OYUN PROGRAMLAMA III

Unity ile Sanal Gerçeklik (VR) Uygulama Geliştirme

Ders Notu

Bilgi	Ayrıntı
Ders	Oyun Programlama III
Program	Bilgisayar Programcılığı, Meslek Yüksekokulu
Kurum	Sinop Üniversitesi
Hazırlayan	Öğr. Gör. İsmail Umut DURAN
Kullanılan araçlar	Unity (URP), XR Plug-in Management, OpenXR, XR Interaction Toolkit (XRI), Input System
Konu sayısı	6 ana konu + uygulama, soru ve sorun giderme bölümleri

Konu Akışı

Bölüm	Başlık
1	Sanal Gerçeklik (VR) ve Artırılmış Gerçeklik (AR) Kavramları ve Özellikleri
2	VR Başlıklarının Yapısı, Sistemi, Özellikleri ve Yetenekleri
3	Unity'de VR Projesi Oluşturma, XR Paketleri, VR Room Sahnesi ve Temel Bileşenler (XR Origin / XR Rig)
4	XR Device Simulator Kullanımı
5	Obje Tutma: XR Grab Interactable ve Attach Transform
6	Snap Turn, Continuous Turn ve Locomotion System
7	Ek: Sık Karşılaşılan Sorunlar, Laboratuvar Uygulaması ve Değerlendirme Soruları

Not (sürüm farkları)
Unity XR Interaction Toolkit (XRI) paketi sürümler arasında bileşen adlarını ve menü yollarını değiştirebilmektedir (örneğin eski "XR Rig" adı sonraki sürümlerde "XR Origin" olmuştur; XRI 3.x'te bazı sınıflar yeni ad alanlarına taşınmıştır). Bu notta bileşen adları genel kullanımıyla verilmiştir; derste kullandığınız paket sürümüne göre küçük farklar olabilir.

1. Sanal Gerçeklik ve Artırılmış Gerçeklik Kavramları

1.1 Temel Tanımlar

Sanal Gerçeklik (Virtual Reality, VR): Kullanıcıyı bilgisayar tarafından üretilmiş, üç boyutlu ve etkileşimli bir ortamın içine yerleştiren teknolojidir. Gerçek dünya görsel olarak (çoğunlukla) kapatılır; kullanıcı kafa ve el hareketleriyle sanal ortamda var olduğunu hisseder.

Artırılmış Gerçeklik (Augmented Reality, AR): Gerçek dünya görüntüsünün üzerine sanal nesne, bilgi veya efektlerin bindirilmesidir. Kullanıcı gerçek ortamı görmeye devam eder (telefon kamerası, şeffaf gözlük vb.).

Karma Gerçeklik (Mixed Reality, MR): Sanal nesnelerin gerçek ortamı "anlayarak" onunla etkileşime girdiği yapıdır (örneğin sanal topun gerçek masaya çarpıp sekmesi).

Genişletilmiş Gerçeklik (Extended Reality, XR): VR, AR ve MR'ı kapsayan şemsiye terimdir. Unity'deki paket ve bileşen adlarında geçen "XR" bu anlama gelir.

Özellik	VR	AR	MR
Gerçek dünya	Kapalı (tamamen sanal)	Görünür, üzerine bindirme	Görünür ve anlaşılır (mekân haritası)
Tipik cihaz	Meta Quest, HTC Vive, PS VR2	Akıllı telefon, AR gözlükleri	HoloLens, Quest passthrough, Vision Pro
Etkileşim	Kontrolcü, el takibi, bakış	Dokunmatik ekran, jest	El, bakış, ses, mekân
Sürükleyicilik	Çok yüksek	Düşük – orta	Orta – yüksek
Örnek	Bir VR korku oyunu, uçuş simülasyonu	Pokémon GO, mobilya yerleştirme uygulamaları	Sanal nesneyi gerçek masaya oturtmak

1.2 Temel Kavramlar

Kavram	Açıklama
Immersion (Sürükleyicilik)	Sistemin kullanıcıyı gerçek dünyadan ne kadar soyutlayabildiğinin teknik ölçüsüdür (ekran, ses, takip kalitesi).
Presence (Varlık hissi)	Kullanıcının "gerçekten orada olduğunu" hissetmesidir; öznel bir deneyimdir.
Interactivity (Etkileşim)	Kullanıcının ortamdaki nesneleri tutma, iterek hareket ettirme, butona basma gibi eylemlerle değiştirebilmesi.
DoF (Degrees of Freedom)	**3DoF:** yalnızca baş dönüşü (yaw, pitch, roll). **6DoF:** dönüşe ek olarak konum hareketi (ileri-geri, sağ-sol, yukarı-aşağı). Modern VR cihazları 6DoF'tur.
FOV (Görüş Alanı)	Kullanıcının aynı anda gördüğü açı. Geniş FOV sürükleyiciliği artırır.
Yenileme hızı (Refresh Rate)	Saniyedeki kare sayısı (72, 90, 120 Hz). Düşük kare hızı baş dönmesine neden olur.
Gecikme (Latency)	Kafa hareketi ile görüntünün güncellenmesi arasındaki süre. Düşük olmalıdır (kabaca 20 ms altı hedeflenir).

