Design It
BİL496 / YAP496
3. Sprint Sonu Raporu

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

Genel Bakış

İkinci sprintte eyleme dökmeye başladığımız konuların ilerleme ve geliştirilmesi üzerine başladığımız bu sprintte, github repomuzun proje üyeleri arasındaki senkronizasyonunu koruyup uygulamamıza planlanan fonksiyonları eklemeye devam ettik.

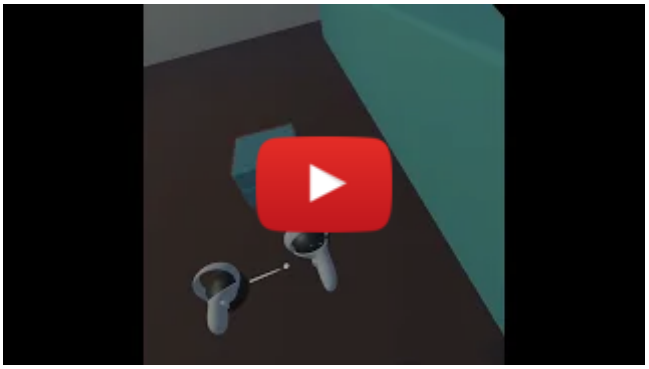
Bu sprintte yapılan işleri konu başlıkları altında açıklayalım:

Yapılan İşler

3D modellerin sahneye eklenip sahneden silinebilmesi

Önceki sprintlerde kullanıcı odasına basit bir küp ekleyebiliyordu. Artık yüklenebilen her modeli belirtilen yerde oluşturup, oluşturulan modelleri de silebiliyoruz. Ayrıca modeller geçersiz yerlerde (duvar içleri vs.) oluşturulamıyor, ayrıca bu issue kapsamında kontrolcüden gönderilen ışının çizilmesi de sağlandı. Bu issue #41 kapsamında ele alınmıştır.

Aşağıdaki videoda işlevi gözlemleyebilirsiniz. Oculus assetlerinden bir sandalye ile test yapıldı.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Obje Manipülasyonu ve Fizik Sistemi

Bu task aslında bu sprint için planlanmamıştı fakat proje yavaş geliştirildiği için bu sprint kapsamında yapıldı. Amacımız kullanıcının odasına eklediği eşyaları manipüle edebilmesini sağlamak. Kullanıcı böylelikle objeleri hareket ettirebiliyor, döndürebiliyor ve büyütüp küçültebiliyor.

Objelerin oluşturulma anında gerekli scriptler objeye ekleniyor, dolayısı ile modellerin yürütme zamanında eklenmesi bir sorun değil. Modelin kaynağından bağımsız olarak mümkün olan en az varsayımla geliştirildi, objenin üzerine objenin tüm parçalarını kapsayan en küçük küp oturtuluyor ve onun üzerinden işlem yapılıyor. Ayrıca objenin üzerinde bulunduğu yüzeyin pozisyonu, rotasyonu ve ölçeği gibi değerlerden etkilenmiyor. Bunlardan dolayı Sketchfab gibi platformdan alınan modeller için de düzgün çalışıyor.

- Kontrolcü obje üzerine tutulduğunda obje çevresinde kapsayıcı küp çiziliyor.
- Sağ kontrolcüyle tutulan obje hareket ettirilebiliyor, farklı yüzeylere çarptığında durabiliyor.
- Sol kontrolcüyle tutulan obje döndürülebiliyor.
- Hem sağ hem sol kontrolcüyle tutulan obje büyütülüp küçültülebiliyor.

Kullanıcı ister kontrolcüden ışın çıkarıp uzaktan, veya objenin yakınına gelip direk temas şeklinde işlevleri gerçekleştirebiliyor. Bu issue #43 kapsamında ele alınmıştır.

Aşağıdaki videoda bahsedilen işlevleri gözlemleyebilirsiniz.

