

Design It

BİL496 / YAP496

4. Sprint Sonu Raporu

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

Genel Bakış

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

Bu sprintte yapılan işleri konu başlıkları altında açıklayalım:

Yapılan İşler

Konuşmadan 3D modele ulaşmayı sağlayan hattın tamamlanması

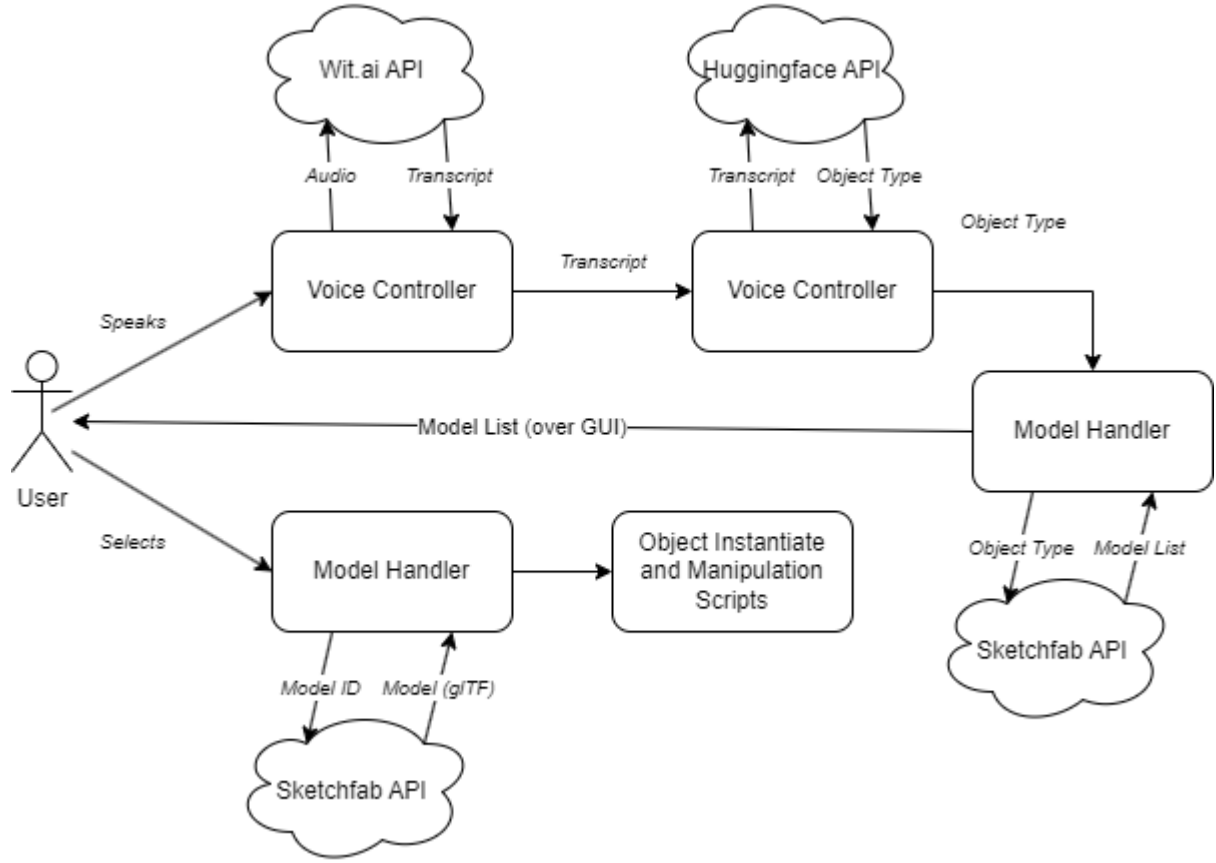
Bu issue kapsamında; geliştirdiğimiz sesten yazıya dönüştürme modülü, geliştirdiğimiz NLP modülü ve geliştirdiğimiz yürütme zamanında model yükleme modülü birleştirilerek kullanıcının istediği model türünü konuşarak belirttikten sonra karşısına çıkan seçeneklerden seçim yapıp sahneye koyabilmesini hedefledik.

Aşağıda sıralanan API'lar birleştirilerek ve önceden gerçekleştirilen geliştirmelerdeki hatalar düzeltilerek işlev gerçekleştirilmiştir.

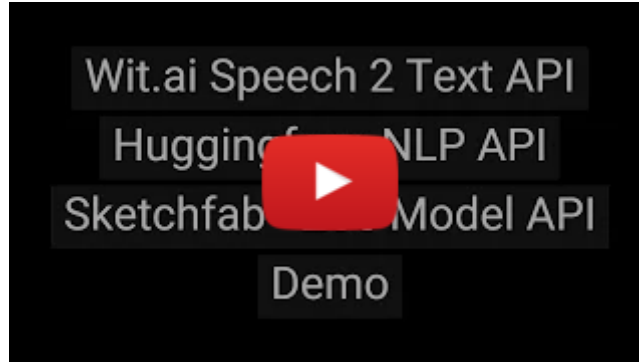
- 1- Wit.ai Speech2Text API
- 2- Huggingface Inference API (NLP)
- 3- Sketchfab-Zoe Runtime Load API

[fe0d2ab](#) altında yapılan geliştirmeleri görebilirsiniz.

