

Beykent Üniversitesi
Yazılım Mühendisliği Bölümü
Yazılım Mühendisliği Tasarım Projesi

Rapor 1

2023 – GÜZ

“THE MIST”

2003013140 FARANGİS SAKAYİ

2003013066 İREM DEMİREĞEN

2103013802 KARDELEN ŞAŞKIN

I. PROJE TANIMI

Bu proje "The Mist" adlı bir korku-gerilim oyununun geliştirilmesini içeriyor. Oyuncular, ana karakter Eva'ya rehberlik ederek Lunavi virüsünden etkilenen tehlikeli hastane dünyasında hayatta kalmaya çalışacaklar. Oyun bulmaca çözme, gizem çözme ve korku unsurlarını içerir ancak aynı zamanda oyuncuya zekayı kullanma yeteneği de sunar. Hikâye, Eva'nın hastaneden kaçma ve dünyayı Lunavi virüsünün etkilerinden koruma hedefini konu alıyor. Proje'nin yenilikçi hikâye anlatımını, heyecan verici deneyimlerini ve bilimsel bağlantılarını birleştirerek oyun endüstrisine katkıda bulunmayı sağlıyor.

1. Projenin Amacı ve Hedefleri

Projemizin amacı Oyun geliştirme süreçlerini uygulama fırsatları sağlayarak oyun tasarımı, hikâye anlatımı, programlama ve oyun geliştirme becerilerini amaçlar. Aynı zamanda bilimsel, etik ve teknolojik konulara dikkat çekmeyi, oyuncuların zekâ ve düşünme kapasitelerini geliştirmeyi amaçlamaktadır. Bu hedefler, projenin değerli deneyimler sunmasını ve oyun endüstrisine katkıda bulunmasını sağlar. Oyununu geliştirirken, oyuncuların bağlılık geliştirmelerini ve kendilerini hikâyenin bir parçası gibi hissetmelerini sağlamak amaçlanır.

- **Yenilikçi oyun deneyimi:** "THE MIST" adlı oyun, video oyunları dünyasına benzersiz ve yenilikçi bir deneyim getirmeyi amaçlıyor. Amaç, oyuncuların korku ve gerilim atmosferinde zorlu bir hikâyeyi deneyimlemeleri ve bu süreçte heyecanlı, korkutucu ve unutulmaz anlar yaşamaları.
- **Gerilim ve Korku Deneyimi:** Korku ve gerilim türünde bir oyun, bu türün hayranlarına heyecan verici bir deneyim sunabilir. Oyunun atmosferi ve tehlikeleri, oyuncuları sürekli olarak tetikte tutabilir.
- **Bulmaca Çözme:** Oyundaki bulmacalar, oyuncuların mantıklı düşünme ve problem çözme becerilerini geliştirmelerine yardımcı olabilir. Bu, eğlence ve eğitimi birleştiren bir bileşen sunabilir.
- **Karanlık Temalar ve Mesajlar:** Oyun, bilim ve etik konuları ele aldığı için oyunculara dünya sorunlarına ve etik meselelere dair düşünme fırsatı sunabilir. Bu, oyunun sadece eğlenceli olmanın ötesinde düşünmeye teşvik edici bir unsuru içerir.
- **Çoklu Platform Erişimi:** Oyunun farklı platformlarda oynanabilir olması, daha geniş bir oyuncu kitlesine ulaşma fırsatı sunabilir. Bu, oyunun daha fazla insan tarafından deneyimlenmesine olanak tanır.
- **Bilimsel ve Tıbbi Farkındalık:** Oyunun tıp ve biyoloji konularına dikkat çekmesi, oyuncuların bilimsel ve tıbbi konularda daha fazla bilgi edinmelerine katkıda bulunabilir.
- **Eğitim Potansiyeli:** Oyun, zekâ oyunları ve bulmacalar içerdiği için eğitim potansiyeli taşıyabilir. Karmaşık sorunları çözme ve mantıklı düşünme becerilerini geliştirmek için kullanılabilir.