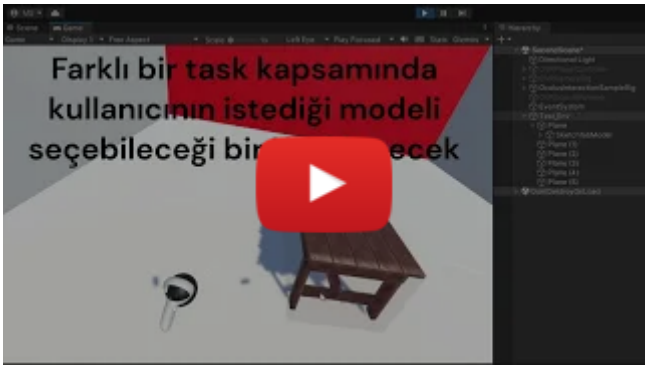
(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Sketchfab yürütme zamanında model yükleme

Önceki sprintlerde kullanıcının odasına ekleyebileceği modeller için lokalde oluşturduğumuz veritabanını kullanmayı planlıyorduk. Dönem ortası demo sunumunda aldığımız geri bildirim üzerine artan 3D model sayısıyla birlikte ölçeklenebilecek yaklaşımlar araştırmaya başladık. Araştırmalarımızda yakın zamanda bağımsız bir platformun Sketchfab modellerini çalışma zamanında indirip sahneye eklemek için bir API oluşturduğunu gördük.

Zoe Immersive platformu tarafından geliştirilen Sketchfab API ile çalışma zamanında verilen keyword için model listesi çekilebiliyor ve istenen model sahneye konulabiliyor. Başarılı bir şekilde bu API kullanıldı ve halihazırda var olan sistemlere entegre edilerek oluşturulan objenin manipülasyonu gibi fonksiyonların da çalışması sağlandı. Bu task kapsamında arayüz oluşturulmadı, bir alttaki başlıkta arayüz işlevini gerçekleştiren tasktan bahsedeceğiz. Bu issue #42 kapsamında ele alınmıştır.

Arayüz oluşturulmadan önceki halini aşağıdaki videodan izleyebilirsiniz.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Dinamik oyun içi model-envanter menüsü

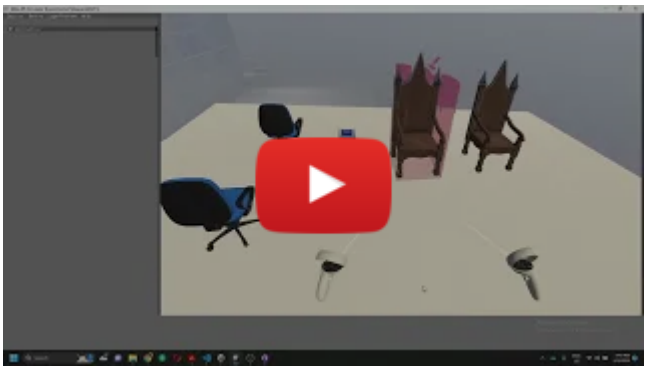
Kullanıcının odaya eşya ekleme/çıkarma işlemlerini gerçekleştirebilmesi için gerekli olan oyun içi envanter ile ilgili çalışmalar gerçekleştirilmiştir.

Az önceki başlıkta objenin dinamik olarak sahneye eklenebilmesi işlevini gerçekleştirdiğimizden bahsetmiştik. Bu başlık altında da aynı API kullanılarak verilen anahtar sözcükler için arama, modeli temsili görsel sağlama ve model seçme vb. işlevlerini sağlayan mekanizma geliştirilmiştir.

Oyun içi menüde kullanıcının indirdiği her 3D modeli görüntüleyebildiği, istediğini aktif edebildiği eşya slotları bulunmaktadır. Her slot, bir adet 3D modeli temsil etmektedir. Kullanıcı araması sonucunda çıkan veya envanterine daha önce eklediği modelleri görüntüleyebilmektedir. Arama sonucunda çıkan modellerden istediğini indirip daha sonra oda içinde yaratabilmektedir.

