BİL496 / YAP496
1. Sprint Sonu Raporu

(Barış Utku Mutan 201104082, Gökçe Başak Demirok 191401005,
Murat Şahin 191101010, Mehmet Can Şakiroğlu 191101001)

Genel Bakış

Genel olarak araştırma için ayırdığımız bu sprintte başarılı bir şekilde araştırmalarımızı gerçekleştirip beklediğimiz çıktılara ulaştık. Kullanmayı planladığımız araçlar hakkında bilgi toplayıp testler gerçekleştirdik, bazı araçlar yerine de farklı araçların daha uygun olabileceğine karar verdik. Genel olarak sağlıklı bir geliştirme sürecine uygun bir altyapı hazırladık.

Bu sprintte yapılan işleri konu başlıkları altında açıklayalım:

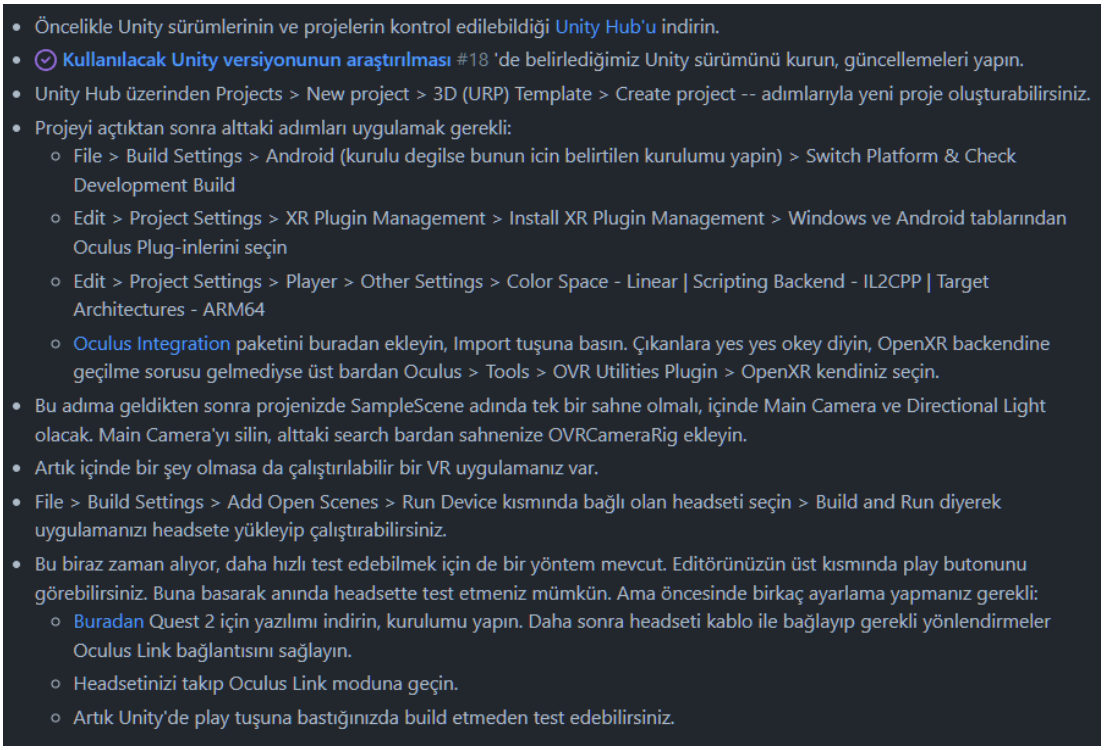
Yapılan İşler

- **Unity ile geliştirme süreci hakkında araştırma**

Bu kısımda öncelikle ilk olarak bir Unity sürüm seçimi yaptık, daha sonra Unity ile temel bir VR/AR uygulaması geliştirmek için nasıl bir süreç izleneceği hakkında araştırmalar yaptık. Ayrıca kullanıcı arayüzünü geliştirebilmek için bir bilgi temeli oluşturduk.

