

**T.C.**

**FIRAT ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ**

**İŞ YERİ EĞİTİMİ**

**BÖLÜMÜ**

**:** YAZILIM MÜHENDİSLİĞİ

**NUMARASI :**

14545026

**ADI ve SOYADI**

**:** MUHAMMED BAHADDİN DEMİR

24 / 09 / 2018 tarihinden 28/ 09 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** | . |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** 40 | |

01 / 10 / 2018 tarihinden 05 / 10 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  | - | - |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

08 / 10 / 2018 tarihinden 12 / 10 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

15 / 10 / 2018 tarihinden 19 / 10 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

22 / 10 / 2018 tarihinden 26 / 10 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  | 22 | 8 |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** 8 | |

29 / 10 / 2018 tarihinden 02 / 11 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

05 / 11 / 2018 tarihinden 09 / 11 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** | . |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** 40 | |

12 / 11 / 2018 tarihinden 16 / 11 / 2016 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  | - | - |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

19 / 11 / 2018 tarihinden 23 / 11 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

26 / 11 / 2018 tarihinden 30 / 11 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** | . |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** 40 | |

03 / 12 / 2018 tarihinden 07 / 12 / 2016 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  | - | - |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

10 / 12 / 2018 tarihinden 14 / 12 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

17 / 12 / 2018 tarihinden 21 / 12 / 2018 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** | . |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  |  |  |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** 40 | |

24 / 12 / 2018 tarihinden 29 / 12 / 2016 tarihine Kadar bir aylık çalışma programı

|  |  |  |  |
| --- | --- | --- | --- |
| **HAFTALAR** | **YAPILAN ÇALIŞMALAR** | **SAYFA NO** | **SAAT** |
| **Pazartesi** |  |  |  |
| **Salı** |  |  |  |
| **Çarşamba** |  |  |  |
| **Perşembe** |  |  |  |
| **Cuma** |  | - | - |
| **Cumartesi** |  |  |  |
| **ONAY** | Kaşe / İmza | **Toplam Saat :** | |

|  |
| --- |
| **STAJ İŞLEMLERİ TAKİP KILAVUZU**   1. Staj için gerekli işlemler, öğrencinin Fırat Üniversitesi Teknoloji Fakültesi ilgili Bölümünün Web Sayfasından elde edeceği dilekçeler ile başlatılır. Öğrenci, **Başvuru Dilekçesi (*Form-1*)** ileStaj yapmak istediği Firmaya/Kuruma başvurur. 2. Staj isteği kabul edilen öğrenci, **Firma/Kurum Bilgi Formunu** **(*Form-2*)** Firmaya/Kuruma onaylatarak **Bölüm** Staj **Komisyonuna** teslim eder. 3. Bölüm Staj Komisyonu toplanarak, **kurumun uygunluğuna** karar verir. 4. Staj **yeri uygun bulunmayan öğrenciler**, ikinci bir kurum bulmak için **1. maddeden itibaren işlemlerini tekrar başlatırlar.** 5. Öğrenci STAJ Komisyonuna **Kabul belgesi**, orijinal imzalı ve mühürlü halini teslim eder. 6. 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanunu’na göre öğrencilerin iş ve meslek hastalıklarına karşı sigortalanması Fırat Üniversitesi tarafından yapılacaktır. 7. Öğrenci sigorta işlemlerini fakültede bulunan ilgili memurdan takip eder ve satajı ile paralel olarak sigortasının başlamasını sağlar. Sigorta evrakı bölüm sekreterliğine teslim edilir. 8. Öğrenci kendisinde muhafaza ettiği **Staj Dosyası** ile birlikte ilgili sömestre başlangıcında kurumuna giderek işlemlerini başlatır. 9. Tüm işlemlerin taja başlamadan 1 hafta önce tamamen bitirilmelidir. 10. Stajını Fırat Üniversitesi Teknoloji Fakültesi **Staj**  Yönergesi kapsamında tamamlayan öğrenci, **hazırladığı** ve **onayları (imza, kaşe vs.) tamamlanmış Staj Dosyasını** varsa ekleri ile birlikte **Bölüm Staj Komisyonuna** teslim eder.   \*Staj hakkında detaylı bilgi için **Staj Yönergesine** başvurulabilir. |

**T.C.**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**STAJ YÖNERGESİ**

**BİRİNCİ BÖLÜM**

**Amaç, Kapsam, Dayanak ve Tanımlar**

**Amaç**

**MADDE 1-** (1) Stajın amacı;

a) Öğrencilere lisans programlarıyla ilgili işyerlerini yakından tanıtmak,

b) Öğrencilerin öğrenim süreleri içinde kazandıkları bilgi ve deneyimlerini pekiştirmek için uygulama yaparak lisans programlarına ilişkin bilgi ve görgülerini arttırma imkânı sağlamak,

c) Öğrencilere almış oldukları teorik bilgileri kullanabilme ve uygulamaya aktarma becerisini kazandırmak,

ç) Öğrencilere Staj yaptıkları kurum veya kuruluşun görevli personeli ve müşterileri ile uyumlu çalışma ve iyi iletişim kurabilme alışkanlığını kazandırmak,

d) Öğrencilerin sektörde yaşanan teknolojik gelişmeleri tanımalarını sağlamak,

olarak tanımlanmıştır.

**Kapsam**

**MADDE 2-** (1) Bu yönerge, Fırat Üniversitesi Teknoloji Fakültesi öğrencilerinin; yurtiçi/yurtdışı, kamuya/özel sektöre veya sivil toplum örgütlerine ait kurum veya kuruluşlarda yapacakları işyeri eğitimleriyle ilgili uygulama esaslarını kapsar.

**Dayanak**

**MADDE 3-** (1) Bu yönerge, 3308 Sayılı Mesleki Eğitim Kanununa, 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanununa ve Fırat Üniversitesi Önlisans ve Lisans Eğitim-Öğretim Yönetmeliği hükümlerine dayanılarak hazırlanmıştır.

**Tanımlar**

**MADDE 4-** (1) Bu yönergede adı geçen;

a) Bölüm İşyeri Eğitimi Komisyonu: İlgili lisans diploma programının işyeri eğitimi işlemlerini yürütmek, yönetmek ve aynı zamanda işyeri eğitimini değerlendirmek üzere, bölüm kurulu tarafından oluşturulan, ilgili lisans programındaki 3 (üç) öğretim üyesinden oluşan komisyonu,

b) Denetçi Öğretim Elemanı: staj izlenmesi ve değerlendirilmesi için işyerlerine giderek staj alan öğrenciyi denetlemekle görevli kişiyi,

c) Fakülte Staj Kurulu: Dekan veya dekan yardımcısı başkanlığında, bölüm başkanları ve Fakülte-Sanayi Koordinatöründen oluşan kurulu,

ç) Fakülte-Sanayi Koordinatörü: Dekan tarafından görevlendirilen ve öğrencilerin bu yönerge doğrultusunda işyeri eğitimi yapmalarını koordine eden öğretim elemanını,

d) İşyeri: Öğrencinin eğitim gördüğü lisans programının gereklerine ve öğrenme çıktılarına uygun olarak uygulama yapabileceği, işyeri eğitiminin gerektirdiği fiziksel ortam, personel ve diğer nitelikleri taşıyan; öğrencilerin eğitimleri süresince kazandıkları bilgi ve deneyimlerini, işyeri eğitimi yoluyla pekiştirdikleri, mal veya hizmet üreten yurtiçi/yurtdışı kamuya, özel sektöre veya sivil toplum örgütlerine ait kurum veya kuruluşları,

e) Staj: Öğrencilere, bir yarıyılda (2 sınıf veya 3. sınıf öğrenim gördükleri lisans programı ile ilgili işyerlerindeki faaliyetlere katılmalarını sağlayarak, öğrenim süresince aldıkları bilgilere dayalı uygulama becerisi kazandırma çalışmalarını,

f),Staj Yetkilisi: İşyeri yöneticisi tarafından görevlendirilmiş ve öğrencinin eğitim alacağı işyerindeki lisans derecesine sahip kişiyi,

g) Staj Yöneticisi: Staj eğitiminin uygulanacağı yurtiçi/yurtdışı kamu, özel sektör veya sivil toplum örgütlerine ait kurum veya kuruluşları adına imzalamaya yetkili kişiyi, ifade eder.

**İKİNCİ BÖLÜM**

**Görev, Yetki ve Sorumluluklar**

**MADDE 5-** (1) Dekan, Staj organizasyonun en üst yetkilisi ve yöneticisidir.

(2) Fakülte Dekanının;

a) Fakülte staj Kurulunu oluşturmak ve Fakülte-Sanayi Koordinatörünü görevlendirmek,

b) Fakülte Staj Kurulunun yaptığı değerlendirme sonuçlarını Öğrenci İşleri Daire Başkanlığına bildirmek,

yetkileri arasındadır.

**Fakülte İşyeri Eğitimi Kurulunun Görevleri**

**MADDE 6:** (1) Fakülte Staj Kurulunun;

a) Staj verilecek işletmeleri belirlemek,

b)Staj planlanmasını, koordinasyonunu ve uygulanmasını sağlamak,

c) Öğrencilere Staj yeri temini hususunda ilgili birimlerle işbirliği yapmak ve gerekli bilgi sistemini oluşturmak,

d) Staj yapacak öğrencilerin sigorta işlemlerini planlamak ve takip etmek,

e) Öğrencileri, bu yönerge hükümleri doğrultusunda işyeri eğitimi ile ilgili bilgilendirmek,

f) Staj kullanılacak evrakları düzenlemek,

g) Öğrencilerin Staj yapacakları işyerlerine dağılımlarını yapmak,

ğ) Öğrencilerin Staj süresince hazırlayacakları çalışma faaliyetlerinin kapsamını ve Staj Dosyasının içeriğini belirlemek ve öğrencilere duyurmak,

h) Staj yönetmelik, yönerge usul ve hükümlerine uygun olarak düzenli sürdürülmesi için gerekli önlemleri almak,

ı) Staj ilgili olarak doğabilecek aksaklık ve problemleri çözmek,

i) Bölüm Staj komisyonlarınca hazırlanan Staj Sicil değerlendirme sonuçlarını karara bağlamak, görevleri arasındadır

**Fakülte-Sanayi Koordinatörünün Görevleri**

**MADDE 7:** (1) Fakülte sanayi koordinatörünün;

a) Staj konusunda fakülte ile işyeri arasındaki iletişimi ve koordinasyonu sağlamak,

b) Stajla ilgili oluşabilecek sorunları çözmek, çözülemeyen hususları fakülte Staj kuruluna iletmek, görevleri arasındadır

**Bölüm İşyeri Eğitimi Komisyonunun Görevleri**

**MADDE 8:** (1) Bölüm Staj Komisyonunun;

a) Satj verilecek işletmeleri belirlemek,

b) Öğrencilerin işyeri eğitimi süresince hazırlayacakları çalışma faaliyetlerini ve Staj dosyasının içeriğini belirlemek,

c) Her yarıyıl başında o yarıyıl için gerekli hazırlıkları yapmak, yarıyıl sonlarında teslim edilmiş Staj dosyalarını ve öğrencileri değerlendirmek,

ç) Fakülte işyeri eğitimi kurulunun belirleyeceği görevleri yerine getirmek,

d) İlgili mevzuat hükümleri kapsamında işyeri eğitiminin yürütülmesini sağlamak,

e) İşyeri eğitimiyle ilgili oluşabilecek sorunları çözmek, çözülemeyen hususları fakülte staj kuruluna iletmek,görevleri arasındadır.

**Denetçi Öğretim Elemanının Görevleri**

**MADDE 9-** (1) Denetçi Öğretim Elemanının;

a) Öğrencilerin stajdaki eğitimlerini denetlemek,

b) Denetim sonucunda Staj Denetim Formunu Bölüm staj Komisyonuna teslim etmek,

görevleri arasındadır.

**İşyeri Yöneticisinin Sorumlulukları**

**MADDE 10-** (1) İşyeri Yöneticisinin;

a) Staj için öğrenci kabul edecek işyerinin kendi personeline sağladığı konaklama, beslenme ve sosyal imkânlardan öğrencilerin de yararlanması için gerekli çabayı göstermek,

b) Öğrenci için gerekli oryantasyon çalışmalarını gerçekleştirmek,

c) Öğrencinin staj bu yönerge esaslarına ve işyeri kurallarına göre yapabilmesi için gerekli şartları taşıyan bir İşyeri Eğitimi Yetkilisini görevlendirmek, sorumlulukları arasındadır.

**Staj Yetkilisinin Görevleri**

**MADDE 11-** (1) Staj Yetkilisinin;

a) Öğrencilerin fakültede almış olduğu teorik ve uygulama esaslı bilgi ve becerilerin iş ortamındaki uygulama çalışmaları ile pekişmesini sağlamak,

b) Öğrenciye haftalık çalışma planı ve sorumluluğu yüklemek,

c) Öğrencinin işyerindeki sorumluluğunu üstlenmek,

ç) Öğrencilere meslek yeterliliğini ve disiplinini kazandırmak,

d) Öğrencinin haftalık hazırladığı işyeri eğitimi dosyasını ve yaptığı faaliyetleri değerlendirmek,

e) İşyeri eğitimi bitiminde, işyeri eğitimi değerlendirme formunu hazırlayıp Bölüm staj Komisyonuna kapalı zarf içinde gizli olarak göndermek, görevleri arasındadır.

**Öğrencilerin Görev ve Sorumlulukları**

**MADDE 12-** (1) Öğrenciler, staj Kabul formunun imzalanan işyerlerinde yapmak zorundadırlar.

(2) Öğrenciler, staj süresince işyeri eğitimi haftalık çalışma planını uygulamakla yükümlüdürler.

(3) Öğrenciler, staj yapacakları işyerinin kurallarına, mevzuatına ve Yükseköğretim Kurumları Öğrenci Disiplin Yönetmeliğine uymak zorundadırlar.

(4) Öğrenciler, günlük çalışma kayıtlarını içeren haftalık çalışma raporunu, sonraki haftanın ilk mesai günü bitimine kadar staj yetkilisine vermek zorundadırlar.

(5) Öğrenciler, işyerinden izinsiz ayrılamaz. staja devam zorunluluğu vardır. Hastalık, birinci derece yakınlarının vefatı veya benzeri acil durumlar dışında izin kullanılamaz. İzinli olarak ayrılması gereken durumlarda işyeri eğitimi yetkilisi tarafından onaylı izin formu düzenlenir ve staj dosyasında sunulur.

(6) Öğrenciler, işyerindeki sendikal etkinliklere katılamazlar.

**ÜÇÜNCÜ BÖLÜM**

**Stajla İlgili Düzenlemeler**

**Stajın Özellikleri**

**MADDE 13-** (1) Öğrenciler, staj son sınıfın güz (2. sınıfl) yaz dönemınde (3. sınıf) yaz dönemınde için devam zorunluluğu gerektiren dersi bulunmaması halinde gerçekleştirir. Öğrencinin 2. veya 2.sınıfında yazdönemınde 20+20 =40 işgünü staj yaparlar

**Staj Süresi**

**İşyeri Eğitimi Başvurusu**

**MADDE 14-** (1) Fakülte-Sanayi Koordinatörü, İşyeri Eğitimi Yetkilisi ve işyeri eğitimi yapacak olan öğrenci tarafından imzalanan İşyeri Eğitimi Sözleşmesi 3 (üç) nüsha olarak çoğaltılır ve taraflara verilir.

(2) Öğrenciler bütün derslerinden başarılı olsalar bile, işyeri eğitimi yapacakları süreler için kayıt yaptırmak ve katkı paylarını ödemek zorundadırlar.

**İşyeri Eğitimi Dosyası Oluşturma**

**MADDE 15-** (1) Her öğrenci, “staj dosyası” hazırlamak zorundadır.

(2) Staja başlayan öğrenci, staj Dosyasını ilgili bölüm sekreterliğinden temin eder.

(3) Staj Dosyası; öğrenci bilgilerinin bulunduğu kapak, staj dosyası, öğrenci tarafından hazırlanmış imzalı ve onaylı haftalık raporlar, işyeri eğitimi denetim formları ve staj yetkilisi tarafından kapalı zarf içerisinde yer alan dönem sonu stasj sicil fişi oluşur.

(4) Staj Dosyası işyeri eğitimi süresince Staj Yetkilisinde bulunur, Staj sonunda değerlendirme amacıyla Bölüm staj Komisyonuna teslim edilir.

**İşyeri Eğitimi Dosyasının Teslimi**

**MADDE 16-** (1) Öğrenci, staj dosyasını işyeri eğitiminin bitiş tarihinden itibaren en geç bir hafta içerisinde Bölüm staj Komisyonuna elden teslim eder veya iadeli taahhütlü posta yoluyla gönderir.

(2) Staj dosyasını, süresi içerisinde teslim etmeyen öğrencinin stajdan geçersiz sayılır.

**Hastalık ve Kaza Halleri**

**MADDE 17-** (1) 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanunu’na göre öğrencilerin iş ve meslek hastalıklarına karşı sigortalanması Fırat Üniversitesi tarafından yapılacaktır. Ancak, iş yerinin kusurundan dolayı meydana gelebilecek iş kazaları ve meslek hastalıklarından, 3308 sayılı Kanunun 25. Maddesine göre, işveren sorumludur.

(2) staj sırasında hastalanan ve resmi kurumlarca belgelenmek üzere hastalığı 7 (yedi) günden fazla süren veya herhangi bir kazaya uğrayan öğrencinin adı, soyadı, hastalığın ve kazanın mahiyeti, işyeri tarafından Fakülte-Sanayi Koordinatörüne bildirilir. Bu durumlarda Sosyal Güvenlik Kurumu hüküm ve uygulamaları geçerli olup, telafi konusunu Fakülte İşyeri Eğitimi Kurulu belirler.