Kavram	Açıklama
IPD	Göz bebekleri arası mesafe. Başlığın lens ayarı bu değere uygun olmalıdır.
Tracking (Takip)	Baş ve kontrolcülerin konumunun algılanması. **Inside-out:** sensörler başlığın üzerinde. **Outside-in:** sensörler/istasyonlar odanın köşelerinde.
VR hastalığı (Motion Sickness)	Görsel hareket ile iç kulak (denge) algısı arasındaki uyumsuzluktan doğan mide bulantısı ve baş dönmesi.

1.3 Kısa Tarihçe

- **1956:** Morton Heilig'in Sensorama makinesi (çok duyulu deneyim).
- **1968:** Ivan Sutherland'ın "Sword of Damocles" adlı ilk kafaya takılan görüntüleme sistemi.
- **1990'lar:** Arcade ve konsol deneme ürünleri (örneğin Virtual Boy); teknoloji ve maliyet nedeniyle yaygınlaşamadı.
- **2012–2016:** Oculus Rift projesi, HTC Vive ve tüketici sürümlerinin çıkışı.
- **2019 sonrası:** Bağımsız (kablosuz) başlıklar (Meta Quest serisi) ve passthrough/karma gerçeklik özellikleri yaygınlaştı.

1.4 Kullanım Alanları

- **Oyun ve eğlence:** VR oyunları, sanal sinema, müzik/konser deneyimleri.
- **Eğitim:** Anatomi, laboratuvar simülasyonu, tarih/mekân gezileri.
- **Sağlık:** Cerrahi eğitim, rehabilitasyon, fobi ve ağrı terapileri.
- **Endüstri ve mimari:** Sanal prototip, bina gezisi, montaj/bakım eğitimi.
- **Askeri ve acil durum:** Simülasyon tabanlı eğitimler.

1.5 Artırılmış Gerçeklik Türleri ve Araçları

- **İşaretçi tabanlı (marker-based):** Bir görüntü/QR işaretçisi algılanınca nesne gösterilir.
- **İşaretçisiz (markerless):** Kamera ve sensörlerle düzlem/yüzey algılanarak nesne yerleştirilir.
- **Konum tabanlı:** GPS ve pusula verisi kullanılır (Pokémon GO gibi).
- **Unity'de AR:** AR Foundation paketi; Android için ARCore, iOS için ARKit üzerinden çalışır.

Ders Notu / Vurgu

VR'da kullanıcı gerçek dünyayı **görmez**, AR'da gerçek dünya **görünür** kalır. Unity'de ikisi de XR altyapısını (XR Plug-in Management, OpenXR) paylaşır; ancak VR için XR Interaction Toolkit, AR için AR Foundation paketi öne çıkar.

2. VR Başlıklarının Yapısı, Sistemi ve Yetenekleri

2.1 Bir VR Başlığının Temel Bileşenleri

Bileşen	Görevi
Ekran(lar)	Her göz için ayrı görüntü sağlayan OLED/LCD paneller. Çözünürlük ve yenileme hızı kaliteyi belirler.
Lensler	Yakındaki ekranı odaklanabilir hale getirir ve geniş görüş alanı sağlar (Fresnel / pancake lens).
Hareket sensörleri (IMU)	Jiroskop, ivmeölçer, manyetometre. Baş dönüşünü çok düşük gecikmeyle algılar.
Kameralar / takip sensörleri	Inside-out takipte ortamı ve kontrolcüleri izler; passthrough için gerçek ortam görüntüsü sağlar.
İşlemci ve bellek	Bağımsız başlıklarda (Quest, Pico) uygulamayı çalıştırır; PC başlıklarında görüntü PC'den gelir.
Kontrolcüler	Konum, rotasyon, buton (trigger, grip, A/B/X/Y), joystick, titreşim (haptic).
Ses sistemi	Dahili hoparlör/kulaklık, uzamsal (3D) ses.
Batarya ve bağlantı	Bağımsız başlıkta batarya; PC başlığında kablo veya kablosuz akış.

2.2 Başlık Türleri

Tür	Örnekler	Özellik
PC'ye bağlı	Valve Index, HTC Vive serisi	Yüksek grafik kalitesi, PC gücü gerekir.
Bağımsız (standalone)	Meta Quest serisi, Pico	Kablosuz, mobil işlemci; performans bütçesi dardır.
Konsol	PlayStation VR2	Konsola bağlı, yüksek kalite, eye tracking gibi özellikler.
Mobil / basit	Google Cardboard benzeri	Telefonu ekran olarak kullanır; sınırlı takip (genelde 3DoF).