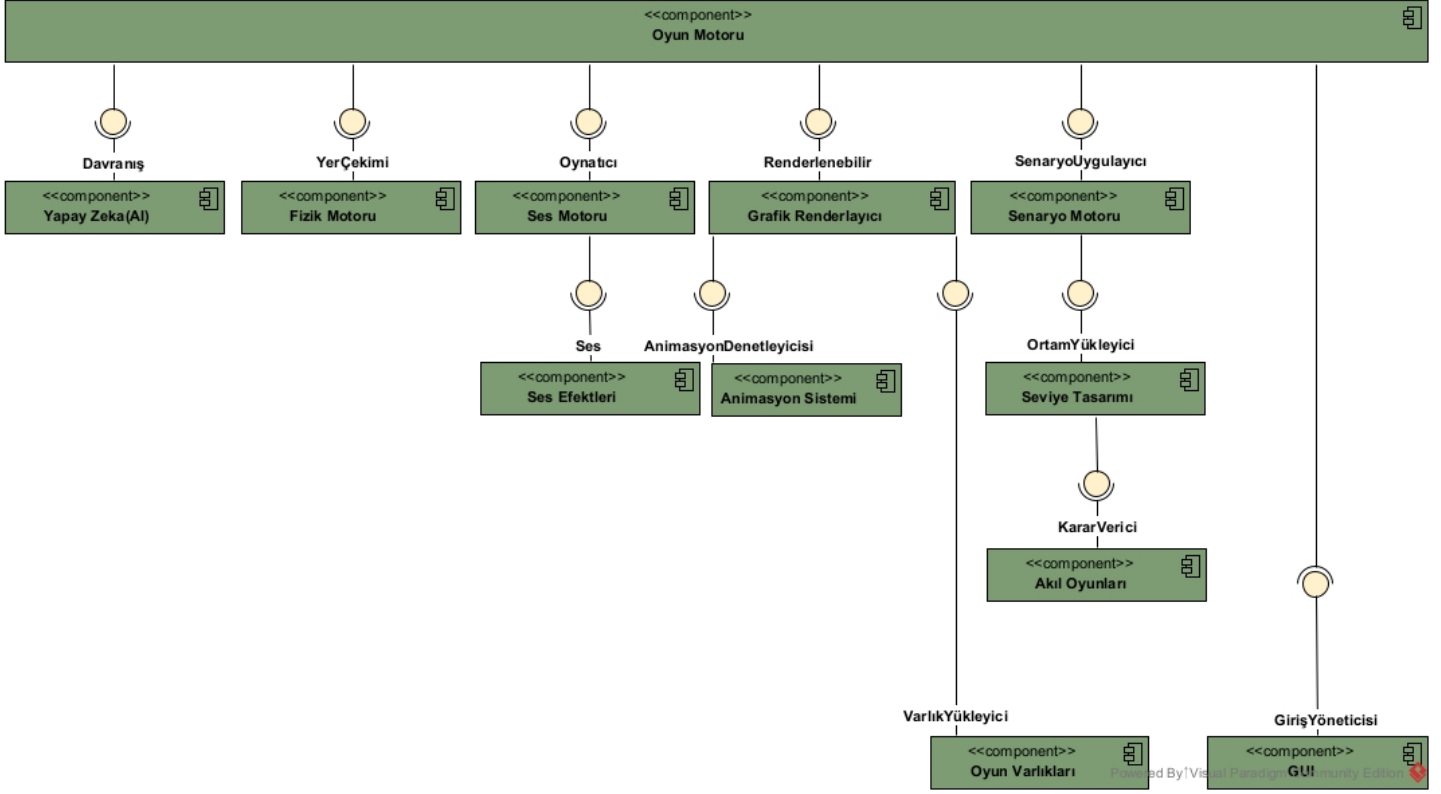
2. Projenin Kapsamı

Geliştirilecek ürün, mevcut oyun endüstrisinde yer alan korku ve gerilim türündeki oyunlara yenilikçi bir yaklaşım sunmayı hedeflemektedir. Oyunun niçin yeniden geliştirilmesi gerektiği ise, mevcut oyunlarda gözlemlenen bazı eksikliklerin giderilmesini gerektiren önemli bir motivasyon kaynağıdır. Mevcut oyunlar arasında, oyun hikayesi, karakter gelişimi ve bilimsel farkındalık unsurlarını bir araya getiren benzersiz bir deneyim sunma konusundaki eksiklikler gözlemlenmiştir. "THE MIST," bu açığı kapatmayı amaçlamaktadır. Oyun, oyunculara korku ve gerilim türünde heyecan verici anlar yaşatmanın yanı sıra karakter gelişimi ve derin hikâye anlatımı ile onları daha fazla içine çekmeyi hedefler.

"THE MIST," oyunun hikayesi ve karakter gelişimi üzerinden bilimsel ve tıbbi konulara dikkat çekmeyi amaçlar. Bu, oyuncuların oyun oynarken bilimsel ve tıbbi bilgi edinmelerine katkıda bulunur. Mevcut sistemlerde bu tür bir bilimsel farkındalık yaratma unsurunun eksik olduğu gözlemlenmiştir. Oyun ayrıca mevcut korku ve gerilim oyunlarının belirli konularda sınırlı kaldığı ve oyunculara daha fazla etkileşim ve düşünme fırsatı sunma potansiyelini değerlendirir. "THE MIST," oyuncuların zekâ kullanma, bulmacaları çözme ve strateji geliştirme becerilerini geliştirmelerini hedefler.

Sonuç olarak, "THE MIST" projesi, mevcut oyun endüstrisindeki eksiklikleri ele almayı amaçlar ve oyunculara hem korku ve gerilim türünde unutulmaz bir deneyim sunmayı hem de bilimsel ve tıbbi konulara dikkat çekmeyi hedefler. Bu nedenle, mevcut sistemlerin bu tür eksiklikleri giderme potansiyelini göz önünde bulundurarak "THE MIST" gibi bir oyunun geliştirilmesine ihtiyaç duyulmuştur.

2.1 Projenin İçeriği



Şekil 1 "THE MIST" component diagram

UML bileşen diyagramları bağlamında, "requested interface" (istenen arayüz), "provided interface" (sunulan arayüz) ve "dependency" (bağımlılık) terimleri, bileşenler arasındaki ilişkileri tanımlamak için kullanılır.

1) Provided Interface (Sunulan Arayüz):

- Bu, bir bileşenin dış ortama açığa çıkardığı veya sağladığı arayüz kümesini ifade eder. Diğer bileşenlere sistemdeki diğer bileşenlere sunduğu hizmetleri, işlevleri veya yetenekleri temsil eder. Diyagramlarda, bu genellikle bileşen üzerinde adlandırılmış bir bağlantı noktası olarak gösterilir.

2) Requested Interface (İstenen Arayüz):

- Bu, bir bileşenin doğru çalışabilmesi için ihtiyaç duyduğu veya gerektiği arayüz kümesidir. İstenen arayüzler, diğer bileşenlerden sağlanan dış işlevselliklere bağımlılığı gösterir. Bir bileşenin sistemdeki diğer bileşenlerden hizmetlerle etkileşimde bulunabilmesi için istenen arayüzler gereklidir.

3) Dependency (Bağımlılık):

- UML bileşen diyagramları bağlamında, bağımlılık; iki bileşen arasındaki ilişkiyi ifade eder ve bir bileşenin diğerine olan bağımlılığını gösterir. Bu ilişki, bağımlanan bileşende yapılan bir değişikliğin bağımlı bileşeni etkileyebileceğini ifade eder. Başka bir deyişle, bir bağımlılık, bir bileşenin başka bir bileşenin hizmetlerini kullandığını belirtir. Bağımlılıklar genellikle kesikli bir çizgi ve bağımlı bileşenden bağımlanan bileşene doğru bir ok ile temsil edilir.

Özetle:

- Provided Interface (Sunulan Arayüz): Bir bileşenin sunduğu hizmetler veya işlevler.
- Requested Interface (İstenen Arayüz): Bir bileşenin ihtiyaç duyduğu hizmetler veya işlevler.
- Dependency (Bağımlılık): Bir bileşenin başka bir bileşenin hizmetlerine bağımlı olduğunu gösteren ilişki.

Bileşenler:**1. Oyun Motoru:**

- Oyunu çalıştıran temel motoru temsil eder.

2. Yapay Zeka (AI):

- Oyuncu olmayan karakterlerin (NPC) ve canavarların davranışlarını kontrol eder.

3. Fizik Motoru:

- Oyun dünyasında gerçekçi fiziksel etkileşimleri simüle eder.
- Yerçekimi, çarpışmalar ve diğer fiziksel olayları yönetir.

4. Ses Motoru:

- Ses öğelerini oynatmayı kontrol eder.
- Ortam sesleri, müzik ve özel efektleri yönetir.

5. Ses Efektleri:

- Ayak sesleri, gıcırdayan kapılar ve korku deneyimini artırmak için diğer sesleri içerir.