**İÇİNDEKİLER**

|  |  |  |
| --- | --- | --- |
|  | | **Sayfa N o** |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| **ONAY** | Kaşe / İmza | |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Kurum Hakkında Bilgi. | | |
| İş Yeri Bilgisi  İş yeri: Nitra Oyun Yazılım Ltd. Şti  İş yeri adresi: KOSGEB Teknoloji Geliştirme Merkezi İstanbul Aydın Üniversitesi H Blok 6022 Beşyol Mahallesi Karadeniz Sokak Küçükçekmece/İstanbul  İş yeri Tel: 0212 422 5007  İş Yeri Tarihçesi  Mobilenter Dönemi    Kurucu ekibimizin oyun geliştirme macerası 2006 yılında Mobilenter Mobil Oyun Stüdyoları'nı kurarak başladı. Mobilenter, sektörün ülkemizdeki ve dünyadaki ilk firmalarından birisidir. 2010 sonuna vardığında, firma 21 oyunluk bir portföye ve 3 ofise sahip olmuş durumdaydı. Ofisler İstanbul, Lancaster/İngiltere ve North San Jose/ABD lokasyonlarında bulunuyordu. O dönemde dünyanın en büyük appstore sitesi olan Getjar'ın tarihinde en çok indirilen 10 oyundan 2 tanesi Mobilenter oyunlarıdır. Firmanın oyunları 11 dilde hazırlıyor ve 46 ülkede 100'ün üzerinde operatörde yayınlanıyordu. İlerleyen yıllarda Java telefonlarının piyasadan kalkması sürecinde Mobilenter küçülmeye gitti ve yurtdışındaki ofislerini kapattı. Ekip 2014 Nisan'ında eski Java oyunlarının Android versiyonlarını yayınlamak üzere Nitra Oyun Yazılım Ltd. Şti. firmasını kurdu.  NİTRA DÖNEMİ  Kurucu ekibimiz 2007 başından 2016 sonuna kadar geçen 11 yıllık süreçte, 1 OpenGL ES benzeri grafik arayüzü, 13 oyun motoru ve 26 oyun geliştirip yayınladı. Nitra, küçük ölçekli bir firma olmasına karşın, bugün kendi alanınında dünyanın en büyük know-how'ına sahip firmalarından bir tanesidir.  2016 yılının Ağustos ayında Bilim, Sanayi ve Teknoloji Bakanlığı AR-GE Yönetmeliği'ni yayınlamasını takiben, KOSGEB tarafından Nitra'ya Türkiye'nin bu sektördeki ilk AR-GE desteği tahsis edildi. Bu destek kapsamında İstanbul Aydın Üniversitesi merkez kampüsü içerisinde yer alan Teknoloji Geliştirme Merkezi'nde bir ofis firmaya ücretsiz olarak verildi. Oyunların motor yazılımı dahil tamamını geliştirmesi sebebiyle, Türkiye'de bu desteği almaya hak kazanan ilk oyun yazılım firması Nitra Oyun Yazılım Ltd. Şti. olmuştur. | | |
| **Tarih:** 24 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 1 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Kurum Hakkında Bilgi. | | |
| VE GAMELAB ISTANBUL KURULUYOR!  AR-GE Yönetmeliği'nin 5g. maddesi uyarınca, oyun yazılımlarını geliştirirken harici bir oyun motoru kullanılması durumunda ürün AR-GE projesi niteliğini kaybetmektedir. Ancak ülkemizde oyun motoru geliştirme konusunda tecrübeli çok az sayıda firma ve uzman bulunmaktadır. Teknoloji Geliştirme Merkezi içerisindeki yeni ofisine taşınan kurucu ekibimiz, ülkenin bu sorununu aşmakta yol gösterici olabilmek amacıyla üniversiteler ile işbirliğine giderek Gamelab İstanbul'un temelini attı.  KOSGEB'in İstanbul Aydın Üniversitesi Teknoloji Geliştirme Merkezi içerisinde ücretsiz olarak tahsis ettiği ofiste, Nitra yöneticileri ve 24 üniversite öğrencisinin katılımıyla kurulan Gamelab Istanbul, 13 Şubat 2017 günü Türkiye'nin en büyük mobil oyun yazılım ekibi olarak faaliyetine başladı.  Kurumun Amaçları  Gamelab Istanbul olarak oyun yazılımları ve oyun motorları geliştiriyoruz. Kuruluş amacımız ve genel hedefimiz, uzun vadede AAA kalitesinde ürünlerden oluşan büyük bir oyun portföyü kurmaktır. Gamelab Istanbul'un 2017 başında faaliyete geçmesine karşın, kurucu ekibimiz mobil oyun sektörünün dünyadaki en eski ve tecrübeli ekiplerinden birisidir. Kurucu ekibimizin tecrübesini üniversite öğrencilerinden oluşan geliştirici grubumuzun dinamizm ve tutkusuyla birleştirerek ülkemizin en büyük mobil oyun geliştirme ekibini Gamelab Istanbul çatısı altında faaliyete geçirdik.  SplashCanvas2 | | |
| **Tarih:** 24 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 2 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Tanışma, şirket ve çalışma sistemi hakkında ve başlangıç için gerekli anlatımların yapılması. | | |
| Stajımın ilk gününde şirket çalışanları ve diğer stajyerlerle tanışıldı. Şirketin çalışma yapısından bahsedildi. Öğren-Yap-Öğret yapısında çalışan şirket ilk olarak bizleri bir eğitime dahil etti. Daha sonrasında ise gerçek projelere dahil edileceğimiz belirtildi. Böylece gerçek bir projede çalışmanın nasıl olduğunu öğreneceğimiz belirtildi. Staj süremizi doldurduktan sonra yeni gelecek olan stajyerlere eğitim vereceğimiz yapıdan bahsedildi.  Daha sonra günlük girişlerimizin tutulacağı hr.gamelab adresini nasıl kullanacağımız hakkında bilgiler verildi. Ve geniş geliştirici gruplarının kullandığı Git sistemi açıklandı. Git sistemi sayesinde proje üzerinde birden fazla kişi eş zamanlı olarak çalışabiliyor. Aynı dosya üzerinde birden fazla kişi değişiklik yapsa bile system bu değişiklikleri uygun bir şekilde yönetiyor ve böylece projemiz düzgün bir şekilde ilerliyor. Ayrıca dosya paylaşımı işlemini de kolaylaştırıyor.  Git sisteminden firmanın eğitim için kullanacağı oyun motorunun nasıl indirileceği ve açılacağı gösterildi. Oyun motorunun bulunduğu projeyi Android Studio üzerinde açabiliyoruz. Motorun temel kodları Java ve C++ dilleri kullanılarak yazılmıştır. İlk başlarda tamamı Java dili ile kodlanan oyun motoru bir süre sonra Android Studio için C++ dili ile kodlama desteği gelmesi ile bir kısmı C++ diline çevrilmiş. Bu sayede oyun motoru sadece Java dili ile yazılan sürümden daha performanslı çalışmaktadır.  Oyun motorunun hem C++ hemde Java dillerinde yazılması sebebiyle Android Studio için gerekli NDK dosyalarını tanımlamamız gerekiyor. Bu dosyalar bilgisayarımızda gerekli olan konuma atıldı ve Android Studio ayarlarımızdan NDK dosyalarının konumu belirtildi. Bu dosyalar sayesinde Android Studio içerisinde C++ dilinde kod yazabiliyoruz.  Oyunumuzun kodlarının yazılacağı GameCanvas.cpp ve GameCanvas.h dosyaları içinde kullanacağımız temel metodlar hakkında bilgiler aldık. GameCanvas.h dosyasında GameCanvas.cpp dosyasında kullanacağımız gerekli değişken ve fonksiyonların tanımlamaları yapıldı. GameCanvas.cpp dosyasında ise bu tanımlamaları kullanarak kodlarımızı yazıyoruz. Burada update() ve draw() metodlarının nasıl kullanıldığını öğrendik.  Bu fonksiyonlar oyunda tekrar tekrar çalışan kısımlar. uptate() kısmında gerekli değişiklikler sağlanıyor ve daha sonrasında draw() kısmı ile ekrana nesnelerimiz çiziliyor. Oyun boyunca saniyede 15 ila 40 defa -oyun motorumuzun hızına bağlı olarak- bu fonksiyonlar çalışıyor. Böylece oyunumuzun döngüsü oluşmuş oluyor. | | |
| **Tarih:** 24 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 3 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Modeller, kamera ve ışığın sahneye yüklenmesi. Ayarlarının yapılması. | | |
| Oyunumuzun kodlarının yazılacağı GameCanvas.cpp ve GameCanvas.h dosyaları üzerinde çalışmaya başlanıldı. GameCanvas.h dosyasında GameCanvas.cpp dosyasında kullanacağımız gerekli değişken ve fonksiyonların tanımlamaları yapıldı. GameCanvas.cpp dosyasında ise tanımlamaları kullanarak kodlarımız yazıldı.  Başlangıçta oyunumuzun yükleniyor veya oyun içinde olduğunu bilebilmek için iki adet değişken tanımlandı. GameCanvas.h dosyamızda tanımladığımız değişkenlere GameCanvas.cpp dosyamızda değerleri verildi.  Const int GameCanvas::GAMESTATE\_LOADING = 0;  Const int GameCanvas::GAMESTATE\_GAME = 1;  Öncelikle stadyum, oyuncu ve top için modelleri yüklendi. Bunun için .h dosyamızda loadAssset adlı bir metot tanımlandı ve .cpp dosyamızda metodumuzun içine gerekli kodlar yazıldı.  Daha sonra bu modelleri ekrana nasıl çizdireceğimizi öğrendik. Bunun için draw() metodu içerisinde bize yardımcı olması için drawSceen() adlı bir metod tanımlandı. drawSceen() metod draw() metodu içinden çağrıldı. Ve kodumuzu çalıştırıldı. Sonuç olarak ekran görüntümüz şu şekilde oluyor.  1  Yüklediğimiz modellerin yönlerini kavramak için kameranın konum ve bakış açısında değişiklikler yapıldı. | | |
| **Tarih:** 24 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 4 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Animasyonlar ve frame’lerin açıklanması ve çalıştırılması. | | |
| İlk olarak oyunumuzda bulunan tüm oyuncuların eklenmesi için gerekli işlemler yapıldı. Eklenmiş olan bir oyuncu modeli yerine 22 adet player modeli eklendi.  for (int t=0; t<teamnum; t++) {  for (int p=0; p<playernum; p++) {  ofPushMatrix();  ofTranslate(px[t][p], py[t][p],  [t][p]);  ofRotateZ(pr[t][p]);  model\_player.getAnimation(0).setFrameNo(playerframeno[t][p]);  model\_player.update();  model\_player.drawFaces();  ofPopMatrix();  }  }  Bunu 11 oyuncu sağ tarafta, 11 oyuncu ise sol tarafta olacak şekilde ayarlandı. Ayrıca tüm oyuncuların yönleri sahanın merkezine bakacak şekilde ayarlaması yapıldı. Bunu sağlamak amacı ile başlangıç değerlerini tanımlayacağımız initializeVariables isimli yeni bir metot oluşturuldu. Bu metot ile oyunumuz başlamadan önce gerekli değerlerin atamaları yapıldı. Bu metot içinde oyuncuların yönleri ve konumları atandı. Ayrıca oyuncuların sahip olduğu animasyonlar da belirtildi.  for (int t=0; t<teamnum; t++) {  for (int p=0; p<playernum; p++) {  px[t][p] = ((t \* -2) + 1) \* 50;  py[t][p] = (-100 + (p \* 20));  pz[t][p] = 0;  pr[t][p] = t \* 180;  playeranimationno[t][p] = ANIMATION\_IDLE;  playerframeno[t][p] = animationFirstFrameNo[playeranimationno[t][p]];  }  }  Oyuncumuz dört temel animasyona sahip. Bunları ANIMATION\_IDLE, ANIMATION\_RUN, ANIMATION\_WALK VE ANIMATION\_CELEBRATE olarak tanımlandı. Animasyonlarımızı oynatmak için öncelikle player modelimize loadAssset() metodunda bir animasyon döngüsü eklendi.  model\_player.getAnimation(0).makeFramedAnimation(71);  initializeVariables() metodumuza animasyonlar için gerekli Frame numarası atamaları yapıldı.  Böylelikle oyuncularımızın animasyonu oynamış oluyor. | | |
| **Tarih:** 25 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 5 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D ve 3D özellikleri. Syntax özellikleri. Dokunmatik ekrandan bilgi alınması. | | |
| 2D ve 3D arasındaki farklar incelendi. Bu farklardan en önemli olanı ise 2D’de baktığımız bir 3D ekranın koordinatlarının aynı olmamasıdır. Bunun için gerekli heseplamalar yapılmalıdır.  Telefonumuzda ekranın koordinat sistemi telefonu dik tuttuğumuzda sağ üst köşede kalmaktadır. Bu nokta (0, 0) noktasıdır. Ekranımızda sadece pozitif koordinatlar bulunmaktadır.    Bunların dışında kodlarımız yazarken dikkat edilmesi gereken hususlardan bahsedildi.   * Class başlığı soldan boşluk verilmeyecek şekilde yazılmalıdır. * Her alt parçanın solunda üst parçaya göre 1 TAB (4 boşluk karakteri) daha fazla boşluk bırakılmalıdır. * Operatör işaretlerinin önünde ve arkasında 1 boşluk bırakılmalıdır. Örneğin if (a < b) …; * Üstteki 3 numaralı kuralın istisnası olarak, for döngüsü tanımlanırken, for operatörlerinin yanlarında boşluk bırakılmamalı, diğer ek operatörler yine boşluklu yazılmalıdır. Örneğin: for (int k=0; k<=m + 5; k++) n = k; * Metotların aralarında bir satır boşluk bırakılmalıdır. * Satır aralarında boşluk bırakılmamalıdır. * Açıklama satırlarının başına konan çift slash işareti (//) satırın en başına konulmalıdır. * System.out.println (Java), cout (C++), Log.i() gibi komutlarla yapılan testlerde “Kontrol (no)” ibaresi kullanılmalıdır. Örneğin System.out.println(“Kontrol 1”); * “+= 1” ifadesi yerine “++” ifadesi kullanılmalıdır. Örneğin x++; * “-= 1” ifadesi yerine “--” ifadesi kullanılmalıdır. Örneğin x--;   Ardından dokunmatik ekrandan bilgi almak için mevcut olarak tanımlı fonksiyonlardan bahsedildi. Bu fonksiyonlardan en çok kullanılanlar,   * touchDown(), parmağımızı ekrana değdirdiğimizde, * touchMoves(), parmağımızı ekranda hareket ettirdiğimizde, * touchUp(), parmağımızı ekrandan kaldırdığımızda, * touchDoubleTab(), ekrana çift dokunduğumuzda, çalışan fonksiyonlardır.   Ayrıca oyuncunun saha sınırlarından dışarı çıkması engellendi.  Rakip takımın forma rengi değiştirildi. | | |
| **Tarih:** 26 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 6 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyuncu modellerinin yönlerinin kontrol edilmesi. Oyuncunun yer değiştirmesi. | | |
| Bugün bizim tarafımızdan yönlendirilen aktif oyuncuyu ekranda çekilen yöne doğru haraket edebilmesi için gerekli olan checkMovement() isimli yeni bir metod yazıldı.  void GameCanvas::checkMovement(){  if (!(touchMoveX == 0 && touchMoveY == 0 && touchDownX == 0 && touchMoveY == 0)){  pr[ownTeam][ap]= 180 - ofRadToDeg(atan2((touchMoveY - touchDownY), (touchMoveX - touchDownX)));  }  }  Bu metot touchDown() ve touchMove() fonksiyonlarından parmağımızı ekrana değdirdiğimiz konumlar alındı ve bunları kullanarak oyuncunun dönmesi gereken açı elde edildi. Bulunan açı oyuncunun açısına eşitlendi. Böylece oyuncunun parmağımızı çektiğimiz yönde dönmesi sağlandı. Program ekrandaki konumları touchDown() ve touchMoved() fonksiyonları ile elde etmemizi sağladı.  void GameCanvas::touchDown(int x, int y, int id) {  touchDownX = x;  touchDownY = y;  }  void GameCanvas::touchMoved(int x, int y, int id) {  touchMoveX = x ;  touchMoveY = y ;  }  Ardından oyuncunun baktığı yönde 40 birim ilerlemesini sağlamak için movePlayer() isimli bir diğer fonksiyonu yazıldı.  if (abs(targetPointX[teamNo][playerNo] - px[teamNo][playerNo]) > abs(playerSpeed \* cos(ofDegToRad(pr[teamNo][playerNo])))) {  px[teamNo][playerNo] -= playerSpeed \* cos(ofDegToRad(pr[teamNo][playerNo]));  } else {  px[teamNo][playerNo] = targetPointX[teamNo][playerNo];  }  if (abs(targetPointY[teamNo][playerNo] - py[teamNo][playerNo]) > abs(playerSpeed \* sin(ofDegToRad(pr[teamNo][playerNo])))) {  py[teamNo][playerNo] -= playerSpeed \* sin(ofDegToRad(pr[teamNo][playerNo]));  } else {  py[teamNo][playerNo] = targetPointY[teamNo][playerNo];}  Oyuncuya bir targetPoint atanarak X ve Y düzlemleri için bu atamaları initializeVariables() metoduna eklendi.  targetPointX[t][p] = px[t][p];  targetPointY[t][p] = py[t][p]; | | |
| **Tarih:** 27 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 7 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyuncunun yer değiştirmesi. Animasyonların harekete göre uyarlanması. | | |
| Oyuncumuzun her seferinde targetPoint değerini değiştirmek için touchUp() fonksiyonuna ekleme yapıldı. Böylece oyuncumuz baktığı yönde 40 birim ilerliyor ve duruyor.  targetPointX[ownTeam][ap] = px[ownTeam][ap] - (40 \* cos(ofDegToRad(pr[ownTeam][ap])));  targetPointY[ownTeam][ap] = py[ownTeam][ap] - (40 \* sin(ofDegToRad(pr[ownTeam][ap])));  Şu ana kadar oyuncumuz çektiğimiz yöne dönüyor ve bunun yanında 40 birim o yönde ilerliyor ve duruyordu. Bugün oyuncumuza atanan animasyonların bu hareketlere uyarlanması gerçekleştirildi. Oyuncunun animasyonları oynatması için playAnimation() adlı bir metot tanımlandı.  void GameCanvas::playAnimation() {  for (int t=0; t<teamnum; t++) {  for (int p=0; p<playernum; p++) {  if (targetPointX[t][p] != px[t][p]) {  if (playeranimationno[t][p] != ANIMATION\_RUN) {  playeranimationno[t][p] = ANIMATION\_RUN;  playerframeno[t][p] = animationFirstFrameNo[playeranimationno[t][p]];  }  } else {  playeranimationno[t][p] = ANIMATION\_IDLE;  playerframeno[t][p] = animationFirstFrameNo[playeranimationno[t][p]];  }  }  }  for (int t=0; t<teamnum; t++) {  for (int p=0; p<playernum; p++) {  playerframeno[t][p]++;  if (playerframeno[t][p] >= animationLastFrameNo[playeranimationno[t][p]]){  playerframeno[t][p] = animationFirstFrameNo[playeranimationno[t][p]];  }  }  }  }  İlk döngümüzde oyuncular için uygulanacak animasyonlar atıldı. İkinci döngümüzde ise bu animasyonları Frame numaralarına göre sırayla oynatıldı. Çalışması için update() fonksiyonumuz düzenlendi.  void GameCanvas::update() {  if (gamestate == GAMESTATE\_LOADING) {  initializeVariables();  loadAsset();  }  else if (gamestate == GAMESTATE\_GAME) {  checkMovement();  playAnimation();  }} | | |
| **Tarih:** 28 / 09 / 2018 | Kaşe / İmza | **Sayfa No:** 8 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Aktif oyuncu hariç kalan oyuncuların hareket ettirilmesi. | | |
| Dün yaptığımız değişiklikler sayesinde artık oyuncumuz çektiğimiz yöne giderken koşma animasyonu ile gitmektedir. Bu sayede doğal bir şekilde hareket etmiş oldu.  Oyuncu saha sınırlarını geçmeye çalıştığı zaman koşma animasyonunu çalışmaya devam ediyordu. Oyuncu saha sınırlarını geçmeye çalıştığı zaman koşma animasyonunun durdurulmasını sağlandı.  Aktif oyuncu hariç diğer oyuncuların konumlarının da değişmesi üzerine çalışıldı. Hedefimiz aktif oyuncu hariç kalan oyuncuların baktıkları yönde ilerleyerek o yöndeki yarı sahanın ortasına gelmeleridir. Aynı zamanda aktif oyuncu da hareket ettirilebilinmelidir.  Bunu yapabilmek için oyuncularımızın targetPoint değerlerine atama yapılması gerekmektedir. Bunun için moveOtherPlayers() isimli yeni bir metot tanımlandı.  void GameCanvas::moveOtherPlayers(){  for (int t=0; t<teamnum; t++) {  for (int p=0; p<playernum; p++) {  if(!(t == 0 && p == 5)) {  targetPointX[t][p] = ((2 \* t) - 1) \* 196;  targetPointY[t][p] = py[t][p];  movePlayer(t, p);  }  }  }  }  Burada aktif oyuncu olarak seçtiğimiz player[0][5]’in konumunun değişmemesi için if koşulunu kullandık. Bu sayede sadece diğer oyuncuların hareket etmesi sağlandı.  Oyuncuların hareketleriyle beraber animasyonlarının oynaması için playAnimation() fonksiyonu gerekli işlevi yerine getirdi.  Ayrıca update() fonksiyonumuz da düzenlendi.  void GameCanvas::update() {  if (gamestate == GAMESTATE\_LOADING) {  initializeVariables();  loadAsset();  }  else if (gamestate == GAMESTATE\_GAME) {  checkMovement();  moveOtherPlayers();  playAnimation();  }} | | |
