hareketini sağlamıştır. 5V'luk adaptörden iki motor güç almaktadır.

Arduino Mega 2560 ile topu dengede tutacak gerekli fasıllar uygulanıp aynı zamanda bilgisayar ile haberleşme işlemleri gerçekleşir.



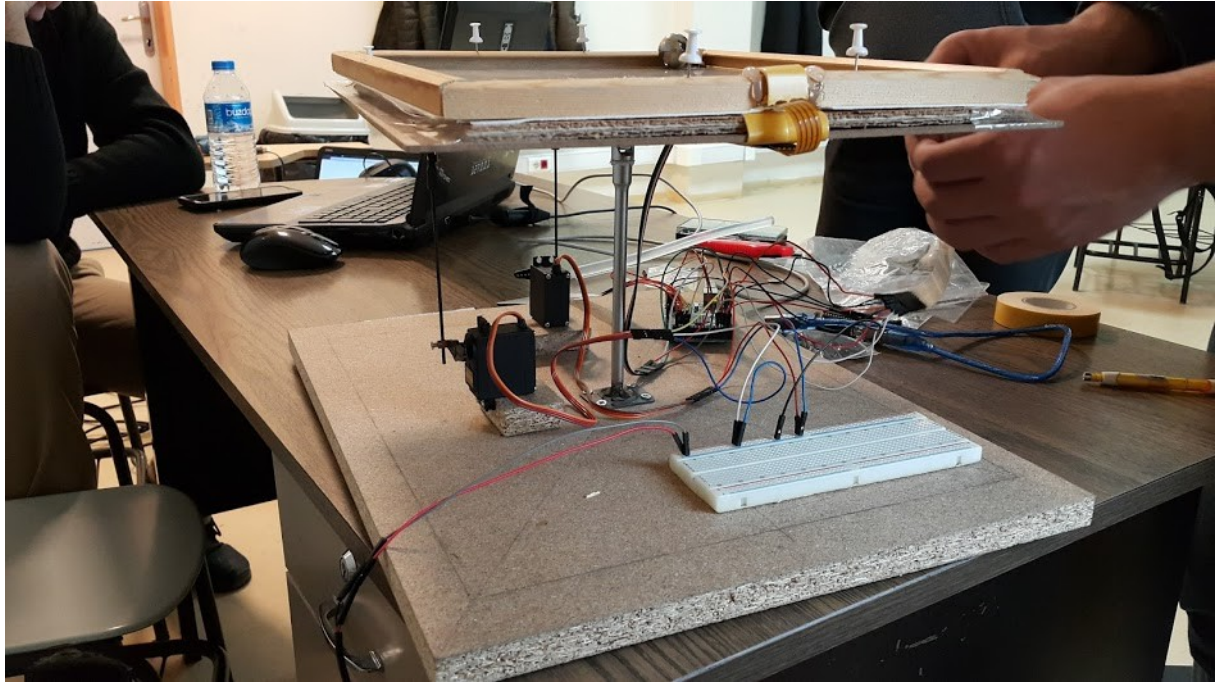
edebilmek üzere iki adet
bağlanmıştır.



Keypad ile düzlem
üzerinde hazırlanmış kale
oyununu (2.2.1) kontrol
Arduino Mega 2560'a