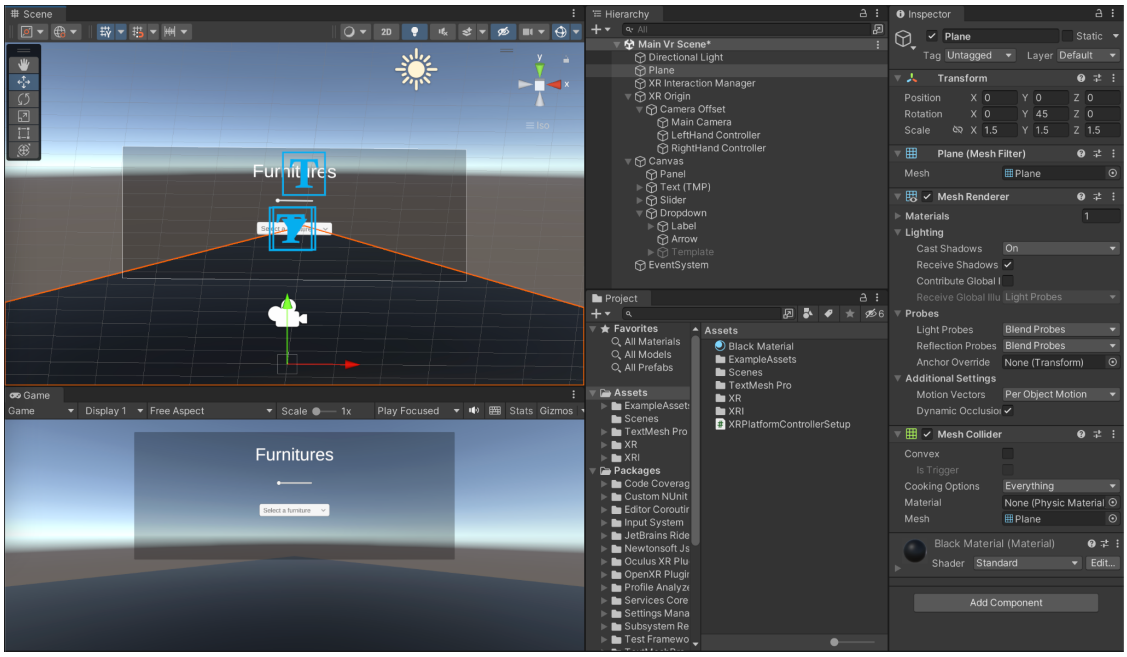
Başlangıç için Oculus tarafından Unity-VR uygulamaları geliştirmek için yayınlanan ve gayet güzel olan bu dokümantasyonu inceledik. Ayrıca bu (1 2 3) tutorialları izledik ve gerçekten çok faydalı oldu.

Sıfırdan çalışan bir uygulama geliştirme adımlarını şu şekilde özetledik:



Kullanıcı Arayüzü

Unity'de kullanıcı arayüzü eklenmesi hakkında araştırma taskı kapsamında https://www.youtube.com/watch?v=yhB921bDLYA "How to Make a VR Game in Unity 2022 - PART 7 - User Interface" videosundaki tutorialı takip ederek Unity içerisindeki bir VR uygulaması için UI nasıl hazırlanır, görsel özellikleri nasıl değiştirilebilir, oluşturulmuş kullanıcı arayüzü ile nasıl etkileşime geçilmesi sağlanılabilir, bu adımlar uygulanırken çıkabilecek problemler nasıl çözülebilir gibi konularda oldukça güzel bir şekilde bilgi edinilebilmekte. Bunlara ek olarak bu kapsamda UI ile etkileşime geçildiği takdirde olması gereken şeylerin kodlanması ile ilgili olarak scriptlerle nasıl çalışılabileceği de videoda örneklerle birlikte mevcut. Dolayısıyla bizim bu taskımız için, videonun izlenip adımların takip edilmesinin ileride yapacağımız UI çalışmalarımızı gerçekleştirebilmemiz için yeterli olacağını, en azından güzel bir temel oluşturmamızı sağlayacağını söyleyebiliriz. Örneğin önümüzdeki sprintte üzerinde çalışmayı planladığımız tespit edilen eşyalarla ilgili modellerin listesinin VR'da listelenmesi taskında bu videodaki gibi bir tutorial'ı takip ederek şu şekildeki gibi basit bir liste UI'ı oluşturmak mümkün:



- **Headset kamerası hakkında araştırma (Passthrough)**

Quest2 Passthrough API araştırma taskı kapsamında bu API hakkında araştırmalar yaptık. Passthrough, gözlük üzerinde bulunan kameraların birleşimi ile oluşturulan dış dünya görüntüsünün kullanıcıya sunulmasını sağlıyor. Uygulama geliştiriciler Passthrough API ile birlikte kendi uygulamalarına bu işlevi katabiliyor. Bu API'ın kullanımı hakkında bilgiler edindik ve basit bir iş akışının nasıl sağlanacağını keşfettik.

Unity üzerinden Passthrough Layer eklenip dış dünya görüntüsü kullanıcıya gösterilebiliyor fakat uygulama tarafından bu görüntü herhangi bir şekilde işlenemiyor (gizlilik nedenlerinden dolayı). Kendi dokümantasyonunda bu konu ile ilgili olarak bir uygulamanın direkt olarak görüntüye ulaşamaması durumuna değinilmekte. Bu nedenlerden dolayı headset görüntüsünden video stream edilerek veya daha inovatif bir çözüm bulunabilirse o şekilde bu görüntüye erişip işlenebiliyor. Örneğin ilerideki tasklardan biri olan, YOLO'nun odadaki hali hazırda gerçek hayatta olan eşyaları detect etmesi, taskında bu durumun üstesinden gelinmesi gerekli.

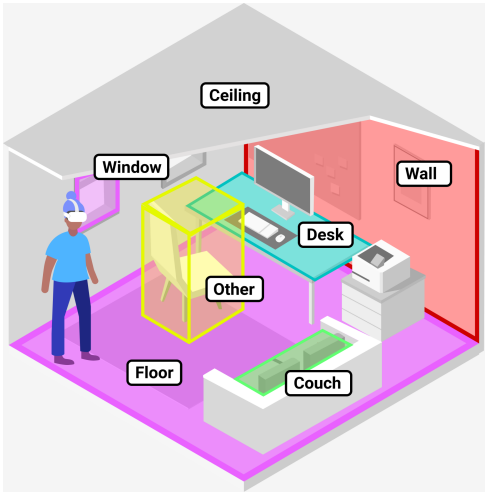
SideQuest adlı, Passthrough API'ın sağladığı etraftaki odanın gerçek görüntüsünün stream edilmesini sağlayabilen, başka bir uygulamanın bizim durumumuza yardımcı olabileceğini düşündük. Dolayısıyla, örneğin YOLO ile odadaki eşyaların türlerinin tespit edilmesi kapsamında, odadaki gerçek görüntüye erişim sağlamak için şu şekilde bir çözüm mevcut: Öncelikle Quest 2 Passthrough API yardımıyla kullanıcının etrafındaki gerçek görüntüyü görür hale gelmesini sağlanıp, sonrasında ise bu görüntünün SideQuest uygulaması yardımıyla stream edilmesi sağlanabiliyor. SideQuest'e stream edilen, kullanıcının gerçek dünyada içinde bulunduğu odanın görüntüleri ise "SideQuest Stream" penceresinin capture edilmesi ile alınıp, oradaki görüntüler üzerinde de YOLO'yu çalıştırılıp asıl amaca bu şekilde ulaşılabilmekte. Ancak

bu çözüm bağlamında VR gözlüğün örneğin bir dizüstü bilgisayara bağlı olması ve SideQuest'in oradan çalıştırılması söz konusu, dolayısıyla aslında önümüzdeki sprintte bu işlemi yapmanın bilgisayara bağlı olmayı gerektirmeyen bir yolunu bulabilirsek o çözümü tercih edeceğiz.