| **Tarih:** 01 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 9 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu**: Oyunun Android Tv ye Uyarlanması | | |
| Sonraki görevim, üzerinde çalıştığımız TASO 2018 uygulamasının Android TV lerde çalışmasını mümkün kılacak bir versiyon hazırlamamdı.  Android, yüksek çözünürlüklü televizyonlar gibi büyük ekranlı aygıtlarda çalışan uygulamalar için optimize edilmiş zengin bir kullanıcı deneyimi sunar. TV'deki uygulamalar, kullanıcılarınızı kanepelerinin konforundan memnun etmek için yeni fırsatlar sunar.  TV uygulamaları, telefonlar ve tabletler ile aynı yapıyı kullanır. Bu yaklaşım, Android için uygulama oluşturma hakkında zaten bildiklerime dayanarak yeni TV uygulamaları oluşturabileceğim veya mevcut uygulamalarımızı TV cihazlarında da çalışacak şekilde genişletebileceğim anlamına gelir. Bununla birlikte, TV için kullanıcı etkileşimi modeli, telefon ve tablet cihazlardan büyük ölçüde farklıdır. Uygulamanızı TV cihazlarında başarılı kılmak için, 10 fit öteden kolayca anlaşılabilecek yeni düzenler tasarlamanız ve yalnızca bir yön pedi ve bir seçim düğmesiyle çalışan gezinti sağlamanız gerekir.  Bugün bu bilgileri elde etmek ve projemizi Android TV’de çalıştırmak için Android’in geliştiricilere kılavuzluk ettiği <https://developer.android.com> adresindeki Android TV bölümlerini okudum.  Ekran Alıntısı2  Resimde görüldüğü gibi mevcut bir android uygulamasını Android TV’de çalıştırmak için gereken bileşenler belirtilmiştir. | | |
| **Tarih:** 02 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Android TV için manifest dosyasında bir activity oluşturmamız gerekiyor. Bu nedenle bugün manifest dosyasındaki komutların ve bölümlerin ne amaçla yazıldığını araştırdım ve manifest dosyasının ne amaçla yazıldığı ve içinde oluşturulan komutların ne amaçla yazıldığı gibi bilgileri öğrendim. AndroidManifest.xml dosyası, herhangi bir Android projesinin olmazsa olmaz dosyasıdır. Uygulamamızın tüm temel bilgileri AndroidManifest.xml içinde belirtilir. Bu dosya [xml](http://tr.wikipedia.org/wiki/XML) formatında oluşturulduğu için hem makina hem de insan tarafından okunabilir.  Aşağıda oyunumuzun manifest dosyasının bir bölümü görünmektedir.  Ekran Alıntısı3  Application etiketinde uygulamamızın temel değişkenlerini ayarlayabiliyoruz. Uygulamanın tema, simge, logo, açıklama ve gereken izinler gibi özelliklerini buradan ayarlayabileceğimiz gibi donanım hızlandırma, kullanıcı alanının yönetimi, yedekleme (backup) gibi bazı özellikler için de bu dosyayı kullanabiliyoruz. Örneğin, aşağıdaki kod parçasında da görüldüğü gibi **Hardware accelerated (Donanım hızlandırma) = true** şeklinde ayarladığımızda, uygulama artık bu özelliğe sahip oluyor.  Ekran Alınt4ısı | | |
| **Tarih:** 03 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Bugün uygulamamızın Android TV de çalışması için manifest dosyasında yapmam gereken değişiklikleri uygulamadım. Öncellikle uygulamamın çalışması için bazı özellikler eklemem gerektiğini öğrendim. Bunlar aşağıdaki resimdeki gibidir.  Ekran Alıntısı5  Android TV’nin telefon ve tabletlerdeki gibi dokumatik özelliği olmadığından touchscreen özelliğinin kapatılması gerekiyor. Manifest içinde leanback özelliğini kullanmam gerektiği için ise “android.sofware.leanback” komutunu kullandım.  Bazı özelikleri ise kullanmam gerekmiyordu. Mevcut sürümde kullanılan özellikler araştırdım ve Android TV için kullanmam gerekmeyen özellikleri tanıdım. Diğer komutlarla çakışmaları engellemek ve hatalara neden olmaması için kullanmam gerekmeyen özellikleri açıklama parantezine aldım.  Ekran Alıntısı6 | | |
| **Tarih:** 04 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| TV cihazlarında çalışması amaçlanan bir uygulama, TV için çalışacağını düşünmesi için bir etkinlik başlatmalıdır. Bunu yapmak için bir  CATEGORY\_LEANBACK\_LAUNCHER filtresi kullanır . Bu filtre, uygulamanızı TV için etkinleştirildiğini ve Google Play'in bir TV uygulaması olarak tanımlamasını sağlar. Bir kullanıcı, uygulamanızı TV ana ekranında seçtiğinde, bu amaç hangi etkinliğin başlatılacağını tanımlar. Bunları kendi uygulamamızda oluşturmamız için birçok araştırma yaptım Android TV için oluşturulmuş örnek uygulamalar buldum Android Studio da çalıştırıp manifest dosyalarını inceledim. Genel mantığı anlamaya çalıştım ve kendi projemize nasıl uygulayacağımı anlamaya çalıştım.  Ekran Alıntısı7  Daha sonra manifest içindeki TV için oluşturulmuş olan activity de uygulamanın TV için çalışması için gerekli diğer özellikleri oluşturmamız gerekiyor. Bunlar aşağıdaki gibidir.  Ekran Alıntısı8 | | |
| **Tarih:** 05 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Uygulamamızı TV üzerinde çalıştırmadan önce yapılasını gereken diğer şeyler ise SDK araçlarını 24.0.0 veya daha yüksek bir sürüme çıkarmam gerekiyordu. Ayrıca kullanacağım Android sürümünün ise API level 21 veya daha üstü olması gerekiyordu. Bu bilgilere android developer kılavuzunu okuyarak temin ettim. Ayrıca developer forumlarında ilgili başlıkları araştırdım.  Ekran Alıntısı9  Gerekli SDK araçlarını indirdim ve API level 27 olan bir Android TV emulatörü oluşturdum.  Ekran Alıntısı10 | | |
| **Tarih:** 08 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Bütün değişiklikleri ve tüm gereklilikleri oluşturdum. Oyunumuzu Android TV emülatöründe çalıştırmaya hazırdı.  Ekran Alıntısı12  Çalıştırırken bir sorunla karşılaştım. Uygulamayı çalıştırınca siyah ekran kalıyordu ve bir tepki alamıyordum. Bütün gün bunun neden kaynaklandığını araştırdım. Yalnız dünya üzerinde Android TV ye uygulama dönüştüren pek fazla geliştirici olmaması işlerin yavaş ilerlemesine neden oluyordu. Bu aksaklık için yaptığım araştırmalar neticesinde sorunun uygulamanın kullandığı görüntü işleyici API si olan OpenGL ES ile ilgili olduğunu öğrendim. Bizim uygulamamız OpenGL ES 1.1 ile çalışan bir uygulamaydı. API level 27 ise desteklemiyordu. Bu yüzden uygulamanın vermesi gereken görüntüyü Android TV ekranında oluşturamıyordu.  Bu yüzden OpenGL ES 1.1 destekleyen Android TV emülatörlerin bulmaya çalıştım. Araştırmalarım neticesinde en son API level 22 nin OpenGL 1.1 ile çalıştığını öğrendim. Daha sonraki API level sürümleri OpenGL ES 2.0 da çalışıyormuş. | | |
| **Tarih:** 09 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| API level 22 kullanan emülatörü çalıştırdığım halde hala görüntü alamıyordum. Emülatör ayarlarından yapılandırma yapmam gerektiğini öğrendim. İnternetten daha önce bu sorunla karşılaşılan konuları okudum. Emülatör ayarlamalarının nasıl yapılması gerektiğini öğrendim.  Ekran Alıntısı13  Emülatörün grafik ayarının GLES 1.1 olması gerektiğini, boot option ın emülatörü tekrar çalıştırdığında kaldığı yerden devam edip etmeyeceğini belirlediğini ve diğer seçeneklerde emülatörün işlemci ram ve bellek değerlerinin ayarlandığını öğrendim. Daha sonra bunlara uygun şekilde değerler verdim. Daha sonra emülatörü tekrar çalıştırdım. Hala görüntü alamıyordum ama oyun açılış sesi artık geliyordu. Arka planda oyun çalışıyordu fakat görüntüyü hala oluşturamıyordu. | | |
| **Tarih:** 10 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Bugün görüntü alamayıp sadece oyun sesinin neden gelebileceğini araştırdım fakat bir sonuç elde edemedim. Bu kouyu mentörümüz Noyan Bey’e anlattım. Bu sorunun, proje dosyalarının emülatörün belleğine doğru şekilde kaydetmemesi veya kaydettiği dosyaları doğru şekilde çekemediği için olabileceğini söyledi.  Araştırmamı bu bilgiyi kullanarak yapmaya devam ettim. Proje dosyalarını emülatörün belleğinde olup olmadığını görmem gerektiğini anladım fakat Android TV emülatörünün dosya yöneticisi olmadığı için göremiyordum. Bu yüzden dosya yöneticisini gösterecek bir programa ihtiyacım vardı. İnternetten bir program buldum ve emülatöre yükledim.  Ekran Alıntısı15  Ekran Alıntısı16  Dosya yöneticisine baktığımda proje dosyalarının doğru şekilde yüklenmediğini gördüm. Bu yüzden proje oyun içi dosyalarını çekemiyor ve görüntü veremiyordu. | | |
| **Tarih:** 11 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Proje içinde proje dosyalarının hangi kod parçasıyla emülatöre gönderdiğini öğrenmem gerekiyordu. Tekrar Noyan Bey’e danıştım. Bana ofAndroid.java sınıfına bakmam gerektiğini söyledi.  Bütün sınıfın içinde istediğim kodu bulmam zamanı alacağını düşündüğüm için daha önce internetten bu dosya yolunun nasıl oluşturulduğunu araştırdım. Bulduğum ipuçları sonuca gitmemi hızlandırdı. Aşağıdaki kod parçasında proje dosyalarının hangi yola yükleneceğini oluşturduğunu anladım. Daha sonra emülatör belleğinde olması gerektiği yerin konumunu yazdım.  Ekran Alıntısı17  Bununla beraber programın proje dosyaları hangi konumdan çekeceğini de bilmesi gerekiyor. Bu yüzden aşağıdaki kod dizisinde de aynı işlemi yaptım. Aynı yolu orada da gösterdim.  Ekran Alıntısı18  Projeyi tekrar build ettim ve yeniden çalıştırdım. En sonunda oyun ekranı açıldı. Ama hala bazı problemler vardı. | | |
| **Tarih:** 12 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun Android Tv ye Uyarlanması. | | |
| Projeyi en son çalıştırdığımda Android TV de çalıştırabilmiştim. Fakat bazı görüntü problemleri vardı. Futbolcu gölgelendirmelerinde ve stadyum çevresinde hatalar göze çarpıyordu. Bunların neden olabileceğini bulmaya çalıştım. Hala bazı objeler bozuk şekilde görünüyordu.  Futbolcu gölgelerinin kaldırılması nasıl bir sonuç vereceğini görmek istedim. Gölgeleri kaldırmak görüntüyü daha güzel bir hale getireceğini düşündüm. Kodlarda bir araştırma yapıp gölgelerin eklendiği yeri bulup kapattım ve sonuç iyiydi.  Ekran Alıntısı14  Ama bu sonuç yeterli değildi hala kapatarak çözemeyeceğim problemler vardı stadyum çevresinde. Kalıcı bir çözüm gerekiyordu. Bu yüzden neden olabileceğini düşünüp ve deneyerek sonuçları test ettim ama başarılı olamadım.  Projeyi sadece tek bilgisayarda çalıştırmıştım. Farklı bir bilgisayarda sonucun nasıl olacağını merak ettim. Bu yüzden ofisteki başka bir bilgisayarla projeyi çalıştırdım.  İkinci bilgisayarda hiçbir görüntü problemi oluşmadı. Her şey olması gerektiği gibiydi. İlk bilgisayarda görüntünün bozuk olmasının grafik kartı ile ilgili bir problem olduğunu düşünüyorum. Ama aynı problem ikinci bilgisayarda yoktu.  Projeyi çalıştırdıktan sonra oyun içi özelliklerini kontrol ettim, oyunu oynadım. Herhangi bir kısmında veya oyun içinde bir problem olup olmadığını test ettim. Her şey sorunsuzdu. Mentörümüz Noyan Bey’e gösterdim beraber değişiklikleri inceledik ve projeyi teslim ettim | | |
| **Tarih:** 15 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun multiplayer kısmına giriş ve multiplayer kodlarının okunması. | | |
| Bugün Noyan bey ekip değişikliği yaptı ve bunun sonucunda ben multiplayer geliştirme ekibine verildim. Burada genel olarak bağlantılar yapılmış, ama oyun ilk olarak singleplayer olarak yapıldığı için multiplayerda çalışmayan kısımları düzeltmemiz gerektiği söylendi.  İlk olarak kullanılan multiplayerda kullanılan canvasların listesi:   1. MenuCanvas – Oyun menüsü 2. ConnectionCanvas – Google Play üzerinden giriş yapma. 3. MultiplayerCanvas – Burda davet atma ve gelen daveti kabul etme işlemleri yapılıyor. 4. TeamSelectionCanvas – Takım seçimi. 5. TasoGameStartCanvas – Taktik ve takım ayarlaması. 6. TasoGameCanvas – Oyunu oynatan canvas.   Multiplayerda Google Real-Time Multiplayer API kullanılıyor. Sunucu olmadığı için veri alışverişini yapabilmek için bir cihaz Host gibi diğeri ise Client gibi kullanılıyor ve verileri API üzerinden cihazlar arasında gönderiliyor. Ancak Google API’si Java ve oyun C++ olduğu için dönüşüm yapılması gerekiyor. Bunun için 3 tane arabirim kullanılıyor.  İlk olarak C++ dan Javaya gönderebilmek ve alabilmek için iki tane metodumuz var. Bunlar sendMultiplayerData() ve receiveMultiplayerData().  sendMultiplayerData() metodunda verilerimizi char\* olarak dönüştürüp bunu JNI arabirimi üzerinden bizden önceki stajyerlerin yazdığı Java API sine gönderiliyor. Java API si üzerinden Google API’sine gönderiliyor. Burada Byte’a dönüşüp diğer cihaza gidiyor.  Diğer cihaz bu veriyi alıp Java’ya çevirip JNI arabirimine gönderiyor. JNI arabirimi Java – C++ dönüşümü yapıp C++ arabirimi ile receiveMultiplayerData() metoduna gönderiyor.  sendMultiplayerData() metodunda gönderdiğimiz sıra ile receiveMultiplayerData() metodunda alıyor ve char\*’dan değişkenlerimizi kendi veri tiplerine dönüştürülüyor.  Sonrasında ise multiplayerda kullanılan canvaslarda çalışma mantığının daha iyi anlaşılması için kodlar incelendi. | | |
| **Tarih:** 16 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Paslaşmadaki ve top sahipliğindeki hataların bulunması ve giderilmesi. | | |
| Oyun içinde paslaşırken davetli takım’da hatalar oluşuyordu. Oyun motoru Openframeworks kütüphaneleri yardımıyla yazılmıştır. Bu yüzden hataların nerden kaynaklandığını bulmak için ofLogNotice() kullanıldı.  Oyunda iki adet değişken bizim için önemli.   1. ballowner 2. ballshooter   ballowner değişkeni takımın değerini tutmaktadır. “0” kendi takımımız, “1” karşı takım, “-1” ise top sahipsiz demektir.  ballshooter ise topa son dokunan takımın değerini tutmaktadır. Bu değişken top aut, korner, taç gibi oyun dışına çıktığında yada paslaşırken biri araya girdiğinde işlemi doğru yapmak için kullanılmaktadır. “0”, “1” topa en son vuran takım, “-1” topun birinde olduğu anlamına gelmektedir.  Çalışma mantığı şu şekildedir;  Durum 1 – Top Sahipliği Durum 2 – Top ortada Durum 3 – Top Sahipliği  Ballowner  0, 1 -1 0, 1    -1 0, 1 -1  Ballshooter  ofLogNotice(“TopSahipliği”) << “ ballowner “ << ballowner << “ ballshooter “ << ballshooter;  Bu kod satırını veri gönderme metodunun sonuna ve veri alma metodunun başına yazdığımızda program çalışırken değişkenlerin anlık verilerini gösteriyor. | | |
| **Tarih:** 17 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 11 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** refreshScene(). Maç sahneleri. | | |
| Bugün maç sahnelerinin gerçekleştiği bölüm olan refreshScene() incelendi.  refreshScene bölümünde multiplayer için bi alan olmadığı gözlemlendi. Sadece single player için sahnelerin kodları vardı. Bu sebepten dolayı multiplayer sahneleri tam olarak gelmemektedir. Bunun sonucunda refreshScene metodunu 2 ye böldük. Birinci kısım multiplayer için ikinci kısım ise single player için ayrıldı.  RefreshScene ikiye böldükten sonra kodları incelemeye başladık.İnceleme sonucunda multiplayer bölümümde yapay zekanın çalışmaması için değişikler yapmaya karar verdik. Bu değişiklikleri loğlardan alıcağımız verilere göre yapıcaktık. | | |
| **Tarih:** 18 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 12 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Logların incelenmesi. Paslaşmadaki ve top sahipliğindeki hatanın tespit edilmesi. | | |
| Bugün refreshScene metonuna loglar attik. Bu loglar ile gol sahnesinin tam larak çelışmadığı anlaşıldı. Gol sahesinde gol atıldğında rakibe gol sahnesi gitmemektedir.  Logların sonucunda rahibe ballowner ve ballshoter değerleri yanlış geldiği anlaşılmıştır. Ballowner ve ballshoter değerlerini doğru bir şekilde almak için sendmultiplaterdata() ve recivemultiplayerdata() da  Verilerin doğru bir şekilde göndermek için Active Scene ekledik. Bu sağyede sahne bilgilerini doğru bir şekilde aldık.  Sendmultiplayerdata();  if (ballowner == 0) {  sprintf(msgscnOWN, "%.2d", scene);  msgscnOWN[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgscnOWN, shortmsgsize);  msgsize = msgsize + shortmsgsize; } else if (ballowner == 1) {  sprintf(msgscnOP, "%.2d", scene);  msgscnOP[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgscnOP, shortmsgsize);  msgsize = msgsize + shortmsgsize; } msgs[msgsize] = '\0'; msgsize++; sprintf(msgsizepointer, "%.4d", msgsize); msgsizepointer[intmsgsize] = '\0'; memcpy(msgs, msgsizepointer, intmsgsize); root->multiplayerConnection.sendMessageU(msgs, msgsize);  receiveMultiplayerData() for (int i = 0; i < shortmsgsize; i++ ){  msgscnOP[i] = msgr[msgsize];  msgscnOP[shortmsgsize] = '\0';  msgscnOP[i] = msgr[msgsize + shortmsgsize];  msgsize++;  } for (int i = 0; i < shortmsgsize; i++ ){  msgscnOWN[i] = msgr[msgsize];  msgscnOWN[shortmsgsize] = '\0';  msgscnOWN[i] = msgr[msgsize + shortmsgsize];  msgsize++;  } | | |
| **Tarih:** 19 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 13 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Logların incelenmesi. Paslaşmadaki ve top sahipliğindeki hatanın tespit edilmesi. | | |
| Bugün, sendMultiplayerData ve receiveMultiplayerData veri logları incelendi. İncelenen loglar ektedir.  Gün boyunca paslaşmadaki ve top sahipliğindeki hataların giderilmesi için yapılan değişiklikler denendi.  Günün sonunda sorunun ballowner verisinin yanlış gönderilmesi sebebiyle oluştuğu tespit edildi. Ballowner yanlış gittiği zaman ballshooter da değiştiği için top ortada gözükmektedir ve karşı takım paslaşırken top ayağa geldiği anda topu alacak oyuncu topu almayıp bırakmaktadır.  screen-0 | | |
| **Tarih:** 22 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 14 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Hatalı kod bloğunun yeniden yazılması ve sorunun çözülmesi. | | |
