**Yazılım Tasarım Dokümanı (SDD)**

**Proje Adı:** Dedektif Pati ile Tarihi Keşfet (AR Eğitici Oyun)

**1. Giriş**

**1.1. Amaç:** Bu doküman, SRS dokümanında tanımlanan gereksinimleri karşılayacak olan "Dedektif Pati ile Tarihi Keşfet" AR oyununun yazılım mimarisini, modül tasarımlarını, veri yapılarını ve arayüzlerini detaylandırmaktadır. Temel konsept, kullanıcının seçtiği tarihi mekanın ve çevresinin 3D modelini Artırılmış Gerçeklik (AR) ile bir yüzeye (masa/zemin) yerleştirmesi ve ardından **ekrandaki sanal kontrolleri kullanarak bu modelin etrafında sanal olarak gezinerek** keşif yapmasıdır. Tasarım, **RAMS (Güvenilirlik, Kullanılabilirlik, Sürdürülebilirlik, Emniyet)** ilkelerini temel alarak sağlam, kullanıcı dostu ve güvenli bir ürün ortaya koymayı; temel tasarım hedefleri ise **SMART (Özgün, Ölçülebilir, Ulaşılabilir, İlgili, Zamanla Sınırlandırılmış)** prensiplerine göre tanımlanmıştır.

**1.2. Kapsam:** Bu tasarım, oyunun ana menüsünden başlayarak, AR mekan seçimi, modelin yüzeye yerleştirilmesi, **sanal navigasyon kontrolleri ile model etrafında keşif**, hedef parça bulma mekaniği, bilgi sunumu ve kullanıcı ilerlemesi gibi SRS'te belirtilen tüm fonksiyonel ve fonksiyonel olmayan gereksinimlerin teknik gerçekleştirimini kapsar.

**1.3. Referanslar:**

* + Yazılım Gereksinim Spesifikasyonu (SRS) - "Dedektif Pati ile Tarihi Keşfet" (İlgili en güncel versiyon)
  + ARKit Documentation (Apple)
  + ARCore Documentation (Google)
  + Unity Engine Documentation (veya seçilen diğer motor)
  + Çocuk Gizlilik Yönetmelikleri (COPPA, GDPR-K vb.)

**2. Sistem Mimarisi (RAMS Odaklı)**

**2.1. Mimari Genel Bakış:** Uygulama, mobil platformlar (iOS, Android) üzerinde çalışacak, Unity oyun motoru kullanılarak geliştirilmiş, modüler bir yapıya sahip olacaktır. Mimari, **Sürdürülebilirliği (Maintainability - M)** artırmak için katmanlı veya modül tabanlı bir yaklaşım (örn. MVVM, MVC veya olay güdümlü mimari) benimseyecektir.

* + **Katmanlar/Modüller:**
    - **Sunum Katmanı (UI/UX):** Kullanıcı arayüzü elemanları (menüler, düğmeler, bilgi kartları), **sanal navigasyon kontrollerinin (örn. joystick, oklar) görselleştirilmesi**, kullanıcı girdisinin alınması (dokunma, sürükleme), Dedektif Pati karakter animasyonları. (**Kullanılabilirlik - A** odaklı).
    - **Oyun Mantığı Katmanı:** Oyun durumu yönetimi (menüde, AR keşifte vb.), **alınan navigasyon girdisine göre sanal kamera hareketlerinin hesaplanması**, görev takibi, hedef parça bulunma kontrolü, puanlama/ödül sistemi. (**Güvenilirlik - R** odaklı).
    - **AR Yönetim Katmanı:** ARKit/ARCore ile etkileşim, yatay **yüzey algılama**, 3D **modelin yüzeye yerleştirilmesi ve sabitlenmesi (Anchoring)**, ortam ışığı takibi, sanal kameranın AR sahnesindeki konum/rotasyonunun güncellenmesi. (**Güvenilirlik - R** [Stabil çapa] ve **Emniyet - S** [Modelin kaymaması] odaklı).
    - **Veri Yönetim Katmanı:** Kullanıcı ilerlemesi (lokal depolama), oyun içeriği (mekan bilgileri, modeller, metinler, sesler) yüklenmesi ve yönetimi. (**Güvenilirlik - R** ve **Sürdürülebilirlik - M** [İçerik güncelleme] odaklı).
    - **Varlık (Asset) Yönetim Katmanı:** 3D modellerin, dokuların, seslerin, animasyonların verimli bir şekilde yüklenmesi, boşaltılması ve yönetimi. (**Kullanılabilirlik - A** [Hızlı yükleme, akıcı performans] odaklı).