Kurulan hat aşağıdaki şemada görülebilmektedir.



Aşağıdaki videoda işlevi gözlemleyebilirsiniz.



(Videoyu açmak için görselin üstüne tıklayıp linke basabilirsiniz.)

Kural Tabanlı Komut Sistemi

Ana hatlarıyla 3. sprintte yazılmış olup 4. sprintte gerekli ihtiyaçlara göre model güncellemesi yapılmıştır. Ayrıca sisteme entegre edilmesi için komutu anlayan model (metod) class haline dönüştürülmüştür. Son halinde modelin çıktısı iki boyutlu array olarak bırakılıp ilk index komut konusu hakkında bilgi verirken ikinci index olumlu ya da olumsuz aksiyon alınması konusunda bilgi verir.

Index 1:

video_chat # label: 0

goruntu #label: 1

ses # label: 2

envanter # label: 3

menu #label: 11

error #label: 12

Index 2:

olumsuz #kapa sonlandir #label: 0

olumlu #aç başlat#label: 1

menu_ozel #menuye don#label: 11

error #label: 12

Web Search

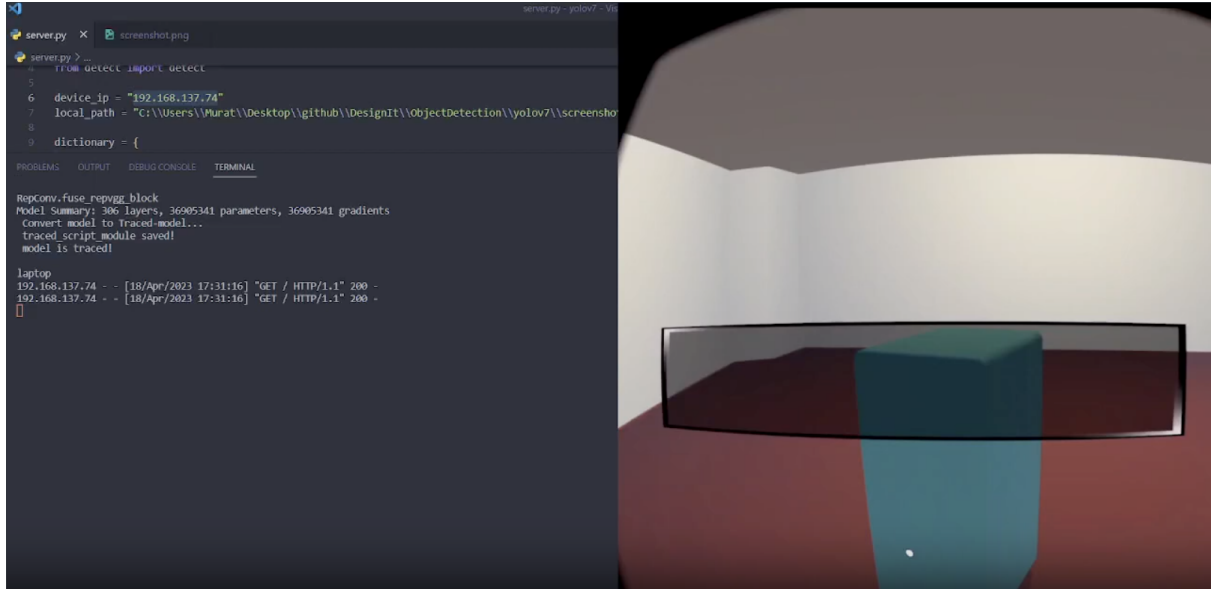
4. sprintte başladığımız bu issue'nun amacı kullanıcının istediği ürünün gerçek hayattaki piyasa fiyat ve örnekleri hakkında bilgi edinmesini sağlamaktır. Güncel piyasa bilgisini çekmek için Google Shopping'i arama sonuçlarını kullandık. Google'ı açmak gerekli komutu yazmak ve arama sonuçlarını lokale çekmek içinse Selenium api kullandık. Çıktılarda alakasız ürünler olabilir, bunun sebebi o kelimeler yazıldığında Google'ın o sonuçları dönüyor olmasıdır.