| Bugün, dün tespit edilen hatanın bulunduğu kod bloğu incelendi. Ballowner top ayaktan çıktığında yanlış değişmektedir. Bu sebeple ballshooter yanlış geliyor ve fixBall() metodunda işlemler yanlış yapıldığı için oyuncu topu alamamaktadır.  sendMutliplayerData() metodunda ballowner’ı gönderdiğimiz kısım.  if (hometeam == TEAM\_OWN) {  if (ballshooter == 1) {  sprintf(msgballowner, "%+.2d", TEAM\_NULL);  msgballowner[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgballowner, shortmsgsize);  msgsize = msgsize + shortmsgsize;  } else {  sprintf(msgballowner, "%+.2d", ballowner);  msgballowner[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgballowner, shortmsgsize);  msgsize = msgsize + shortmsgsize;  }  } else if (hometeam != TEAM\_OWN) {  if (ballshooter == 1) {  sprintf(msgballowner, "%+.2d", TEAM\_NULL);  msgballowner[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgballowner, shortmsgsize);  msgsize = msgsize + shortmsgsize;  } else {  sprintf(msgballowner, "%+.2d", ballowner);  msgballowner[shortmsgsize] = '\0';  memcpy(&msgs[msgsize], msgballowner, shortmsgsize);  msgsize = msgsize + shortmsgsize;  }  }  Bazı kısımlar değiştirilip denendiğinde topda ışınlanmalar ya da top görünüyor ama kimse alamıyor gibi sorunlar meydana gelmektedir. Bu son şeklinde çözüme ulaşıldı. | | |
| **Tarih:** 23 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 15 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Geri tuşuna basıldığında iki taraftada oyunu durdurup devam ettirme. | | |
| Oyunun multiplayer kısmındaki bir diğer sorun ise oyunu oyunculardan biri durdurduğunda oyunun karşı takımda durmamasıdır. Bunun sorun olmasının sebebi ise biri durunca süreside duruyor ama diğerinde durmadığı için bir tarafta erkenden yarı/oyun bitmekte ve karşı takımın oyuncuları yok olmaktadır. Diğer taraf ise karşı takım olmadan tek başına oynamaktadır.  Oyunu durduran metod bool TasoGameCanvas::backPressed() metotudur. Ancak bu metod singleplayera göre çalıştığı için multiplayera göre düzenlendi.  const int TasoGameCanvas::GAMESCREEN\_LOAD = -1;  const int TasoGameCanvas::GAMESCREEN\_GAME = 0;  const int TasoGameCanvas::GAMESCREEN\_MENU = 1;  const int TasoGameCanvas::GAMESCREEN\_EXIT = 2;  Gamescreenler oyunun durumunu belirtmektedir. İlk başta GAMESCREEN\_LOAD ile asset yüklemeleri yapılmakta, GAMESCREEN\_GAME oyun oynanırken çalışmakta, GAMESCREEN\_MENU oyun içerisinde menüye girince çalışmakta, GAMESCREEN\_EXIT ise oyundan çıkarken çalışmaktadır.  Oyun içerisinde iken geri/durdurma tuşlarına basıldığında backPressed() metodu çalışmaktadır. Oyunun iki tarafta da durdurulması için burda işlemler yapıldı.  ownTeamBackPressed = false;  opTeamBackPressed = false;  ownTeamBackPressed değişkeni oyunu kuran takım, opTeamBackPressed değişkeni ise karşı takım için mesajlaşırken kullanılacak kontrol değişkenleridir. Menüye girince true dönmekte, menüden çıkınca ise false dönmektedir.  if (gamescreen == GAMESCREEN\_GAME) {  if (gametype == GAMETYPE\_MULTIPLAYER) {  ownTeamBackPressed = true; //set this value true when backpressed for the first time.  gamescreen = GAMESCREEN\_MENU;  }else{  gamescreen = GAMESCREEN\_MENU;  }  if (scene > SCN\_MATCHSTARTING && scene < SCN\_ENDOFHALFTIME)  matchPlayedTime = matchMinute;  } else if (gamescreen == GAMESCREEN\_MENU) {  if(gametype == GAMETYPE\_MULTIPLAYER){  ownTeamBackPressed = false; //set this value false when backpressed for exiting menu.  }  if (menusubno == SUBMENU\_MAIN) {  if(gametype == GAMETYPE\_MULTIPLAYER) {  }else {  gamescreen = GAMESCREEN\_GAME; } | | |
| **Tarih:** 24 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 16 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Geri tuşuna basıldığında iki taraftada oyunu durdurup devam ettirme. | | |
| Bugün, geri tuşuna basıldığında iki taraftada oyunu durdurup devam etmesi üzerine yapılan çalışmaya devam edildi. En son backPressed() metodu içerisine gerekli kodlar yazıldı.  if (gametype == GAMETYPE\_MULTIPLAYER) {  ownTeamBackPressed = true; //set this value true when backpressed for the first time.  gamescreen = GAMESCREEN\_MENU;  }else{  gamescreen = GAMESCREEN\_MENU;  }  Yukarıda bulunan kod parçası geri tuşuna basıldığında oyunu kuran takıma ait olan değişkeni true yapmakta ve menüyü açmaktadır.  Update() içerisinde, alttaki kodları kullanarak bir taraf menüye girdiği zaman diğer tarafın da menüye girmesi sağlandı.  if(gametype == GAMETYPE\_MULTIPLAYER) {  if (gamescreen == GAMESCREEN\_GAME) {  if(opTeamBackPressed){  ownTeamBackPressed = true;  gamescreen = GAMESCREEN\_MENU;  matchPlayedTime = matchMinute;  }  }  }  Burdan sonra iki tarafında oyuna dönüşte aynı anda oyuna girmesi gerektiği için backPressed() de altta yazılan kodlar kullanıldı.  else if (gamescreen == GAMESCREEN\_MENU) {  if(gametype == GAMETYPE\_MULTIPLAYER){  ownTeamBackPressed = false; //set this value false when backpressed for exiting menu.  }  if (menusubno == SUBMENU\_MAIN) {  if(gametype == GAMETYPE\_MULTIPLAYER) {  }else {  gamescreen = GAMESCREEN\_GAME;  }  Bu kodlar tekrar geri butonuna bastığımızda backPressed() de çalışmaktadır. Eğer oyun menüdeyse oyunu kuran takıma ait değişkeni false yapmakta ve kaşı tarafa göndermektedir. if (menusubno == SUBMENU\_MAIN) kısmında ise multiplayerda bir şey yapılmamakta çünkü bu işlemin farklı kodlarla gerçekleştirilmesi gerekmektedir. | | |
| **Tarih:** 25 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 17 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Geri tuşuna basıldığında iki taraftada oyunu durdurup devam ettirme. | | |
| Normalde update() metodu, oyun GAMESCREEN\_GAME durumunda değilse return edilmektedir. İlk önce aşağıda yazan kısım değiştirildi.  if (gamescreen != GAMESCREEN\_GAME){  if (connectiontype != CONNECTIONTYPE\_LOCAL) {  if (gametype == GAMETYPE\_MULTIPLAYER) {  if (ofxAndroidMultiplayerConnection::haveMessage) {  receiveMultiplayerData();  }  if(gamescreen == GAMESCREEN\_MENU){  if (ownTeamBackPressed == false && opTeamBackPressed == false){  gamescreen = GAMESCREEN\_GAME;  }  }  sendMultiplayerData();  }  }  return;  }  Burada durdurulmasına rağmen cihazlar arası mesajlaşma işlemi yapılması gerekiyor ki iki tarafın da menüden çıkabilmesi sağlansın. Bu yüzden returnden önce karşıdan gelen mesaj alındı ve menüdeyse işlemi yapıldıktan sonra tekrar mesaj gönderilmesi sağlandı. Buradaki if kontrolü ile iki tarafta menüden çık dediyse ikisinin de aynı anda oyuna dönmesi sağlandı.  ownTeamBackPressed verisinin karşıya gönderilmesi gerekmektedir. Bu işlem şu şekilde sağlandı: ofxAndroidMultiplayerConnection.h’a (C++ Arabirimi) iki değişken tanımlandı. Bu değişkenler verileri gönderirken ve alırken en başta olan oyundaki state’i belirleyen değişkenlerdir.  const static char WAIT= 'W';  const static char WAITREADY= 'D';  sendMultiplayerData() metodunda üstteki iki değişkeni kullanarak ownTeamBackPressed değişkeni mesaj üzerinden gönderildi.  sendMultiplayerData()  else if (ownTeamBackPressed) {  prcsmsg[firstelement] = root->multiplayerConnection.WAIT;  prcsmsg[secondelement] = '\0';  } else if (ownTeamBackPressed == false) {  prcsmsg[firstelement] = root->multiplayerConnection.WAITREADY;  prcsmsg[secondelement] = '\0';  } | | |
| **Tarih:** 26 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 18 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Karşı takımda kameranın çalışmaması sorununun giderilmesi. | | |
| Bugün, geri tuşuna basıldığında iki taraftada oyunu durdurup devam etmesi üzerine yapılan çalışmanın son kod bloğu yazıldı ve yapılan iş başarı ile sonuçlandı.  Burada iki cihazda kendini ownTeam olarak gördüğü için receiveMultiplayerData() metodunda bunun opTeam olarak alınması sağlandı.  receiveMultiplayerData()  else if(msgr[intmsgsize] == root->multiplayerConnection.WAIT) {  //Opponent ready for the game.  opTeamBackPressed = true;  msgsize++;  } else if(msgr[intmsgsize] == root->multiplayerConnection.WAITREADY) {  //Opponent not ready for the game.  opTeamBackPressed = false;  msgsize++;  }  Sonrasında, oyunda bir başka sorun olan karşı takımda kemaranın çalışmaması sorunu üzerinde çalışmaya başlandı. Bu sorunun çözülmesi uzun zaman aldı. Bunun sebebi kameranın çok farklı yerde kullanılması ve statik olması.  Kodlar incelenip nerde kullanıldığı, hangi sahnelerde kullanıldığı, nasıl kullanıldığı ve nasıl hareket ettirildiği hakkında araştırmalar yapıldı.  Gün boyunca kodlar okundu ve kameranın çalışma mantığı öğrenildi. | | |
| **Tarih:** 30 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 19 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Karşı takımda kameranın çalışmaması sorununun giderilmesi. | | |
| Bugün, kameranın kullanıldığı yerlere log koyuldu ve hangi durumlarda hangi kodların çalıştığı araştırıldı. Kamera pozisyonları nasıl ayarlandığına, nasıl hareket ettirildiğine bakıldı.  Sonrasında kamerayı karşı takımda hareket ettirmek için yavaş yavaş değişiklikler yapıldı ve loglar ile kontrol edildi. Bu kısım uzun sürdü. Çünkü yapılan her değişikliği çalışıyor mu, çalışmıyorsa neden çalışmıyor gibi soruları kontrol ederek çözüme gidildi.  Kamerada önemli kısımlardan biri top sahipliğidir. Kamera topa göre, top ise oyunculara göre hareket etmektedir.  Loglar ile kodlara bakıldığında sadece top oyunu kuran takımdayken kameranın çalıştığı gözlemlendi. Bundan sonra ise kamera kodları karşı takım için düzenlenmeye başlandı.  Kamera, topa sahip oyuncuya göre hareket ettiği için iki seçenek vardı. Ya kamera verisi karşı tarafa gönderilecekti ya da kamera kodları başka bir şekilde düzenlenecekti. Kamera verisini karşı tarafa göndermek için çok fazla veri gönderilmesi gerektiği için mevcut olarak giden top verisine göre hareket ettirmeye karar verildi.  Sonrasında ise kodlar değiştirilerek denenmeye başlandı.  TasoGameCanvas2 | | |
| **Tarih:** 31 / 10 / 2018 | Kaşe / İmza | **Sayfa No:** 20 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Karşı takımda kameranın çalışmaması sorununun giderilmesi. | | |
| Kodlara bakıldığında kamera hareketi için iki tane önemli metod mevcut:   * moveBall() – Top hareketi. * moveCamera() – Kamera hareketi.   Bu iki metod update() metodunda sürekli olarak çalışmaktadır.  İki metodun da her kısmına log koyarak nerelere giriyor veya girmesi gereken kısma neden girmiyor gibi soruların cevabı arandı.  Yapılan her değişiklikten sonra bir çok durum için logların incelenmesi uzun sürdü.  moveBall metodu topu hareket ettiren metoddur. Burada kamera kodu kullanılmıştır. Bu kodlar düzenlendi.  if (ballowner == TEAM\_OPP || ballshooter == TEAM\_OPP) {  cameraix = ballix;  cameraiy = balliy;  }  if(ballowner == TEAM\_OWN || ballshooter == TEAM\_OWN) {  cameraix = player[ballowner][ap[ballowner]].positiondiff.x;  cameraiy = player[ballowner][ap[ballowner]].positiondiff.y;  }  moveCamera metodu ise tamamen kamera hareketleri üzerine yazılmıştır. Sahnelerde kameranın baktığı yön, pozisyon ve kamera hareket kodları bu metodda bulunmaktadır.  if (ballowner == TEAM\_OPP) {  ofLogNotice("movecamera") << “ movecamera else teamopp camera codes" ;  cameraix = ballix;  cameraiy = balliy;  }  if (ballowner == TEAM\_OWN) {  ofLogNotice("movecamera") << " movecamera else teamown camera codes " ;  cameraPlacex = player[ballowner][ap[ballowner]].position.x;  cameraPlacey = player[ballowner][ap[ballowner]].position.y;  }  Burada if (ballowner == TEAM\_OWN) kısmı önceden if (ballowner != -1) olduğu için her iki cihazda da çalışmaktadır. Ancak kodlar aktif oyuncuya göre çalıştığı için top sahibi olmayan cihazda kamera çalışmamaktadır. Kod üstteki şekilde yazılarak çözüme ulaşıldı. | | |
| **Tarih:** 01 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 21 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** İlk yarının bitişinde oyunun düzgün başlatılması. | | |
| İlk yarının bitiminde multiplayerda iki taraf oyuna aynı anda girememektedir. Bu sebepten dolayı ilk giren taraf diğeri girmeden topu alıp oynayabilmektedir. Bugün bu sorun üzerinde çalışılmaya başlandı.  if (scene > SCN\_MATCHSTARTING) {  if (scene != SCN\_ENDOFHALFTIME) {  matchMinute = (ofGetSystemTimeMicros() - matchHalfTimeStarter) / 1000000;  if (scene == SCN\_NORMALGAME && matchMinute > matchHalfTime + matchAdditionalTime) {  if ((!isBallShooted && abs(ballx) < fieldHalfLength - 100.0f) ||  matchMinute > matchHalfTime + 20) {  utils.playSound(TasoGameCanvasUtils::SOUND\_WHISTLE2);  scene = SCN\_ENDOFHALFTIME;  stage = 0;  thereIsMessage = true;  msgNo = 11;  ownTeamReady = false;  opTeamReady = false;  if (halftimeno == 1) msgNo = 12;  else gameIsStaring = true;  }  }  matchMinute += halftimeno \* matchHalfTime;  }  }  Bu kodlarda ilk yarı bitiminde sahneyi SCN\_ENDOFHALFTIME yapıldı. Yani oyun sahnesini durdurup ara sahneye geçişi sağlandı. Buraya ownTeamReady ve opTeamReady değişkenlerini koyup false yaparak iki taraftada devam etmesi engellendi ve msgNo = 11; ile waiting mesajı verildi.  touchDown() metoduna alttaki kodlar yazıldı ve ekrana dokunulduğu an oyunu kuran takımın durumu true yapılıp karşıya gönderildi.  else if (scene == SCN\_GOAL || scene == SCN\_ENDOFHALFTIME) {  if(gametype == GAMETYPE\_MULTIPLAYER){  if(scene == SCN\_ENDOFHALFTIME) {  ownTeamReady = true;  }  }  } | | |
| **Tarih:** 02 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Menu giriş ekaran görünümü değiştirme. | | |
| Taso 2018 giriş yaparken normalde sol üst köşede dostluk maçı, yanında haftalık maçlar altında ise turnuva butonu bulunmakhtadır. Multiplayer görünümü ayarlı olmadığı için mağaza butonundan giriş yapıyoruk.  Bugün menu görünümüne multiplayer ekledik. Buton adını çok oyunculu olarak ayaradık.  İlk başta dostluk maç adının hemen oyna olarak değiştirdik. Bu değişiklikleri ofdataresources dosyası içinde bulunan bölümde değişiklik yaptık. Hemen Oyna yazısını ingilizce,fransızca,ispoanyolca,italyanca,almanac,potrekizce,türkçe ve rusça olarak ekledik.  Bunun yanınca dostlık maçı yazısınıda aynı şekilde 8 dilde Çok oyunculu olarak değiştirdik.  Daha sonra MenuCanvas.cpp dosyası 2 yerde bulunnan kodalra eklme ve düzenleme yaptık.  Screenshot_2018-12-23-15-19-12  else if (selectedkeyno == 1) {  iskeysactivated = false;  MostPlayedCanvas \*mc = new MostPlayedCanvas(root);  root->isGoingToFlagPage = true;  root->setCurrentDisplay(mc);  } | | |
| **Tarih:** 03 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Menu giriş ekaran görünümü değiştirme. | | |
| else if (selectedkeyno == 1) {  iskeysactivated = false;  MostPlayedCanvas \*mc = new MostPlayedCanvas(root);  root->isGoingToFlagPage = true;  root->setCurrentDisplay(mc);  }  MostPlayedCanvas bölümüne connectionCanvas ekledik.  else if (imbuttouched == 1) {  iskeysactivated = false; *// BluetoothCanvas \*mc = new BluetoothCanvas(root);* MostPlayedCanvas \*mc = new MostPlayedCanvas(root);  root->isGoingToFlagPage = true;  root->setCurrentDisplay(mc);  }  Aynı şekilde buradada Connection canvas Ekledik.  Yapılan bu değişiklikler sayesinde haftalık maçlar yerine multiplayer ekledik.  Screenshot_2018-12-23-15-20-33 | | |
| **Tarih:** 04 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |
| **Yapılan Çalışmanın Konusu:** Menu giriş ekaran görünümü değiştirme. | | |
| Bugün daha önce yapmış olduğumuz menü görünümü için ayarlama yaptık. Bu ayarlama ile oyunu single player iin yüklediğimizde haftalık maşlar , multiplayer olarak yüklediğimizde haftalık maçlar yerine çok oyunculu yazacak ve multiplayer moduna giriş yapacak.  Bunun için ilk önce ofApp.cpp canvasına WORKMODE\_MULTIPLAYER ekledik ve değeri olarak 5 verdik. Bu sayede workmode çalışma şeklini ayarlayabileceğiz. Aynı şekilde afApp.h bölümüne WORKMODE\_MULTIPLAYER ekledik. OFAndroid.java bölümüne WORKMODE\_MULTIPLAYER eklemesini yaptık.  Bu değişiklikleri yaptıktan sonra çağıracağımız canvasa şartımızı yazık.  else if (imbuttouched == 1) {  if(workmode==WORKMODE\_MULTIPLAYER){  iskeysactivated = false; *// BluetoothCanvas \*mc = new BluetoothCanvas(root);* ConnectionCanvas \*mc = new ConnectionCanvas (root);  root->isGoingToFlagPage = true;  root->setCurrentDisplay(mc);  }else{  }  iskeysactivated = false; *// BluetoothCanvas \*mc = new BluetoothCanvas(root);* MostPlayedCanvas \*mc = new MostPlayedCanvas(root);  root->isGoingToFlagPage = true;  root->setCurrentDisplay(mc);  } | | |
| **Tarih:** 05 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** SIGSEGV Hatası | | |
| Bugün Bi hata ile harşılaştık. Bu hata SIGSEGV hatası.  SIGSEGV hatası temelinde pointer değerinin boş yani null dönmesi sonucu oluşmaktadır. Biz ekip olarak daha önce yapmış olduğumuz deişiklikleri inceledik ve hatayı aramaya başladık.  SIGSEGV hatasını bulmak için çalışmış olduğumuz bütün alanlara log atmamız gerekmektedir.  TasoGameCanavas.cpp bölümünden başladık.  İlk olarak TasoGameCanvas.cpp içinde bulunan setup(),draw() ve update() metodlarına log attık.  Atılan bu loglar sayesinde hatanın kayanağına gideceğiz.  İlk hatayı setup() metodunda aldık. Draw metodunun her satırıa log yazdık ve Nerede patlıyacağına baktık.  if (ofxAndroidMultiplayerConnection::haveMessage) {  ofLogNotice("log") << "";  receiveMultiplayerData(); }  Gerekli incelemeler sonucunda refreshScene(); kısmında patladığı için refreshScene (); metoduna log yazıldı. | | |
| **Tarih:** 06 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** SIGSEGV Hatası | | |