- Oda bilgilerinin çekilmesi hakkında araştırma

Kullanıcının bulunduğu oda hakkındaki bilgilerin (boyut, duvar konumu, eşya konumları) çekilebilmesi taskını gerçekleştirebilmek için RoomMapper eklentisinin kullanılabileceğini düşünmüştük. Fakat daha sonradan Oculus'un Presence platform adı altında duyurduğu gerçek ve sanal dünyayı birleştirmeyi hedefleyen bir oluşumu gördük. Bu hedefe yönelik de Scene Capture denilen bir sistem oluşturmuşlar.

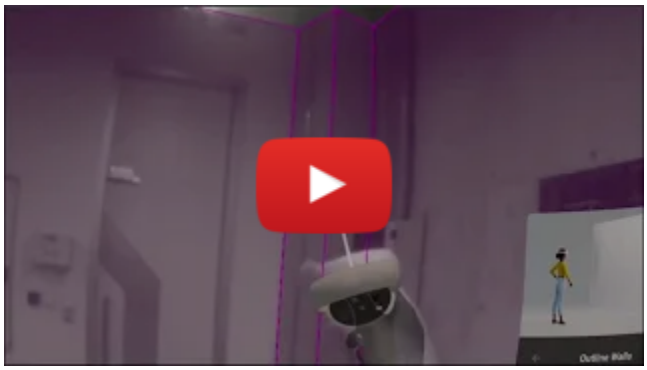
Bu sistem sayesinde kullanıcı odasını daha hiçbir uygulamaya giriş yapmadan tanıtabiliyor, biz de bu bilgiyi Oculus tarafından sunulan yardımcı bileşenler yardımıyla çekebiliyoruz. Böylelikle odanın ve eşyaların seçilmesi işlevi daha basit bir hale geliyor. Bu konu hakkında örnek bir görseli aşağıda görebilirsiniz.



Bu sistemi gördükten sonra, RoomMapper eklentisini tamamen bir kenara bırakıp bu eklentiyi araştırmaya odaklandık ve çok iyi sonuçlara ulaştık. Bu sistem kullanıcının oda bilgisini OS düzeyinde headset'e aktarabilmesini sağlıyor. Uygulama geliştiriciler bu bilgileri API'lar aracılığıyla kullanabiliyor. Sistemin internette yeterli bir dokümantasyonu maalesef bulunmamakta, kaynak kodu kendimiz okuyarak çoğu şeyi anlayabildik.

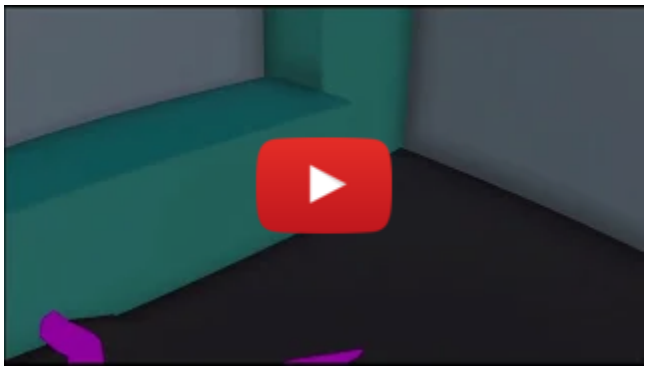
Oculus Integration paketi sayesinde Unity'e eklenen *OVRSceneManager* bileşeni genel olarak bu işlevleri yönetiyor. Uygulamamız kullanılmadan önce kullanıcılardan oda seçim işlevini tamamlamalarını talep ediyoruz. *OVRSceneManager.RequestSceneCapture()* çağrılarak bu talep iletilebiliyor. Daha sonra bu oda bilgisini de *OVRSceneManager.LoadSceneModel()* çağırarak uygulamamıza çekebiliyoruz. Bilgiler Unity'e çekildikten sonra yazdığımız scriptler ile bunları kullanabiliyoruz.