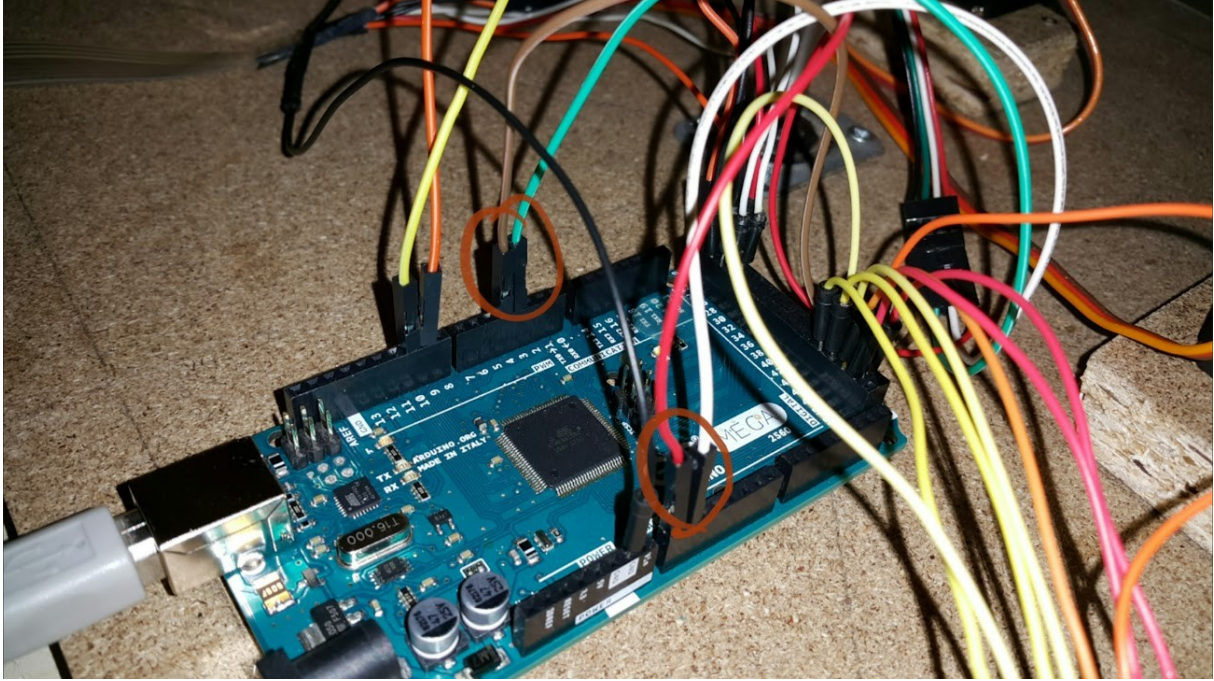
2.1.2 Donanım Birleştirme

Kalın tahta bir düzlem donanımın tabanını oluşturması için seçildi. Daha sonra dokunmatik ekranın yapıştırıldığı ince tahta düzlem ile kalın tahta düzlem metal bir ayak ile vidalanarak sabitlendi.

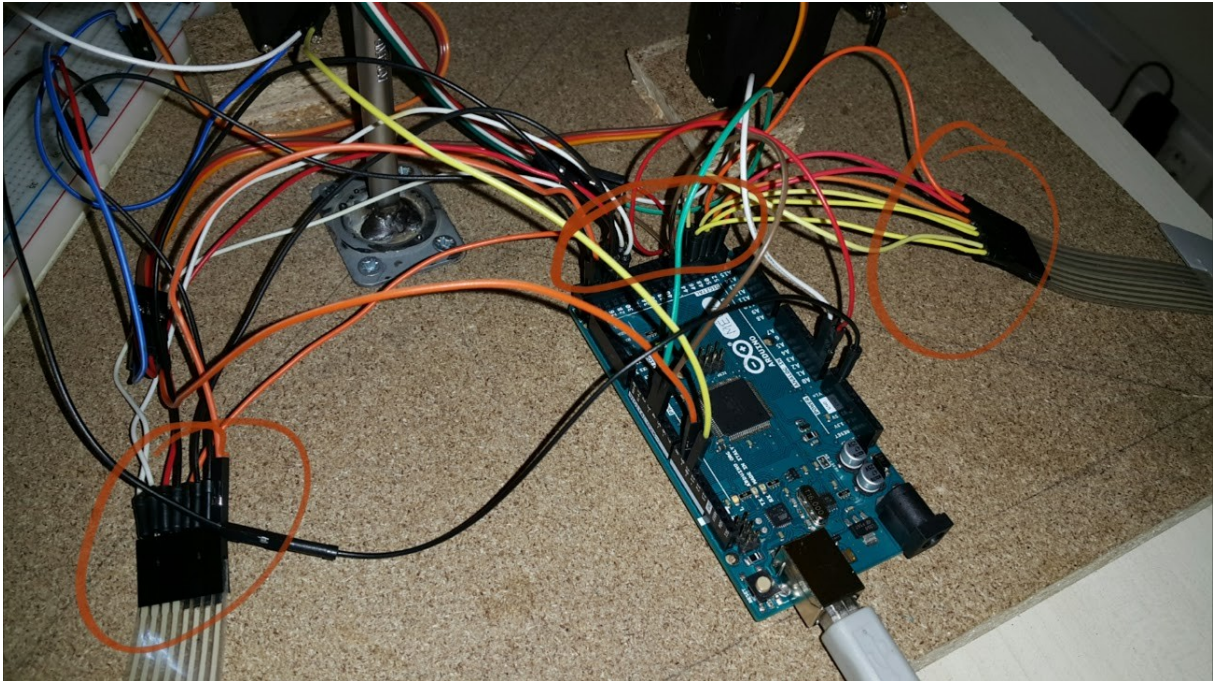


Servo motorlar zemindeki kalın tahta düzleme ahşap objeler yardımıyla sabitlendi. Servo motorlarının ilk olarak sahip oldukları açı 90 derece olup zemine paralel bir görümüne sahiptirler. Servo motorlara takılan düz başlıklar takılarak, eklenecek Servo eklemleri için hareket alanı sağlanmıştır. Servo eklemlerinin diğer ucu da ekranın bağlı olduğu ince tahta düzleme sabitlenmiştir. Böylelikle Servo motorunun gerçekleştirdiği dönme hareketleri ile dokunmatik ekran düzlemi üç boyutlu uzayda hareket ettirilebilmektedir. Bahsedilen görümüne yukarıdaki şekilden ulaşılabilir. Servo bağlantıları 9. ve 10. dijital pinlerden gerçekleştirilmiştir.