2.3 Takip (Tracking) Sistemleri

- **Inside-out:** Kameralar başlığın üzerindedir; ek donanım gerekmez, kurulumu kolaydır (Quest).
- **Outside-in:** Odaya yerleştirilen istasyonlar başlığı ve kontrolcüleri izler; çok yüksek hassasiyet sağlayabilir (Vive/Index'in base station'ları).
- **Kontrolcü / el takibi:** Fiziksel kontrolcü veya kameraların elleri doğrudan izlemesi (hand tracking).

2.4 Başlık Yetenekleri

- Baş ve kontrolcü takibi (6DoF).
- El takibi ve parmak hareketlerinin algılanması.

- Passthrough: Gerçek ortamı kamerayla göstererek karma gerçeklik deneyimi.
- Eye tracking ve foveated rendering: Bakılan bölgeyi yüksek, çevreyi düşük çözünürlükte çizerek performans kazancı.
- Haptic geri bildirim: Kontrolcü titreşimiyle dokunma hissi.
- Uzamsal ses: Sesin yönü ve mesafesinin algılanması.
- Guardian / Boundary: Güvenli oyun alanı sınırı tanımlama.

2.4.1 Performans ve Konfor Hedefleri

Konu	Öneri
Kare hızı	Cihaza göre 72 / 90 / 120 FPS hedeflenir; kare düşmesi baş dönmesine yol açar.
Mobil VR (Quest)	Poligon sayısı, gölge, dinamik ışık ve post-processing sınırlı tutulmalıdır.
Hareket	Ani kamera hareketi ve ivmeli hareket yerine ışınlanma veya snap turn tercih edilebilir.
Ölçek	Dünya 1 birim = 1 metre ölçeğinde kurulmalıdır; yanlış ölçek gerçeklik hissini bozar.

2.5 OpenXR Standardı

OpenXR, Khronos Group tarafından geliştirilen açık bir standarttır. Farklı donanımlar (Quest, Vive, Index, Windows Mixed Reality, Pico vb.) için tek bir API sunar. Unity'de "OpenXR Plugin" paketi ile uygulama bir kez geliştirilir, farklı cihazlarda çalıştırılabilir. Cihaza özel girdi eşlemeleri **Interaction Profile** olarak tanımlanır (örneğin Oculus Touch Controller Profile).

3. Unity'de VR Projesi Oluşturma, XR Paketleri ve Temel Bileşenler

3.1 Proje Oluşturma

1. Unity Hub'da **New Project** seçilir; 3D (URP) şablonu veya Unity'nin VR şablonu (VR / VR Core) kullanılabilir.
2. Android/Quest hedefleniyorsa Unity Hub'da ilgili sürüme **Android Build Support** (OpenJDK, Android SDK & NDK Tools) modülleri yüklenmiş olmalıdır.
3. Proje açıldıktan sonra **File > Build Profiles / Build Settings** üzerinden hedef platform seçilir (Windows veya Android).

3.2 Gerekli XR Paketleri

Paket	Görevi
XR Plug-in Management	XR sağlayıcılarını (OpenXR vb.) etkinleştiren yönetim katmanı.
OpenXR Plugin	Cihazlarla konuşan standart XR çalışma zamanı katmanı.
XR Interaction Toolkit (XRI)	Hazır etkileşim sistemi: XR Origin, Interactor/Interactable, Locomotion, UI etkileşimi.
Input System	Kontrolcü girdilerini Input Action'lar üzerinden okur (eski Input Manager yerine).
XR Hands (isteğe bağlı)	El takibi verisini ve el eklemlerini sağlar.
Starter Assets (XRI örneği)	Hazır Input Action'lar, Preset'ler ve örnek prefab'lar. Package Manager > XR Interaction Toolkit > Samples bölümünden içe aktarılır.

Kurulum adımları

1. **Window > Package Manager** açılır, **Unity Registry** seçilir.
2. XR Interaction Toolkit, XR Plug-in Management ve (varsa) OpenXR paketleri yüklenir. Input System paketi de yüklenir; Unity "Active Input Handling" için yeniden başlatma isteyebilir.
3. XRI paketi seçili iken **Samples** sekmesinden **Starter Assets** içe aktarılır.
4. **Edit > Project Settings > XR Plug-in Management** bölümünden hedef sekmeye (PC veya Android) geçilir ve **OpenXR** işaretlenir.
5. **OpenXR** alt bölümünde **Interaction Profiles** altına kullanılacak kontrolcü profili eklenir (örneğin Oculus Touch Controller Profile). Android için Meta Quest desteği/özellikleri ilgili sekmede etkinleştirilir.
6. **Project Settings > Player > Active Input Handling** değerinin **Input System Package (New)** (veya Both) olduğu kontrol edilir.
7. XR ayarları uyarı verirse **Project Validation** penceresi üzerinden önerilen düzeltmeler uygulanır.