Bu görevle ilgili çalışmalar #36 ve #44 nolu issuelar kapsaminda Listing Sketchfab Thumbnails , Download & Instantiate Feature Added, Object Spawning Functionality Improved ve Follow camera utility added. commitleriyle ilgili branch #42 44 Sketchfab Runtime ‘a eklenip 46 nolu pull request ile ve main branch’e eklenmiştir.

Aşağıda menüyle ilgili demo videosunu izleyebilirsiniz. Demo videosunun çekilmesi hedeflenen 4 Nisan 2023 gününde laboratuvar kapalı olduğundan ötürü demo Meta XR simulator kullanılarak kaydedilmiştir.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Headset - Bilgisayar arası obje tespiti için API oluşturulması

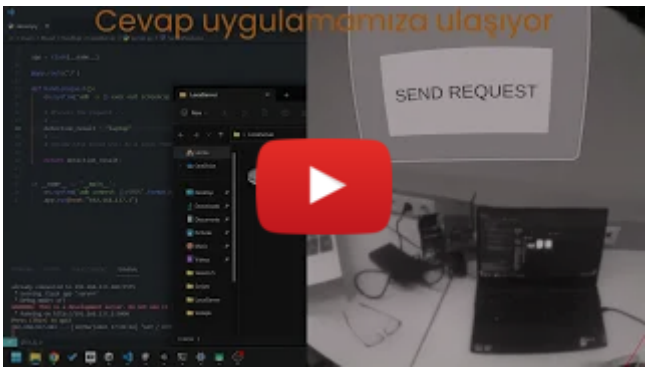
Geliştirilen obje tanıma modülünün çalışması için tamamen headsete gömülü bir sistem oluşturamıyoruz, çünkü önceki raporumuzda da bahsettiğimiz gibi Meta güvenlik nedeniyle uygulamaların Passthrough API yardımıyla gelen görüntüleri işlemesini mümkün kılmamış. Dolayısı ile görüntülerin headsetten bilgisayara, daha sonra da sonuçların bilgisayardan headsete aktarılması gerekliydi.

Bu taskın gerçekleştirilmesi için ADB (Android Debug Bridge) aracını kablosuz olarak kullandık. Bilgisayarla headseti aynı ağa bağlayıp ADB yardımıyla komut bağlantısı kurduk. İki cihaz arasında sürekli mesajlaşmalarla sonuç olarak headsetin ilettiği bir fotoğrafa karşılık headsete bir etiket dönebiliyoruz.

Çalışma adımları aşağıdaki şekilde:

1. Kullanıcı gerçek hayattaki bir objenin türünü öğrenmek istediğinde bir metodun triggerlanıp bilgisayarda çalışan API'a istek atması.
2. API'da istek alınınca Wireless ADB yardımıyla headset'e screenshot için komut gönderilip kameradan 1 frame alınması.
3. Gelen framenin YOLOv5 modeline verilip 1 adet etiket alınması. (bu adım şu an yok)
4. Elde edilen etiketin headsetin ilk adımda göndermiş olduğu POST requeste response olarak dönülmesi.

Aşağıdaki videoda bağlantının çalışmasını gözlemleyebilirsiniz, 3. adım daha sonra gerçekleştirileceğinden dolayı API sabit bir değer dönüyor.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