Dokunmatik ekran 3 ve 4 dijital ve A0 ve A1 analog pinlerinden sağlanmıştır. Dört dirençli dokunmatik ekrandaki bu dört pin sırasıyla X+, X-, Y+, Y- kablolarını temsil etmektedir.



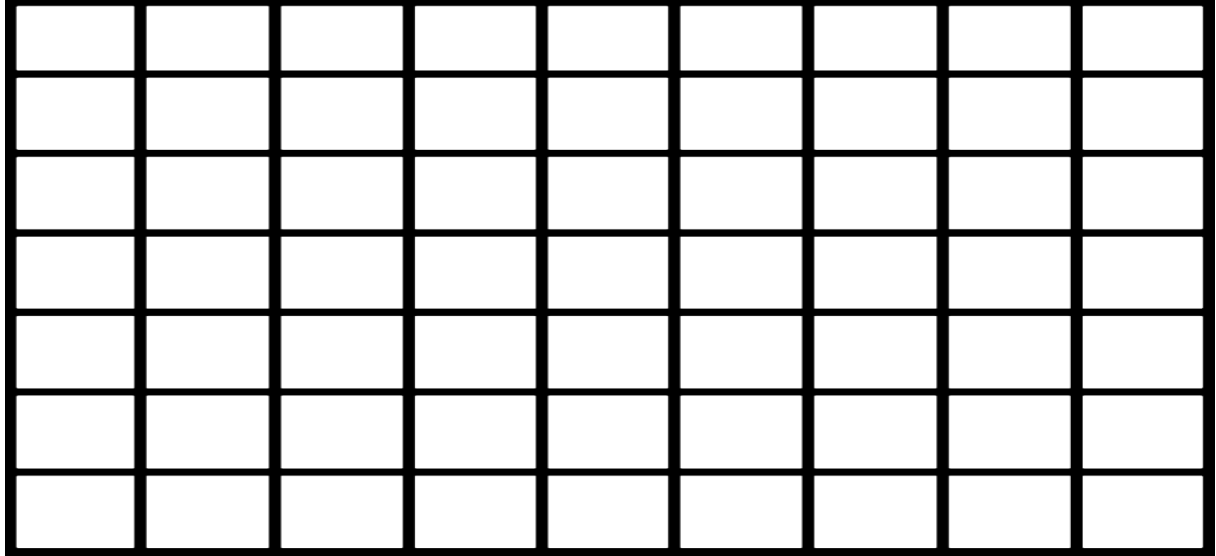
Oyun oynamak için eklenmiş iki adet keypad, Arduino Mega 2560'ın keypad1 için 26, 27, 28, 29. Kolon 22, 23, 24, 25. satır, keypad2 için 30, 31, 32, 33 kolon, 37, 36, 35, 34. satır dijital pinler arasına takılmıştır. Böylelikle kullanıcılar keypad ile kurdukları etkileşimlerde girdilerini sisteme aktarabilmektedirler.



2.2 Yazılım Modülü

2.2.1 Modüller

MAP: Haritalama fonksiyonu temel olarak belirli bir değer aralığında olan tam sayı değerlerini başka bir değer aralığına çevirir. Projede, dokunmatik ekrandan gelen konum değerlerindeki gürültüyü azaltmak için, aralık daha küçük bir aralığa çekildi. Konum aralıkları, her iki eksen için 0 – 1024 bandındayken haritalama yöntemi ile bu değerler X eksenini için 0 - 400, Y eksenini için 0 – 300 aralığına çekildi. Haritalama işleminin ardından düzlem aşağıdaki gibi hayali ızgaralara ayrılarak her bir birim alan için yaklaşık motor tepkileri oluşturulur.



PID denetleyicisi: Topun sürekli hareketinden ötürü ortaya çıkan gecikme ile topun konumları için gerekli tepkilerin üretilmesi geç gerçekleşmektedir. PID denetleyicisi ile topun olacağı yer ile şu an olduğu yer bilgileri üzerinden hata hesaplaması yapılır. Hatanın büyüklüğüne göre oluşturulan daha keskin Servo motor hareketleri ile geç tepki süreleri kısaltılmış olur. [3]

Touchscreen.h: Arduino için hazırlanan dokunmatik ekran kütüphanesi ile *getPoint()* vb. fonksiyonlar ile X ve Y noktalarına ait bilgiler alınabilmektedir [4].

