

YAZILIM MÜHENDİSLİĞİ

GÜNCEL KONULAR

Programcı Kılavuzu



Hazırlayan

Muhammed Samet AKGÜL – 200541002

| | |
|--------------------------------|---|
| 1. Giriş | 3 |
| 1.1. Amaç | 3 |
| 1.2. Kapsam | 3 |
| 2. Yazılım Mimarisi | 3 |
| 2.1. Genel Mimari | 3 |
| Temel Bileşenler: | 3 |
| 2.2. Bileşenler | 3 |
| 2.3. Yazılım Tasarımı | 4 |
| 3. Geliştirme Araçları | 4 |
| 3.1. Kullanılan Teknolojiler | 4 |
| 3.2. Yazılım Geliştirme Süreci | 4 |
| 3.3. Kodlama Standartları | 4 |
| 4. Oyun Mantığı | 4 |
| 4.1. Oyun Akışı | 4 |
| 4.2. Kullanıcı Etkileşimleri | 5 |
| 4.3. Geri Bildirimler | 5 |
| 5. Test ve Hata Ayıklama | 5 |
| 5.1. Test Süreci | 5 |
| 5.2. Hata Ayıklama | 5 |
| 6. Kurulum | 5 |
| 6.1. Yazılım Kurulumu | 5 |
| 6.2. Sistem Gereksinimleri | 6 |
| 7. Bakım | 6 |
| 7.1. Yazılım Güncellemeleri | 6 |
| 7.2. Hata Düzeltmeleri | 6 |

1. Giriş

1.1. Amaç

Bu kılavuz, artırılmış gerçeklik (AR) tabanlı bir eğlence sisteminin geliştirilmesinde kullanılan yazılım bileşenlerini, araçları, kodlama standartlarını ve test süreçlerini açıklamak amacıyla hazırlanmıştır. Geliştirici olarak projeyi anlamanızı sağlamak, bakımı ve gelecekteki güncellemeleri kolaylaştırmak için gerekli tüm teknik bilgileri sağlar.

1.2. Kapsam

Bu doküman, yazılım geliştirme sürecinde kullanılan tüm teknolojilere, yazılım bileşenlerine ve projedeki mimari yaklaşıma odaklanmaktadır. Ayrıca, projenin test süreçlerini ve yazılım kurulumunu açıklayan detaylara da yer verilmiştir. Bu kılavuzun amacı, AR deneyimi sağlayan oyunların geliştirilmesiyle ilgilenen yazılımcılara rehberlik etmektir.

2. Yazılım Mimaris

2.1. Genel Mimari

Projede kullanılan yazılım mimarisi hakkında detaylı bilgi sağlanmıştır. Uygulama, **Unity** oyun motoru üzerinde geliştirilmiş olup, **Vuforia SDK** kullanarak artırılmış gerçeklik (AR) özellikleri entegrasyonu yapılmıştır. C# dilinde yazılım geliştirilmiştir.

Temel Bileşenler:

- **Unity Oyun Motoru:** Tüm oyun sahneleri, kullanıcı etkileşimleri ve AR deneyimleri burada yönetilmektedir.
- **Vuforia SDK:** AR hedeflerinin algılanması ve üzerine sanal içerik eklenmesi işlevini yerine getirir.
- **C# Programlama Dili:** Unity üzerinde oyun mantığını yazmak ve işlevsellik sağlamak için kullanılır.