6. Grafik Renderlayıcı (Grafik Oluşturucu):

- 2D ve 3D grafiklerin renderlenmesini yönetir.
- Karakterleri, ortamları ve özel efektleri renderlemek için bileşenler içerir.

7. Animasyon Sistemi:

- Karakter ve nesne animasyonlarını yönetir.
- Hareketleri, jestleri ve tepkileri kontrol eder.

8. Oyun Varlıkları:

- Oyunda kullanılan görsel, ses ve script varlıklarını içerir.

9. Senaryo Motoru:

- Yazılı olayların ve dizilerin uygulamasına olanak tanır.
- Oyuncu eylemleri ve tetikleyicilere bağlı olarak oyunun akışını kontrol eder.

10. Level Tasarımı:

- Oyun akışına yönelik oyundaki zorluk seviyesini yönetir ve kontrol eder.

11. Akıl Oyunları:

- Oyunun akışını sağlamak için olan akıl oyunları ve puzzle oyunlarını içerir.

12. Grafiksel Kullanıcı Arayüzü:

- Menüler, sağlık çubuklarının ve diğer UI öğelerinin görüntülenmesi için bileşenleri içerir.
- Navigasyon ve etkileşim için kullanıcı girişini yönetir.

İlişkiler:

- Oyun Motoru; *Şekil 1* de olduğu gibi oynanış, grafikleri ve genel oyun mekaniğini koordine eden tüm bileşenlerle etkileşimde bulunur.

- Oyuncu Denetleyici; oyuncunun girişlerini ve eylemlerini iletmek için Oyun Motoru ile iletişim kurar.

- Karakter Yapay Zekâ (Enfekte Hastalar) bileşeni; Oyun Motoru ile etkileşimde bulunarak oyun mantığına dayalı enfekte hastaların davranışını kontrol eder.

- Oyun Seviyeleri; Oyun Motoru tarafından yönetilir ve her seviye için özgü olan varlıklar ve bulmacalar içerir.

- Bulmaca Sistemi; Zihinsel zorlukları ve bulmacaları sunmak ve yönetmek için Oyun Motoru ile etkileşimde bulunur.
- Ses Sistemi; ses efektleri ve müziği oyunun ürkütücü atmosferini oluşturmak için Oyun Motoru ile çalışır.
- Envanter Sistemi; Oyun Motoru ile etkileşimde bulunarak oyuncunun topladığı öğeleri yönetir.
- Kayıt Sistemi; oyuncuların ilerlemelerini kaydetmelerine ve kaydedilmiş oyunları yüklemelerine olanak tanır ve Oyun Motoru ile etkileşimde bulunur.
- Hikâye Sistemi; Oyun Motoru ile etkileşimde bulunarak oyunun hikayesini ve ilerlemesini sunar.
- Oyun Kaynakları; farklı bileşenler tarafından kullanılır, bu bileşenler arasında Oyun Motoru, Karakter Yapay Zekâ ve Bulmaca Sistemi bulunur.

Oyun Motoru ile Oyun Seviyeleri Arasındaki İlişki:

- Oyun Motoru, oyun seviyelerini yönetir. Her bir seviye, Oyun Motoru tarafından kontrol edilir ve yönetilir.

Oyuncu Denetleyici ile Oyun Motoru Arasındaki İlişki:

- Oyuncu Denetleyici, oyuncunun girişlerini alır ve bu girişleri Oyun Motoruna ileterek oyuncunun oyunla etkileşimde bulunmasını sağlar.

Karakter Yapay Zekâ ile Oyun Motoru Arasındaki İlişki:

- Karakter Yapay Zekâ, Oyun Motoru ile etkileşimde bulunarak oyun mantığına dayalı enfekte hastaların davranışlarını kontrol eder.

Oyun Seviyeleri ile Bulmaca Sistemi Arasındaki İlişki:

- Her seviye, kendi içinde bulmacaları içerir. Bulmaca Sistemi, Oyun Seviyeleri tarafından kullanılarak zihinsel zorlukları ve bulmacaları yönetir.

Bulmaca Sistemi ile Oyun Motoru Arasındaki İlişki:

- Bulmaca Sistemi, Oyun Motoru ile etkileşimde bulunarak bulmacaların çözümünü kontrol eder.

Ses Sistemi ile Oyun Motoru Arasındaki İlişki:

- Ses Sistemi, Oyun Motoru ile iş birliği yaparak oyunun atmosferini oluşturan ses efektleri ve müziği sağlar.

Envanter Sistemi ile Oyun Motoru Arasındaki İlişki:

- Envanter Sistemi, Oyun Motoru ile etkileşimde bulunarak oyuncunun envanterini ve topladığı öğeleri yönetir.

Kayıt Sistemi ile Oyun Motoru Arasındaki İlişki:

- Kayıt Sistemi, Oyun Motoru ile etkileşimde bulunarak oyuncuların ilerlemelerini kaydetmelerine ve yüklemelerine olanak tanır.

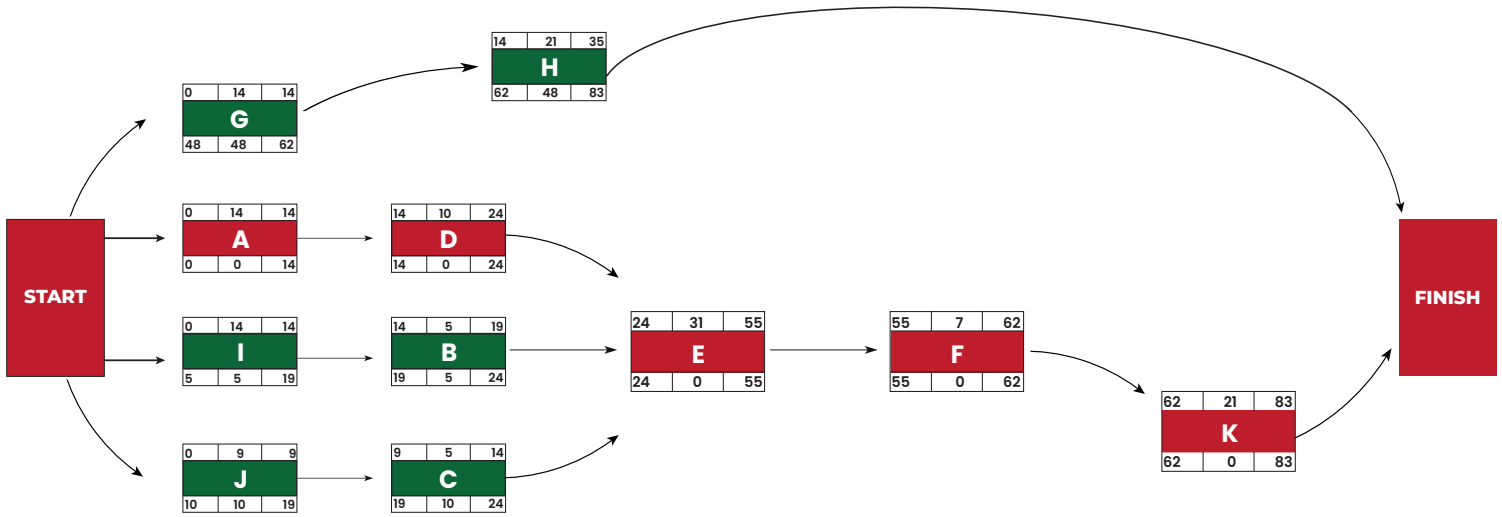
Hikâye Sistemi ile Oyun Motoru Arasındaki İlişki:

- Hikâye Sistemi, Oyun Motoru ile etkileşimde bulunarak oyunun hikayesini ve ilerlemesini yönetir.

Oyun Kaynakları ile Oyun Motoru, Karakter Yapay Zeka ve Bulmaca Sistemi Arasındaki İlişki:

- Oyun Kaynakları, Oyun Motoru, Karakter Yapay Zekâ ve Bulmaca Sistemi gibi bileşenler tarafından kullanılır. Bu bileşenler, oyun içindeki grafikler, sesler, medya dosyaları ve diğer kaynaklara erişim sağlar.

3. Proje Ağ Diyagramı



Şekil 2 The MIST Ağ Diyagramı

KRİTİK YOL : A – D – E – F – K

□ Yapılan ağ diyagramında kırmızı kutucukla işaretli olan yerler kritik yolu temsil eder.

A = OYUN MOTORU	14 GÜN
B = SES SİSTEMİ	5 GÜN
C = MÜZİK MOTORU	5 GÜN
D = GİRİŞ MOTORU	10 GÜN
E = GENEL PROGRAMLAMA	31 GÜN
F = MATEMATİK / FİZİK	7 GÜN
G = 2D ÇİZİM	14 GÜN
H = 3D ÇİZİM	21 GÜN
I = SES EFEKTLERİ	14 GÜN
J = MÜZİK / SES KAYDI	9 GÜN
K= TASARIM	21 GÜN

4. Fonksiyonel Olmayan Gereksinimler

Güvenilirlik:

Yüksek Hata Toleransı: Sistem, kullanıcıların etkilenebilecek her türlü hata durumunda otomatik olarak geri yüklenebilir durumda olmalıdır.

Güvenlik Duvarları: Sistem, güvenilirlik ve dayanıklılık sağlamak adına düzenli olarak güvenlik duvarları tarafından taranmalı ve gerekli güvenlik önlemleri alınmalıdır.

Yedekleme Sıklığı: Kritik veri tabanları ve sistem bileşenleri düzenli aralıklarla yedeklenmeli ve bu yedekleme süreçleri otomatik olarak gerçekleşmelidir.

Performans:

Maksimum Kullanıcı Yüğü: Sistem, en yoğun kullanım anlarında bile minimum gecikme ve en yüksek performans seviyelerini sağlamak için optimize edilmelidir.

Yanıt Süreleri: Kullanıcının herhangi bir eylemine sistemin hızlı bir şekilde yanıt vermesi için, sistem yanıt süreleri belirli bir eşiğı geçmemelidir.

Platform Bağımsızlık: Farklı cihazlarda (bilgisayarlar, tabletler, akıllı telefonlar) ve işletim sistemlerinde tutarlı bir performans sağlanmalıdır.

Kullanılabilirlik:

Kullanıcı Dostu Arayüz: Kullanıcı arayüzü, kullanıcıların oyunu kolayca anlamalarını ve kullanmalarını sağlayacak şekilde tasarlanmalıdır.

Bağlantı Durumu Bildirimleri: Veri girişi yapılırken internet bağlantısının sağlıklı olduğu kabul edildiğinde, kullanıcılara bağlantı durumu hakkında anlık geri bildirim sağlanmalıdır.

Çoklu Dil Desteğı: Oyun, farklı dil ve kültürleri destekleyerek geniş bir kullanıcı kitlesine hitap etmeli ve kullanıcılar arasında dil bariyerini ortadan kaldırmalıdır.

Güvenlik:

Gizlilik: Kullanıcı bilgileri güvenli bir şekilde depolanmalı ve işlenmelidir. Bu, gelişmiş şifreleme ve güçlü kimlik doğrulama yöntemleri kullanılarak sağlanmalıdır.

Güvenlik Güncellemeleri: Güvenlik güncellemeleri düzenli olarak uygulanmalı ve sistem, bilinen güvenlik açıklarına karşı sürekli olarak taranmalıdır.

Yetkilendirme Kontrolleri: Sisteme erişim seviyeleri, kullanıcı türüne bağlı olarak doğru bir şekilde belirlenmeli ve izlenmelidir.

Sürdürülebilirlik:

Modüler Yapı: Sistem, yeni özellikler eklemek ve mevcutları güncellemek için modüler bir yapıya sahip olmalıdır.

Düşük Bakım Gereksinimi: Sistem, arızaları en aza indirmek ve bakım süreçlerini kolaylaştırmak adına düşük bir bakım gereksinimiyle tasarlanmalıdır.

Geliştirici Dostu: Geliştirici belgeleri ve dokümantasyonlar eksiksiz olmalıdır, bu sayede yeni geliştirmeler ve güncellemeler kolayca uygulanabilir.

Taşınabilirlik:

Çoklu Platform Desteği: Oyun, farklı işletim sistemlerinde (Windows, macOS, Linux) sorunsuz bir şekilde çalışabilmeli ve farklı ekran çözünürlüklerine uyum sağlamalıdır.

Mobil Uyum: Oyun, mobil cihazlarda da oynanabilir olmalı ve mobil ekran boyutlarına uyum sağlamalıdır.

Dokümantasyon:

Kapsamlı Kullanıcı Kılavuzu: Kullanıcılar ve sistem yöneticileri için kapsamlı bir kullanıcı kılavuzu sağlanmalıdır.

Geliştirici Belgeleri: Sistem mimarisi, API dokümantasyonları ve geliştirici belgeleri eksiksiz ve anlaşılır olmalıdır.