İletişim: İletişim Arduino ve bilgisayar olarak iki farklı bölüme ayrılmıştır. Her iki bölümde de alıcı (rx) ve gönderici (tx) fonksiyonlar yer almakla beraber bağlı olunan seri COM portu ile haberleşme gerçekleştirilir. Arduino tarafında sistem fonksiyonu olan *Serial.read()* [5] ve *Serial.write()* [6] fonksiyonları kullanılarak bayt bayt okuma ve yazma yapılarak cstring oluşturulur. Bilgisayar tarafında ise RS-232 kütüphanesi [7] kullanılarak seri port üzerinden karşılıklı haberleşme gerçekleştirilmiştir. Bu bölümün devamında sırasıyla bu fonksiyonlardan bahsedilecektir.

- **İletişim protokolleri:** Üç kısma ayrılmıştır.
 - o Arduino, bilgisayara mesaj gönderirken “<x>,<y>x” şeklinde bir string oluşturulup gönderilir. <x> ve <y> topun dokunmatik ekran üzerindeki konumunu bildirmektedir. String sonundaki x ise bitiş karakteri olarak tanımlanmıştır.

- o Bilgisayardan Arduino'ya mesaj gönderilirken <setPointX,setPointY>x formatı uygulanmaktadır. Böylece bilgisayar üzerinden kullanıcı dokunmatik ekran üzerinde yeni merkezler belirleyebilme imkânına sahip olmaktadır.
- o Kale oyunu için eklenen keypad'lerin bilgisayardan bağımsız Arduino ile iletişim kurması sağlanmıştır. Keypad üzerine basılan karakter Arduino üzerinden okunup tepki üretilmektedir.
- **Serial.read():** Arduino üzerinde 'x' olarak belirlenen string mesaj sonu karakterine kadar okunan karakterler sırasıyla bir string üzerinde biriktirilmektedir. Mesaj sonu görüldüğü zaman 'x' yerine '\0' konularak alınan mesaj ile işlem yapılmaktadır.
- **Serial.write():** Dokunmatik ekran üzerinde bulunan metal topun konum bilgileri bu fonksiyon aracılığı ile birer bayt şeklinde seri porta yazılmıştır.
- **RS232_PollComport():** RS-232 kütüphanesinden olan bu fonksiyon ile açılan seri port üzerinden belirtilen boyuta kadar karakter dizisi alınabilmektedir.
- **RS232_cputs():** RS-232 kütüphanesinden olan bu fonksiyon ile açılan seri port üzerinden istenilen mesaj aynı anda birer bayt olacak şekilde seri porta yazılabilmektedir.

Ara yüz: OpenSceneGraph'ın sunduğu görsel imkânlarla üç boyutlu bir uzayda objelerden oluşan tasarım uygulanmıştır. OpenSceneGraph altında her bir obje ağaç hiyerarşisine sahiptir. En tepede (kök) *camera HUD* objesi yer almaktadır. Oluşturulan her yeni obje bu kökün içine eklenmelidir. Aşağıdaki fonksiyon açıklamalarıyla ara yüzün temel çalışma yöntemi hakkında fikir vermek amaçlanmaktadır.

- **addDrawable():** Bu fonksiyona gönderilen veri yapıları ile bir üst düğüme mevcut özellikler atanabilmektedir. Projede kullanılan veri yapıları arasında üç boyutlu vektör, çizgi, devamlı çizgi, düzlemler vb. yer almaktadır. Örneğin oyun için tasarlanan kale yapısı iki tane üçer çizgi objesinin birleştirilmesiyle gerçekleştirilmiştir.
- **updateCallback():** Dinleyici olan bu fonksiyon, başka fonksiyonlar içinde çağırıldığında belli frekansta atanan işi yürütmektedir. Bu fonksiyon ile animasyonlu (hareketli) objelerin yönetimi gerçekleştirilmiştir.
- **update():** Donanımın görsel ifadesine ek olarak pencerede sunulan gerçek zamanlı x ve y tablosunun kullandığı bu fonksiyon ile tablo düzenli olarak güncellenmektedir.
- **pickHandler class:** Pencereye eklenen düzlem objesine atanan tıklama dinleyicisi ile kullanıcı tıklamaları ile dokunmatik ekranın merkez noktasının değiştirilmesi sağlanmaktadır. GUIEventHandler sınıfını ve setPointsXYfromPano fonksiyonunu kullanan bu sınıf ile fare tıklamaları ile kullanıcı etkileşimi gerçekleştirilmiştir.