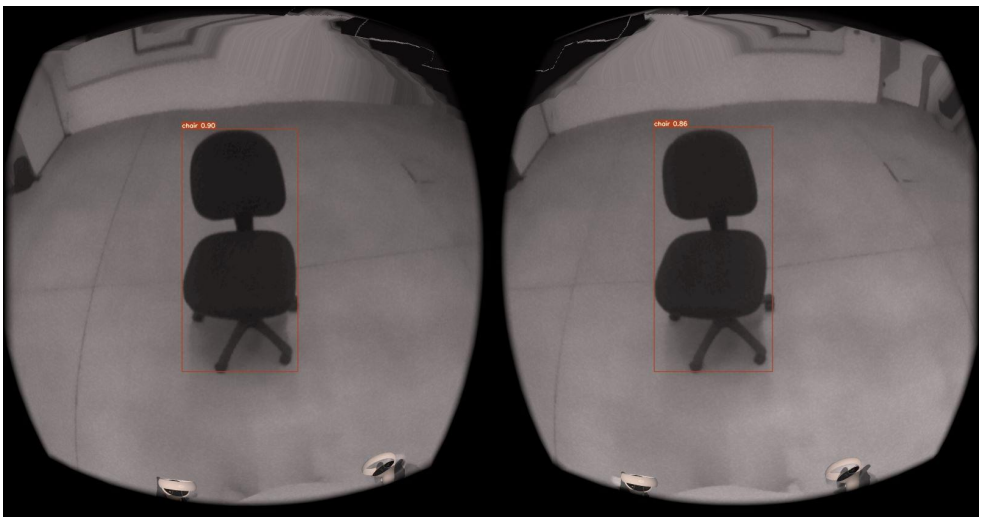
YOLO'nun, VR'dan Gelen Görüntülerle Çalışma Performansının İyileştirilmesi & Proje Codebase’ine Eklenmesi

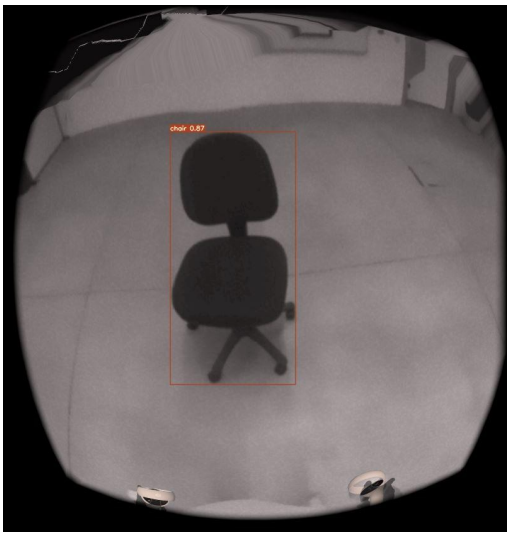
Bu issue kapsamında, VR'dan gelen görüntüler üzerinde YOLO modelinin çalışma başarımını yükseltme ile ilgilenildi. Bu noktadaki temel sorun VR'dan gelen görüntülerin düşük çözünürlüklü ve düşük kaliteli görüntüler olması. Bunlara ek olarak fisheye formatında olması, noisy olması gibi sorun kökeni ihtimalleri de mevcut. Bu sebeple, bu gibi sorunlara address edebilecek çözümler denendi. Sorunun, görselin fisheye olmasından kaynaklanması ihtimali üzerine defisheye yöntemleri denendi ancak performansta kayda değer bir değişim elde edilemedi. Defisheye yöntemleri özellikle görselin kenar ve köşe bölgelerinde kayda değer değişikliklere sebep olabildiği için ve bizim ilgilendiğimiz nesne görselin merkezinde bulunduğu için bunun tam bir çözüm olmaması aslında mantıklı. Sonrasında sorunun, görselin noisy olmasından kaynaklanması ihtimali üzerine basit blurlama, opening-closing tabanlı morfolojik operasyon tabanlı çeşitli denoising yöntemleri denendi ancak bu da performansta kayda değer bir yükseliş sağlamadı, tersine bazen düşüşe sebebiyet verdi.