3.3 VR Sahnesinin Temel Hiyerarşisi

Bir VR sahnesinde klasik "Main Camera" yerine, kullanıcıyı temsil eden **XR Origin** yapısı kullanılır. Menüden **GameObject > XR > XR Origin (VR)** ile eklenir. Sahnede ayrıca **XR Interaction Manager** ve (UI için) **EventSystem** bulunur.

```
XR Interaction Manager
XR Origin (XR Rig)
  └─ Camera Offset
       ├─ Main Camera            (kafa: Tracked Pose Driver)
       ├─ Left Controller        (XR Controller + Interactor)
       └─ Right Controller       (XR Controller + Interactor)
Locomotion System / Providers    (Snap Turn, Continuous Move, Teleportation ...)
EventSystem (XR UI Input Module) (VR arayüzü için)
```

Bileşen	Görevi
XR Origin (XR Rig)	Kullanıcının sanal dünyadaki kök noktasıdır. Kamera ve kontrolcüler onun altındadır. Kullanıcıyı hareket ettirmek/döndürmek için **bu nesne** taşınır/döndürülür, kamera değil.
Camera Offset	Kamera yüksekliği ofsetidir. Tracking Origin Mode'a göre (Floor / Device) kullanıcının boyunu doğru yansıtır.
Main Camera + Tracked Pose Driver	Başlığın konum ve rotasyon verisini kameraya aktarır; kullanıcının "kafasıdır".
Left / Right Controller	Kontrolcünün konum/rotasyonunu ve buton girdisini (select, activate, UI press) taşır; üzerinde XR Controller bileşeni ve bir interactor bulunur.
XR Interaction Manager	Interactor'lar ile Interactable'lar arasındaki etkileşimi yöneten merkezî yöneticidir. Sahnede bulunmazsa etkileşim çalışmaz.
Interactor (XR Direct / Ray / Poke / Socket)	Etkileşimi **başlatan** taraf: elin yakın tutması (Direct), uzaktan ışınla seçme (Ray), parmakla dokunma (Poke) veya nesne yuvası (Socket).
Interactable (XR Simple / Grab Interactable)	Etkileşime **giren** nesne: seçilebilir, tutulabilir vb.
Interaction Layer Mask	Hangi interactor'ın hangi interactable ile etkileşeceğini katman bazlı filtreler.
Locomotion System	Hareket sağlayıcılarını (Snap/Continuous Turn, Move, Teleportation) bir arada ve çakışmadan yöneten sistem.
Teleportation Area / Anchor	Işınlanma yapılabilecek yüzey veya sabit nokta.
XR UI Input Module / Tracked Device Graphic Raycaster	World Space Canvas üzerindeki arayüzle kontrolcü ışınının etkileşmesini sağlar.

XR Rig – XR Origin farkı

Eski sürümlerde bu nesneye **XR Rig** denirdi; güncel XRI sürümlerinde **XR Origin (XR Rig)** adıyla bulunur. İşlevi aynıdır: kullanıcıyı sahnede konumlandıran kök nesnedir. **Tracking Origin Mode** ayarı **Floor** olduğunda kullanıcı gerçek boyuyla zemine göre konumlanır; **Device** (Eye Level) modunda başlığın ilk konumuna göre konumlanır ve Camera Y Offset ayarı önem kazanır.

3.4 VR Room Örnek Sahnesi

Unity'nin resmi VR örnek içeriği (VR Room / XRI Starter Assets örnek sahneleri) hazır bir oda, etkileşilebilir nesneler ve ayarlanmış XR Origin içerir. Öğrencilerin sıfırdan başlamadan önce çalışan bir örneği incelemeleri önerilir.

1. Örnek içerik (Samples / VR Room) projeye içe aktarılır.
2. Örnek sahne açılır ve **Hierarchy** incelenir: XR Origin, Interaction Manager, Locomotion System, Grab edilebilir nesneler, Teleportation Area.
3. Cihaz yoksa XR Device Simulator (Bölüm 4) sahneye eklenerek test edilir.
4. Sahnedeki bir nesne kopyalanıp kendi tasarladığımız modelle değiştirilir; Rigidbody, Collider ve XR Grab Interactable bileşenleri korunur.
5. Kendi VR odamız oluşturulur: Plane/Terrain ile zemin, duvarlar ve masa üzerine eşyalar yerleştirilir.

3.5 Sıfırdan Basit Bir VR Sahnesi (Uygulama)

1. Yeni sahne açılır; varsayılan **Main Camera** silinir.
2. **GameObject > XR > XR Origin (VR)** eklenir; **XR Interaction Manager** otomatik gelir.
3. **3D Object > Plane** ile zemin oluşturulur, ölçeği ve Y konumu ayarlanır (zemin Y = 0).
4. XR Origin'in Y konumu zeminle uyumlu olacak şekilde 0 yapılır; Tracking Origin Mode Floor olarak ayarlanır.
5. Birkaç Cube/Sphere eklenir (sonraki bölümde tutulabilir yapılacak).
6. Cihaz bağlıysa Play'e basılır; değilse Bölüm 4'teki simülatör kullanılır.