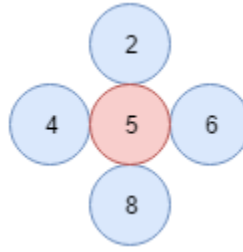
Keypad yöneticisi: Kale oyunu için kullanılan iki keypad'den sisteme veri girişi sağlanması için *getKey()* isimli fonksiyon kullanılmıştır. Bu fonksiyon ile alınan değer oyun yapısının içine dâhil edilir. Oyun yapısından daha ayrıntılı bir sonraki alt başlıkta bahsedilmiştir.

Kale oyunu: Kullanıcının ara yüz aracılığı ile başlattığı bu oyunun kuralları aşağıdaki gibidir.

- o Top, kullanıcı1'in yarı sahasında ise kullanıcı1, kullanıcı2'nin yarı sahasında ise kullanıcı2 düzlem kontrolünü keypad aracılığı ile kontrol edebilmektedir.



- o Keypad ile atanmış tuşlar aşağıdaki şekilde gibidir. Kullanıcı mavi tuşlar ile kendi bakış açısı yönünde sırasıyla (ileri, sol, sağ, geri) hareketlerini Servo motorlarına 20'şer derecelik tepki göndererek oluşturmaktadır.



- o Kullanıcı kırmızı tuş ile etkileşime geçtiği zaman karşı kale yönünde (2 yönünde) 40 derecelik Servo motoru hareketi gerçekleştirir.
- o Kullanıcılar arası skor kayıt altında tutulup gol anında ara yüzde golü ifade edecek eylemler gerçekleştirilir.

2.2.2 Genel Algoritma

- Arduino'ya yüklenen program ilk olarak düzgün PID'ler ile topu dengede tutmaktadır. PID denetleyicisi ile topun hızı ve ivmesi hesaplanarak Servo motorların yapacakları dönüş hesaplanır. Topun hızı, ivmesi ve konumuna göre topun yaptığı birkaç salınımdan sonra top merkez olarak belirlenen noktada kalmaktadır.
- Dokunmatik ekran üzerine top konulduğunda X ve Y koordinatları Arduino aracılığı ile seri porta yazılmaktadır.
- Dokunmatik ekranın üzerinde top yok iken Servo açıları zemine paralel hale gelip sistem denge konumuna ulaşmaktadır.
- 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 merkez noktası (200,150) olarak alınmaktadır. Kullanıcı istediği zaman bu merkez noktası ile ilgili değişikliğe gidebilmektedir.

- Kullanıcı oyun oynayıp oynamak istemediği girişini ara yüz aracılığı ile ifade edebilmektedir.

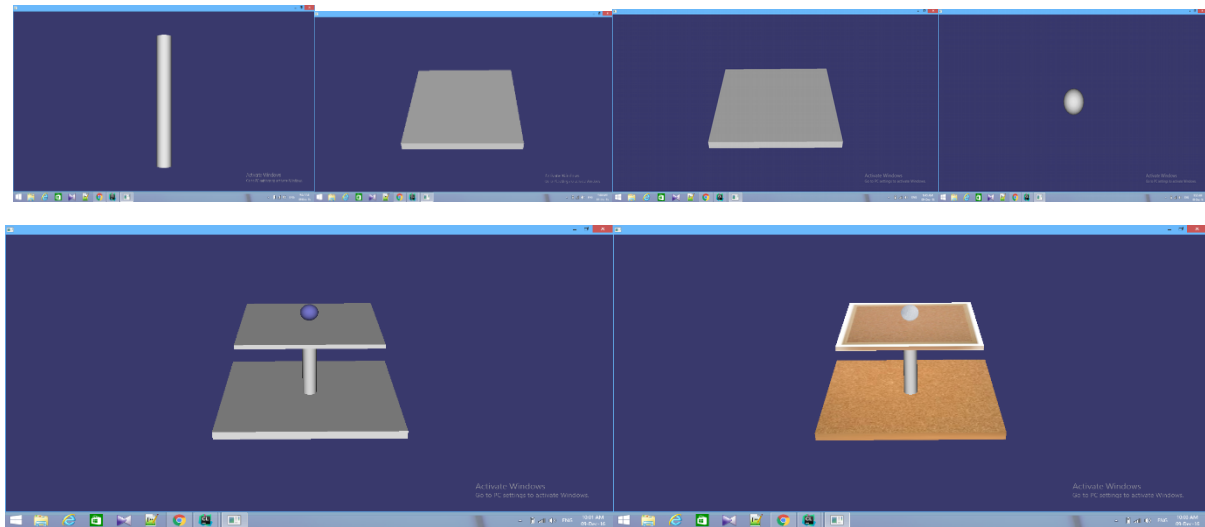
2.3 Arayüz Modülü

2.3.1 Gereksinimler

OpenSceneGraph proje ara yüzünün temelini oluşturmaktadır. OpenSceneGraph kütüphaneleri kullanılarak görsel ara yüz ve üç boyutlu modellemeler gerçekleştirilmiştir. Buna ek olarak kullanıcı girdilerini Arduino'ya gönderebilen düğme yapıları konularla kullanıcı etkileşimi sağlanmıştır. Bilgisayar için yazılmış iletişim kodlar modelleme programına dâhil edilerek ara yüz aracılığı ile Arduino ile haberleşme gerçekleştirilmektedir.

