

BIL 431 – Tasarım 1 Dersi
2020-2021 Güz Dönemi Final Çalışması

PROJE DÖKÜMANTASYONU

Abdullah TUNÇER
1721012045
Bilgisayar Mühendisliği

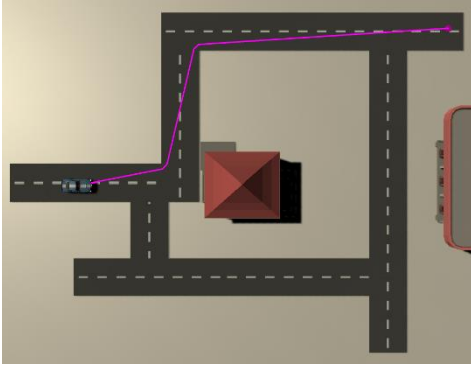
Isparta – 2020

1. GİRİŞ

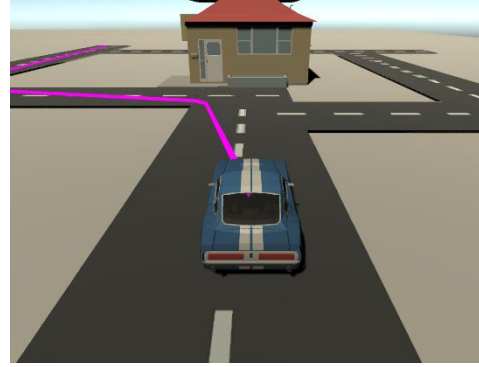
Unity oyun motoruyla 3 boyutlu tek oyunculu araç sürüş simülasyonu yapılacaktır. Oyundaki scriptler C# ile yazılmaktadır. Proje yedeklemesi Unity Cloud üzerinden yapılmaktadır. Oyun sadece bilgisayar ortamında oynanabilecektir. Oyunda araçla görevler yapıp para kazanılabilecek ve parayla daha gelişmiş araçlar alınabilecektir.

Görevleri oyun içerisindeki tableti açarak seçilebilir. Görevler belli ağırlığı olan eşyaları bir yerden alıp bir yere götürme üzerinedir. Görevleri seçerken aracın taşıyabileceği kapasiteye dikkat edilmelidir. Oyun tabletinde 4 ana görev türü vardır; lokanta(hafif), market(orta), nakliye(ağır) ve yolcu taşımacılığı. Ana görev türü seçildikten sonra sistem görevler üretir ve listeler. Görevlerin başlangıç noktaları sabittir (yolcu taşıma hariç). Hedef nokta ise evlerin konumları olduğu bir veri tabanından rastgele seçilir. Ücret ve yük ise ana görev türüne göre belirli aralıkta rastgele belirlenir. Görev seçildiğinde haritada başlangıç ve bitiş noktaları işaretlenir. Görev başarıyla tamamlandığında ödeme oyuncuya aktarılır.

M tuşuna basarak harita açılabilir ve gidilmek istenilen yere sağ tıklayarak rota oluşturulabilir. Rota oluşturma NavMesh yapay zekası ile sağlanır ve rota Line Renderer ile yolun üzerine çizilir.



Şekil 1.1. Harita, M Tuşu ile Açılır/Kapatılır



Şekil 1.2. Oyuncu Kamerası

Araç kaza yaptıkça hasar almaktadır. Hasar algılaması, araç colliderı farklı bir collidera çarptığında tetiklenecektir. Collider, modellerin birbirlerinin içinden geçmesini engelleyen iskelet denilebilir. Araç tamiri servise giderek yapılabilir. Eğer araç hareket edemeyecek düzeye gelir ise oyun içindeki tableten kurtarıcı çağrılabilir. Kurtarıcı aracı servise bırakır. Kurtarıcı aracın yakıtı bittiği zaman da çağırılarak servisten bir miktar yakıt alınabilir. Araç deposunu doldurmak için benzinlikte pompa ile etkileşime girerek doldurulabilir.