5. Tanımlamalar

Temel Terimler:

- *Lunavi Virüsü (LV)*: Oyunun merkezindeki tehdit. İnsanlar arasında bulaşıcı ve etkilerini gösterdiğinde delirme ve saldırganlık gösterme özellikleri bulunan bir virüs.
- *VRC Şirketi*: Virus Research Center'ın kısaltması. Oyunun geçtiği askeriye şirketi. Lunavi virüsüne dair gizli projeleri ve karanlık geçmişi içerir.
- *Howrinskaya Hastanesi*: Oyunun ana geçtiği yer. Virüsün etkilerini incelemek ve yayılmasını kontrol altında tutmak adına kullanılan bir araştırma hastanesi.

Karakter ve Nesneler:

- *Eva (E)*: Oyunun ana karakteri. Bilim ve araştırmaya olan tutkusu, Lunavi virüsü salgını sırasında test edilecek ve hayatta kalmak için mücadele edecektir.
- *Hastalar (H)*: Lunavi virüsü tarafından etkilenmiş düşmanlar. Normalde insanlar, ancak virüs nedeniyle saldırgan hale gelmişlerdir.
- *Anahtar (K)*: Oyuncunun ilerlemek ve kapıları açmak için kullanabileceği nesne.

Oyun Mekaniği:

- *Bulmaca (P)*: Oyuncunun ilerlemesi için çözmesi gereken zorlu mantık sorunları.
- *Gizli Geçiş (GG)*: Oyuncunun keşfetmesi ve kullanması gereken gizli bir yol veya geçiş.

Oyun Stilleri ve Atmosfer:

- *Retro (SKY)*: Oyunun geçtiği atmosfer, eski dönemlere referans. Teknolojik ve görsel olarak eski moda bir his uyandırır.
- *Gerilim (⚡)*: Oyunun içindeki tehlikeli durumları temsil eder. Oyuncuyu tetikte tutar.
- *Korku (👻)*: Oyunun korku unsurları içerdiğini ve oyuncuya korku hissiyatı kazandırmayı amaçladığını belirtir.

Proje İçindeki Özel Konseptler:

- *Lunatic Virüsü (LV)*: Oyunun özgün virüs konsepti. Lunavi virüsünün karakteristik özelliklerini ve etkilerini tanımlar.
- *Toplama Görevi (TG)*: Oyuncunun belirli nesneleri toplamasını gerektiren bir görev türü. Hayatta kalırken kullanacağı araç ve gereçlerdir.

Kullanılacak Platform veya Teknolojiler:

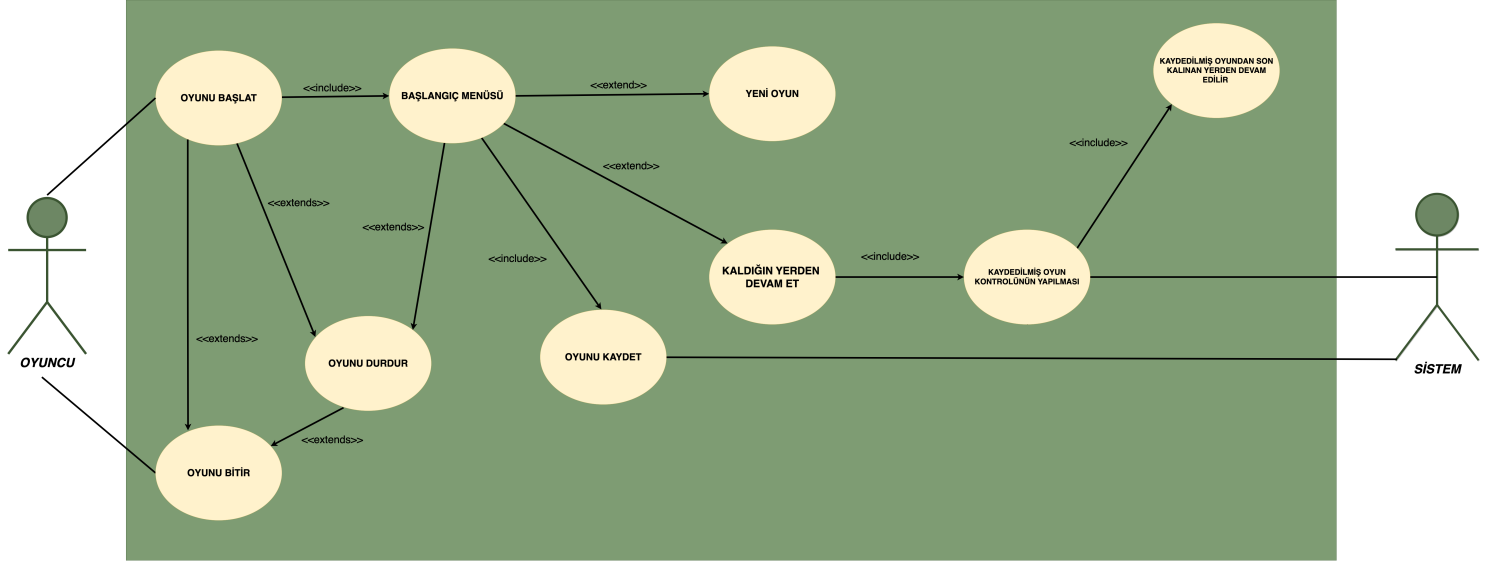
- *Unity (U)*: Oyun geliřtirmek için kullanılan oyun motoru.
- *Steam* : Oyunun satıřa sunulcađı platform

Proje Ekibi ve Roller:

- *Proje Yöneticisi (PM)*: Projenin genel yönetiminden sorumlu kiři.
- *Geliřtirici (G)*: Oyunun programlama ve teknik kısmından sorumlu kiři.
- *Tasarımcı (T)*: Oyunun görsel ve kullanıcı deneyimi tasarımından sorumlu kiři.