**2.2. Teknoloji Seçimi:**

* + **Oyun Motoru:** Unity - Geniş platform desteği, güçlü AR entegrasyonları (AR Foundation), zengin varlık mağazası ve geniş geliştirici topluluğu. (**Sürdürülebilirlik - M, Güvenilirlik - R**).
  + **AR Kütüphaneleri:** Unity AR Foundation (ARKit ve ARCore'u soyutlar). (**Güvenilirlik - R, Sürdürülebilirlik - M** [Tek kod tabanı]).
  + **Programlama Dili:** C# (Unity için standart).
  + **Lokal Depolama:** PlayerPrefs (basit veriler için) veya SQLite (daha yapısal veriler için). (**Güvenilirlik - R, Kullanılabilirlik - A** [Offline çalışma], **Emniyet - S** [Veri cihazda kalır]).

**3. Detaylı Modül Tasarımı (SMART Hedeflerle)**

**3.1. AR Model Yerleştirme & Sabitleme Modülü (SRS FR-X uyumlu)**

* + **Tasarım:** Cihaz kamerası aracılığıyla yatay düzlemleri (masa, zemin) AR Foundation kullanarak algılar. Kullanıcıya görsel geri bildirimle (örn. hedef göstergesi, algılanan yüzeyin vurgulanması) uygun bir yüzey gösterir. Kullanıcının ekrana dokunmasıyla veya otomatik algılama sonrası, seçilen Mekan+Çevre 3D modelini uygun bir masaüstü ölçeğinde (gerekirse kullanıcı tarafından ayarlanabilir) hedef yüzeye bir AR Anchor kullanarak yerleştirir. Bu çapa, modelin yüzeyde stabil kalmasını sağlar.
  + **SMART Hedefler:**
    - **(S) Özgün:** Yatay düzlem algıla, görsel ipucu sun, kullanıcı girdisiyle modeli yerleştir, AR Anchor ile sabitle.
    - **(M) Ölçülebilir:** Yeterli ışık ve dokudaki yüzeylerde 5 sn içinde düzlem algılama başarı oranı >%95. Model yerleştirildikten sonra 1 dakika boyunca anchor kayma oranı <%5 (ideal ışıkta).
    - **(A) Ulaşılabilir:** AR Foundation'ın standart düzlem algılama ve çapa (anchor) yetenekleri ile gerçekleştirilebilir.
    - **(R) İlgili:** Oyunun AR temelini oluşturur, stabilite kullanıcı deneyimi için kritiktir.
    - **(T) Zamanlı:** Prototipi Sprint 1'de, stabil versiyonu Sprint 2'de tamamlanacak.

**3.2. Sanal Navigasyon Modülü (SRS FR-Y uyumlu)**

* + **Tasarım:** Ekranda (örn. sol alt köşe) sabit bir sanal joystick veya alternatif olarak yön okları bulunur. Kullanıcının bu kontrollerle etkileşimi (joystick'i hareket ettirme, oklara basma) alınır. Bu girdi, önceden tanımlanmış bir hız ve dönüş oranı ile AR sahnesindeki sanal kameranın transform (pozisyon ve rotasyon) bileşenini günceller. Hareket, yerleştirilen 3D modelin çevresinde gerçekleşir. İsteğe bağlı olarak, kamera modelin içine girdiğinde basit bir çarpışma algılama uygulanabilir (örn. kamera geri itilir).
  + **SMART Hedefler:**
    - **(S) Özgün:** Sanal joystick/düğme girdisini al, sanal kamerayı AR modelinin etrafında akıcı bir şekilde hareket ettir ve döndür.
    - **(M) Ölçülebilir:** Kullanıcı girdisi ile kamera hareketi arasındaki gecikme < 50ms. Hedef kitle (6-9 yaş) ile yapılan testlerde, kullanıcıların %85'i 3 denemede belirlenen bir hedefe navigasyon kontrolünü kullanarak ulaşabilmeli.
    - **(A) Ulaşılabilir:** Unity'nin UI ve Input sistemleri, basit kamera kontrol scriptleri ile gerçekleştirilebilir. Kullanılabilirlik testleri gerektirir.
    - **(R) İlgili:** Oyunun temel keşif ve etkileşim mekanizmasıdır.
    - **(T) Zamanlı:** Kontrol prototipi Sprint 2'de, kullanılabilirlik iyileştirmeleri Sprint 3'te tamamlanacak.