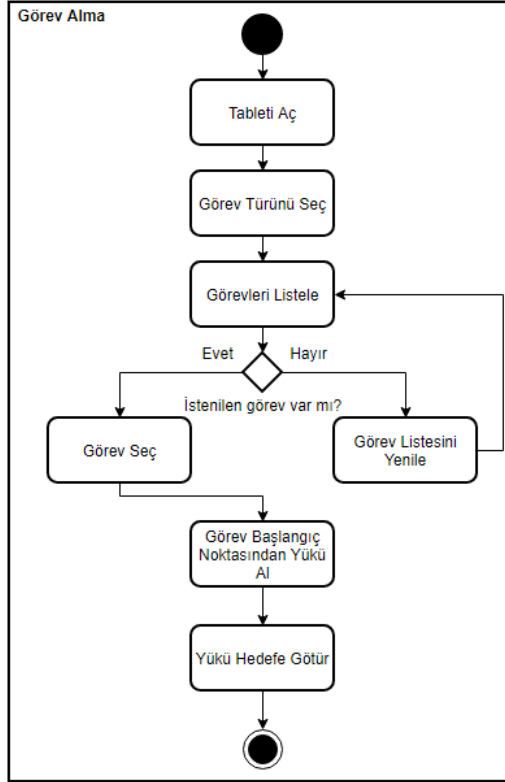
Araç marketi, oyun tableti içindeki araç reklamına tıklayarak erişilir. Oyuncuda mevcut olmayan araçlar araç veri tabanından alınır ve listelenir. Araçların hız, yük kapasitesi, depo büyüklüğü ve dayanıklılık gibi farklı özellikleri vardır. Oyuncunun parası yettiği bir aracı alındığında araç oyuncunun garajına eklenir eklenir ve araç veri tabanından alındığına dair true değeri gönderilir.

Oyuncu, garajına otoparktan erişebilir. Sistem sahip olunan araçlara, araç veri tabanından erişir. Mevcut araç değiştirilmek istenilirse garaj menüsünden araç seçilir. Mevcut araç garaja gönderilir seçilen araç garajdan çıkarılır.

2. DAVRANIŞSAL TASARIM

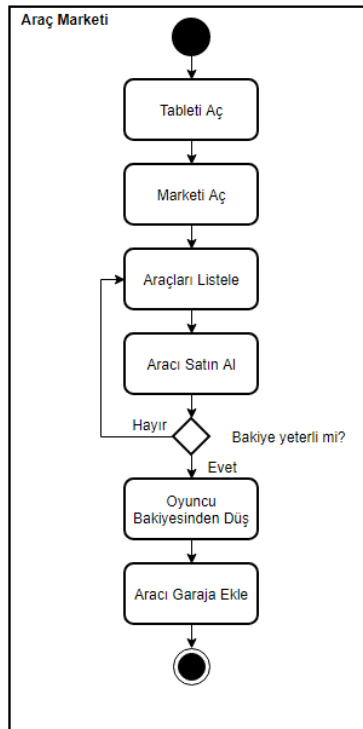
2.1. Activity Diagram

Şekil 2.1.1’de oyun tabletinden görev almayı gösterir. Görev türü 4 tanedir bunlar lokanta(hafif), market(orta), nakliye(ağır) ve yolcu taşımacılığıdır.



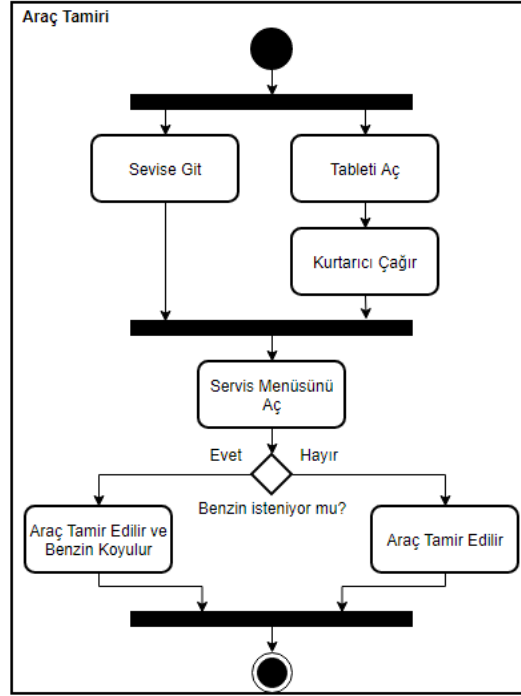
Şekil 2.1.1. Görev Alma Activity Diagram

Şekil 2.1.2’de oyun tabletinden araç alımını gösterir. Araçlar, araç veri tabanından getirilir ve listelenir. Alınan araç garaja gönderilir.