II. Gereksinimlerin Analizi

6. Use Cases ve Tablo Açıklamaları

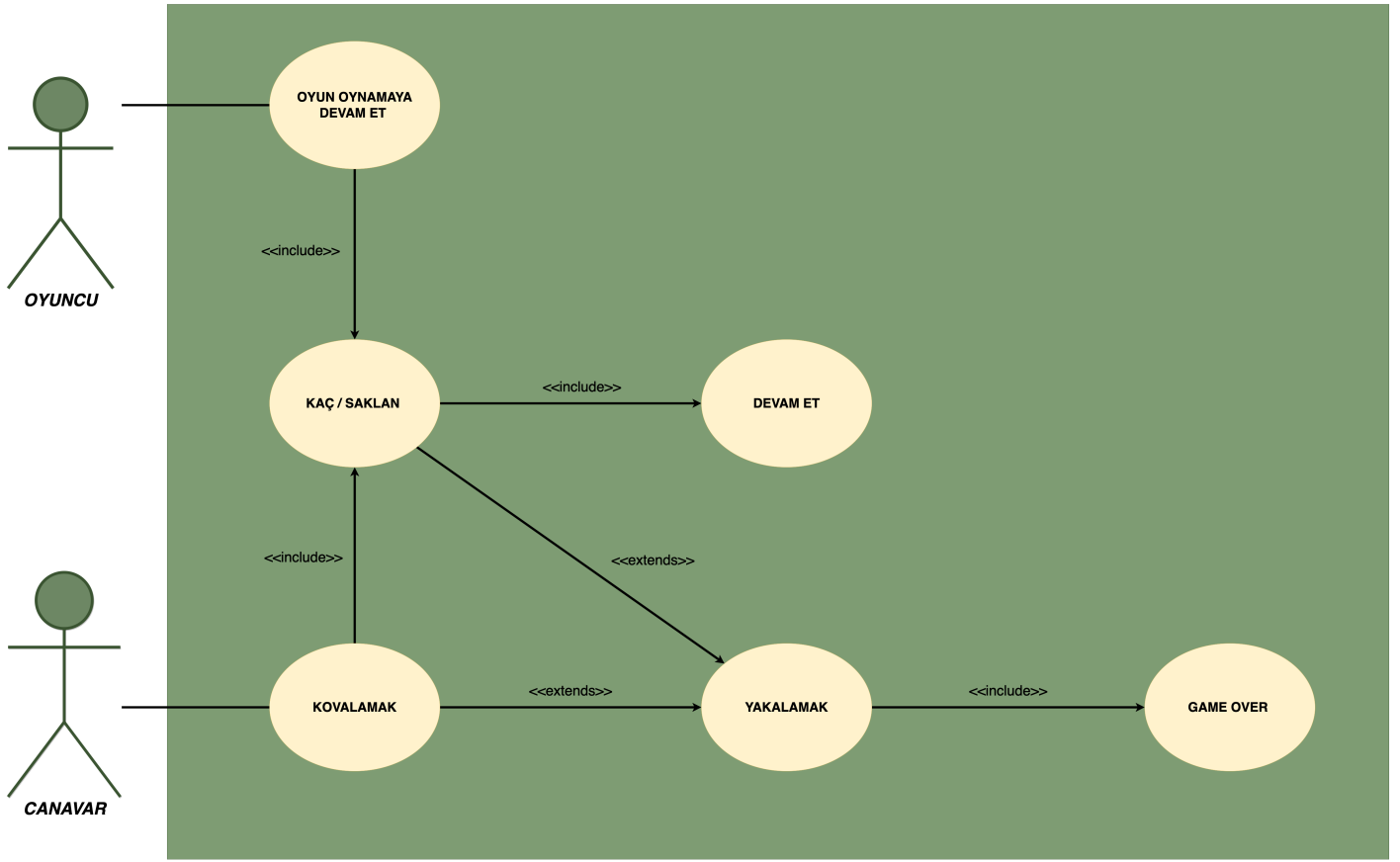


Şekil 3 THE MIST BAŞLANGIÇ MENÜSÜ KULLANIMI

- BAŞLANGIÇ MENÜSÜ KULLANIMI

Tablo 1 BAŞLANGIÇ MENÜSÜ KULLANIMI

Use Case Numara:	1
Use Case İsmi	Başlangıç Menüsü Kullanımı
Ön Koşul	Oyuncunun oyunu başlattığı varsayılır
Son Koşul	Oyuncu Ana menü ekranını kullanarak oyununu başarılı bir şekilde başlatmıştır
Temel yol	1. Kullanıcı Oyunu başlat butonuna tıklar 2. Kullanıcı karşısına çıkar ara yüzden yeni oyun seçeneğini seçer. 3. Oyun yazılımı oyunu başlatır.
Alternatif Yol	1. Kullanıcı Oyunu Başlat butonuna tıklar 2. Kaldığın yerden devam et butonuna tıklar 3. Oyun yazılımı kaydedilmiş bir oyun var mı kontrolü yapar 4. Oyun yazılımı kaydedilmiş bir oyun dosyası bulur ise 5. Oyunu son kalan yerden devam ettirir
Alternatif Yol:	1. Kullanıcı Oyunu Başlat butonuna tıklar 2. Kaldığın yerden devam et butonuna tıklar 3. Oyun yazılımı kaydedilmiş bir oyun var mı kontrolü yapar 4. Oyun yazılımı kaydedilmiş bir oyun dosyası bulamaz ise 5. Oyun yazılımı başlangıç menüsüne geri döndürür.
Alternatif Yol:	1. Oyuncu kaydet butonuna tıklayarak oyunu sonlandırır
Alternatif Yol:	1. Kullanıcı oyunu başlat butonuna tıklar 2. Kullanıcı oyunu durdur butonuna tıklar 3. Kullanıcı oyunu bitir butonuna tıklar