TM-210 VR Labındayken bulunduğumuz odayı başarıyla seçebildik ve bu sürecin videosunu çektik (Video SideQuest yardımıyla çekildi). Aşağıdaki videodan bunu izleyebilirsiniz:



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Ayrıca bu seçimi yaptıktan sonra Unity'deki örnek uygulamamızda bilgileri headsetten çekebildik. Seçtiğimiz duvarları ve 2 eşyayı (bir adet masa ve dolap) Unity'e çektiğimiz demoyu da aşağıdaki videodan izleyebilirsiniz. Yukarıdaki videoda seçilenlerin bu videoda yüksek bir doğrulukla yansıtıldığını gözlemleyebilirsiniz.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Şu anda Oculus Link üzerinden başarılı bir şekilde bu işlevi sağlamamıza rağmen bizden kaynaklı olmayan sebeplerden ötürü Build edilmiş uygulamada sorun yaşıyoruz. Bu sorunla ilgili olarak birkaç aydır Oculus tarafına iletilmiş geri bildirimler var, araştırmalarımız sürüyor.

- Obje Tespiti hakkında araştırma

Odada olan eşyaların türlerinin tespit edilmesi taskının bir alt taskı olan YOLO sürümlerinin araştırılması taskı tamamlandı. Bu task kapsamında yapılan araştırmalar sonucu projemiz için kullanılması en mantıklı olacak YOLO sürümünü YOLOv5 olarak seçildi. Üzerinde eğitildiği dataset itibari ile pretrained modellerinin hali hazırda 15 furniture objesini tanıyabilen modeller olması, daha öncesinde android için uygulamalar kapsamında da implement edilmiş olması, uzun bir süredir kullanılan ve üzerindeki birçok konuda daha rahat kaynak bulunabilen,

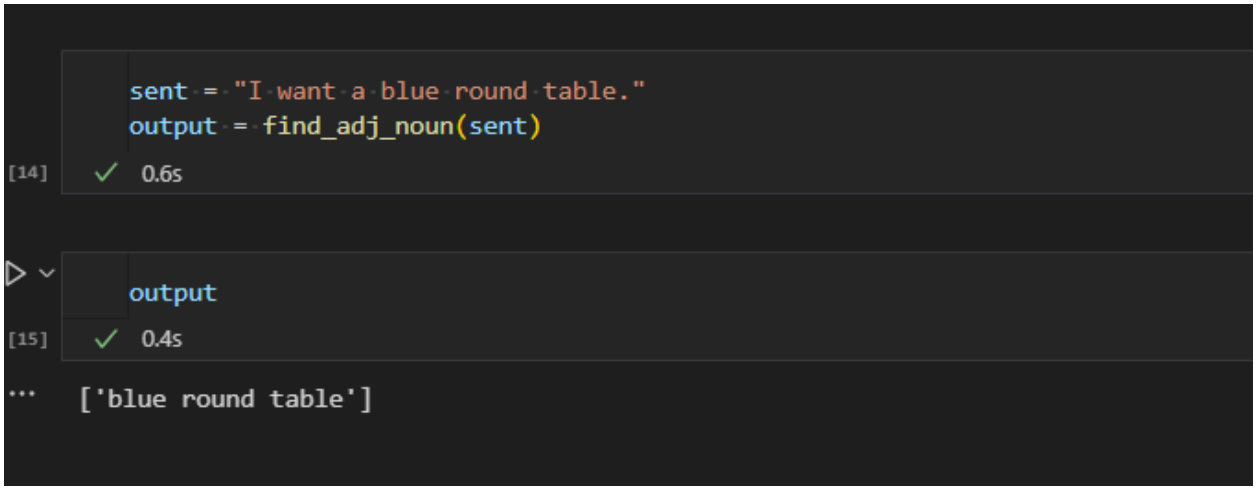
rahatlıkla üzerinde çalışılabilecek güzel bir GitHub reposu olan (GitHub - ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite) bir version olması, pretrained modellerinin oldukça güzel başarımlarla furniture objelereini detect edebiliyor olması, üzerine belki ek bir training yapılmasına gerek duyulmayacak belki de gerek duyulacak olursa hali hazırda benzer işleri oldukça iyi yapan bir pretrained model finetune edileceği için YOLO'nun 5. versiyonun projemiz kapsamında seçilmesi en mantıklı olan versiyon olduğuna karar verdik.