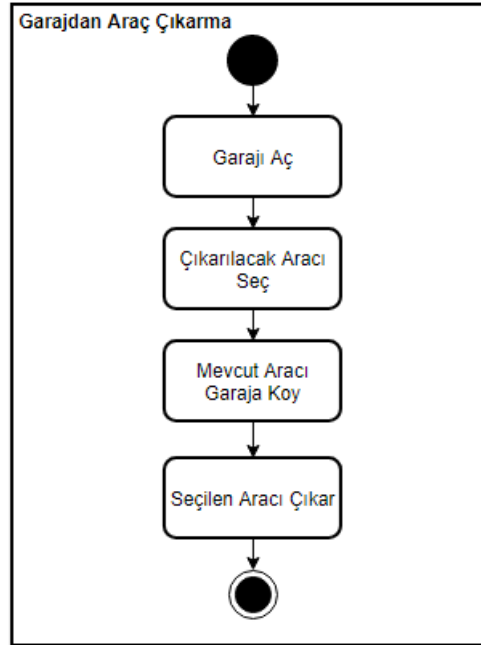
Şekil 2.1.2. Araç Marketi Activity Diagram

Şekil 2.1.3’de aracı tamir etme yollarını göstermiştir. Kurtarıcı çağrılarak veya direk servise giderek araç tamir edilebilir.



Şekil 2.1.3. Araç Tamir Activity Diagram

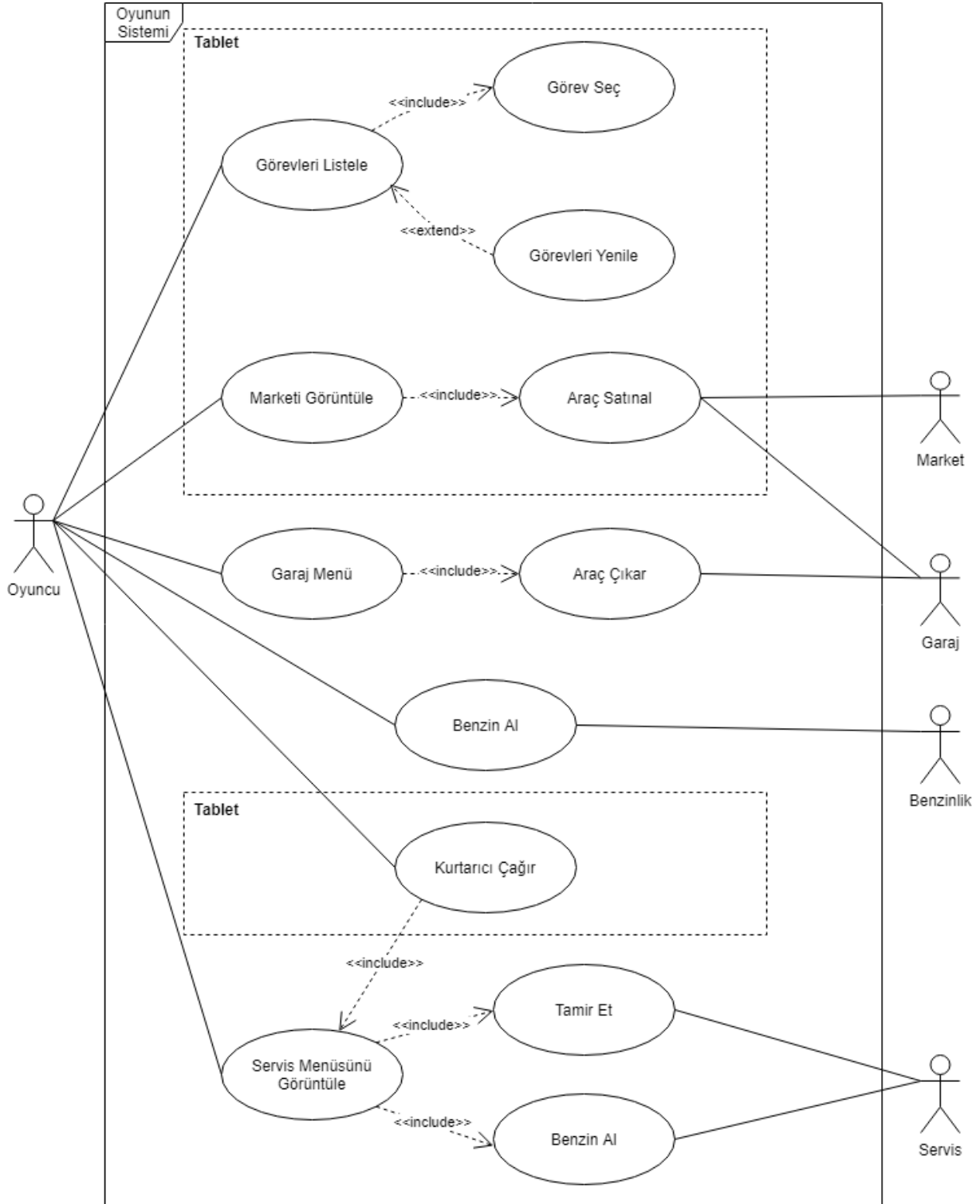
Şekil 2.1.4’de garajdan araç çıkarmayı göstermiştir.