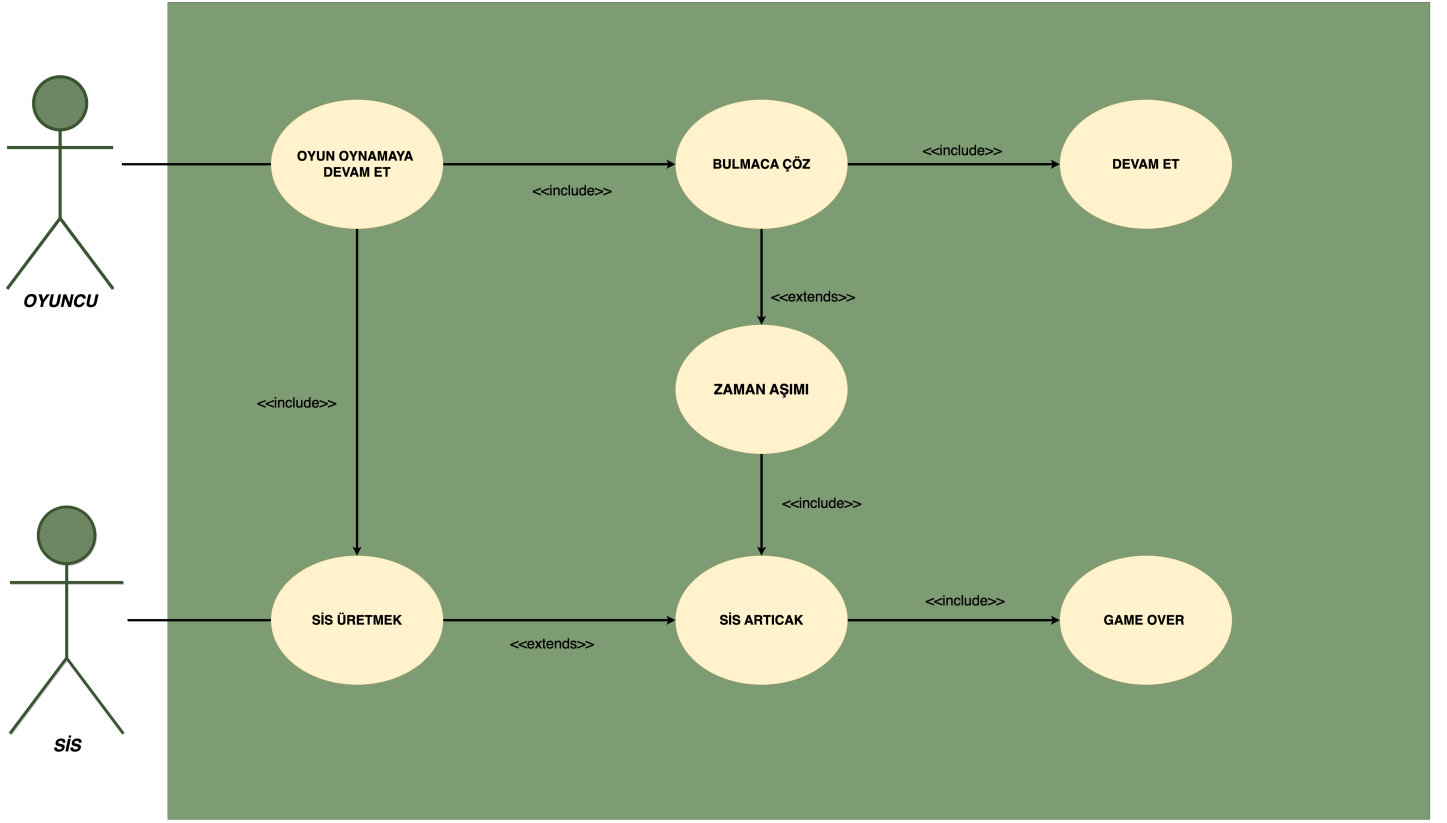


Şekil 4 Canavar Aktörünün Oyuncu Aktörü ile İlişkisi

- CANAVARIN AKTÖR OYUNCUSUYLA İLİŞKİSİ

Tablo 2 Canavar Aktörünün Oyuncu Aktörü İle İlişkisi

Use Case Numara:	2
Use Case İsmi	Canavar Aktörünün Oyuncu Aktörü İle İlişkisi
Ön Koşul	Canavarın oyuncu aktörünü farketmesi
Son Koşul	Canavarın oyuncuyla ilişkisinde canavar oyuncuyu yakalarsa game over durumuna bağlanması
Temel yol	<ol style="list-style-type: none"> 1. Canavar saldırmaya çalışır 2. Oyuncu(Eva) kaçmayı başaramaz ise 3. Canavar oyuncuyu yakalar ve game over olur.
Alternatif Yol	<ol style="list-style-type: none"> 1. Canavar Saldırmaya Çalışır 2. Oyuncu(Eva) kaçıp saklanır 3. Canavar yakalamayı başaramaz.
Alternatif Yol	<ol style="list-style-type: none"> 1. Canavar Saldırmaya Çalışır 2. Oyuncu(Eva) kaçır ve saklanır 3. Canavar oyuncuyu yine de bulursa 4. Canavar oyuncuyu öldürür 5. Oyun game over olur.

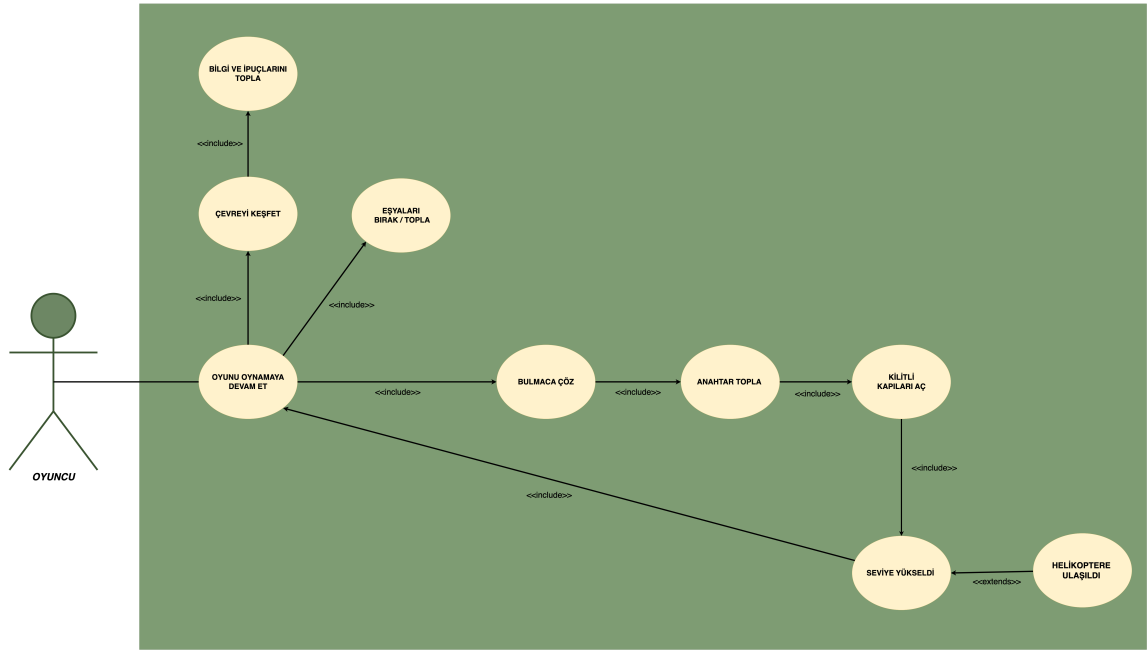


Şekil 5 Sis Aktörünün Oyuncu Aktörü İle İlişkisi

- SİS AKTÖRÜNÜN OYUNCU AKTÖRÜYLE İLİŞKİSİ

Tablo 3 Sis Aktörünün Oyuncu Aktörü İle İlişkisi

Use Case Numara:	3
Use Case İsmi	Sis Aktörünün Oyuncu Aktörü İle İlişkisi
Ön Koşul	Belirli süre içerisinde bulmaca çözülmez ise sistin artması
Son Koşul	Sis artarak oyuncuyu etkisi altına alıp game over durumuna bağlanması
Temel yol	<ol style="list-style-type: none"> Oyuncu (Eva) bulmacayı verilen sürede çözemez ise Sis aktörünün etkisi artar Oyun game over olur.
Alternatif Yol	<ol style="list-style-type: none"> Oyuncu (Eva) bulmacayı verilen sürede çözer ise Oyuncu (Eva) bir sonraki aşamaya geçecek Verilen sürede çözer ise devam edecek