**3.3. Hedef Parça Etkileşim & Bilgi Sunum Modülü (SRS FR-5 & FR-6 uyumlu)**

* + **Tasarım:** Model üzerindeki hedef parçalar, etkileşim için bir Collider bileşenine sahip olur ve görsel olarak (örn. hafif parlama shader'ı) ayırt edilebilir. Kullanıcı ekrana dokunduğunda, **sanal kameranın konumundan** dokunulan ekran noktasına doğru bir ışın (Raycast) gönderilir. Bu ışın hedef parçanın collider'ına isabet ederse, başarı durumu tetiklenir. Başarı durumunda görsel/işitsel efektler oynatılır ve o parçayla ilişkili Bilgi Kartı UI paneli (metin, resim, sesli okuma butonu içeren) gösterilir.
  + **SMART Hedefler:**
    - **(S) Özgün:** Ekrana dokunmayı algıla, sanal kameradan Raycast gönder, hedef parça ile çarpışmayı kontrol et, başarı efektlerini/bilgi kartını göster.
    - **(M) Ölçülebilir:** Hedef parçaya isabetli dokunma algılama oranı >%98. Bilgi kartının görüntülenmesi < 0.5 sn. Gösterilen bilginin içeriğinin SRS ile %100 uyumu.
    - **(A) Ulaşılabilir:** Standart Unity Raycasting, UI ve ses sistemleri ile gerçekleştirilebilir. İçeriklerin (bilgi, resim, ses) eksiksiz olması gerekir.
    - **(R) İlgili:** Oyunun temel hedefi olan nesne bulma ve bilgi verme mekanizmasını gerçekleştirir.
    - **(T) Zamanlı:** Mekanizma geliştirmesi Sprint 3, içerik entegrasyonu Sprint 4'te tamamlanacak.

**3.4. Dedektif Pati Rehberlik Modülü (SRS FR-4 uyumlu)**

* + **Tasarım:** Oyunun mevcut durumuna (başlangıç, görev verme, ipucu zamanı, parça bulundu) göre uygun diyalogları, seslendirmeyi ve animasyonları (muhtemelen ekranda sabit duran 2D sprite animasyonu) yöneten basit bir state machine veya event tabanlı sistem. İpuçları, oyuncu belirli bir süre hedefi bulamazsa veya zorluk seviyesine göre tetiklenebilir. İpuçları sanal navigasyona yönelik de olabilir ("Yaklaşmak için yukarı oku kullan!").
  + **SMART Hedefler:**
    - **(S) Özgün:** Oyun durumunu takip et, doğru diyalog/animasyon/seslendirmeyi oynat, zamanlı veya duruma bağlı ipuçları ver.
    - **(M) Ölçülebilir:** Tüm kritik oyun durumlarında ilgili rehberlik mesajının gösterilme oranı %100.
    - **(A) Ulaşılabilir:** Unity Animator, UI ve ses sistemleri ile yapılabilir. Karakter çizimleri, animasyonları ve seslendirme dosyaları gereklidir.
    - **(R) İlgili:** Kullanıcı deneyimini iyileştirir, oyunu çocuklar için daha ilgi çekici ve anlaşılır kılar.
    - **(T) Zamanlı:** Karakter assetleri Sprint 2, durum makinesi ve entegrasyon Sprint 4'te tamamlanacak.