Ancak bu noktada, daha önce projemiz amacı kapsamında yeterli gözüken en küçük model olması sebebiyle seçtiğimiz yolov5 yerine daha büyük ve daha kapsamlı modeller denendi. Bununla beraber yolov7 modelinin, yine detect edemediği görseller olsa da, daha önce denenenlere göre daha yüksek bir başarıma ulaşabildiği gözlemlendi. Bu nedenler ile yolov7 modeli temel alınarak çalışılmaya devam edildi.

Verilen bir görsel için ilgili label'ı çıktı olarak sağlayabilen bir code-base, yolov7 repository'sini temel alan ve projemiz kapsamında gerekli yerlerde değişiklikler (mainly detect.py) ve gerekli yerlerde eklemeler (mainly query.py) ile elde edildi. Bahsedilen geliştirmeler "ObjectDetection" adlı directory altında, "Object Detection - YOLO" adlı commit ile projeye eklendi.

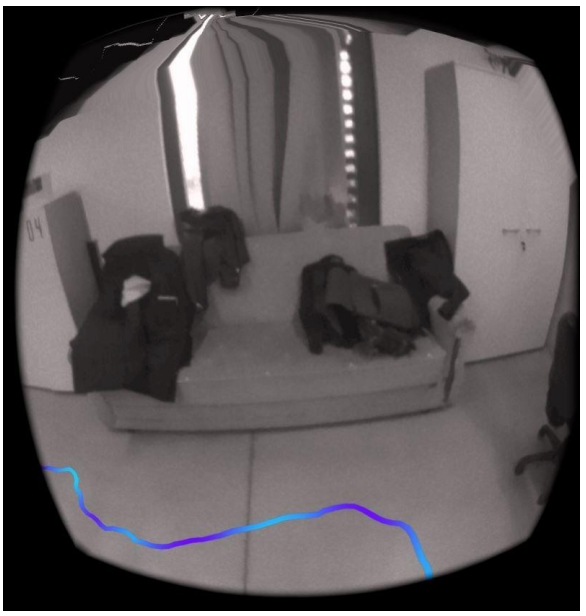
Bir önceki sprint sonunda detect edilemeyen ancak şu anki sistem ile detect edilebilir hale gelmiş bir örnek şu şekilde (chair):





Farklı açılardan bir detect edilebilen ve bir de detect edilemeyen iki örnek (couch):

Positive (Correct Detection)



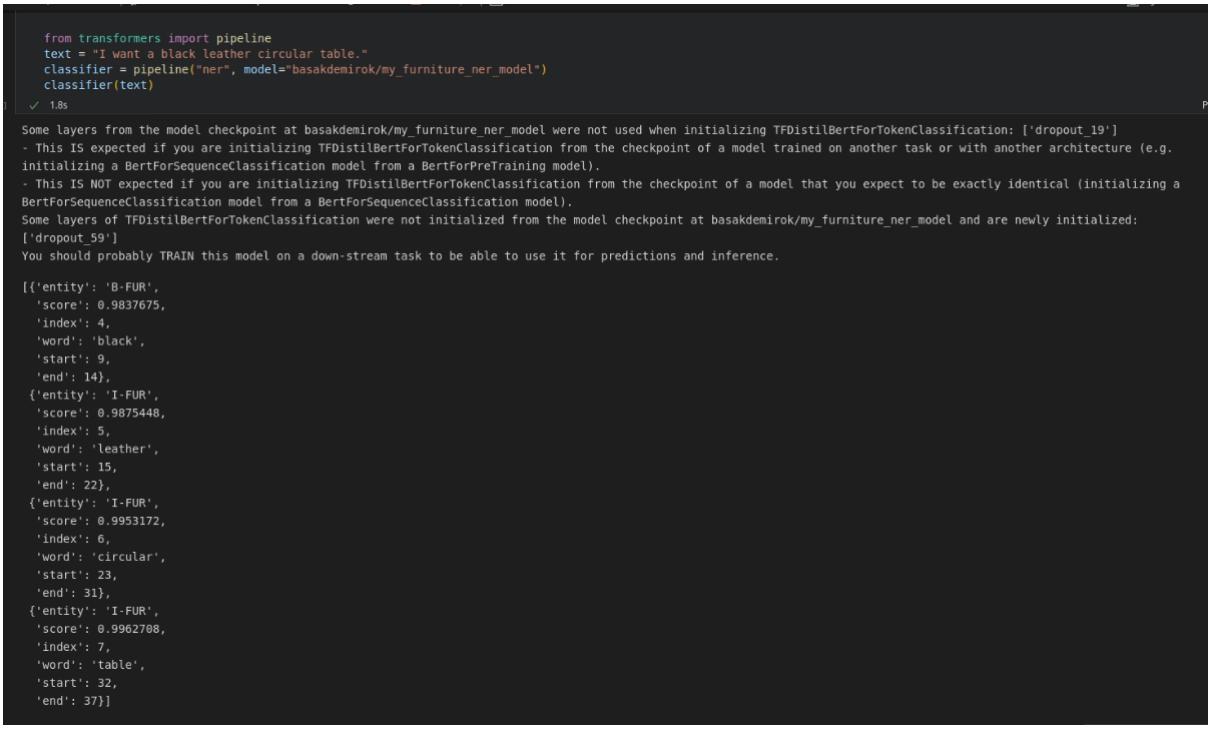
Negatif (No Detection)