Şekil 6 Oyuncunun oyun içindeki durumları

- OYUNCUNUN OYUN İÇİNDEKİ DURUMLARI

Tablo 4 Oyuncunun oyun içindeki durumları

Use Case Numara:	4
Use Case İsmi	Oyuncunun oyun içindeki durumları
Ön Koşul	Oyunun başarılı bir şekilde başlayıp Ana karakterin uyanması
Son Koşul	Ana karakterin Helikoptere başarılı bir şekilde ulaşması
Temel yol	<ol style="list-style-type: none"> 1. Eva uyanır 2. Bulmacaları çözer 3. Anahtarları toplar 4. Kapıları Açar 5. Ve helikoptere ulaşır.
Alternatif Yol	<ol style="list-style-type: none"> 1. Eva uyanır 2. Çevreyi keşfeder ve ipuçlarını toplar 3. Karşısına çıkan bulmacaları verilen sürede başarılı bir şekilde çözer 4. Canavarla karşılaşır 5. Başarılı bir şekilde kaçmayı başarır 6. Son aşamaya ulaşır 7. Ve helikopterle kaçmayı başarır.

Alternatif Yol:	<ol style="list-style-type: none"> 1. Eva uyanır 2. Çevreyi keşfeder, ipuçlarını bulur inceler 3. Etrafındaki eşyaları toplar 4. Verilen ipuçlarını öğrenir. 5. Canavarla Karşılaşır 6. Canavardan kaçıp saklanmayı başaramaz ise 7. Canavar onu yakalar ve game over olur.
Alternatif Yol:	<ol style="list-style-type: none"> 1. Eva uyanır 2. Çevreyi Keşfeder, ipuçlarını bulur inceler 3. Etrafındaki eşyaları toplar 4. Canavar ile karşılaşır ve canavardan kaçıp saklanmayı başarır 5. Verilen süre içerisinde bulmacaları çözemez 6. Ve sis artar 7. Oyun game over olur.
Alternatif Yol:	<ol style="list-style-type: none"> 1. Eva uyanır 2. Çevreyi Keşfeder, ipuçlarını bulur inceler 3. Etrafındaki eşyaları toplar 4. Canavar ile karşılaşır ve canavardan kaçıp saklanmayı başarır 5. Verilen Süre içerisinde bulmacaları çözer 6. Anahtarları başarılı bir şekilde bulup kapıları açar 7. Tüm seviyeleri (katları) başarılı bir şekilde tamamlar 8. Helikoptere ulaşır ve kaçmayı başarır.

□ Tüm diyagramlar draw.io ile çizilmiştir.

7. Fonksiyonel Gereksinimler

"Projemizin temel fonksiyonel gereksinimleri, kullanıcıya etkileyici bir oyun deneyimi sunmak üzerine odaklanmaktadır. Oyun, oyunculara karakterlerini kullanma ve kontrol etme yeteneği sunmaktadır. Bu bağlamda, karakter hareketleri, saldırı ve savunma mekanizmaları ile ilgili temel kontroller oyunculara sunmaktadır. Ayrıca, oyun içi görevler ve bulmacalar, kullanıcıları mekanik olarak meşgul ederek oyunun çeşitliliğini arttırmaktadır. Oyunun hikayesi, kullanıcıları bağlamak ve ilgilerini sürdürmek adına derinlik ve çeşitlilik içermektedir. Grafikler ve ses efektleri, oyunun görsel ve işitsel açıdan çekici olmasını sağlamak için yeterli düzeydedir. Ayrıca, oyuncuların ilerlemelerini takip etmelerini sağlamak için kayıt ve checkpoint sistemleri gibi temel ilerleme mekanizmaları bulunmaktadır. Bu temel fonksiyonel gereksinimler, projenin başarılı bir oyun deneyimi sunmasını sağlamak için önemlidir."

Tablo 5 Problem Gereksinimlerinin Tanımı

Gereksinim ID	Gereksinim Tanımı
FG.1.0	Kullanıcı oyun yazılımını başlatır.
FG.1.0.1	Kullanıcı başlangıç menüsü butonuna basarak yeni bir oyun başlatabilir yada kaldığı yerden oyuna devam edebilir
FG.1.0.2	Kullanıcı kayıtlı oyunu bulamazsa tekrardan başlangıç menüsüne dönebilir
FG.1.0.3	Kullanıcı oyunu kaydet butonuyla oyunu kaydedebilir.
FG.1.0.4	Kullanıcı oyunu durdurabilir ya da oyunu bitir butonu ile oyunu bitirebilir.
FG.2.0	Canavar aktörü oyunun başlamasıyla sistem tarafından aktifleşir.
FG.2.0.1	Canavar aktörü oyuncuyu öldürebilir.
FG.2.0.2	Canavar aktörü oyuncuyu kovalayabilir.
FG.2.0.3	Canavar aktörü oyuncuya saldırabilir.
FG.3.0	Sis aktörü oyunun başlamasıyla sistem tarafından aktifleşir.
FG.3.0.1	Sis aktörü süreye bağlı olarak artmaktadır.
FG.4.0	Sistemin oyunu başlatmasıyla Eva karakteri uyanır
FG.4.0.1	Oyuncu oyunu keşfedebilir
FG.4.0.2	Oyuncu bilgileri ve ipuçlarını toplayabilir.
FG.4.0.3	Oyuncu itemleri toplayıp, bırakabilir.
FG.4.0.4	Oyuncu kaçıp saklanabilir.
FG.4.0.5	Oyuncu karakteri verilen sürede bulmacaları çözebilir.
FG.4.0.6	Oyuncu karakteri anahtarları toplayıp, kapıları açabilir
FG.4.0.7	Oyuncu level(kat) atlayabilir.
FG.4.0.8	Oyuncu canavar ve sis tarafından ölebilir.
FG.4.0.9	Oyuncu helikoptere ulaşip kaçabilir ve oyunu tamamlayabilir.