**3.5. Ana Menü Yönetim Modülü (SRS FR-1 uyumlu)**

**Tasarım:** Oyunun giriş noktası olarak ayrı bir Unity sahnesi (MainMenuScene) veya ana sahnede özel bir Canvas grubu kullanılır. UI Elemanları (Butonlar, Resimler, Metinler - Tercihen TextMeshPro) ile oluşturulur. Tarihi mekanların listesi, bir ScrollView içinde dinamik olarak (prefab'lardan veya veri kaynağından oluşturulan) butonlar/ikonlar kullanılarak gösterilir. Mekan seçimi yapıldığında, seçilen mekan ID'si kaydedilir ve AR sahnesine geçiş tetiklenir (SceneManager.LoadSceneAsync). Ayrıca "Ayarlar", ("Rozetlerim" gibi isteğe bağlı bir ilerleme ekranı) ve "Çıkış" (platforma özgü - örn. Android'de) butonları bulunur ve ilgili sahnelere/panellere yönlendirme yapar. Arka plan müziği ve belki Dedektif Pati'nin basit bir hoşgeldin animasyonu eklenebilir.

**SMART Hedefler:**

**(S) Özgün:** Kullanılabilir mekan listesini görsel olarak sun, mekan seçimine izin ver, Ayarlar ve diğer ekranlara navigasyonu sağla.

**(M) Ölçülebilir:** Menü yükleme süresi < 1.5 saniye (ilk açılış sonrası). Tüm butonların dokunma tepki süresi < 100ms. Sahne/Panel geçişleri %100 başarılı olmalı.

**(A) Ulaşılabilir:** Standart Unity UI bileşenleri (Button, ScrollView, Image), Scene Management ve basit animasyon sistemleri ile gerçekleştirilebilir.

**(R) İlgili:** Oyunun başlangıcı ve temel navigasyonu için zorunludur.

**(T) Zamanlı:** Tasarım ve temel uygulama Sprint 1'de tamamlanacak.

**RAMS Etkisi**: Net, anlaşılır düzen (Kullanılabilirlik - A). Hızlı yükleme (Kullanılabilirlik - A). Kolayca yeni mekan ekleyebilme (Sürdürülebilirlik - M). Hata vermeyen navigasyon (Güvenilirlik - R).

**3.6. Kullanıcı İlerleme Kaydı (Lokal) Modülü (SRS FR-7.1 uyumlu)**

**Tasarım:** Kullanıcının tamamladığı görevleri ve kazandığı ödülleri (rozetler vb.) cihazda kalıcı olarak saklamak için Unity'nin PlayerPrefs sistemi veya (ileride veri karmaşıklaşırsa) JSON serileştirme ile basit bir dosya kullanılır. Kesinlikle kişisel veri saklanmaz. Saklanacak anahtarlar önceden tanımlanır (örn. const string COMPLETED\_LANDMARKS\_KEY = "CompletedLandmarks";). Tamamlanan mekanların veya kazanılan rozetlerin ID'leri bu anahtar altında bir string olarak (virgülle ayrılmış: "1,3,5") veya basit bir formatta (örn. bitmask) saklanır. ProgressManager adında statik veya Singleton bir sınıf oluşturulur. Bu sınıf, SaveProgress(), LoadProgress() (oyun başında çağrılır), IsLandmarkCompleted(int landmarkId), MarkLandmarkAsCompleted(int landmarkId), GetEarnedBadges() gibi metodlar içerir. İlerleme, bir mekan başarıyla tamamlandığında veya yeni bir rozet kazanıldığında SaveProgress() çağrılarak kaydedilir.

**SMART Hedefler:**

**(S) Özgün:** Tamamlanan mekan/rozet ID'lerini lokal ve kalıcı olarak kaydet/yükle. Belirli bir mekanın tamamlanma durumunu sorgula.

**(M) Ölçülebilir:** Kaydetme/Yükleme işlemi süresi < 50ms. Oturumlar arası veri kalıcılık oranı >%99.9 (normal kullanımda veri kaybı olmamalı). Tamamlanma durumu sorgulama doğruluğu %100.