Şekil 2.1.4. Garaj Activity Diagram

2.2. Use Case Diagram

Şekil 2.2.1’de garaj menüye otoparka giderek, benzin ala benzinlikte ve servis menüye servise giderek erişilebilir.

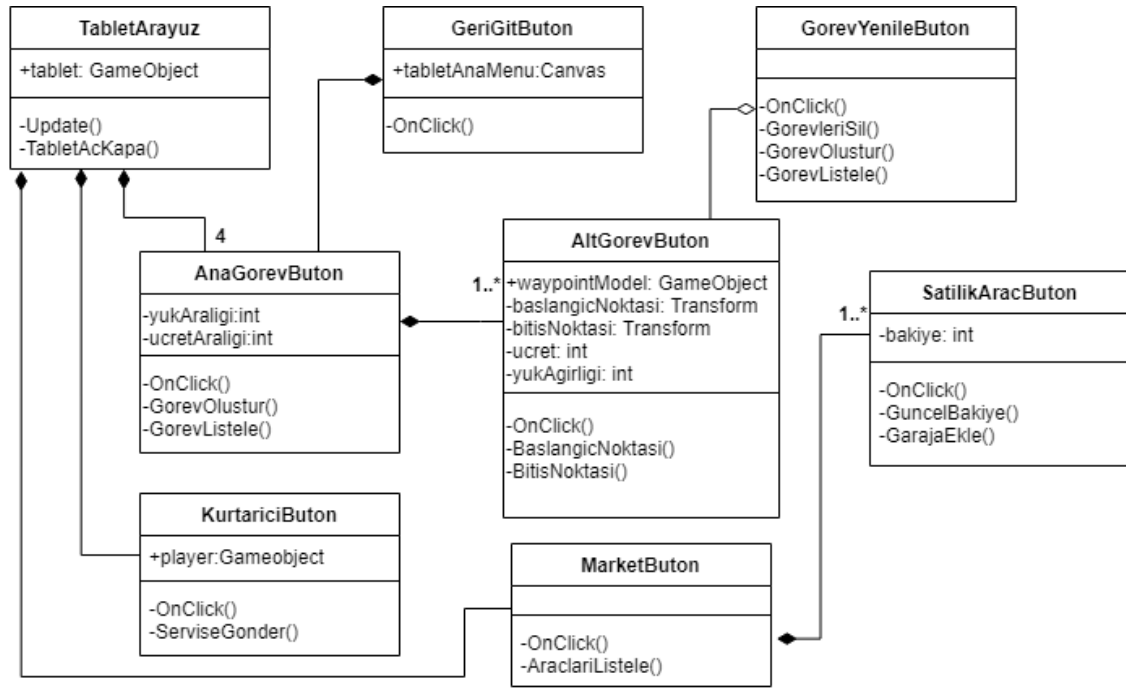


Şekil 2.2.1. Use Case Diagram

3. YAPISAL TASARIM

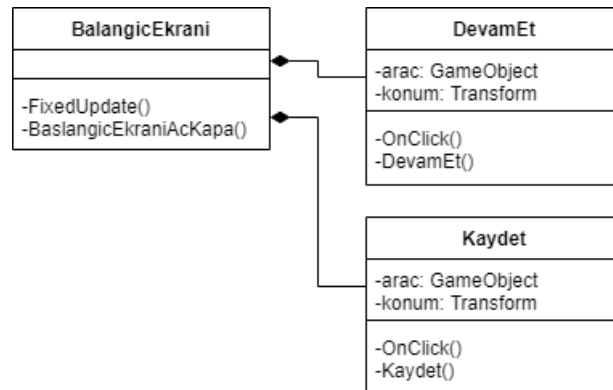
3.1. Class Diagram

TabletArayuz sınıfı tableti açıp kapar. 4 tane AnaGorevButon'u kapsayan görev butonu vardır. Ana görev seçildikten sonra GorevOlustur() fonksiyonu görevler oluşturur ve GorevListele() görevleri listeler. İstenilen görev yoksa görev yenileme butonuna tıklattılır GorevleriSil() görevleri siler ve GorevOlustur() yeni görevler üretir. İstenilen görev seçildiğinde