NLP Model Geliştirmesi

Cümlede geçen ev eşyalarının tespiti için 2.sprintte geliştirdiğimiz Bert tabanlı modelin ev eşyalarının sıfatlarını yakalamada yetersiz kaldığını gördük. Bunun için modeli sıfat içeren veriyle besledik. Bu sıfatlar şekil (shape), materyal (material) ve renk (color) olarak modele eklenmiştir.

Yeni halinde:

Input: “I want a leather black circular table.” -> Output: leather black circular table)



Aynı zamanda önceki modelimiz bir cümlede birden fazla farklı furniture objeleri içerdiğinde tek bir furniture objesi varmış gibi davranıp bunu oluşturan kelimeler bunlardır şeklinde bir output veriyordu. Yeni halinde ise birden fazla ev eşyası geçtiğinde ayrı ayrı bu objelerin başlangıç kelimelerine “B-FUR” etiketi atıyor.

```
text = "I want a table lamp and a tv."
classifier = pipeline("ner", model="basakdemirok/my_furniture_ner_model")
classifier(text)
```

2.4s Python

```
Some layers from the model checkpoint at basakdemirok/my_furniture_ner_model were not used when initializing TFDistilBertForTokenClassification: ['dropout_19']
- This IS expected if you are initializing TFDistilBertForTokenClassification from the checkpoint of a model trained on another task or with another architecture (e.g.
initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing TFDistilBertForTokenClassification from the checkpoint of a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
Some layers of TFDistilBertForTokenClassification were not initialized from the model checkpoint at basakdemirok/my_furniture_ner_model and are newly initialized:
['dropout_79']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.

[{'entity': 'B-FUR',
  'score': 0.9799975,
  'index': 4,
  'word': 'table',
  'start': 9,
  'end': 14},
 {'entity': 'I-FUR',
  'score': 0.96997136,
  'index': 5,
  'word': 'lamp',
  'start': 15,
  'end': 19},
 {'entity': 'B-FUR',
  'score': 0.9536613,
  'index': 8,
  'word': 'tv',
  'start': 26,
  'end': 28}]
```

Kural tabanlı komut algılama sisteminin oluşturulması

Bu sprintte aşağıda bulunan konularla ilgili sesli komutların algılanması için kural tabanlı NLP modeli geliştirilmiştir:

*Video chat/ görüntüyü aç kapa, sesi aç kapa, çağrıyı başlat çağrıyı bitir
*Envanteri aç kapa / oyun için menüyü aç kapa
*Menüye dön

Bu model iki indexli array döndürüyor ve if-else bloklarından oluşuyor. İlk indexte bulunan sayı komutun hangi konuyla alakalı olduğunu belirtirken ikinci indexte ise bu konuyla alakalı olumlu (1) bir aksiyona mı yoksa olumsuz (0) bir aksiyona mı geçilmek istendiğini belirtiyor.