2.2. Bileşenler

Projenin her bir bileşeninin işlevselliği detaylandırılmıştır:

1. **Kullanıcı Arayüzü (UI):** Oyunun ana ekranı, oyun içi menüler ve diğer etkileşimli öğeler Unity'nin UI araçları ile tasarlanmıştır. Butonlar, geçiş animasyonları ve oyun durumları gibi öğeler burada yer alır.
2. **AR Özellikleri:** Vuforia kullanılarak, kullanıcının çevresindeki nesnelerle etkileşim kurması sağlanır. Çevredeki hedef görseller algılandığında, sanal içerikler ekrana yerleştirilir.
3. **Veri Yönetimi:** Kullanıcı ilerlemesi ve oyun skorları gibi veriler, veritabanında saklanır. Ayrıca, oyuncu bilgileri yerel olarak cihazda depolanabilir.
4. **Görsel ve Ses Efektleri:** Kullanıcı geri bildirimleri için görsel animasyonlar ve ses efektleri tasarlanmıştır.

2.3. Yazılım Tasarımı

Proje, kullanıcı etkileşimi ve oyun içi mantık için modüler bir yapıya sahiptir. Oyun seviyeleri, kullanıcı seçimleri ve AR hedeflerinin tanınması bağımsız modüllerle işlenir. Bu sayede her bir bileşen birbirinden bağımsız olarak test edilebilir ve geliştirilebilir.

3. Geliştirme Araçları

3.1. Kullanılan Teknolojiler

Projede kullanılan yazılım ve donanım araçları aşağıda listelenmiştir:

1. **Unity:** Oyun geliştirme platformu olarak kullanılmıştır. Unity'nin güçlü araçları sayesinde oyun tasarımı ve AR entegrasyonu kolaylıkla yapılabilmektedir.
2. **Vuforia SDK:** AR özelliklerini entegre etmek için kullanılmıştır. Görsel tanıma ve hedeflere sanal içerik ekleme işlevselliği sağlar.
3. **C#:** Unity için ana yazılım dili olarak kullanılmıştır.
4. **Trello:** Proje yönetimi ve görev takibi için Trello kullanılmıştır. Takım üyeleri arasında işbirliği sağlamak ve projeyi takip etmek için bu araç kullanılmıştır.
5. **Git/GitHub:** Proje dosyalarının sürüm kontrolü için Git ve GitHub kullanılmıştır. Bu, yazılım geliştirme sürecinde takım içi işbirliğini sağlar ve kodun geçmişine dair izleme yapılmasını mümkün kılar.

3.2. Yazılım Geliştirme Süreci

Projenin yazılım geliştirme süreci şu aşamalarda gerçekleştirilmiştir:

1. **Tasarım Aşaması:** Projenin temel gereksinimleri ve kullanıcı etkileşimi planlandı. UI/UX tasarımı ve AR entegrasyonunun nasıl yapılacağına dair kararlar alındı.
2. **Geliştirme Aşaması:** Unity ortamında oyun sahneleri, menüler ve AR hedeflerinin işlevselliği kodlandı.
3. **Test ve Hata Ayıklama Aşaması:** Projede kullanılan her bileşen (AR hedef algılama, UI etkileşimleri, oyun mantığı) bağımsız olarak test edilerek entegrasyon sağlandı.

3.3. Kodlama Standartları

- **Kod Düzeni:** Kodlar, açıklayıcı yorumlarla ve düzgün bir biçimde yazılmıştır. Fonksiyonlar ve sınıflar, belirli işlevlere sahip olacak şekilde modüler hale getirilmiştir.
- **Versiyon Kontrolü:** Git, tüm proje dosyalarının sürüm kontrolünü sağlar. Her yeni özellik eklendiğinde ya da hata düzeltildiğinde, değişiklikler bir commit ile kaydedilir.

4. Oyun Mantığı

4.1. Oyun Akışı

Proje, iki ana oyundan oluşur:

1. **Pinwheel Game:** Kullanıcılar, benzer pinwheel'leri (rüzgar gülü) bularak puan toplar.
2. **PineTree Game:** Kullanıcılar, ağaç üzerinde gizlenmiş sayıları arar ve bu sayede eğitimsel bir oyun deneyimi sunulur.