görev alınmış olur. Tablet arayüzüne geri dönmek için GeriGitButon kullanılır. KurtarıciButon aracı servise gönderir ve kurtarıcı ücreti alır. MarketButon veri tabanından oyuncuda olmayan araçları listeler. Oyuncu bir araç aldığı anda ise araç ve oyuncu veri tabanı güncellenir.



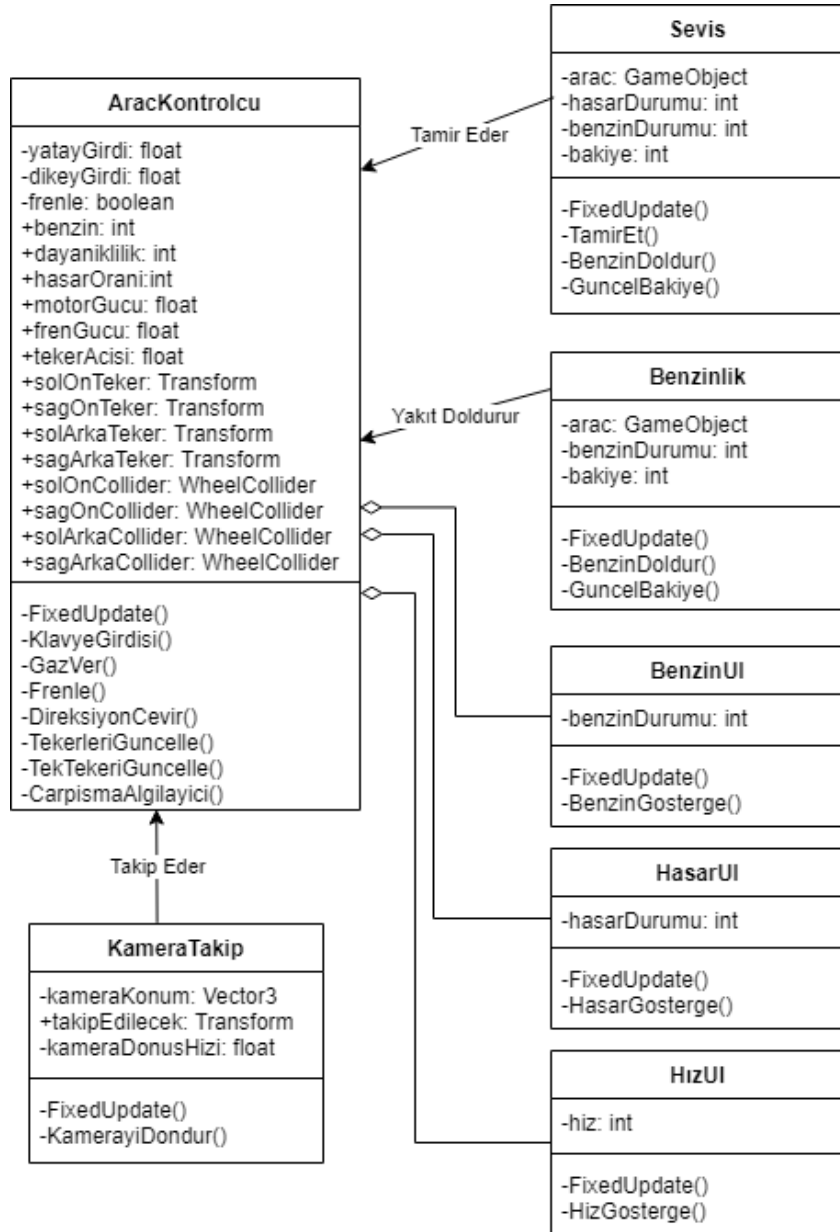
Şekil 3.1.1. Class Diagram-Tablet

BaslangicEkranı classı başlangıç ekranını açıp kapatır. Devam Et butonuna tıklattıldığında oyuncunun konum bilgisi oyuncu veri tabanından, kullandığı araç ise araç veri tabanından alınır. Sonra araç kaydedilen en son konuma bırakılır. Kaydet butonuna tıklattıldığında oyuncunun o anki konumu oyuncu veri tabanında güncellenir.