Örnek olarak:

```
sentence = "Turn the video chat off."
label_command(sentence)
```

[34] 0.0s

```
[0, 0]
```

```
sentence = "Turn on the video."
label_command(sentence)
```

[35] 0.0s

```
[1, 1]
```

```
sentence = "Close the inventory."
label_command(sentence)
```

[36] 0.0s

```
[3, 0]
```

```
sentence = "shut down the sound."
label_command(sentence)
```

[37] 0.0s

```
[2, 0]
```

Sonraki Sprint Planlaması

- İstenilen ev eşyaları için web arama sonuçlarının fiyat ve görsellerinin sunulması

Kullanıcının istediği ürünlerin (örneğin mavi masa) güncel olarak piyasada bulunan örnekleri ve bu örneklerin fiyatlarına bakabilmesi için Google Alışverişler'den çekilen ürün örnekleri ve fiyatları kullanıcıya sunulacaktır.

Bu kapsamdaki issue GitHub'da https://github.com/cansakiroglu/DesignIt/issues/12 olarak tanımlanmış ve Gökçe Başak Demirok'a atanmıştır.

- Sesli komutların (görüntü aç/kapa vs.) unity kısmında anlaşılabilirliği

Şuanki durumda bu model dışarıda yazılmış olup unity'e entegrasyonu sağlanmamıştır. 4. sprint sonuna kadar bu komut modelinin unity tarafındaki istemci ile sunucu arasında iletişimi sağlanacaktır.
Bu kapsamdaki issue GitHub'da https://github.com/cansakiroglu/DesignIt/issues/49 olarak tanımlanmış ve Gökçe Başak Demirok'a atanmıştır.

- Eşya Label'ı Belirlenmesi

Eşyaların türlerinin belirlenmesi eşyalara ait Unity'de kullanılabilecek modelin aratılması için gereklidir, dolayısıyla eşyaların türlerine ait label'ların belirlenmesi gereksinimi mevcuttur. Bu noktada, dönem ortası demomuzda aldığımız feedback'e de istinaden YOLO'nun bazen detection yapamaması ihtimali de mevcut olduğu için, bu durumlarda object detection ve ses algılama sistemlerinin beraber kullanılması ile eşya label'ının belirlenmesi amaçlanmaktadır.

Bu kapsamdaki issue GitHub'da https://github.com/cansakiroglu/DesignIt/issues/48 olarak tanımlanmış ve Mehmet Can Şakiroğlu'na atanmıştır.

- Entegrasyon, Refaktoring ve Optimizasyonların Tamamlanması

Task kapsamında ana Unity projesinin dışında geliştirilmiş olan ses tanıma, nesne tanıma, doğal dil işleme gibi özelliklerin projeye entegre şekilde çalışması için gerekli mekanizmaların gerçekleştirilmesi sağlanacaktır. Kod tabanında event yönetimine dair belirli mekanizmaların refaktör işlemleri gerçekleştirilecek. Obje render ve etkileşim tarafında simülasyonun hızlandırmasına yönelik optimizasyon işlemleri gerçekleştirilecektir. Proje genelinde ortaya çıkmış-çıkacak olan bugların çözülmesi sağlanacaktır. Genel olarak bu task projeye son halini vermek üzere tasarlanmıştır.

Bu kapsamdaki issue GitHub'da https://github.com/cansakiroglu/DesignIt/issues/50 olarak tanımlanmış, Barış Utku Mutan ve Murat Şahin'e atanmıştır