YOLO için furniture veri seti araştırması yapılması taskı kapsamında YOLO Sürümlerinin Araştırılması taskı üzerine çalışırken bulunmuş hali hazırda 15 furniture objesini yüksek bir başarımla detect edebilen bir pretrained model (YOLOv5) var olması sebebiyle YOLO'nun train edilmesi için bir dataset gerekliliği problemi büyük ölçekte ortadan kalktı. Projemiz kapsamında algılanması beklenilen bir odanın içinde bulunabilecek furniture objelerini hali hazırda tanıyabilen bir model mevcut. Dolayısıyla bu task kapsamında bir dataset bulunması gerekliliği artık yok. İleride yukarıda bahsedilmiş olan 15 furniture objesi dışında bir furniture eklemek istememiz durumunda, COCO dataset'i üzerinde eğitilmiş olan YOLOv5 modelinin kendi eklemek istediğimiz eşyalara ait ve önceki 15 furniture objesine ait küçük çaplı kendimiz oluşturup etiketleme işlemini yapabileceğimiz bir dataset ile finetune edilmesi söz konusu olabilir, bu durumda ise bu sorunu YOLO Training taskı kapsamında (önümüzdeki, 2. sprintte) ele alıp modeli, kendimiz oluşturacağımız ve etiketleyeceğimiz küçük bir custom dataset ile finetune etme yoluyla çözme yoluna gidebiliriz. Çünkü furniture objelerine ait available dataset'ler hep o 15 furniture kapsamında ve YOLO modeli de o 15 furniture'ı zaten hali hazırda tanıyabilir bir durumda. Ayrıca VR kısmından gelen Passthrough görüntüleri ile hazırlanan görüntülerle küçük bir finetuning yapma işlemini önümüzdeki sprintte ele alacağımız YOLO training taskı kapsamında yapmamız sadece yeni bir furniture eklemek için değil belki de VR Passthrough'dan gelen görüntülerin çözünürlük açısından çok da günümüz standartlarında görüntüler olmamaları sebebiyle de gerekli olabilir. Olması durumunda bahsedildiği üzere önümüzdeki sprintlerde YOLO taskı kapsamında bunlarla ilgilenilecek.

- ## Sesli Komut Sistemi hakkında araştırma

Bu konu hakkında önceliğimiz verilen komuttaki ev eşyasının anlaşılabilmesi idi. Doğal dil işlemede (Natural Language Processing- NLP) bu konuya indirgenebilir en uygun task'in Adlandırılmış Varlık Tespiti (Named Entity Recognition- NER) olduğu düşünüldü. Bunun için ilk aşamada ev eşyası tespitiyle ilgili önceden geliştirilmiş bir model olup olmadığı araştırıldı, ancak ne yazık ki daha önceden bu konuda eğitilmiş bir deep model bulunamadı. Bu noktada son teknoloji (state of the art) NLP modellerden biri olan BERT (Bidirectional Encoder Representations from Transformers) modelinin "base"'ini kullanarak ev eşyası tespiti yapılması planlandı. Bu base modeli kullanarak herhangi bir nlp task'ine çevirmek düşünüldü, ya da eğitilmiş ner modelini transfer learning yaparak adlandırılmış varlık bulan ağların son katmanlarını ev eşyası bulma üzerine eğitimek düşünüldü.