| Bugün refreshScene(); atılan loğların incelenmesine devam edildi. İncelemler sonucunda gol olduğunda maç detaylarını göstermede hata meydana gelmektedir.  macolayidetayi.push\_back(utils.zamaniFormatla(matchMinute, 1));  zamaniFormatla metodunun çalışırken daç dakikasını tam almadığı ve bazı dk boş geldiği anlaşılmış ve düzltilmiştir.  Daha Sonra başka hatalara bakılmıştır. refreshScene() de tekrar hata meydana gelmektedir. Hata al-nlamak için tekrardan log atıldı.  Atılan loglar ile hatanın gol sahnesinde meydana geldiği anlaşıldı. | | |
| **Tarih:** 07 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** SIGSEGV Hatası | | |
| Bir önceki gün yapılan incelemere devam ettik. Hata son olarak gol sahnesinde meydana geldi.  else if (scene == SCN\_GOAL) {  if (stage == 1) {  if (cameraDirection == 0 && acam != 4 &&  abs(camerax) > abs(shootingAreax) \* 300.0f) {  cameraix = -shootingAreax \* 4.0f;  if (acam != 1) cameraiy = 3.0f;  }  if ((cameraDirection == 1 || acam == 4) && cameray < 80.0f) {  cameraix = 0.0f;  cameraiy = 4.0f;  if (cameraDirection == 1 && goalScorer == 0 &&  ofSign(camerax) == shootingAreax) {  if ((acam != 8 && abs(camerax) > abs(shootingAreax) \* 200.0f) ||  (acam == 8 && abs(camerax) > abs(shootingAreax) \* 270.0f))  cameraix = -shootingAreax \* 4.0f;  }  }  } else if (stage == 2) {  stage = -1;  stagecount = -1;  scene = SCN\_MATCHSTARTING;  startingCount = 0;  startingCountLimit = 30;  gameStarter = 1 - goalScorer;  setBallowner(TEAM\_NONE);  cameraiscut = false;  isStartedManually = true;  isStartManualFirstCycle = true;  ballix = -ballx;  balliy = -bally;  } }  Gol sahnesine loğları attık ve oyunu yeniden yükledik. Hata moveball() metodunda alındı. moveBall() metodunun olduğu her yere log attık.Daha sonra oyun tekrar yükledik. Oyun update() metodunda patladı. Update metoduna log attık daha sonra moveBall() Metodunun içinde olduğunu anladık.  moveBall() metodunun içine log attığızda hatanın maç olaylarının gösterilmesinde olduğunu anladık.  macolayidetayi.push\_back(utils.zamaniFormatla(matchMinute, 1));  Hata diğer hata ile aynı idi. Maç olaylarını gösteririken dk bilgisni eksik almasından kaynaklı idi. Hata başarılı bir şekilde düzltildi. | | |
| **Tarih:** 08 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** VR görevine geçiş ve hataların belirlenmesi | | |
| Bugün mentörümüz Noyan Bey tarafından görev değişikliği yapıldı. Multiplayer, VR, android tv ve optimizasyon olan gruplardan vr grubuna dahil edildim.  Taso oyunun acil tamamlanması gereken VR, kod eklentileri, hataları ve OpenGL sürüm yükseltilmesi durumları açıklandı.  VR için bizden önce yapılan çalışmalardan bahsedildi. VR çalışmasında Google‘ın VR API’si kullanılmış ve gerekli kütüphaneler uygulamaya eklenmiştir. Eklenen kodlarda birtakım sorunlar ortaya çıkmıştır.  Bu sorunlar:  VR için açılan iki göz alanından birinin çalışıp diğerinin çalışmaması.  Ekranı ikiye bölme hatası.  Işıkların açılmaması ve Framebuffer’a çizdirme hatası.  Yüklenen görüntünün Renderlanmaması.  Bu hataların çözümümü için çalışmaya başlandı.  Daha sonra git hesabından lokal ağa bağlanıp proje tekrar klonlandı. Sdk ve Ndk kurumlarını gerçekleştirdik. Kurulumları yapılırken Cmake ve Gradle hataları alındı. Problemleri internetten aratarak ve ekip arkadaşlarına danışarak çözümü gerçekleştirildi.  Apk’yı Android telefonda açmak için Gradle Scripts’e girilip build gradle ‘armeabi-v7a’ kısmı açılıp çalıştırıldı.  Android Studio’dan projeyi çalıştırıp gelen görüntü bozuklukları tespit edildi.  C:\Users\burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2018-08-14 at 11.21.58 (2).jpeg C:\Users\burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2018-08-14 at 11.21.58 (1).jpeg    C:\Users\burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2018-08-14 at 11.21.57.jpeg C:\Users\burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2018-08-14 at 11.21.58.jpeg | | |
| **Tarih:** 11 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 8** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** VR ile ilgili araştırma yapıldı | | |
| Google’ın Android ndk ve tutoriallarını araştırıldı. Google’ın kendi yayınladığı makeleler incelendi. Google’ın örnek VR API’si GitHub’dan indirilip bilfisayarda Android studio’da açıldı. Googlevr/Gvr-android-sdk dosyasındaki Ndk-Treasurehunt örneği dâhil edildi.  Googlevr/Gvr-android-sdk dosyanın içerisinde farklı örnekler mevcuttur. Bunlar;   * Ndk-hellovr * Ndk-Treasurehunt * Sdk-controllerpaint * Sdk-hellovr * Sdk-simplepanowidget * Sdk-simplevideowidget * Sdk-treasurehunt * Sdk-video360 * Sdk-videoplayer   Üzerinde çalışılan proje C ++ dili ile yazıldığı için Ndk dosyalarının seçilmesi gerekiyor. Üzerinde çalışılan proje için uygun olan Ndk-Treasurehunt örnek uygulaması seçildi.  Üzerinde çalışılan projede yazılan kodların diğer çalışanları etkilememesi için workmode değişkenleri bulunmaktadır. Örneğin forma isimleri konusunda çalışan kişi workmode değerini WORKMODE\_KITNAME diye tanımlayarak kod yazmaya başlamaktadır. Çünkü eğer bir hata olursa bu hatanın diğer çalışanları etkilememesi gerekmektedir. Bunun için workmode değişkeninin değerini WORKMODE\_VR\_OWN olarak atandı ve proje üzerinde çalışmaya başlandı. | | |
| **Tarih:** 12 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 9** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Google‘ın örnek VR API’si incelendi | | |
| Google’ın örnek API’si olan Ndk-Treasurehunt örneği telefona yüklenip açıldı. VR’ın mantığını anlamak ve üzerinde çalışılan projedeki hataları çözmek için kodlar tek tek incelendi.  Google’ın örnek API’sindeki fonksiyonların çalışma mantığı öğrenildi.  google api  Ndk-Treasurehunt Ndk-Treasurehunt -2 | | |
| **Tarih:** 13 / 10 / 2018 | Kaşe / İmza | **Sayfa No: 10** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** OpenGL 1.1 hatası | | |
| Google’ın örnek API’si kullanılarak çizdirildiğinde balıkgözü şeklinde çizdirmesi başarıyla tamamlandı fakat bunun sonucunda bazı sıkıntılara neden oldu. Örneğin oyun başladığında oyuncuların formaları çizdirilmemesi, menülerde görsel bozuklukların oluşması. Önce bufferSizeOut değişkenini kullanarak çizdirilen ekranın boyutları değiştirilmeye çalışıldı. Sonrasında üzerinde çalışılan projede OpenGL 1.1 kullanıldığı için bu hataların meydana geldiği fark edildi.  Google VR API’sinde Render hatası olduğu için ve OpenGl 1.1’i desteklemediği için toplantı yapılarak şirketin kendi APİ’sinin yazılması gerektiğine karar verildi.  C:\Users\burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\opengl_logo.png | | |
| **Tarih:** 14 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 11** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Firmanın kendi VR API’sinin yazılması | | |
| Üzerinde çalışılan projedeki oyunda ekranı küçültüp ikiye bölerek vr kısmındaki opengl 1.1 hatasının giderilmesine karar verildi.  BufferSpec adında bir class bulundu ve buffer fonksiyonunun çalışma mantığını öğrenildi. Buna göre oyunu vr olarak görebilmek için ekrana çizdirilen her şeyi bir buffer oluşturup ona atanması gerekmektedir. Ekranın sol tarafına çizdirdikten sonra soldaki her şeyi buffer yardımıyla sağ tarafa da çizdirilince ekranın iki tarafında da aynı ekran çizdirilmiş olacaktı. Böylece sensörler hariç vr hakkında eklenmesi gereken bir şey kalmayacaktır.  Bu projede en çok kullanılan kütüphane openFrameWorks kütüphanesidir. Ve içinde VR için yazılmış bazı kodlar mevcuttur. Araştırmalar sonucunda ofxFBOTexture adında bir class bulundu ve tüm OpenGL sürümlerini desteklediği öğrenildi. Bu sınıfın içinde ekranın boyutunu ayarlama gibi methodlar da mevcuttur. | | |
| **Tarih:** 15 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 12** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Şirketin kendi VR API’sini yazılmaya devam edildi | | |
| Üzerinde çalışılan projeye aşağıda bulunan kodlar yazıldı ve ekran küçültülüp ikiye bölündü. Böylece sağ ve sol göz çizdirildi.  if(workmode == WORKMODE\_VR\_OWN){  ownvrmode = new ofxFBOTexture;  ownvrmode->allocate(ofGetWidth(), ofGetHeight()); // VR için ayrılacak boyut.  vrmodetopspace = ofGetHeight() \* 0.25;  vrmodedisplayspace = ofGetHeight() \* 0.50;  vrmodewidth = ofGetWidth() / 2;  }  if (workmode == WORKMODE\_VR\_OWN) {  ownvrmode->begin(); // start to ownvr  drawBackground();  drawCanvas();  ownvrmode->end(); // stop to own vr  ownvrmode->draw(0, vrmodetopspace, vrmodewidth, vrmodedisplayspace); //rigth eyes draw  ownvrmode->draw(vrmodewidth, vrmodetopspace, vrmodewidth, vrmodedisplayspace); // left eyes draw  **Screenshot_2018-08-15-10-31-05-744_com.nitragames.football** **Screenshot_2018-08-15-13-58-13-859_com.nitragames.football** | | |
| **Tarih:** 16 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 13** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** ProfilePicture ve LikeButton Vr\_Own kısmından belirli canvaslardan kaldırıldı. | | |
| Vr\_Own modunda orantısız bir şekilde karşımıza gelen ProfilePicture ve LikeButton kısımlarını if bloğu altında, sadece Relase modunda çalıştırılarak Vr\_Own kısmından kaldırıldı.  if (root->workmode == root->WORKMODE\_RELEASE) { //showProfilePicture Vr\_Own kısmından kaldırdık  root->showProfilePicture();  }  //MenuCanvas.cpp’den ve TokenStoreCaanvas.cpp’ye yazıldı  if (root->workmode == root->WORKMODE\_RELEASE) {  root->showLikeButton();  }  **Screenshot_2018-08-15-11-37-27-957_com.nitragames.football.taso2016**  Screenshot_2018-08-16-14-04-24-223_com.nitragames.football.taso2016 | | |
| **Tarih:** 19 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 14** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyun içindeki kırmızılık hatası düzeltildi. | | |
| Oyun başlatıldıktan sonra kırmızılık sorunu için TasoGameCanvas.cpp içerisindeki ofSetColor değerlerini değiştirip oyunun görüntüsü çalışması gereken haline döndürüldü.  void TasoGameCanvas::drawGUI() {  ofSetGlobalAmbientColor(ofColor(255, 255, 255));  ofSetColor(255, 255, 255);  }  Screenshot_2018-08-15-10-32-28-851_com.nitragames.football.taso2016  **Screenshot_2018-08-15-13-59-59-001_com.nitragames.football** | | |
| **Tarih:** 20 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 15** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Derinlik testi incelendi. | | |
| VR\_Own modda oyun içinde taraftarlar ekrana çizdirilirken ve oyuncuların sahaya çıkarken ekrana çizdirilmesi durumunda hataların oluştuğu gözlemlendi.  Bu durumun derinlik hatasını olduğu tespit edildi ve internetten DepthTest işlemi ile ilgili araştırma yapıldı. Uzun araştırma ve denemeler sonucunda OpenGl 1.1’de derinlik sorunu çözme başarısız oldu ve mentörümüz Noyan Bey’e danışılıp yardım alındı.  Bütün hatalar çözülüp şirketin kendi API’sinin yazımı tamamlandı. Böylece projenin VR kısmı tamamlanmış oldu.  Screenshot_2018-07-27-15-13-42-584_com  Screenshot_2018-07-27-15-13-34-722_com | | |
| **Tarih**: 21 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D OYUN PROGRAMLAMA EĞİTİMİ | | |
| Gamelab İstanbul, firma olarak Başakşehir ve Küçükçekmece Belediyeleri ile iş birliği halinde olan bir firma. Belediyelerin ayarladığı eğitim programlarına destek vermektedir. Genel olarak orta okul seviyesindeki öğrencilere kodlama eğitimleri ve mobil programlama eğitimleri veriyor.  Trt Çocuk’un düzenlemiş olduğu Trt Çocuk Medya Konferansı’nda orta okul seviyesindeki öğrencilere kodlama eğitimi vereceğimiz planlandı. Firmanın daha önceden yapmış olduğu ‘çevreci kaplumbağa’ oyununu çocuklar için tekrar yapacaktık  Oyun, firmanın kendi ürettiği oyun motoru olan Ngdroid üzerinde yapılacak. Bu oyun motorunda oyunlar, Java programlama dili kullanılarak yapılmaktadır.  Genel olarak oyunu özetlemek gerekirse; oyun, küçük yaştaki çocuklara atık çöplerin türüne bakılarak hangi çöp kutusuna atılması gerektiğini eğlenceli bir şekilde öğretmeyi amaçlıyor.  Oyun içerisinde yukarıdan aşağıya atık çöpler düşmektedir. Alt kısımda ise çeşitli renklerde farklı atıkları temsil eden çöp kutuları bulunmaktadır. Oyuncu uygun çöp kutusunu seçerek puan toplamaya çalışır. Yanlış seçim yaptığı takdirde can seviyesi azalır. Can seviyesi sıfıra ulaştığında oyun biter ve ekranda alınan puan yazdırılır.  Bu hafta boynca vereceğimiz eğitim üzerine yoğunlaştık. Eğitim sırasında kullanılacak materyaller ve kodlar gözden geçirildi. Eğitim sürecindeki yapılacaklar ve görevlendirilmeler konuşuldu. Eğitim sırasında herhangi bir aksaklık yaşanmaması için hazırlıklar yapıldı. | | |
| **Tarih**: 22 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D OYUN PROGRAMLAMA EĞİTİMİ | | |
| İlk olarak oyunun ana menüsünü yapmaya başladık.        Başlangıç olarak bir arka plan değişkeni tanımladık. Bitmap tanımı değişkenin resim olduğunu belirtir. Ardından arka plan resminin nerde olduğunu programa söylememiz gerekiyordu. Bunu loadImage komutunu kullanarak gerçekleştirdik. ‘root’ bilgisayarı ifade ediyor. ‘main\_menu.png’ ise kullandığımız resmin dosya adı. Ardından draw fonksiyonu içerisinde tanımladığımız resmin çizme işlemini gerçekleştirdik. Bunu drawBitmap komutu ile gerçekleştirdik. Arkaplan değişken ismi, sonrasında gelen iki değişken x ve y koordinatları, ardından gelen değişken ise resmin olmasını istediğimiz rengi. Resmin orijinal rengini istediğimiz için null kullandık. Yazdığımız kodları çalıştırdığımızda aşağıdaki görüntü oluştu. | | |
| **Tarih**: 23 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D OYUN PROGRAMLAMA EĞİTİMİ | | |
| Ardından ana menüden oyun ekranına geçmek için buton koyduk. Bunu yaparken ekran resmi çizdirirken yaptığımız işlemleri tekrarladık.        Bu işlemlerden sonra aşağıdaki görüntü elde edildi.      Bu komut oyna butonunu tıklanabilir hale getirir. Oyna butonunun olduğu koordinatlara dokunduğumuzda GameCanvas’a yani oyun içi ekranına geçiş yapmamızı sağlar. | | |
| **Tarih**: 26 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D OYUN PROGRAMLAMA EĞİTİMİ | | |
| Oyun içi ekranında önce arkaplan çizdirdik. Ardından düşen çöp objesini, yerde duran çöp kutularını, can seviyelerini ve puan yazan kısmı çizdirdik. Bunu yaparken aynı işlemleri tekrar ettik. Önce elimizdeki objeleri programa tanıttık, sonra objelerin hangi dosya konumunda bulunduğunu programa bildirdik ardından objeleri ekrana çizdirdik.  Update() fonksiyonu içerisine oyunun oynanış şeklini yazdık.    copuret() fonksiyonu ile ekranın üst kımında rastgele bir alanda rastgele bir atık çöp üretilir.  copdussun() fonksiyonu ile oluşturulan çöpün dikey konumu azaltılarak düşme efekti verilir.  copkutusunageldimi() fonksiyonu ile çöpün kutulara gelip gelmediği ve doğru çöp kutusuna gelip gelmediği kontrol edilir. Eğer doğru kutuya gelirse puan artar eğer yanlış kutuya gelirse can seviyesi 1 eksilir.  oyunbittimi() fonksiyonu ile oyunun durumu kontrol edilir. Eğer can seviyesi sıfıra düşerse oyunu bitirir ve ekrana elde edilen puanı çizdirir. | | |
| **Tarih**: 27 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 2D OYUN PROGRAMLAMA EĞİTİMİ | | |
|  | | |
| **Tarih**: 28 / 11 / 2018 | Kaşe / İmza | **Sayfa No: 16** |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun gamepad kısmına giriş ve gamepad kodlarının okunması. | | |
| Bugün, VR API’si yazımı bitti ve projenin VR kısmı başarılı bir şekilde tamamlandı.  Sonrasında Noyan bey ekip değişikliği yaptı ve bunun sonucunda ben gamepad geliştirme ekibine verildim. İlk olarak gamepad bağlantısının temel fonksiyonları hakkında bilgiler edinildi. Gamepad tuşları projeye onkeydown methodu üzerinden eklenmiştir. Yön tuşları Key\_dpad\_right, key\_dpad\_left, key\_dpad\_down, key\_dpad\_up şeklinde projeye eklenirken; sağ analog key\_axis\_x, sol analog key\_axis\_y şeklinde projeye eklenmiştir. Futboldaki şut, pas, ara pas, uzun pas işlemleri için sırasıyla key\_rectangle, key\_x, key\_triangle, key\_circle işlemleri dahil edilmiştir. Key\_x butonu ise gerek oyun içi geçişlerde gerekse menüler arasındaki geçişlerde kullanılmıştır. Gamepad’in arka tuşları olan L1, LT, R1, RT de aynı şekilde onkeydown methodu içerisinde tanımlanmıştır.  Butonların özellikleri ise şöyledir:   * X tuşu - Herhangi bir seçeneği seçmek, menülere girmek ve oyunda pas atmak. * Daire tuşu – Herhangi bir menüden geriye dönmek, oyun içinde orta açmak. * Kare tuşu – Oyun içinde şut çekmek. * Üçgen tuşu – Herhangi bir menüdeyken, kullanıcı profiline erişim sağlamak. * Yön tuşları (Yukarı, Aşağı, Sağ, Sol) – Menülerde gezinme, seçeneklerde gezinme, oyunda oyuncunun koşu yönünü belirleme. * L1, L2, R1, R2 – Seçenekleri değiştirmede ve oyun içi oyuncu değiştirme gibi kontrollerde kullanılır.   Bu tuşların hepsi, oyun içerisinden algılanması için gönderdikleri sinyallere göre, ana sınıflarda tanımlanarak oyun içinde algılanabilir hale getirilmiştir. Oyun kolunda basılan her tuşa karşılık, cihaz tarafından algılanan bir numarasal kod vardır. Bu kodları, ana sınıfların içerisinde static değişkenlerde tutarak, tüm oyun kanvasları içerisinde kullanılabilir hale getirilmiştir.  Oyun kolunda bulunupta, açıklaması yapılmamış herhangi bir tuş olup olmadığı kontrol edildi.    Gamepadde kullanılan canvaslarda, çalışma mantığının daha iyi anlaşılması için kodlar incelendi. | | |
| **Tarih:** 29 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 10 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Gamepad joystik’e çapraz yönlerin eklenmesi ve var olan hatanın giderilmesi. | | |