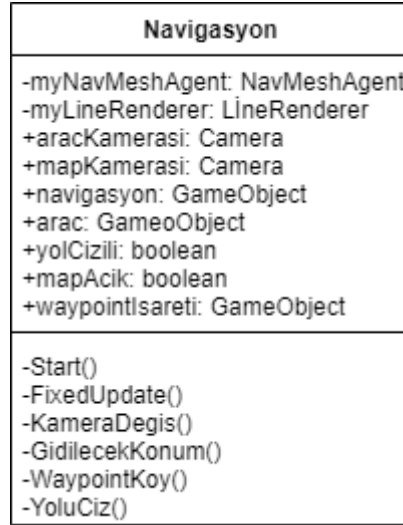
Şekil 3.1.2. Class Diagram-Başlangıç Ekranı

AracKontrolcu sınıfı araçla ilgili herşeyi üstlenir. Aracın gelen girdilerle hareketi, hasar kontrolü, benzin durumu, hızlanması ve frenlemesi gibi işlemler bu sınıfta bulur. UI'lar AracKontrolcu'den gelen verilere göre oyuncu arayüzünde değişiklik gösterir. Benzinlik classı benzinlikteki pompalara tanımlıdır ve etkileşime girildiğinde araçtaki benzin değerini depo kapasitesine kadar artırır. Servis classı aracı tamir edecek olan servisçilerde tanımlıdır ve etkileşime girildiğinde hasar oranını sıfırlar ve istenilirse bir miktar benzin değerini artırır. KameraTakip oyuncunun sürdüğü aracı belirli bir uzaklıktan takip etmesini sağlar.



Şekil 3.1.3. Class Diagram-Araç

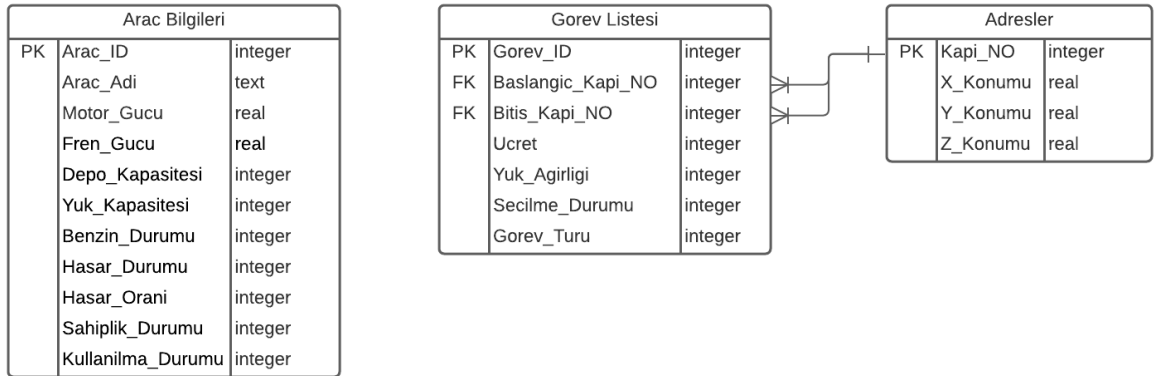
Navigasyon classı oyuncu kamerasıyla harita kamerası arasında geçiş yapar. Harita kamerasında istenilen yere sağ tıklatıldığında tıklanılan yerin konumu alınır ve waypoint işareti koyulur. YoluCiz() sonksiyonunda NavMesh rota oluşturur ve LineRenderer yolu görünür hale getirir. Eğer bir daha sağ tıklanırsa rota kaldırılır.



Şekil 3.1.4. Class Diagram-Navigasyon

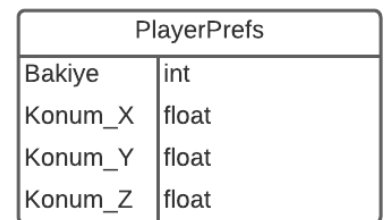
4. VERİTABANI

Projede, Şekil 4.1'deki SQLite veritabanı kullanılacaktır. Görev oluşturucu adresler ve görev listesi tablolarını kullanır. Market ve garaj sistemi de araç bilgileri tablosunu kullanır. SQLite'da veri tipleri integer → int, text → string, real → float olarak kullanılır.