> **Sık Hata**
> Sahnede hem eski Main Camera hem de XR Origin içindeki kamera varsa iki kamera çakışır. Sahnede **yalnızca bir** Main Camera bulunmalıdır (XR Origin içindeki).

4. XR Device Simulator Kullanımı

4.1 Amaç

VR başlığı olmadan, bilgisayarda klavye ve fare ile kafa ve kontrolcü hareketlerini taklit ederek VR sahnesini test etmeyi sağlar. Laboratuvarda başlık sayısı sınırlı olduğunda hızlı geliştirme ve hata ayıklama için çok kullanışlıdır.

4.2 Kurulum

1. **Window > Package Manager > XR Interaction Toolkit > Samples** sekmesinden **XR Device Simulator** örneği (bazı sürümlerde **XR Interaction Simulator**) içe aktarılır.
2. Assets içinde örnek klasörü altındaki **XR Device Simulator** prefab'ı bulunur ve sahneye sürüklenir.
3. Input System tarafında simülatörün kendi Input Action'larının (XR Device Simulator Controls) etkin olduğundan emin olunur.
4. Sahnede gerçek XR cihazı yoksa simülatör, cihaz yerine veri sağlar. **Play** moduna geçilir.

4.3 Kullanım Mantığı

İşlem	Açıklama
Kafa (kamera) hareketi	Fare ile bakış yönü, klavye (W/A/S/D) ile ilerleme/yan hareket.
Sol / sağ kontrolcüyü seçme	Belirli bir tuş basılı tutularak (varsayılan olarak sol ve sağ el için ayrı tuşlar) o kontrolcü fareyle kontrol edilir.
Buton girdileri	Fare tuşları ve klavye tuşlarıyla trigger, grip, primary/secondary buton taklit edilir.
Eğitim ipucu	Tuş atamaları paket sürümüne göre değişebilir. Güncel atamalar simülatörün Input Action Asset'inde ve oyun ekranındaki yardım arayüzünde görülebilir.

Uygulamada İpucu
Tutma (grab) denemesi için önce sağ kontrolcüyü nesnenin yanına getirin, sonra **grip/select** karşılığı olan tuşa basılı tutun. Nesne ele yapışıyorsa Interactor-Interactable eşleşmesi doğrudur.

4.4 Sınırlılıklar

- Gerçek takip kalitesi, ölçek hissi, derinlik ve konfor (baş dönmesi) **simülatörde test edilemez**.
- Performans (FPS) ölçümleri gerçek başlıkta yapılmalıdır.
- Android build'i alıp Quest üzerinde son kontrol yapılması gerekir.

5. Obje Tutma: XR Grab Interactable ve Attach Transform

5.1 Mantık

XRI'da etkileşim **Interactor** (elin tarafı) ve **Interactable** (nesnenin tarafı) arasında olur. Bir nesnenin tutulabilmesi için nesne üzerinde **XR Grab Interactable** bileşeni, kontrolcü tarafında ise bir **interactor** (XR Direct Interactor veya XR Ray Interactor) bulunmalıdır. İkisini **XR Interaction Manager** eşleştirir.

5.2 Adım Adım: Küpü Tutulabilir Yapma

1. Sahneye bir **Cube** eklenir; boyutu gerçekçi seçilir (örneğin 0.2 × 0.2 × 0.2 m).
2. Cube'a **Rigidbody** eklenir (fizik için). Collider zaten vardır.
3. Cube'a **XR Grab Interactable** bileşeni eklenir.
4. Sol/Sağ kontrolcüde **XR Direct Interactor** bulunduğundan emin olunur. Direct Interactor, nesneye değmek için bir **Sphere Collider (Is Trigger)** kullanır; yoksa eklenir.
5. Kontrolcünün **Select** (grip veya trigger) Input Action'ının atanmış olduğu doğrulanır.
6. Play'e basılır; kontrolcü küpe yaklaştırılıp grip tuşuna basılınca küp ele yapışmalıdır.

5.3 XR Grab Interactable Ana Ayarları

Ayar	Açıklama
Interaction Manager	Boş bırakılırsa sahnedeki Interaction Manager otomatik bulunur.
Interaction Layer Mask	Hangi interactor'ların bu nesneyi seçebileceğini belirler.
Colliders	Etkileşim algılama için kullanılan colliderlar; boş ise nesne ve çocuklarındakiler kullanılır.
Distance Calculation Mode	Interactor'a en yakın noktanın nasıl hesaplanacağı (özellikle çok parçalı nesnelerde önemlidir).
Movement Type	**Instant:** Hemen konuma ışınlanır, fizik gecikmesi yok. **Kinematic:** Rigidbody kinematik hareket eder, hafif gecikmeli ve yumuşak. **Velocity Tracking:** Fizik hızlarıyla takip eder; çarpışmalarla doğal etkileşir, nesne duvardan geçmez.
Attach Transform	Nesnenin elde **nereden ve hangi yönde** tutulacağını belirleyen boş çocuk nesne (aşağıda).
Attach Ease In Time	Nesnenin ele yumuşakça gelme süresi; 0 olursa anında yapışır.
Track Position / Track Rotation	Nesnenin elin konum/rotasyonunu takip edip etmeyeceği.
Smooth Position / Smooth Rotation	Takip sırasında yumuşatma uygular.
Throw On Detach	Bırakınca nesneye elin hızı verilir; fırlatma mümkün olur.
Force Gravity On Detach	Bırakıldığında yerçekimi zorla açılır.
Select Mode	Single: bir el tutar; Multiple: iki el aynı anda tutabilir.