**(A) Ulaşılabilir:** PlayerPrefs veya temel dosya G/Ç işlemleri kullanılarak kolayca gerçekleştirilebilir.

**(R) İlgili:** Oyuncuya ilerleme hissi verir, motivasyonu artırır, potansiyel olarak içerik kilitleme için kullanılabilir.

**(T) Zamanlı:** Temel mekanizmanın tasarımı Sprint 1, uygulama ve entegrasyonu ise görev tamamlama mantığı ile birlikte (örn. Sprint 4) yapılacak.

**RAMS Etkisi:** Veri Gizliliği (Emniyet - S) kritik: Asla PII saklanmamalı. Basitlik ve lokal depolama veri kaybı riskini azaltır (Güvenilirlik - R). Hızlı yükleme/kaydetme (Kullanılabilirlik - A). Kolay anlaşılır veri yapısı (Sürdürülebilirlik - M).

**3.7. Ayarlar Modülü (SRS FR-8 uyumlu)**

**Tasarım:** Ana Menü'den erişilebilen ayrı bir UI Paneli veya Sahne. Ses Efektleri (SFX), Arka Plan Müziği ve (varsa) Seslendirme için açma/kapama anahtarları (Unity Toggle bileşeni) içerir. Her Toggle, durumu değiştiğinde (onValueChanged olayı) ilgili PlayerPrefs anahtarını ("SfxEnabled", "MusicEnabled", "VoiceoverEnabled") günceller. Ayarlar ekranı açıldığında, Toggle'ların başlangıç durumu mevcut PlayerPrefs değerlerine göre ayarlanır. Ses çalmaktan sorumlu merkezi bir AudioManager sınıfı, sesleri çalmadan önce ilgili PlayerPrefs değerini kontrol eder. Ayarlar ekranından çıkmak için belirgin bir "Geri" veya "Kapat" düğmesi bulunur.

**SMART Hedefler:**

**(S) Özgün:** Ses/Müzik/Seslendirme için ayar kontrolleri sun, durumlarını PlayerPrefs'e kaydet/yükle, bu ayarları uygulama genelindeki ses çalma işlemlerine uygula.

**(M) Ölçülebilir:** Ayar değişikliği anında kaydedilmeli. Yapılan ayar değişikliğinin ses çalmaya etkisi < 0.5 saniye içinde fark edilmeli. Kaydedilen ayar ile Toggle durumu arasında %100 tutarlılık olmalı.

**(A) Ulaşılabilir:** Standart Unity UI Toggle, PlayerPrefs ve temel ses yönetimi scriptleri ile gerçekleştirilebilir.

**(R) İlgili:** Kullanıcıya ses deneyimi üzerinde kontrol imkanı sunar, temel bir kullanılabilirlik özelliğidir.

**(T) Zamanlı:** Tasarım ve uygulama Sprint 2 içinde tamamlanabilir.

**RAMS Etkisi:** Kullanıcı kontrolü sağlar (Kullanılabilirlik - A). Ayarlar kalıcıdır (Güvenilirlik - R). AudioManager gibi merkezi bir yapı kod tekrarını önler ve bakımı kolaylaştırır (Sürdürülebilirlik - M).

**4. Veri Tasarımı**

**4.1. Oyun İçeriği Veri Yapısı:**

* + **Yapı:** Her tarihi mekan için Unity ScriptableObject veya JSON/XML dosyaları kullanılacak. Örnek Alanlar: MekanID, MekanAdi, MekanAciklamasi, ModelReferansi (Prefab yolu), CevreModeliReferansi (varsa), HedefParcaListesi [Her parça için: ParcaID, ParcaAdi, ParcaModelIcindekiKonumu (Vector3), BilgiKartiIcerigi (metin, resim yolu, ses dosyası yolu)].
  + **RAMS Etkisi:** İçeriğin merkezi yönetimi (Sürdürülebilirlik - M), tutarlı veri yapısı (Güvenilirlik - R). Yeni mekan eklemeyi kolaylaştırır.

**4.2. Kullanıcı İlerleme Verisi:**

* + **Yapı:** Cihaz üzerinde PlayerPrefs veya basit bir dosya. Saklanacaklar: TamamlananMekanIDleri listesi, KazanilanRozetIDleri listesi.
  + **RAMS Etkisi:** **Kesinlikle PII içermez (Emniyet - S).** Veri kaybı riskini azaltır (lokal olduğu için - Güvenilirlik - R). Offline çalışır (Kullanılabilirlik - A).