Araştırmalara devam edilirken ev eşyasını sıfat tamlaması olarak ele almak fikri oluştu. Daha kolay olan bu yolda SpaCy "en_core_web_sm-3.0.0" modeli kullanılarak cümledeki sıfat tamlaması bulundu.



Bunun sonucunda daha zor olan ve ilk paragrafta anlatılan data - model oluşturma ilk sprint için askıya alınmıştı fakat ilk sprintin sonuna gelinen günlerde sıfat tamlaması bulmanın yetersiz kaldığı alanlar fark edildi, bu alanlar:

-Kullanıcı sıfat tamlaması vermek yerine direkt isim olarak istekte bulunursa, örneğin:

“Mavi masa” yerine “masa” demesi

-Kullanıcı sıfat tamlaması yerine isim tamlaması olan bir eşya isterse, örneğin:

Masa lambası

Bu fark edildikten sonra ilk paragrafta anlatılan yolun, yani cümlenin içindeki ev eşyası tespiti için model geliştirmenin gerekli olduğu kararına varıldı. İkinci sprintte bunun üzerine uğraşılacaktır.

…

- ## 3D modeller veritabanının oluşturulması ve ilgili araçların geliştirilmesi

Bir önceki paragtafta bahsedilen menüde listelenecek ve simülasyonda kullanılacak olan 3D modellerin elde edilmesiyle iligli çalışmalar büyük ölçüde tamamlanmıştır. SketchFab API kullanılarak geliştirilen modelde gerekli API izinleri alınmıştır ve ekibin üstünde mutabık olduğu bir kelime listesi belirlenmiştir. Geliştirilen API connector ile kelime listesi verilerek listedeki her eşya için sorgu gerçekleştirerek simülasyonda derleme zamanında hazır bulunacak veritabanı oluşturulmuştur. Eşyaların yürütme zamanında load ve render edilmelerine dair zaman kısıtı gibi zorluklar bulunduğundan bu tasarım kararının daha derin araştırılması ve geliştirilmesiyle ilgili konular ikinci sprintte tartışılacaktır.

- ## Video Sohbet ve Sesli Sohbet

Video chat sistemi Agora Real Time Chat SDK kullanılarak gerçekleşmesi için ilgili API izinleri alınmış, gerekli scriptler yazılmış ve temel arayüz geliştirilmiş; video chat, voice chat gibi kabiliyetler projeye eklenmiştir.

Sonraki Sprint Planlaması

- Bilgisayarla Görü Tabanlı Eşya Tanıma Sisteminin Geliştirilmesi, Eşya Modellemeleri İle Etkileşime Geçilmesi

YOLO ile odada olan eşyaların türlerinin tespit edilmesi ana taskı (https://github.com/cansakiroglu/DesignIt/issues/2, bu task kapsamında VR headsetinden odaya dair almış olduğumuz bir görüntüdeki furniture objelerini detect edebilen bir model elde etmek amaçlanmaktadır) ve bunun bir alt taskı olan YOLO training taskı (https://github.com/cansakiroglu/DesignIt/issues/23), tespit edilen eşyalarla ilgili modellerin listesinin VR'da listelenmesi taskı (https://github.com/cansakiroglu/DesignIt/issues/9, bu task kapsamında ise YOLO ile tespit edilen eşyalara ait modellemelerin bir listesinin görülebilir ve etkileşime geçilebilir bir biçimde oluşturulması amaçlanmaktadır) ve Unity içerisinde eşyalarla seç-sürükle-bırak-rotate gibi etkileşime geçilebilmesi taskı (https://github.com/cansakiroglu/DesignIt/issues/13) ikinci sprintte yapılması planlanmış olan mevcut tasklardandır. Bu taskların ETA'sı 2. Sprint'in sonu olup, Mehmet Can Şakiroğlu'na assign edilmiştir.
[Bunlara ek olarak VR headsetinden gelen görüntülerin çözünürlüğünün yetersiz olmasından dolayı sorun çıkması durumunda küçük ek bir dataset ile finetuning yapma taskı da YOLO taskları kapsamında ele alınacaktır, bu durumun söz konusu olması durumunda bu ayrı bir task olarak açılacaktır ve ETA'sı ise büyük ihtimalle 3. Sprint'in sonu olacaktır. Dolayısıyla bu durumun ortaya çıkması durumunda buna ait planlama 2. Sprint'in sonunda yapılacaktır.]