Ayar	Açıklama
Use Dynamic Attach	Nesne, tutulduğu noktadan ele bağlanır; Attach Transform yerine dinamik nokta kullanılır.

5.4 Attach Transform

Attach Transform, tutulan nesnenin elle ilişkisini belirleyen referans noktasıdır. Nesnenin altına boş bir GameObject eklenir (örneğin `AttachPoint`), bu nesne konumlandırılıp döndürülür ve XR Grab Interactable bileşenindeki **Attach Transform** alanına sürüklenir.

- **Konum:** Nesnenin hangi noktasından tutulacağı (kılıcın sapı, tabancanın kabzası, kupanın kulpu).
- **Rotasyon:** Nesnenin elde hangi açıyla duracağı (namlu ileri bakacak şekilde).
- **Önemli:** Attach Transform noktasının kontrolcünün tutma noktasıyla üst üste oturduğu varsayılır; bu yüzden nesne tutulduğunda attach noktası ele hizalanır.
- **İki el:** Çift elle tutulan nesnelerde (tüfek, çift kollu kılıç) ikinci bir (secondary) attach noktası tanımlanabilir. Sürüme göre bileşen adı ve davranış farklı olabilir.

```
Sword (Rigidbody, XR Grab Interactable)
  ├─ Model
  └─ AttachPoint      ← Attach Transform alanına atanır
```

5.5 Olaylar (Events)

Olay	Ne zaman tetiklenir?
Hover Entered / Exited	Interactor nesnenin üzerine geldiğinde / ayrıldığında (vurgulama için kullanılır).
Select Entered / Exited	Nesne tutulduğunda / bırakıldığında.
Activated / Deactivated	Nesne tutulurken **Activate** girdisine (genelde trigger) basıldığında / bırakıldığında (silah ateşleme, el feneri açma).

Bu olaylar Inspector'dan **UnityEvent** olarak bağlanabilir. Aşağıda "tutarken tetiğe basınca ateş etme" için script yazmadan önce kullanılabilecek basit bir yapı verilmiştir: script içindeki herkese açık metot, **Activated** olayına Inspector'dan atanır.

```
using UnityEngine;

public class SimpleGun : MonoBehaviour
{
    public GameObject bulletPrefab;
    public Transform muzzle;        // Namlu ucu (boş obje)
    public float bulletSpeed = 15f;

    // XR Grab Interactable > Interactable Events > Activated olayına bağlanır
    public void Fire()
    {
        GameObject b = Instantiate(bulletPrefab, muzzle.position, muzzle.rotation);
        b.GetComponent<Rigidbody>().linearVelocity = muzzle.forward * bulletSpeed;
        Destroy(b, 5f);
    }
```

```
}
```

> **Not**
> Unity 6 öncesi sürümlerde `Rigidbody.linearVelocity` yerine `Rigidbody.velocity` kullanılır.

5.6 Sık Yapılan Hatalar

- Rigidbody yoksa Grab Interactable çalışmaz.
- Interactor'un Interaction Layer Mask değeri ile nesnenin değeri uyuşmuyorsa nesne seçilemez.
- Direct Interactor'daki collider "Is Trigger" değilse veya çok küçükse nesneye değmez.
- Select Input Action etkin değilse (Input Action Manager yok) buton basımı algılanmaz.
- Movement Type **Instant** iken duvar gibi nesnelerden geçme olabilir; fiziksel doğallık gerekiyorsa **Velocity Tracking** düşünülmelidir.

6. Snap Turn, Continuous Turn ve Locomotion System

6.1 Locomotion Nedir?

VR'da **locomotion**, kullanıcının sanal dünyada yer değiştirmesi ve dönmesidir. Fiziksel alan sınırlı olduğundan yönlendirme ve hareket için kontrolcü girdilerinden yararlanılır. Konfor açısından dikkatli seçilmelidir; çünkü yanlış hareket tasarımı baş dönmesine neden olur.

6.2 Locomotion System

Locomotion System bileşeni, hangi hareket sağlayıcının (provider) ne zaman XR Origin'i hareket ettirebileceğini yönetir. Böylece aynı anda birden fazla sağlayıcının çakışması engellenir. Sahnede tek bir Locomotion System bulunur ve **XR Origin** referansı verilir.