| Bugün eski stajyerlerin oyun konsolu üzerine oyunda yaptıkları bir takım geliştirmelerin hatalı olduğu saptandı ve bunlar üzerine düzeltme çalışmalarına başlandı. Bazı semantik hatalarında oyunun hata verip, kapanmasına yol açtığını görmek bu hataların bulunmasını kolaylaştırdı. Bulunan hatalar düzeltildi ve kod dosyaları Git sistemine onaylanması üzerine yüklendi.  Sonrasında gamepad joystik’e dört yön dışında, çapraz yönler de eklendi. Joystik sağ, sol, yukarı ve aşağı olmak üzere dört yönde çalışmaktadır. Oyunda joystik’in dört yön dışında çapraz olarak dört yöne daha gitmesi durumu da ele alındı. Çapraz gitmesi için, bir tane olan key parametresini iki parametre olarak ele alınması gerektiğine karar verildi. Bu yüzden keepLastData adında önceki değeri tutan bir değişken tanımlandı. Böylece gamepad joystik ile çapraz yönlerde ilerleme oyun içine başarıı bir şekilde eklenmiş oldu. | | |
| **Tarih:** 30 / 11 / 2018 | Kaşe / İmza | **Sayfa No:** 11 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** X tuşu ile oyuna direk geçiş sağlanması. | | |
| Bugün Gamepad’de X (key\_x) tuşu ile direk oyuna geçiş işlevi üzerinde çalışıldı.  Taso2018 futbol oyununda, oyuna geçmeden önce karşılaşma yapacak olan takımların ekranı, takımların diziliş ekranı ve sonrasında oyuncuların diziliş ekranı gelmektedir. Andorid tabanlı olan mobil oyunumuzda takım dizilişini, oyuncuların dizilişini görmek istemiyorsak bu ekranlara parmağımızla dokunduğumuzda (bu esnada çalışan method touchup) direk santra ekranına geçip maçı oynamaya başlayabiliyoruz. Bu işlemin aynı zamanda gamepad üzerinde de yapılması gerekmektedir. Çünkü oyun gamepad ile oynanmak istendiği zaman, sadece gamepad üzerindeki tuşları kullanarak aynı işlevleri gerçekleştirmelidir.  Gamepad üzerindeki X tuşu ile direk başlama vuruşu ekranına gelindi. Takım dizilişi de burada görüldü. Bunun için gerekli olan kod keyPressed methodu içerisine yazıldı.  Void TasoGameCanvas::keyPressed()  if(gamescreen == GAMESCREEN\_GAME && key == 96) {  // GAMEPAD -> x tuşuna basıldığında ekranı atla.  counterfordrawfield = 100;  fbOwnKeyPressed = false;  key = -5;  }  counterfordrawfield sayaçtır. Ekran geçişlerinde hata olmaması için sayaç ile geçişler kontol altına alındı. | | |
| **Tarih:** 03 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 11 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun dil seçimi kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün ilk menülerin gamepad ile yönetimi sağlandı. Oyunun başlangıcında dil seçimi menüsüne gidene kadar üç tane bölüm vardır. İlk bölüm nitragames hakkında bilgi verdiği için bu ekran geçilememektedir. Sonrasında gelen canvaslar SplashWarningsCanvas ve SplashCanvas dır. Bu iki canvas yönetimi de bir arada sağlanmaktadır. Gamepad üzerinden dil seçimi ekranına gidebilmek için SplashCanvas içerisindeki keypressed methoduna ilgili kodlar yazıldı.  void SplashCanvas::keyPressed (int key) {  if (key == root->KEY\_X) {  root->playSound(root->SOUND\_NEXT);  if (splashvideo.isLoaded() && splashvideo.getPosition() >= logoshowstart / 4) {  // && nin sağındaki kısım videonun bekleme süresini belirtiyor.  if (screenno == 0) {  // screenno 0 ise ilk video, 1 ise 2.video demektir.  screenno = 1;  // screenmode 0 olduğu için, screenmode 1'e geçerek, 2. video oynatılır.  isScreenModeChanged = true;  } else if (screenno == 1) oyunuBaslat();  //Ekranda zaten 2.video var ise x e basınca menü gelecek.  }  }  }  root->SOUND\_NEXT ile bir sonraki ekrana yani dil seçimi ekranına geçişi sağlandı. Kod bloklarının çoğuna, yol gösterici nitelikte yorum satırları eklenilmesine dikkat edildi. Yazılan kod sayesinde splashcanvas üzerinden bir sonraki ekrana geçiş, gamepad üzerindeki X tuşu ile sağlandı. Ana ekrandaki geçiş işlemini tamamladıktan sonra dil seçimi ekranına sıra geldi. SplashCanvas’dan ilerledikten sonra SelectLanguageCanvas’a geçmektedir. Bu yüzden menüler üzerinde sıradaki canvas dil seçimi canvasıdır.  Dil seçimi ekranında birçok dil mevcuttur. Bu diller arasındaki geçiş, birer birim şeklindedir. Dil seçimi ekranının en yukarısında ki, ilk dile gelindiğinde o dil üzerinden sadece aşağı hareket mevcuttur. İlk dilden en sonuncu dile geçiş bulunmamaktadır. Aslında gamepad’i burada aktif etmek, parmağımızla dokunduğumuzda aktif olan kodlarla aynı mantıktadır. touchUp method işleyişini anladıktan sonra, gamepad entegrasyonu için işlem yapmak daha kolay olmaktadır.  selectLanguage canvas’da diller arasında yukarı, aşağı geçiş kontrolü ve sayfa geçişi vardır. Bir dili seçip diğer canvasa geçişi sağlandı.  Oyunun normal işleyişinde diller arasındaki geçiş bir koordinat sistemi mantığı ile çalışmaktadır. Yukarı yön geçişi eksi koordinat sistemi, aşağı yön geçişi ise artı koordinat sistemi olarak alınmıştır. Koordinat sistemine göre bir işleyiş tasarlandığı için yukarı ve aşağı yön için birim, birim ilerleme mantığı kullanıldı. Yukarı yöne giderken -1, aşağı yöne giderken +1 birim ilerleme sağlandı. Aynı koordinasyon gamepad üzerinde de etkili olmaktadır. Farklı bir yöntem izlenmesine gerek kalmadı. Gamepad üzerinde de koordinat sistemi şeklinde ilerlenmektedir. | | |
| **Tarih:** 04 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 12 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun hakkında kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün oyunun “Hakkında” bölümü üzerinde çalışıldı. Hakkında bölümü tanıtım ve dil ekranlarından sonra gelen ana menünün bir canvasıdır. İçeriğinde oyunun hangi firmaya ait olduğu, kimler tarafından geliştirildiği gibi genel bilgiler bulunmaktadır. Bu menüde istenen, dokunmatik ekrana dokunarak yukarı ve aşağı yöne yapılan kaydırma işleminin, gamepadin yukarı (KEY\_DPAD\_UP) ve aşağı (KEY\_DPAD\_DOWN) tuşları ile yapılmasıdır.  Taso2018 futbol oyunu projesinde, Android studio içerisinde yazılan kodlar cpp ve header dosyası olmak üzere iki ayrı yerde yazıldı. Bu yüzden cpp canvasın içerisinde initialization işlemi yapılmamalıdır. Tanımlamalar header dosyasında yapılmalıdır. Kod yazarken önceden cpp dosyası içerisinde yapılmış olan tanımlamalar header dosyasına alındı. Bu tanımlamaların header dosyasında tutulması oyuna hız olarak kazanç sağlamaktadır.  Hakkında kısmı yapılırken tıpkı diller arası geçişte olduğu gibi koordinat sistemi mantığından yararlanıldı. Yukarı yöne giderken -1, aşağı yöne giderken +1 birim ilerleme sağlandı. Koordinat sistemi üzerinden gidecek olursak ekranın y ekseni üzerinde yukarı aşağı hareket etmesi gerekiyordu. Sistemin y koordinatı textbacky değişkenine atandı. Header dosyasında y1 isimli değişken tanımlandı ve cpp dosyası üzerinde işlem yapıldı. Tuşlar ile çalışılacağı için cpp dosyasındaki keypressed fonksiyonunun içine görevler atandı.  void AboutCanvas::keyPressed(int key) {  //ofLogNotice("AboutCanvas.cpp") << "y1" << ofToString(y1);(yazmasanız da olur)  switch(key){  case root->KEY\_DPAD\_DOWN:  y1-= textbacky;  break;  case root->KEY\_DPAD\_UP:  y1+= textbacky;  break;  default:  break;  }  }  Görevlerin atanmasının ardından, y ekseni üzerinde yukarı ve aşağı giderken hata yaşanmaması için sınır değerler belirlendi ve oyunun hakkında kısmı tamamlandı. | | |
| **Tarih:** 05 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 13 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun turnuva kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün turnuva kanvası adında ki yeni bir menüye konsol kontrolleri eklemesi üzerine çalışıldı. Bu kanvasa bağlı, 3 adet daha alt kanvas olduğu tespit edildi. Bu kanvaslar:   1. TournamentStartCanvas.cpp 2. TournamentSetupCanvas.cpp 3. TournamentCanvas.cpp kanvaslarıdır.   TournamentStartCanvas içerisinde, iki seçenek bulunmaktadır. Yeni turnuvayı başlatacak “Yeni” butonu, ya da mevcut turnuvaya devam edecek “Devam” butonudur. Bu butonlara “Sağ” ve “Sol” yön tuşlarıyla erişim sağlanırken, “X” tuşu ile seçilebilmesi sağlandı.  Turnuvaya girildiğinde eğer hali hazırda mevcut bir turnuva yoksa, “Devam” butonunun inaktif olduğunun anlaşılması için, karartılmış şekilde çizdirildi. Bunlar halledildikten sonra, turnuvaya ikinci girilişte devam butonu da aktif hale getirildi. Kontrolün hangi butonda olduğunu öğrenmek için “countercolorkey” isimli sayaç oluşturuldu. Cpp dosyasındaki draw fonksiyonunun içine, sayaç sıfırda ise “devam” butonunun aktif gösterilmesi, sayaç 1’e eşitse “yeni” butonunun aktif gösterilmesi için kodlama yapıldı.  Çizim bölümündeki iş bittikten sonra sıra butonlara görev atamaya geldi. Cpp dosyasının içindeki “keypressed” fonksiyonunun içine gamepad’in sağ ve sol yön tuşlarına (KEY\_DPAD\_RIGHT ve KEY\_DPAD\_LEFT) sayacın hangi butonda olduğunu belirtmesi için görevler atandı. Keypress fonksiyonunda birden fazla butona görev atandığı için switch case yapısı kullanıldı.  case root->KEY\_DPAD\_LEFT:  countercolorkey = countercolorkey - 1;  break;  case root->KEY\_DPAD\_RIGHT:  countercolorkey = countercolorkey + 1;  break;  Oyun konsolunda “X” tuşuna basıldığında, sanki iki kere basılmış gibi algılandı. Bunun sonucunda oyun çökmesine sebep olan bir hata ile karşılaşıldı. Kontrol bloklarıyla “X” tuşunun iki kere basılması engellenerek bu hata çözüldü. Tuşlara birden fazla kez basıldığında oyunun hata vermemesi için switch case yapısının dışına sınır değerler atandı.  if(countercolorkey <= -1) countercolorkey = 0;  if(countercolorkey >= 2) countercolorkey = 1;  Gamepad’in X tuşuna (KEY\_X) seçili olduğu butonun üzerinde işlem yapması için görevler atandı ve projenin “Turnuva Başlangıç Ekranı (TournamentStartCanvas)” tamamlandı. | | |
| **Tarih:** 06 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 14 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun turnuva kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün ilk olarak TournamentSetupCanvas’ta, yeni bir turnuvanın ilk kurulumunun yapılabilmesi için kıta seçme işlemi L2-R2 tuşlarına, bayrak seçme işlemi L1-R1 tuşlarına atandı.  z  Turnuva kanvasının ilk menüsü tamamlandı ve alt kanvaslarında ki kontroller eklenmeye başlandı.  TournamentCanvas’ta, görüntülenebilir tablolara erişim sağlandı. Open Frameworks kütüphanesine ait, ofDrawRectangle fonksiyonu kullanılarak, bu menülerde oyun konsoluyla üzerine gezilebilen seçeneklerin arkasına, mavi bir pencere çizdirildi. Bunun sebebi, oyuncunun kol ile sağ ve sol yönlere gittiğinde hangi seçeneğin üzerinde durduğunu net bir şekilde görebilmesini sağlamaktır.  if(selectedbuttonno == 0){ // oyna ikonunun arkasına mavi kare yaratır.  ofSetColor(root->buttoncolorr, root->buttoncolorg ,root->buttoncolorb, 200);  ofDrawRectangle(iconx[0] \* 0.95 ,icony[0] \* 0.95,iconw \* sizeRatio ,iconh \* sizeRatio);  } else if (selectedbuttonno == 1) { // skorlar ikonunun arkasına mavi kare yaratır.  ofSetColor(root->buttoncolorr, root->buttoncolorg ,root->buttoncolorb, 200);  ofDrawRectangle(iconx[1] \* 0.98 ,icony[1] \* 0.95,iconw \* sizeRatio ,iconh \* sizeRatio);  } else if (selectedbuttonno == 2) { // maçlar ikonunun arkasına mavi kare yaratır.  ofSetColor(root->buttoncolorr, root->buttoncolorg ,root->buttoncolorb, 200);  ofDrawRectangle(iconx[2] \* 0.95 ,icony[2] \* 0.976,iconw \* sizeRatio ,iconh \* sizeRatio);  } else if (selectedbuttonno == 3) { // puan durumu ikonunun arkasına mavi kare yaratır.  ofSetColor(root->buttoncolorr, root->buttoncolorg ,root->buttoncolorb, 200);  ofDrawRectangle(iconx[3] \* 0.98 ,icony[3] \* 0.976,iconw \* sizeRatio ,iconh \* sizeRatio);  }  Başı “of” ön-eki ile başlayan fonksiyonların hepsi, Open Frameworks’e aittir. Yukarıda ki kodda, ofSetColor ise, çizdirme fonksiyonundan hemen önce kullanıldığında, sırasıyla “Red”, “Green”, “Blue” ve “Opacity” olarak 4 adet parametre alarak renk ve saydamlık seviyesini belirlemektedir.  Daha sonra bu seçenekler seçildiği zaman, mesela turnuvada ki puan durumunu gösteren seçenek, gerekli bilgileri getirip ekrana yazdırılması için, verileri tutan sınıf ile iletişime geçerek güncel bilgiler aktarıldı. Karşılaşılan hatalardan birisi, seçeneklerin arkasına çizdirmek istendiğinde mavi pencerelerin, seçeneklerin üstüne çizdirilmesiydi. Bu sorun, program akışında pencere çizimini, seçenek çiziminin önüne alınarak çözüldü. Bu sayede, program akışının oyun programlarken, çizim fonksiyonu içerisinde ki önemi öğrenildi. | | |
| **Tarih:** 07 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 15 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun kadro yönetimi gamepad ile yönetiminin sağlanması. | | |
| Bugün oyunun kadro yönetimi kısmı olan “Management Squad Canvas” üzerinde çalışıldı. Ekranın sol tarafında takımın ilk 11i ve yedekleriyle birlikte 23 kişilik kadrosu yer almaktadır. Listenin altında, liste üzerinde düzenlemeler yapan ve oyuncuların güç ve kabiliyetlerini gösteren 6 tane düğme yer almaktadır. Bunlar oyuncu bilgisi, oyuncu değiştir, oyuncu çıkar, oyuncu ekle, kadroyu kaydet ve oyuncu sil olmak üzere 6 bölümden oluşmaktadır.  Ekranın sağ tarafında maçlara göre kadro dizilişleri ve en altta takımın sahaya dizilişini gösteren kısım yer almaktadır.  Kadroda hangi oyuncunun üzerinde işlem yapıldığını belirtmek için kadro listesine belirteç eklendi. Liste altındaki butonlara da kadroda değişimler yapmaları için görevler atandı. Gamepadin L1 ve R1 tuşları ile tuşlar arasında gezinme sağlandı. Sağdaki kadro işlemleri arasında gezinmek için de kadro ve diziliş kısımlarında gezinmeyi sağlayan belirteçler eklendi. Sağ ve sol tuşlarıyla taktik ve kadro seçimleri sağlandı. Sağ alt kısımda ise seçilen kadronun sahaya dizilişi gösterildi. | | |
| **Tarih:** 10 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 16 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Maç ön ayarları kısımlarının gamepad ile yönetiminin sağlanması. | | |
| Bugün oyunun “Taso Game Start Canvas” bölümü, yani oyuna başlamadan önce takım dizilişleri, iç saha ve deplasman takım forma seçimleri, oyunun kaç dakika süreceği ve oyunun gece veya gündüz oynanması gibi temel ayarların olduğu kısım üzerinde çalışıldı. İlk olarak gamepadin hangi ayarında olduğunu belirtmek için ayarların olduğu yerin etrafına dikdörtgen şeklinde gösterge yapıldı. Ayarlar düğmeler ve butonlar olarak iki kısımdan oluşturulmuştur. Oyunun cpp bölümünün “draw” fonksiyonuna belirteçler tanımlandı.  ofSetColor(35,34,91); //belirtecin rengi belirlendi.  ofDrawRectangle(switchx[counterswitch]-5, switchy[counterswitch]-5, 145, 80); //belirtecin x y konumları, genişlik ve yükseklikleri belirlendi.  Belirtecin ardından düğmelere görevler atandı. İlk düğmeye ev sahibi takımın ya da deplasman takımının dizilişlerinin izni için görevler atandı. İkinci düğmeye de maçın gece ya da gündüz oynanmasını belirlemesi için görev atandı. Kontroller belirtecin sağ ve sol tuşlarına atandı. Ardından maçın oynanacağı zamanı belirlemek için aynı şekilde sağ ve sol tuşlara görevler atandı. Düğmeler ve butonlar arasında geçişi sağlamak için yukarı ve aşağı (KEY\_DPAD\_UP ve KEY\_DPAD\_DOWN) tuşları ile çalışan belirteç kullanıldı. | | |
| **Tarih:** 11 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 16 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun ana menü kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün oyunun ana menüsünü içeren “MenuCanvas.cpp” üzerinde çalışıldı. Yazılan kodlara tek tek yorum satırı eklendi.  void MenuCanvas::keyPressed (int key){  if(loadingStopControl){ // Eğer loading olayı bitti ise, yani menü draw olduysa true döndürüp tuşları etkinleştirir.  switch (key) {  case ofApp::KEY\_DPAD\_LEFT: // sola basıldı  selectedbuttonno = selectedbuttonno - 1;  if (selectedbuttonno >= 8 || selectedbuttonno <= 0) { selectedbuttonno = 0; }  break;  case ofApp::KEY\_DPAD\_RIGHT: // sağa basıldı  selectedbuttonno = selectedbuttonno + 1;  if (selectedbuttonno >= 8 || selectedbuttonno <= 0) { selectedbuttonno = 0; }  break;  case ofApp::KEY\_X: // X tuşuna basıldı  case ofApp::KEY\_BL\_X: // Bluetooth cihazdan X'e basıldı  if (selectedbuttonno == 0) { // Ana menüde ilk sırada bulunan alt menü seçeneği  iskeysactivated = false;  TournamentStartCanvas \*mc = new TournamentStartCanvas(root); // İlk menü olan, turnuva kanvasına ait bir pointer oluşturulur  root->setCurrentDisplay(mc); // Oluşturulan bu pointer, gösterime sokulur.  } else if (selectedbuttonno == 1) {  // more code blocks going here...  }  // more code blocks going here…  }  Ana menüde “Sağ” “Sol” ve “X” tuşuna gerekli atamalar yapıldı. “X” tuşuna basıldığı zaman, “selectedbuttonno” değişkeni kontrol ediliyor ve 0’dan 7’e kadar, toplamda 8 tane olan alt menülerden seçili olana göre, hedef alınan alt menüyü bir pointer’a atayıp, ekrana çizdirildi. Hedef alınan menü ise, sağ ve sol tuşlar basıldığı zaman “selectedbuttonno” değişkeninin arttırılıp, azaltılması yoluyla belirlendi. | | |
| **Tarih:** 12 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 17 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun ülke seçiminin gamepad ile yönetiminin sağlanması. | | |
| Bugün “SelectCountryCanvas” ve “MostPlayedCanvas” üzerinde çalışıldı. “SelectCountryCanvas” içerisinde oyuncunun oynamak istediği takımın ülkesini seçebileceği, seçeneklerin bayrak resimleri olarak sunulduğu bir menüdür. “MostPlayedCanvas” ise, haftalık maçları gösteren ve yine aralarından seçim yapılabilinen bir menüdür. İlk olarak, bu sınıflarda daha önceden yazılmış kodlar incelenmeye başlandı. Kodlar okunarak, oluşturulan değişkenlerin ve yazılmış fonksiyonların işlevleri çözülmeye çalışıldı. Bu arada yapılan çıkarımlar da, kod bloklarının yanına yorum satırı olarak eklenerek ilerlendi.  Eskiden yazılmış kodlar incelendikten sonra, “SelectCountryCanvas” üzerinde çalışılmaya başlandı. Bu kanvasta, yapılması gereken şeyler, bayraklar arasında yön tuşları ile geçiş yapılması, listenin en sonuna gelindiğinde tekrar başa dönmesi, liste en baştayken direk olarak en sona gidilebilmesi ve “X” tuşuna basıldığında seçili bayrağı, oynanmak istenen ülke olarak seçmesidir. Bu kontroller switch-case blokları kullanılarak, oyun konsolu üzerinde ki yön tuşlarına atandı ve if bloklarıyla sona gelindiğinde en başa ya da başa geldiğinde en sona gidebilmesi sağlandı. Son olarak, “X” tuşuna gerekli görevi atandı. Böylece “SelectCountryCanvas” üzerinde oyun konsolu kontrolleri tanımlamaları tamamlandı. | | |