- 3D modellerin render edilmesi ve odada instantiate edilmesi

Veritabanından çekilmiş 3D modellerin seçilerek odadaki koordinat sistemine riayet edecek şekilde renderlanması ve kullanıcı girdisiyle beraber gerekli pozisyonda instantiate edilmesiyle ilgili gerekli araştırmalar yapılması ve proof of concept düzeyinde bir örneğin gerçekleştirilmesi hedeflenmektedir. İlgili issue https://github.com/cansakiroglu/DesignIt/issues/28 olarak açılmış ve Barış Utku Mutan'a atanmıştır.

- Ana Giriş Menüsünün Tasarımı ve Geliştirilmesi

Kullanıcıyı simülasyon ilk çalıştığında karşılayıp, ‘’odayı tanıtmaya başla’’, ‘’mevcut odayı yükle’’, ‘’ayarlar’’ ve ‘’çıkış’’ gibi temel opsiyonların olduğu ana giriş menüsüyle ilgili çözümlerin araştırılması, menünün tasarlanması ve ilgili componentlerle birlikte hazır edilmesi hedeflenmektedir. İlgili issue https://github.com/cansakiroglu/DesignIt/issues/29 olarak tanımlanmış ve Barış Utku Mutan'a atanmıştır.

- ## Auth0 ile Kimlik Doğrulama Sağlanması

Kullanıcı uygulamayı ilk açtığında ve kullanıcıya ana menü sunulduğunda kimlik doğrulama için bir mekanizma kurulması hedeflenmiştir. Öncelikle Auth0 sistemine kayıt yapılacak ve platformun sunmuş olduğu API'ların Unity-VR geliştirme sürecine dahil edilmesi ile ilgili araştırma yapılacak. Daha sonra da yapılan araştırmalar ışığında projemize işlev eklenecek. İlgili issue https://github.com/cansakiroglu/DesignIt/issues/6 olarak tanımlanmış ve Murat Şahin'e atanmıştır.

- ## Speech-to-text sağlanıp çıktının kullanıma uygun hale getirilmesi

Oculus tarafından sunulan Voice SDK yardımıyla veya kendi oluşturacağımız ASR metodları ile kullanıcının konuşmasını yazıya dökme işlevini gerçekleştireceğiz. Yazıya dökülen bu konuşma ise ileride NLP modellerinde kullanılmak üzere scriptlerde kullanılmaya hazır hale getirilmiş olacak. İlgili issue https://github.com/cansakiroglu/DesignIt/issues/10 olarak tanımlanmış ve Murat Şahin'e atanmıştır.

- ## Çıktı cümlenin içinden ev eşyası tespiti için data ve model oluşturma

Bert base modeli kullanılacak olup, ner task'i etiketleri değiştirilerek ve transfer learning yapılarak model oluşturulacaktır.

Bu model için gerekli data el ile oluşturulacak olup en az iki bin cümle içerecektir. Bunun için ev eşyası listesine ihtiyaç vardır. En az 50 cümle manuel şekilde yazılıp cümlenin içindeki ev eşyası kısımları değiştirilerek data augmentation ile data çoğaltılacaktır.

- ## Model sonucundaki ev eşyasının eş anlamlısı olma ihtimaline karşılık eş anlam bulabilen model araştırması yapılması

Süreçte fark edilen noktaların birisi de kullanıcı eş anlamlı bir kelime kullandığında anlamamız gerektiği oldu. Örneğin kullanıcı çöp kutusu için "bin", "trash", ya da "trash box" kullandığında aynı obje dönmeli. Bunun için de yine nlp model araştırması yapılmalıdır.