Eşya Label'ı Belirlenmesi

Bu issue kapsamında eşyaların türlerinin ("chair", "table"... gibi) belirlenmesi problemi ele alındı ve bu issue üzerinde çalışarak problemi çözümlendirdik. Issue kapsamında yapmış olduğumuz geliştirmeleri, issue ile aynı ismi taşıyan "#48 Eşya Label'ı Belirlenmesi" adlı commit ile beraber codebase'e ekledik.

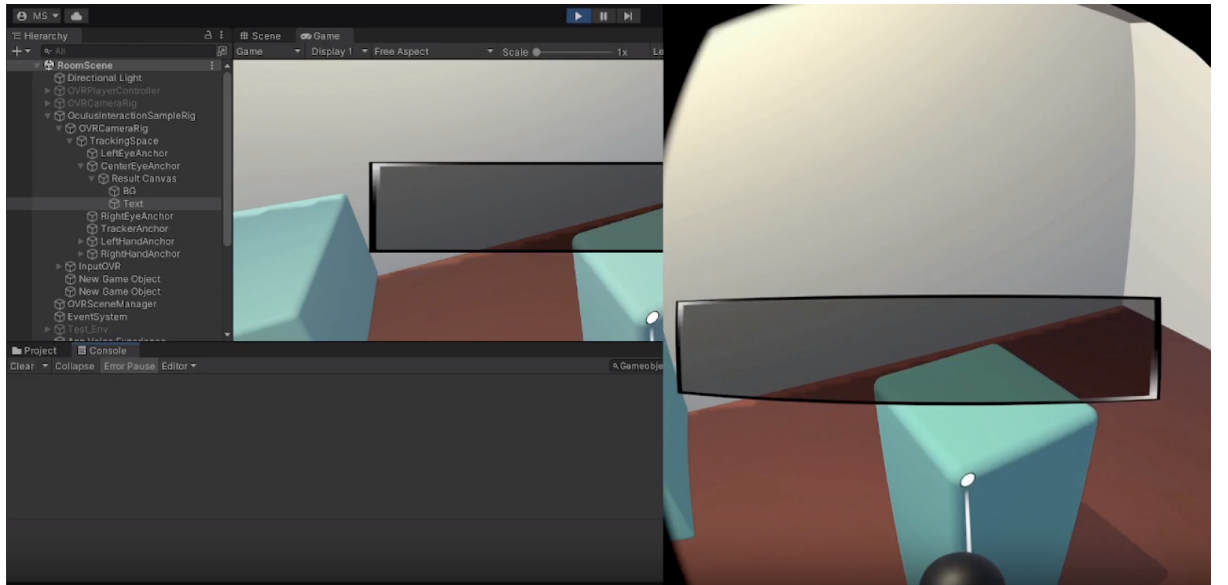
Bu problemin çözümü noktasında, daha önce de almış olduğumuz geri bildirimler yardımıyla şu şekilde bir çözüm planı kurguladık ve gerçekleştirdik:

- Kullanıcı bir eşyanın türünün ne olduğuna dair etiketin object detection modeli yardımıyla otomatik olarak algılanmasını isterse, sol controller'ı ile eşyayı point ederken sol controller'ındaki Y tuşuna basıyor. Bu noktada, cihazın passthrough özelliğine geçiliyor, elde edilen görüntü ise server'a gönderiliyor. Server kısmında object detection modeli ile o görüntüdeki eşyaya ait etiket bulunuyor ve dönülüyor. Bu etiket ise o kullanıcının başta işaret etmiş olduğu eşya modeline atanıyor. Bu kapsamındaki geliştirmelerimizi göstermiş olduğumuz video şu şekilde: (Sağda VR headset'ten kaydedilen görüntü mevcutken, solda ise çözümümüzün işleyişini/akışını gösteren ilgili log çıktıları mevcuttur.)



(Object Detection Demo Video - <https://youtu.be/9JzOiWiPwtA>)

- Kullanıcı bir eşyanın türünün ne olduğuna dair etiketi kendisi manual bir şekilde sesli komut yardımıyla belirlemek isterse, sol controller'ı ile eşyayı point ederken sol controller'ındaki X tuşuna basıyor. Bu noktada ise, kullanıcının diyecekleri dinlenilmeye başlanıyor ve sonrasında voice transcript yapılarak kullanıcının söyleyerek belirtmiş olduğu etiket elde ediliyor. Bu etiket ise o kullanıcının başta işaret etmiş olduğu eşya modeline atanıyor. Bu kapsamındaki geliştirmelerimizi göstermiş olduğumuz video ise şu şekilde: (Sağda VR headset'ten kaydedilen görüntü mevcutken, solda ise çözümümüzün işleyişini/akışını gösteren ilgili log çıktıları mevcuttur.)



(Voice Transcript Demo Video - https://youtu.be/h2TLcV_DBIQ)

Yapmış olduğumuz bu geliştirmeler sonucunda, issue kapsamındaki çözümümüzü gerçekleştirmiş olduk.