4.2. Kullanıcı Etkileşimleri

- **Dokunma Etkileşimi:** Kullanıcılar, mobil cihazlarının ekranına dokunarak oyun içindeki nesnelerle etkileşimde bulunurlar. Bu etkileşimler, kullanıcıya görsel geri bildirim ve animasyonlarla desteklenir.
- **AR Algılama:** Vuforia SDK, kullanıcıların çevresindeki nesneleri algılar ve bu nesnelere sanal öğeler ekler. Kullanıcılar, AR hedeflerine doğru bakarak etkileşimde bulunabilirler.

4.3. Geri Bildirimler

Kullanıcı etkileşimleri sonrasında, oyun ses efektleri ve animasyonlarla geri bildirim sağlar. Örneğin, kullanıcı doğru bir seçim yaptığında "Kazandınız!" gibi bir ses duyulur ve ekranda animasyonlar görünür.

5. Test ve Hata Ayıklama

5.1. Test Süreci

Projede kapsamlı testler yapılmıştır:

1. **Manuel Testler:** Oyun mantığı ve kullanıcı etkileşimleri, manuel olarak test edilmiştir. Herhangi bir hatanın tespiti durumunda, yazılımcılar tarafından gerekli düzeltmeler yapılmıştır.
2. **Entegrasyon Testi:** Projede kullanılan farklı yazılım bileşenleri (Unity, Vuforia, kullanıcı arayüzü) arasında entegrasyon sağlanmış ve her bileşenin uyumlu bir şekilde çalışıp çalışmadığı test edilmiştir.
3. **Sistem Testi:** Projenin tamamlanmış haliyle tüm bileşenleri bir araya getirilerek sistemin genel işlevselliği test edilmiştir.

5.2. Hata Ayıklama

- **Hata İzleme:** Hatalar, Unity ve Vuforia logları ile takip edilmiştir. Hatalar tespit edildikçe, kod üzerinde gerekli düzenlemeler yapılmıştır.
- **Sorun Çözme Yöntemleri:** Performans iyileştirmeleri, bellek yönetimi ve kullanıcı etkileşimi hataları gibi konularda çözüm yolları araştırılmıştır.

6. Kurulum

6.1. Yazılım Kurulumu

1. **Unity İndir:** Unity'nin en son sürümünü [Unity Resmi Web Sitesi](#) üzerinden indiriniz.

2. **Vuforia SDK İndir:** Vuforia'nın Unity ile uyumlu olan en son sürümünü indiriniz ve projeye entegre ediniz.
3. **Projeyi İndirin ve Çalıştırın:** GitHub'dan projeyi indirip Unity ortamında açarak oyun uygulamasını başlatın.

6.2. Sistem Gereksinimleri

- **İşletim Sistemi:** Windows 10 veya macOS 10.15 ve üzeri
- **Donanım:** Minimum 8 GB RAM, NVIDIA GTX 1060 veya eşdeğer bir ekran kartı
- **Unity ve Vuforia Uyumluluğu:** Projenin Unity ve Vuforia'nın en son sürümleriyle uyumluluğu sağlanmıştır.

7. Bakım

7.1. Yazılım Güncellemeleri

- **Yazılım Güncellemeleri:** Unity veya Vuforia'nın yeni sürümleri yayınlandığında, proje bu sürümlerle uyumlu hale getirilip test edilmelidir. Ayrıca, kullanıcı geri bildirimlerine dayanarak yeni özellikler eklenebilir.
- **Yeni Özellikler:** Kullanıcı deneyimini geliştirmek amacıyla yeni özellikler eklenebilir.

7.2. Hata Düzeltmeleri

- **Hata İzleme ve Çözme:** Kullanıcılar tarafından bildirilen hatalar veya test sırasında tespit edilen sorunlar giderilmelidir.
- **Performans İyileştirmeleri:** Oyun performansını iyileştirmeye yönelik güncellemeler yapılabilir, özellikle işlemci ve bellek yönetimine odaklanılmalıdır.
-