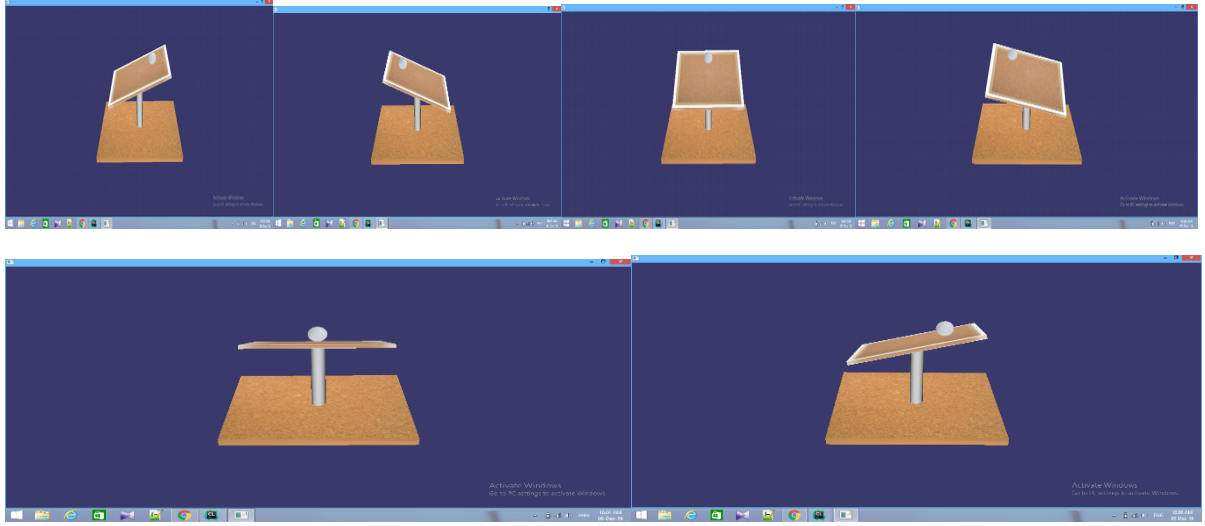
2.3.2 Modelleme

Proje için öncelikle bahsedilen *addDrawable()* metodu yardımıyla donanımın görsel bir kopyası OpenSceneGraph için yaratıldı. Yapılan objelerin birbirleri ile ilişkilendirilmesi sağlandı. Bundan ötürü objeler birbirlerine bağımlı gibi davranış sergilemektedirler.