| **Tarih:** 13 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 18 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Haftanın maçlarının gamepad ile yönetiminin sağlanması. | | |
| Bugün “MostPlayedCanvas” üzerinde çalışarak devam edildi. Genel mantığı, bir önce ki çalışılan kanvasla aynıydı. Çünkü kaydırmalı bir menüde bayrak seçerek çalışıyordu. Ancak, bu kanvasta bir takım hatalar ve buglar ile karşılaşıldı. Bu hata, kanvas açıldığı zaman “Yeni maçlar indirilsin mi ? “ diye bir soru soran, ve evet cevabı verildiğinde internet üzerinden en güncel maçları oyuna çeken bir açılan pencere ile alakalıydı. Oyun konsolu kontrolleriyle, bu açılan pencerede bir seçenek seçildiği zaman, internet üzerinden çekilen güncel maçların bayrakları, oyun penceresinde boş beyaz bir ekran olarak çizdirilmekteydi. Bu sorunun çözülebilinmesi için, oyun motorunun ana sınıflarının incelenmesine karar verildi. Bayrak resimlerini, oyuna yüklemek için ve daha sonra ekrana çizdirmek için kullanılan fonksiyonlar okunarak anlamaya çalışıldı. Oyun motorunun ana sınıflarını üzerinde değişiklik yapılması önerilmediği için alternatif bir yöntem denenerek sorun çözüldü. Bu yöntem, resim dosyalarının “draw” fonksiyonu içerisinde tekrar çizdirilerek, eksik yüklenen beyaz resimlerin üstü olması gereken gerçek resimlerle kaplandı.  Maçların arasında gezilmesi için, yön tuşları kullanıldı. Listelenen maçlar, satır başına iki tane maç gelecek şekilde dizilmişlerdir. Bu sebeple, eğer toplam maç sayısı tek sayı olursa, en alta kalan satırda bir adet boşluk oluşmaktadır. Bu değişkenlik, esnek bir kod yazılmasını gerektirdi. Bu sebeple, kontrol blokları kullanılarak toplam maç sayısının tek sayı olması durumu için ekstra bir kod yazıldı.  case ofApp::KEY\_DPAD\_DOWN: // aşağı tuşa basıldıysa  listselectedslotno += 2;  if (listselectedslotno == slotmatchnum + 2)  listselectedslotno = firstmatchno; // Eğer aşağıda maç yoksa en son maça gider.  else if (listselectedslotno > slotmatchnum) listselectedslotno = slotmatchnum;  if ((matchnum % 2 == 1 && listselectedslotno == slotmatchnum - 1)) // eğer 2'ye bölümünden kalan 1 ise yani tek sayı ise ve en sondan bir önceki bayrak seçiliyse  listselectedslotno = slotmatchnum;  mylist-> | | |
| **Tarih:** 14 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 18 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun takım seçimi kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün “TeamSelectionCanvas” üzerinde çalışıldı. Takım seçiminin yapıldığı ve ana menüdeki “Dostluk Maçı” alt menüsünden ulaşılabilen bu kanvasa, eklenilmesi gereken birçok işlevsellik eklendi. Örneğin, L2 ya da R2 tuşlarına basılması durumunda, kıta seçeneklerini sağa ya da sola doğru değiştirebilme işlevselliği eklendi. L2 butonu için yazılmış olan kod şu şekilde;  case ofApp::KEY\_BUTTON\_L2: // L2 tuşuna basılırsa  case ofApp::KEY\_BL\_BUTTON\_L2: { // Bluetooth cihazdan L2’ye basılırsa  int i = countermenu;  listedleague[i]--; // kıtalar sola kaydırılır  if (listedleague[i] < 0) listedleague[i] = 0; // eğer en soldaki sınıra gelindiyse, daha fazla gidilemez  selectlistbuttoninactive[i][0] = false;  selectlistbuttoninactive[i][1] = false;  if (listedleague[i] == 0) {  selectlistbuttoninactive[i][0] = true;  selectlistbuttonhovered[i][0] = false;  }  refreshMenu(i);  teamtacticname[i] = getTeamFormation(  leagueteam[i][listedleague[i]][selectedkeyno[i]]);  root->playSound(root->SOUND\_CLICKKEY);  }  Bu mantık, R2 tuşu içinde aynen uygulandı. Daha sonrasında, L1 ve R1 tuşlarına, takım seçimini belirlemek için bayraklar arasında gezebilmesi işlevselliği eklendi. Buraya kadar “TeamSelectionCanvas” içerisinde yapılan değişikliklerin tümü, daha öncesinde “TournamentSetupCanvas” üzerinde yapılan ve yine kıta değiştirme, takım değiştirme, ayarlar içerisinde gezinebilme gibi işlemleri kapsayan kısımlardan örnek alınarak yapıldı. Bu yol ile, proje ilerledikçe daha önce yapılan çalışmalar, ekibe yol gösterici oldu ve geliştirme süreci hızlanmış oldu.  TeamSelectionCanvas | | |
| **Tarih:** 17 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 19 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun mağaza kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün mağazaya ait sınıflar üzerinde çalışıldı. “ItemStoreCanvas” ve “TokenStoreCanvas” aynı anda geliştirildi. Bunun sebebi, içerisinde satışa sunulan öğeler hariç menü sisteminin tamamen aynı olmasıdır.  “ItemStoreCanvas” içerisinde, oyun içi satın alma özelliklerine sahip ürünler sergilenmektedir. Ürünlerin sergilenmesi için kullanılan kayan menü, oyun motoruna ait bir sınıf ile ekrana çizdirilmektedir. Bu sınıf incelenerek, kayan menünün çalışma mantığı kavranmaya çalışıldı. Sınıfın içerisinde bulunan fonksiyonlar ile, kayan menüye yeni bir ürün ekleme gibi bir çok özelliğe sahip olunduğu keşfdildi. Sonrasında, oyun konsolu ile kayan menü üzerinde gezilip, belirli bir ürünü seçip, onu satın almaya kadar giden yolların kodları yazılmaya başlandı. Oyun konsolu kontrollerin bir çoğu eklensede resimlerin yüklenmesi ile alakalı bir takım sorunlar yaşandı. Menü içerisinde gezinme sırasında, oyun kolu ile erişim yapıldığında satıştaki öğelerin resimleri yerine beyaz ve boş bir dikdörtgen çizdirilmektedir. Bu sorunu MostPlayedCanvas’ta olduğu gibi, resimleri yükleyen fonksiyonu tekrar çağırmakla çözülemediği için, alternatif metodlara yönelindi. Bu kez sorunun çözülebilmesi için oyun motoruna ait ana sınıflara müdahele edildi. Bu ana sınıflar, Open Frameworks’ten alınıp geliştirilmiş olup, değiştirilen sınıfın ismi “ofxSlidingList”dir. Bu sınıf ile, oyun menüsü sınıfları arasında bağlantı kurmak istenildiğinde, belli başlı hatalar alındı. Örnek olarak, iki sınıfı birbirlerine “include” etmek isterken ikisini de “header” dosyalarından yapılmaya çalışıldığı zaman, “Cycle” denilen bir derleme hatasına sebep olmaktadır. İki sınıfta, birbirini sürekli “include” etmeye çalışarak sonsuz bir döngüye girmektedirler. Bu sorunu, sınıflardan birinde “include” işlemini “.cpp” dosyası içerisinde yapıp, “header” dosyası içerisinde de “forward declaration” yaparak o sınıfa ait bir obje de “header” içerisinde tanımlanabildi. Mağazaya ait menülerde, satıştaki öğeler arasında sağa sola hareket edilebilinmesi için kontroller eklendi. Örneğin kaydırmalı menüde sağa kayabilmek için, oyun kolundaki sağ tuşa basıldığında işlevselliğin çalışması aşağıdaki kod ile sağlandı;  case ofApp::KEY\_DPAD\_RIGHT:  if (isonbutton) { // eğer yukarı ya da gir butonları üzerindeyse  selectedbutton++;  if (selectedbutton > buttonnum - 1) selectedbutton = 0;  touchDown(buttonx[selectedbutton] +  btx[selectedbutton] +  dialogue->getButtonFontShadowDiff(),  buttony[selectedbutton] +  bty[selectedbutton] +  dialogue->getButtonFontShadowDiff(), 0);  } else {  selectedslotno++;  if (selectedslotno > totalpacknum - 1) selectedslotno = 0;  mylist->setSelectedSlotNo(selectedslotno);  }  break; | | |
| **Tarih:** 18 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 20 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun ayarlar kısmının gamepad ile yönetiminin sağlanması. | | |
| Bugün “SettingCanvas” yani oyunda ayarlar olarak geçen kısım üzerinde çalışıldı. Bu noktaya gelinene kadar, artık oyun motoru içerisinde menü oluşturmak ve görselliği arttırmak için bulunan fonksiyon ve sınıfların bir kısmına hakim olunmuştur. Ayrıca üç boyutlu bir oyun yapılırken, kullanılan programlama dilinin alçak-düzey (low level) bir dil olmasının performans açısından avantajları, oyun içerisinde deneyimlendi. C++ dili her ne kadar yazılması daha çok zaman alan ve karmaşıklık seviyesi daha fazla olan bir dil olsa da, performans konusunda oyun alanında tercih edilen ilk dil olmasının sebebi anlaşıldı.  Ayarlar kanvasında, 10 adet ayar seçeneği bulunmaktadır. Bu seçenekler arasında, oyun konsolunun yön tuşlarıyla gezilebilinmesi için kontroller eklendi. Bazı ayar seçenekleri, aç-kapa mantığıyla çalışırken, bazılarında kamera açısı seçmek gibi alt seçenekleri mevcuttur. Hepsi için ayrı kontroller eklenerek, bunları olabilecek en verimli ve okunabilir hale indirgemeye özen gösterildi. Kanvastaki tüm kontrollerin eklenmesi bittiğinde, her zaman olduğu gibi yapılan tüm değişikliklere açıklayıcı yorum satırları eklendi ve ana projeye eklenmek üzere git sistemine gönderildi. Projenin ilerleme hızı geçen haftalara göre daha düşük olsa da bunun sebebi, üzerinde çalışılan kanvasların daha karmaşık olması ve çok daha fazla sorun ile karşılaşılmasıdır. Böylece ana menüden ulaşılıp, oyun konsolu kontrolü kalmayan kanvas sayısı 3’e indi.  Şu ana kadar yapılan çalışmalar doğrultusunda, artık bu tarz işlemler daha kolay kodlanabilinmektedir.  Sonrasında, daha önce üzerinde çalışılan “Taso Game Start Canvas” isimli kanvasta eksik olduğu tespit edilen özellik üzerinde çalışıldı. İç saha ve dış saha formalarının belirlenmesi için yön tuşları ve belirteçlerden yararlanıldı. Gamepadin X tuşuna basıldığında da ayarların kaydedilerek maça başlanması sağlandı.  Case root->KEY\_X:  root->playSound(root->SOUND\_NEXT);  iskeysactivated = false;  root->videoad2.pause();  root->unloadMainmenuBackground();  root->stopMusic();  TasoGameCanvas \*mc = new TasoGameCanvas(root);  root->setCurrentDisplay(mc);  Break;  Kuruldan önce, geliştirilmeye açık kısımlar hala mevcuttu ancak oynanabilirlik aşamasına gidene kadar kullanılan tüm bölümleri oyun konsoluyla erişilebilir hale getirmek için ek mesai harcandı. | | |
| **Tarih:** 19 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 21 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Oyunun yönetim alt menüsü kısmında bulunan hataların giderilmesi ve oyuna yeni özellik eklenmesi. | | |
| Bugün ana menüden erişiminde sorun olan alt menülere erişim sağlanması için, hata kodları incelendi. Bu aşamada, “ManagementCanvas” yani yönetim alt menüsüne oyun koluyla ulaşılmaya çalışdığı zaman, oyun çökmektedir. Çökme anında oluşan hata çıktıları uzun bir süre incelendi ve üzerinde fikir paylaşımı yapıldı. Sonunda hatanın kaynağı bulundu ve gerekli düzenlemeler yapıldı.  Son olarak, menülerden ayrı şekilde geliştirilen “TasoGameCanvas”, yani oyunun oynanış aşamasını içeren bölüm ele alındı. Burada, daha önceden yapılmış bir çalışma mevcuttur. Ekip olarak, ne gibi geliştirmeler yapılabilineceği ise daha önceden konuşulmuştur. Tasarım detaylarında belirtilen tüm işlevsellikler incelendi, eksik olan kısımlar eklendi, düzeltimesi gereken kısımlar onarıldı ve optimize edildi. Optimize etmek, sonradan katılan ekip çalışanları tarafından, oyun kolundaki “Analog” yön belirleyiciye 15 derecelik açılarla yön verebilme işlevselliği kazandırmaktır. Bunun sebebi, oyun kollarında analog yönlendiricilerin 15 derecelik açılarla sinyaller gönderebilmesidir ve bu özelliğin oyuncuların yönünü belirlemekte daha ince kararlar verilmesine olanak sağlayacağının kararlaştırılmasıdır. Oyun içine yapılan eklemeler, matematiksel hesaplamaların da yardımı ile yapıldı. Özellikle “Analog” kontrolün, 15 derecelik açılarını futbolcunun hareketleriyle bağlamak için, dikkatli hesaplamalar yapıldı.  ManagementSquadCanvas | | |
| **Tarih:** 20 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** KOSGEB AR-GE ve İnovasyon Kurulu | | |
| Bugün şirketin KOSGEB tarafından destek alarak geliştirdiği TASO 18 oyunu ve oyun motoru geliştirme projesinin final sunumunu yapmak üzere KOSGEB İstanbul müdürlüğüne gidildi. Tamamlanan proje teknik ayrıntıları ile beraber kurula sunuldu ve heyetin sorduğu sorular cevaplandı. Yapılan sunumda projenin android tv, vr, gamepad ve multiplayer kısımlarının bitmiş hali heyet önünde canlı olarak oynanarak gösterildi. Oyunun multiplayer kısmının geliştirilirken verinin nasıl gönderildiği anlatıldı. Kullanılan API’ler ve gönderilen veri boyutları açıklandı. Böylece heyetin sorduğu multiplayer verileri nasıl gönderiyorsunuz sorusu cevaplanarak kafalardaki soru işareti giderildi. Daha sonra yapılan sunum için teşekkür edilerek sunum bitirildi ve ofise geçilerek günün geri kalanında geliştirdiğimiz proje üzerinde çalışmaya devam edildi. | | |
| **Tarih:** 21 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Rusça Dil Desteği Eklenmesi | | |
| Bugün Noyan bey, Kristina’dan rusça çeviri dosyalarını alıp oyuna entegre etmemi istedi. Böylece TASO 18 oyununa Rusça dil desteği eklenmiş olacaktı. İlk olarak dosyaları ofdataresources adlı zip dosyasının içine atarak oyun başlatıldı. Oyun açıldığında kelimeler bozuk gözüküyordu. Dil dosyalarını kendi bilgisayarım üzerinde açtığım zaman bütün yazıların bozuk gözüktüğü fark edildi ve bunun, dil dosyalarının macOS işletim sistemi üzerinde hazırlanmasından kaynaklandığı tespit edildi. Dil dosyaları tekrar düzeltildi ve ofdataresources adlı zip dosyasının içine atıldı. Bunlara ek olarak dil dosyalarının yanına oyunun Rusça yazıları tanıması için kiril alfabesinin karakterleri eklendi. Daha sonra oyun başlatıldı ve oyun hatasız açıldı fakat oyunun dil seçeneklerinde Rusça seçeneği hala görülmedi. Bunun üzerine oyunun dil seçiminin yapıldığı SelectLanguageCanvas üzerinde kodlar incelendi. Hatanın dil seçeneklerinin tutulduğu vektör dizisinin boyutundan kaynaklandığı tespit edildi. Vektör dizinin boyutu bir arttırıldı ve değiştirilen kodlar cihaza yüklenerek oyun başlatıldı. Oyun tekrar başlatıldığında Rusça dil seçeneğinin dil seçim menüsüne başarılı bir şekilde eklendiği gözlemlendi. | | |
| **Tarih:** 24 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 3B OYUN GELİŞTİRME | | |
| Bugün bana Noyan bey tarafından yeni bir görev verildi. Görev oyunun giriş bölümünde gelen bayrakların boyutunun küçültülmesi yapılacak.  Bunu düzeltmek için oyunun bayrakları çektiği dosyayı bulduk. Bayrak görsellerini küçülterek bu işi halledebileceğimiz anladık. 212x212 olan bayrak görsellerini 32x32 boyutuna küçülttük. Bunu yaparken Photoshop programını kullandık. Program içerisinde ayarlamayı yaptık ve programın diğer tüm bayrakları aynı şekilde küçültmesini sağladık.    Bu işlemi yaptıktan sonra Haftalık maçlar kısmında bir hata oluştuğunu fark ettik. Bayrakları küçülttüğümüz için maç seçme bölümlerindeki görüntüde bozulma meydana geldi. Bunu düzeltmek için kodu tekrar inceledik. Uygun alanı bulup düzelttik. | | |
| **Tarih:** 25 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 3B OYUN GELİŞTİRME | | |
| Daha sonra oyun içerisinde dil hatasını fark ettik. Oyun, telefon dilinden bağımsız olarak İngilizce dilini seçiyordu. Bizim istediğimiz ise telefondan dil ve konum bilgisini alarak oyun içerisindeki dil ve ülke seçimini otomatik olarak yapmasıydı.  Bu kısım bizi çok zorladı. Çünkü bu işlemi yapan kodun nerde olduğunu bulmak çok vaktimizi aldı.  Öncelikle ülke seçimini yapmaya odaklandık.    Bu kodda, öncelikle telefonun konum bilgisine dayanarak hangi ülkede olduğunu saptadık. Daha sonra bu ülkenin ülke seçim listesinde kaçıncı sırada olduğunu saptadık. Programa bunu kodladık ve oyunun o sıradaki seçimi, yani telefonun bulunduğu ülkeyi seçmesini sağladık. | | |
| **Tarih:** 26 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 3B OYUN GELİŞTİRME | | |
| Otomatik ülke seçimini bitirdikten sonra oyunun otomatik dil seçimini yapmaya başladık.    Dil seçimini otomatik olarak telefonun diline ayarlanmasını sağladık. | | |
| **Tarih:** 27 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** 3B OYUN GELİŞTİRME | | |
| Bu işlemleri bitirdikten sonra genel olarak oyun içi hızı nasıl arttırırız diye düşünmeye başladık.  Oyunun hızını arttırmak için oyun içinde yüklenen bayrakları, resimleri vb. objeleri; ayrıca program içerisinde çok defa tekrar edilen döngüleri oyunun en başına alırsak programın hızlanacağını düşündük ve bunun için çalışmalara başladık.  Programın içinde yeni bir dosya tanımladık. Bu dosyanın içinde kullanılan objeler tanımladık ve diğer dosyaların içinde tekrar tekrar kullanılmasını önledik. | | |
| **Tarih:** 28 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Gradle Versiyon Yükseltme | | |
| Gradle android uygulamaları geliştirirken kullandığımız build sistemidir. Gradle ile debug, test ve release olaylarını kolaya hale getirmektedir. Android uygulamalarda tercih edilmesinin sebebi java resim, font ve xml dosyaları gibi android uygulamasında olması gereken bileşenleri barındıran paketleri tutabikmektedir. Aşağıda proje kurulurken karşımıza çıkacak klasörler ve dosyalar vardır.    **app**: .idea klasörü klasörü altında bulunan app klasörü yapılan projenin modülüdür. Eğer büyük ve karmaşık projeler yapılmayacaksa sadece bu klasörde çalışılacakatır.  **build**: build dosyalarının tutulduğu klasördür  **libs**: kütüphanelerin tutulduğu klasördür  **src**: Java. Res ve androidManifest dosyalarının bulunduğu klasördür  **build.gradle(app):** Bu dosyanın içinde kütüphane, proguard ve anahtarla ilgili bilgiler tutulur. Burada minSdkVersion, targetSdkVersion ve compileSdkVersion bulunur. | | |
| **Tarih:** 29 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |

|  |  |  |
| --- | --- | --- |
| **Yapılan Çalışmanın Konusu:** Gradle’da targetSdkVersion’un son sürüme güncellenmesi ve appcompat araştırması | | |
| Google play’e oyun eklemek veya hazırda olan oyununun güncelleme yapılması için targetSdkVersion’un son sürümde olması lazımdır. İlk olarak bende targetSdkVersion ve compileSdkVersion ve compileSdkVersion sürümleri 28’e güncellendi ek olarak buildToolsVersion 28.0.3 eklendi.    Bu işlemlerden sonra dependencies kısmındaki appcompat kütüphanesi altı çizili olduğu tespit edildi.  Appcompat SDK 21 ile gelmiştir. Android’e gelen her api seviyeleriyle beraber Android’e yeni özellikler geliyor. Bu yeni elemanlar veya özellikler önceki api seviyeleriyle kullanılmaz oluyor. Google da bu sorunu kaldırmak için Support Library adını verdiği kütüphanesi ile, yeni gelen özellikleri eski api lerde kullanabilmemizi sağlıyor. Geriye dönük uygulama desteğini sunmak adına gerçekten çok iyi bir kütüphane. Appcompat da, Android 4.0 ile gelen ActionBar‘ın geriye dönük desteğini sağlamak adına geliştirilmiş bir Support Library elemanı. Appcompat sayesinde ActionBar özelliğini Android’in Gingerbread versiyonlarından itibaren kullanabiliyoruz.    Buradaki hata bütün com.android.support kütüphanelerinin doğru çalışabilmesi için aynı versiyonda olması gerekir. Appcompat 28.0.0 deyken kullanılan diğer android support kütüphaneleri 28.0.0 olcak şekilde kodlara eklendi.    Bu günden sonra staj süresi bitmiştir. | | |
| **Tarih:** 29 / 12 / 2018 | Kaşe / İmza | **Sayfa No:** 22 |