Provider	Görevi
Snap Turn Provider	Kullanıcıyı belirli açı kadar (örneğin 45°) ani döndürür.
Continuous Turn Provider	Kullanıcıyı sürekli, belirli hızla döndürür.
Continuous Move Provider	Joystick ile sürekli ileri-geri/yan hareket (smooth locomotion).
Teleportation Provider	Teleportation Area/Anchor üzerinde ışınlanmayı gerçekleştirir.
Grab Move Provider (isteğe bağlı)	Çevreyi tutup çekerek hareket etme (tırmanma/ittirme benzeri).

6.3 Snap Turn

Joystick'e bir kez itildiğinde kullanıcıyı sabit açıyla **anında** döndürür. Kamera sürekli kaymadığı için baş dönmesine en az sebep olan döndürme yöntemidir.

Ayar	Açıklama
Turn Amount	Her dönüşte uygulanan açı (genelde 30°–45°).
Debounce Time	İki dönüş arasında beklenmesi gereken süre; art arda istem dışı dönmeyi engeller.
Enable Turn Around	Joystick geri çekilince 180° dönmeyi açar.
Left/Right Hand Snap Turn Input	Dönüş için kullanılacak Input Action'lar (genelde sağ joystick X ekseni).

6.4 Continuous Turn

Joystick itildiği sürece kullanıcı **sürekli ve yumuşak** biçimde döner. Daha doğal bir his verir; fakat duyarlı kullanıcılarda baş dönmesine neden olabilir.

Ayar	Açıklama
Turn Speed	Derece/saniye olarak dönüş hızı (başlangıç için 60–90 uygundur).

Ayar	Açıklama
Left/Right Hand Turn Input	Dönüş için joystick Input Action'ı.

6.5 Snap Turn – Continuous Turn Karşılaştırması

Kriter	Snap Turn	Continuous Turn
Dönüş şekli	Ani, sabit açılı	Yumuşak, sürekli
Konfor	Yüksek (VR hastalığı riski düşük)	Daha düşük (baş dönmesi riski var)
Doğallık	Kesik hissi verir	Daha doğal
Kime uygun?	Yeni başlayanlar, uzun oturumlar	Deneyimli kullanıcılar
Ayarlanabilir	Turn Amount, Debounce	Turn Speed

Tasarım Önerisi

Oyunda ayarlar menüsüne **Turn Type** seçeneği (Snap / Continuous) koyun. Kullanıcıya seçim hakkı vermek konforu artırır. Aynı anda ikisi etkin olmamalıdır; biri açıkken diğer Provider devre dışı bırakılmalıdır.

6.6 Move ve Teleportation

- **Continuous Move:** Sol joystick ile hareket. Ayarlar: Move Speed (yürüme hızı genelde 1–2 m/s), Enable Strafe, Use Gravity, Forward Source (kafa yönü veya el yönü).
- **Teleportation:** Zemine **Teleportation Area**, belirli noktalara **Teleportation Anchor** eklenir. Kontrolcüde ışınlanma için bir Ray Interactor (Teleport Ray) ve Teleportation Provider gerekir. Işınlanma en konforlu hareket türüdür.

6.7 Uygulama: Dönüş ve Hareket Kurulumu

1. Starter Assets içindeki **XRI Default Input Actions** asset'inin sahnede etkin olduğundan emin olunur.
2. XR Origin altındaki (veya ayrı) **Locomotion System** nesnesine XR Origin referansı verilir.
3. **Snap Turn Provider** ve **Continuous Turn Provider** bileşenleri eklenir; her ikisine sağ kontrolcünün joystick (Turn) Input Action'ı atanır.
4. Önce yalnızca Snap Turn'ü etkin bırakıp test edin (Turn Amount 45°).
5. Snap Turn'ü kapatıp Continuous Turn'ü etkinleştirin ve Turn Speed değerini 60–90 aralığında deneyin.
6. **Continuous Move Provider** ekleyip sol joystick'i atayın; Move Speed ve Use Gravity ayarlarını deneyin.
7. İsteğe bağlı: Basit bir ayar paneli veya script ile Turn Type'ı çalışma zamanında değiştirin.

```
// Turn tipini çalışma zamanında değiştirmek için kavramsal örnek
// (provider sınıfı adı ve ad alanı XRI sürümüne göre değişebilir)
public GameObject snapTurn;
public GameObject continuousTurn;
```

```
public void SetTurnType(bool useSnap)
{
    snapTurn.SetActive(useSnap);
    continuousTurn.SetActive(!useSnap);
}
```

Not

Provider'ların ayrı GameObject'ler üzerinde tutulması, bu şekilde SetActive ile kolayca açıp kapatmayı mümkün kılar.