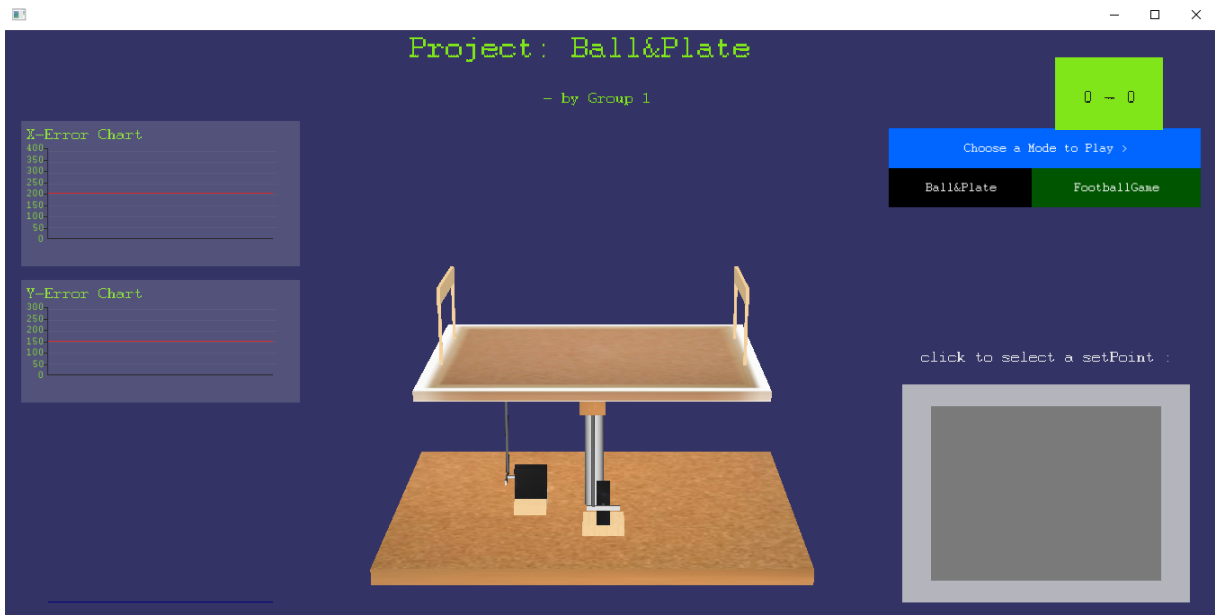


Birbirleriyle ilişkilendirilen objeleri daha güzel gözükmeleri için doku kaplama tekniği uygulanmıştır. Doku kaplama ile iki boyutlu bir grafik üç boyutlu bir cisme uygulanmıştır (yukarıdaki son resim). Böylelikle daha gerçekçi görüntüler elde edilmiştir.

Oluşturulan objelerin hareketleri sağlandı. Alttaki tahta düzlem ve çubuk hareket etme kabiliyetine sahip değildir. Top ve dokunmatik ekran düzlemi Arduino üzerinden gelen konum değerlerine göre hareket etmektedir. Böylece donanımda gerçekleşen işlemlerin görsel benzetimi de sağlanmış olur.



Işığın yansımalarının cismin yüzeyine hangi açıyla geldiği belirlenerek düzenlenmesini ve kamera ayarlamasını yapıldı. Eklenen Servo modelleri ile daha gerçekçi bir görünüm hedeflenmiştir. Ekranın sağ alt tarafına dokunmatik ekran üzerindeki merkezi değiştirecek tık dinleyici düzlem sol tarafına da anlık x ve y değişimlerinin ifadesi eklenmiştir. Ara yüzün son görünümü aşağıdaki gibidir.



3.KURULUM VE ÇALIŞTIRMA

Projenin tüm süreci Github üzerinden yürütülmüştür. Aşağıda verilen link üzerinden süreci inceleyebilirsiniz:

https://github.com/alozta/395_Project1

Yalnızca FINAL_DEMO klasöründe bulunan içerik teslim edilmiştir. Diğer dosyalar geliştirme ekibinin sürekli değişen çalışmalarıdır. Her bir klasörde Readme.md dosyasında gerekli yönergeler verilmiştir. Yönergeleri takip ediniz. Bu yönergeler aşağıda da verilmiştir. Kurulum için aşağıdaki yönergeleri takip ediniz:

#ARDUINO İÇİN

Arduino kodunu yüklerken gerekli kütüphaneleri yüklemediğiniz hatası alırsanız, Libraries klasöründen bunları sisteminize ekleyiniz.

Bunun ardından arduino kodunu yüklemeye hazırsınız.

Final klasörü içindeki Final.ino yu yükleyiniz.

Ball&Plate klasörünü Clion ide'si ile açınız. Çalıştırmadan önce openscenegraph kütüphanesinin yüklü olup olmadığını kontrol ediniz. Eğer yüklü değilse :

#LINUX İÇİN

Linux için Open Scene Graph kütüphanesinin kurulumundan önce bağımlılıklarını kurmak isteyebilirsiniz. Bunun için şu linkteki adrese gidin:

<http://trac.openscenegraph.org/projects/osg/wiki/Support/PlatformSpecifics/Debian-Dependencies>

Bundan sonra şu komutları uygulayarak kütüphaneyi kurun:

```
# check out OSG
```

```
svn checkout http://svn.openscenegraph.org/osg/OpenSceneGraph/trunk OpenSceneGraph
```

```
# compile the OSG
```

```
cd OpenSceneGraph
```

```
./configure
```

```
make -j 8
```

```
cd ..
```

Bu komutlardan sonra uygulamanın cmakeist.txt dosyasında linux için openscenegraph pathini belirtin. Unutmayın bu path home/user klasöründeki değil usr/local/lib32 veya lib64 dizinleridir. Varsayılan olarak bu projede /usr/local/lib64/ kullanılmıştır.

#WINDOWS İÇİN

OpenSceneGraph 'ın kurulması:

1. OpenSceneGraph klasörü C disk içerisindeki ProgramFiles(x86) 'ın içine konulacaktır.
2. OpenSceneGraph klasörü içerisindeki bin klasörünün path' i System Properties in içindeki Environment Variables path' ine eklenecektir.
3. OpenSceneGraph\bin içerisindeki osgPlugins-3.4.0 klasörünün path' i System Properties' in içindeki Environment Variables path' ine eklenecektir.

MinGW 0.6.0 'ın kurulması :

1. Bilgisayarda kurulu olan herhangi bir minGW sürümü varsa silenecek.
2. MinGW 0.6.0 installer indirilip bilgisayara kurulacak (link: <https://sourceforge.net/projects/mingw/>) .
3. Kurarken en azından şu package ler seçilmeli : gcc, g++, make, Cmake, gdb .
4. MinGW\bin klasörünün path'i i System Properties in içindeki Environment Variables path ine eklenecektir.