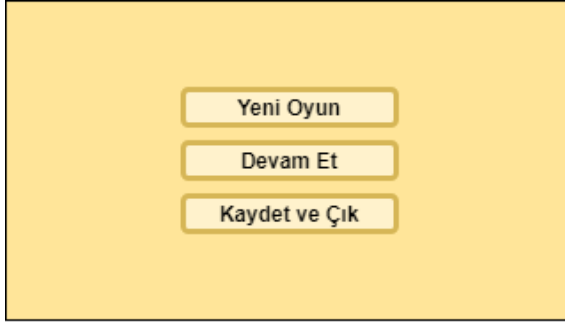
Şekil 4.1. SQLite Veritabanı

Oyuncunun bakiye ve konum bilgileri tek satırlık bir bilgi olacağı için Unity oyun motorunun depolaması olan PlayerPrefs (Şekil 4.2) kullanılacaktır.

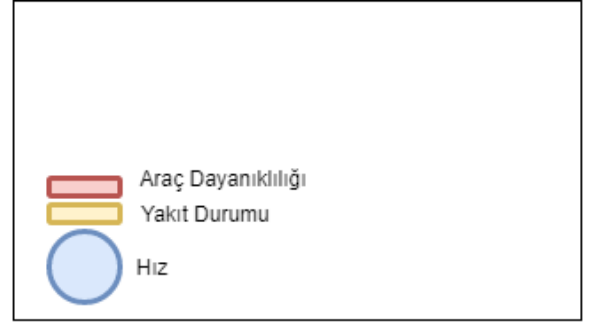


Şekil 4.2. PlayerPrefs

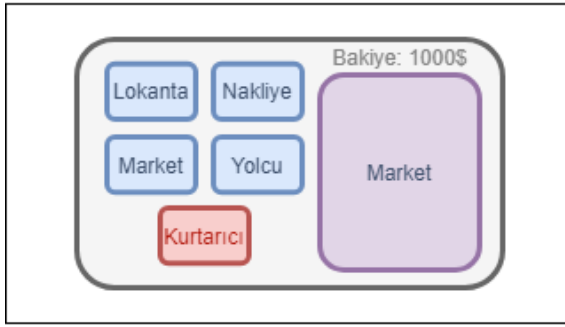
5. Arayüz Tasarımı



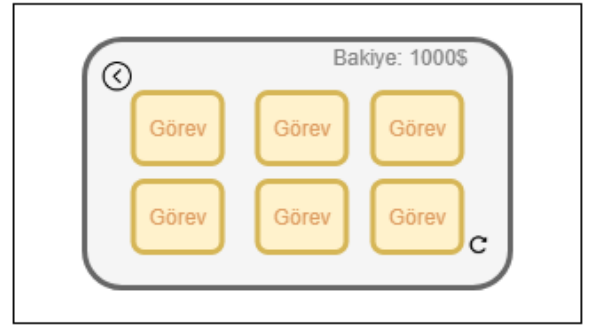
Şekil 5.1. Başlangıç Ekranı



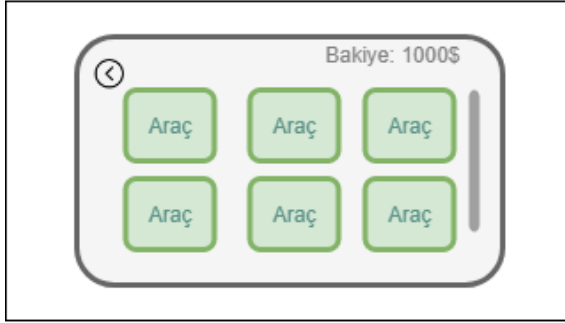
Şekil 5.2. Oyun Arayüzü



Şekil 5.3. Oyun Tableti - Anamenü



Şekil 5.4. Oyun Tableti - Görev Listesi



Şekil 5.5. Oyun Tableti - Araç Mağazası



Şekil 5.6. Servis Menüsü

6. Demo Videosu

https://drive.google.com/file/d/1_uz7qhFA7TkN3j3IFuoatrq-lx_q9E1h/view?usp=sharing