**4.3. Ayarlar Verisi:**

* + **Yapı:** PlayerPrefs. Saklanacaklar: SesEfektleriAcik (bool), MuzikAcik (bool), SeslendirmeAcik (bool).

**5. Arayüz Tasarımı (UI/UX)**

**5.1. UI Akış Şeması:** Uygulamanın ana ekranlarından (Splash, Ana Menü, Mekan Seçimi, AR Ekranı, Bilgi Kartı, Ayarlar) geçişleri gösteren görsel diyagram.

**5.2. Ana UI Elemanları:**

* + Canlı, çocuk dostu renk paleti ve grafik stili.
  + Büyük, kolay dokunulabilir düğmeler ve ikonlar.
  + Okunabilir, basit yazı tipleri.
  + **Sanal Navigasyon Kontrolleri:** Konum (örn. sol alt), boyut ve görsellik (joystick mi, oklar mı?) çocuklar için optimize edilmiş. Başlangıçta nasıl kullanılacağına dair kısa bir görsel ipucu eklenebilir.
  + Dedektif Pati Karakteri: Ekranda genellikle sabit bir konumda (örn. sağ üst köşe), konuşurken animasyonlu.
  + Bilgi Kartı: Net resim alanı, kısa metin alanı, belirgin kapatma düğmesi, "Sesli Oku" butonu.

**5.3. UX Prensipleri (RAMS ile ilişkili):**

* + **Sezgisel Kullanım (Usability - A):** Menü navigasyonu ve özellikle **sanal kontroller** minimum açıklama ile anlaşılabilir olmalı.
  + **Anlık Geri Bildirim (Reliability - R, Usability - A):** Her dokunma, kontrol hareketi ve olay (parça bulma vb.) anında görsel ve/veya işitsel olarak teyit edilmeli.
  + **Tutarlılık (Maintainability - M, Reliability - R):** Farklı ekranlardaki benzer işlevler (örn. kapatma düğmeleri) aynı görünüm ve konumda olmalı.
  + **Basitlik (Usability - A, Safety - S):** Gereksiz karmaşıklıktan kaçınılmalı. Çocukları strese sokacak zaman sınırları gibi mekaniklerden kaçınılmalı.
  + **Yardımcı Rehberlik (Usability - A):** Dedektif Pati'nin rolü sadece görev vermek değil, oyuncu zorlandığında nazikçe yol göstermek olmalı.

**6. RAMS Analizi ve Tasarım Kararları Özeti**

**6.1. Güvenilirlik (Reliability):** Stabil AR çapa sistemi (anchor) kullanımı. Akıcı sanal kamera hareketi. Kapsamlı hata yakalama (örn. model yüklenemezse). Tutarlı veri yönetimi. Otomatik testlerin (birim/entegrasyon) proje sürecine dahil edilmesi.

**6.2. Kullanılabilirlik (Availability & Usability):** Optimize edilmiş varlık yükleme süreleri. Basit, sezgisel UI ve **özellikle sanal navigasyon kontrolleri üzerinde kullanıcı testleri yapılması**. Offline çalışabilirlik. Dedektif Pati ile sürekli rehberlik. Yaşa uygun dil kullanımı.

**6.3. Sürdürülebilirlik (Maintainability):** Modüler kod yapısı (örn. her mekan için ayrı konfigürasyon). Kodlama standartlarına uyum. Açıklayıcı yorum satırları. ScriptableObject/JSON ile içerik yönetiminin kolaylaştırılması. Sürüm kontrolü (Git) kullanımı.

**6.4. Emniyet (Safety):** **Kesinlikle PII toplamama politikası.** COPPA/GDPR-K uyumu. Uygulama içinde reklam, uygunsuz içerik veya harici bağlantı bulundurmama. Sadece oyuna odaklı, dikkat dağıtıcı olmayan tasarım. Sanal navigasyonun ani veya baş döndürücü olmaması.