JetBrains Clion'ın kurulması:

1. JetBrains Clion indirilip bilgisayara kurulacaktır (link : <https://www.jetbrains.com/clion/download/#section=windows-version>) .
2. Clion kurulduktan sonra şu şekilde toolchains açılacak : File -> Settings -> 'Build, Execution, Deployment' -> 'Toolchains'.
3. Environment kısmında MinGW'ın kurulduğu klasörün path'inin doğru gösterilmesi lazım.
4. 'CMake executable' kısmı için 'Use bundled CMake x.x.x' 'i seçilecek.
5. 'Debugger' kısmı da ' bundled debugger x.x.x' 'i göstermesi lazım.
6. Alt kısmında tüm kategoriler önünde doğru işareti olması ve 'make', 'C compiler' and 'C++ compiler' ların doğru path'I gösterilmiş olması lazım.
7. 'Run' -> 'Edit configuration' kısmına gittiğinizde projenizin ismi sol tarafta Configuratin da sağ tarafta gösterilecektir.
8. Target : all target, configuration: Debug, executabal: Ball&Plate(executable'un ismi) şeklinde ayarlanmış olacaktır.
9. 'Working Directory' kısmında projenin bulunduğu klasörün path'I seçilecektir.

4. SONUÇ

Yapılan projede metal bir topun bir düzlem üzerinde sürekli bir şekilde dengelenmesi sağlanmıştır. Bu süreç üç ana bölüm altında özetlenebilir: donanım, yazılım ve ara yüz.

Donanım sürecinde öncelikle platformu oluşturabilmek adına kaliteli ve hesaplı araç arayışına gidilmiştir. Bunun için Arduino Mega 2560, dokunmatik ekran, iki Servo motor, iki keypad ve destekler bir araya getirilmiştir. Küçük test programlarıyla ekranın doğru değerleri okuduğundan emin olunmuştur.

Yazılım sürecinde öncelikle dokunmatik ekran ve Servo motor arasındaki ilişkiyi ortaya çıkarmaya yönelik çalışmalar yapılmıştır. Bu süreçte grubun ürettiği kodlardan elde edilen performanslar yetersiz kalmaktaydı. Bundan ötürü hazır kütüphaneler incelenerek PID ve haritalama yöntemleri bulundu. Proje uygulanan bu yeni algoritma ile artık metal top donanım üzerinde sabit şekilde durabilmekteydi.

Projenin iki ana kısmı bittikten sonra son olarak ara yüz kısmı üzerinde yoğunlaşıldı. Böyle bir isteği gerçekleştirebilmek için öncelikle iki farklı platformun (Arduino ve bilgisayar) haberleşmesi gerekliliği ortaya çıkmıştır. Geliştirilen iletişim protokolü yöntemleri ile iki farklı platform aynı protokolü kullanarak sağlıklı bir şekilde çift taraflı iletişim kurabildi. Bu duruma paralel bir şekilde geliştirilen üç boyutlu platform içine donanıma benzer objeler yerleştirilip benzer görüntüler elde edilmiştir. Daha sonra üç boyutlu ara yüzün etkileşim testleri yapılmıştır. Farklı değerlere göre farklı görünüme kavuşan (gerçek donanımda olduğu gibi) etkileşimli platformun testi gerçekleştirilmiştir. Son olarak karşılıklı iletişim sonucu elde edilen gerçek zamanlı verilerin bu üç boyutlu sanal uzayda gösterimi gerçekleştirilmiştir.

Hem donanım hem yazılım hem de ara yüz tarafı başarı ile gerçekleştirilen bu projeye farklı bir tat katması açısından bir oyun kavramı projeye eklenmiştir. Kale oyunu (futbol) ile iki farklı kullanıcının, kuralları önceden belirlenmiş bu oyunu eş zamanlı olarak keypad'ler aracılığı ile oynayabilmesi sağlanmıştır.

5 GÖREV DAĞILIMI

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

6. REFERANSLAR

- [1] *Arduino Mega 2560*, <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [2] *OpenSceneGraph*, <http://www.openscenegraph.org/>
- [3] *PIDLibrary*, <http://playground.arduino.cc/Code/PIDLibrary>
- [4] *TouchScreen.h*, <https://github.com/adafruit/Touch-Screen-Library>
- [5] *Serial.read()*, <https://www.arduino.cc/en/Serial/Read>
- [6] *Serial.write()*, <https://www.arduino.cc/en/Serial/Write>
- [7] *RS-232*, <http://www.teuniz.net/RS-232/>