7. Ek Bölüm: Sorun Giderme, Laboratuvar Uygulaması ve Sorular

7.1 Sık Karşılaşılan Sorunlar

Belirti	Olası neden ve çözüm
Kafa hareket etmiyor / sahne donuk	XR Plug-in Management'ta OpenXR etkin değil; Tracked Pose Driver Input Action'ı atanmamış; Input Action Manager veya Input System kurulmamış.
Kontrolcüler görünmüyor	Interaction Profile eklenmemiş; kontrolcü model prefab'ı atanmamış; XR Controller girdileri eksik.
Nesne tutulmuyor	Rigidbody/XR Grab Interactable yok; Layer Mask uyuşmuyor; Interactor collider'ı yok; Interaction Manager eksik.
Nesne ele yapışıp yamuk duruyor	Attach Transform yanlış konum/rotasyonda; nesne ölçeği hatalı.
Kullanıcı yerin altında / havada	Tracking Origin Mode ve XR Origin Y konumu uyumsuz; zemin yüksekliği hatalı.
Dönüş çalışmıyor	Provider'lara Input Action atanmamış; Locomotion System'de XR Origin referansı yok; iki provider çakışıyor.
Quest'te build açılmıyor	Android modülleri/OpenXR Android ayarları eksik; Meta Quest özellik grubu etkin değil.
Düşük FPS	Gölge, post-processing, yüksek poligon; Quest için URP mobil ayarlarına geçin, batching ve baked light kullanın.

7.2 Laboratuvar Uygulaması (Mini Proje)

Aşağıdaki görevleri sırayla yapan bir VR odası geliştiriniz:

1. Yeni bir Unity projesi oluşturun; XR Plug-in Management, OpenXR, XR Interaction Toolkit ve Input System kurun.
2. Bir oda kurun (zemin, 4 duvar, masa) ve XR Origin ekleyin.
3. Masanın üzerine en az 3 tutulabilir nesne koyun: bir küp, bir kupa/silindir ve bir "alet" (Attach Transform ile doğru tutulan).
4. En az bir nesnede Movement Type olarak Velocity Tracking, bir diğerinde Instant kullanın ve farkı gözlemleyin.
5. Nesneyi tutarken Activate ile ışık açıp kapatın (Activated olayı).
6. Snap Turn ve Continuous Turn'ü kurun; bir UI düğmesi veya klavye tuşuyla aralarında geçiş yapın.
7. Tüm testleri XR Device Simulator ile yapın; mümkünse gerçek başlıkta doğrulayın.

Değerlendirme ölçütü	Puan	Açıklama
Doğru XR Origin ve paket kurulumu	20	Cihazda/simülatörde çalışması
Tutulabilir nesneler ve Attach Transform	30	Doğru tutuş noktası ve rotasyonu
Dönüş sistemi (Snap / Continuous)	20	Provider ve Input Action doğru atanmış

Değerlendirme ölçütü	Puan	Açıklama
Sahne tasarımı ve ölçek	15	Gerçekçi ölçek, düzenli hiyerarşi
Rapor / kısa sunum	15	Karşılaşılan sorunlar ve çözümleri

7.3 Değerlendirme Soruları

1. VR, AR ve MR arasındaki farkları örneklerle açıklayınız.
2. 3DoF ile 6DoF arasındaki fark nedir? Hangi kullanım senaryosunda hangisi yeterlidir?
3. Inside-out ve outside-in takip sistemlerini avantaj ve dezavantajlarıyla karşılaştırınız.
4. OpenXR'ın amacı nedir? Interaction Profile ne işe yarar?
5. XR Origin nedir? Neden kullanıcıyı hareket ettirirken kamerayı değil XR Origin'i hareket ettiririz?
6. XR Interaction Manager'ın görevi nedir? Sahnede bulunmazsa ne olur?
7. Interactor ve Interactable kavramlarını örneklerle açıklayınız.
8. XR Device Simulator'ın faydaları ve sınırlılıkları nelerdir?
9. Bir nesnenin tutulabilmesi için gereken minimum bileşenleri sıralayınız.
10. Movement Type seçeneklerini (Instant, Kinematic, Velocity Tracking) karşılaştırınız.
11. Attach Transform ne işe yarar? Bir kılıç için nasıl ayarlarsınız?
12. Snap Turn ve Continuous Turn arasındaki farkı konfor açısından yorumlayınız.
13. Locomotion System'in görevi nedir? İki provider'ın aynı anda aktif olması hangi soruna yol açar?

7.4 Kısa Özet

- VR kullanıcıyı sanal dünyaya yerleştirir; AR gerçeğe sanal öğe ekler; XR ikisini kapsar.
- Unity'de VR geliştirmek için **XR Plug-in Management + OpenXR + XR Interaction Toolkit + Input System** kullanılır.
- Sahnenin kalbi **XR Origin** ve **XR Interaction Manager**'dır.
- **XR Device Simulator** başlık olmadan hızlı test sağlar; gerçek konfor testi başlıkta yapılmalıdır.
- Obje tutmak için **Rigidbody + XR Grab Interactable + Interactor**; tutuş yeri **Attach Transform** ile ayarlanır.
- Dönüş için **Snap Turn** konforludur, **Continuous Turn** doğaldır; hepsini **Locomotion System** yönetir.