

1. GİRİŞ.....	2
1.1. Proje Genel İstatistikleri	2
1.1.1. Veri Tabanı.....	2
1.1.2. Yardımcı Programlar	2
1.1.3. Web Tabanlı Oyun	2
2. PROJENİN TANIMI VE PLANI.....	3
1. Projenin Amacı	3
2. Projenin Tanımı	3
3. Projenin Kapsamı	4
4. Projeye İlişkin Tahminler	5
5. Risk Yönetimi.....	6
6. Zamanlama	7
7. Proje Kaynakları	7
8. Proje Modülleri ve Görev Paylaşımı	7
3. KURAMSAL BİLGİLER	8
1. Yapay Zekâ Ödül Yöntemi ile Öğrenme (Reinforcement Learning) Algoritması ..	8
2. En Kısa Yol Bulma Algoritması (A*).....	9
4. ANALİZ VE MODELLEME.....	15
1. Analiz	15
2. Sistem Modeli.....	16
5. TASARIM, GERÇEKLEME VE TEST	17
1. Tasarım	17
1.1. Veritabanı Tasarımı.....	17
1.2. Yazılım Tasarımı	32
2. Yardımcı Programlar	44
2.1. Pokemon Bilgileri Veritabanı Doldurucusu.....	44
2.2. Dil Ekleme Programı	45
2.3. Harita Hazırlama Programı	46
6. DENEYSEL SONUÇLAR.....	53
7. SONUÇ VE ÖNERİLER.....	54
8. KAYNAKLAR	55

1. GİRİŞ

Web tabanlı yapay zekâ ve yol bulma algoritmaları içeren stratejiye dayalı çok oyunculu bir oyun tasarlanacaktır. Oyun dünyaca ünlü olan Pokemon oyununu temel almaktadır. Oyun mekanikleri[17] tamamen Nintendo'nun çıkarttığı orijinal pokemon oyununa dayanmaktadır.

Yapılan araştırmalar sonucunda daha önce Pokemon temel alınarak yapılmış akademik düzeyde web tabanlı oyun çalışmasına rastlanmamıştır.

Yapılan çalışmalar sonucunda Nintendo Game Boy Advance konsolu için çıkarttığı pokemon oyunlarının özelliklerini birleştiren ve 4. jenerasyon pokemon konsol oyununu temel alan online web tabanlı çoklu oyuncu desteğine sahip bir oyun yapılmıştır. Oyunda akademik düzeyde algoritmalar kullanılmıştır. Yapay zekâ ve yol bulma algoritmaları ile oyunun içeriği zenginleştirilmiştir.

1. Proje Genel İstatistikleri

1.1. Veri Tabanı

Toplam 43 adet tablo

Toplam 936 adet sütun

Herhangi bir kullanıcı bilgisi eklenmeden toplam veri tabanı boyutu 12800 KB

1.2. Yardımcı Programlar

Toplam 3 adet yardımcı program

Toplam 9000 satır kod yardımcı program

1.3. Web Tabanlı Oyun

Toplam 22 adet Sınıf

Toplam 21 adet sayfa

Toplam 45000 adet satır kod

2.PROJENİN TANIMI VE PLANI

1.Projenin Amacı

Türk oyun sektörü, son yıllarda yapılan profesyonel ve amatör oyunlarla gelişme çabası içinde, fakat bilgisayar oyunları oynama konusundaki aktifliğimizi üretme konusunda gösteremiyoruz. Bu eğlenceli hobimizi kariyerimize katmayı hedeflemekteyiz. Bu amaçla projemizde dünyaca ünlü anime filmi Pokemon'un web tabanlı oyunu yapılmıştır.

2. Projenin Tanımı

Karışık yer şekillerinin bulunduğu oyun haritalarında kullanıcının gitmek istediği yere en kısa yoldan gitmesini sağlayan en kısa yol bulma algoritmalarından yararlanılmıştır. Gerek savaşa girmeden önce gerek savaş sırasında verilecek kararlar, savaşın sonucunu doğrudan önemli bir şekilde etkilemektedir. Bu gibi çevrimiçi strateji oyunları karar verme yeteneğini, problem çözme kabiliyetini ve düşünme gücünü artırmaktadır.[1] Ayrıca savaş modüllerinde yapay zekâ algoritmalarından öğrenme (Learning)[16] kullanılarak, rakiplerin giderek güçlenmesi sağlanmış, galibiyet için kullanıcıların savaşlarda daha etkin bir strateji ile hamle yapmaları gerekli hale getirilmiştir.

Oyuncu oyuna kayıt sırasında verilen başlangıç pokemonlarından istediğini seçerek başlamakta, bu pokemonu ve yakaladığı diğer pokemonları geliştirerek oyuna devam etmektedir.

Oyundaki diğer en önemli birimler eğitmenler, vahşi pokemonlar, eşyalar, haritalar ve kullanılabilir binalardır.[15][36][14][25]

Eğitmenler, oyun boyunca haritalarda karşılaşılan bilgisayarın yönettiği karakterlerdir. Her birinin haritalarda farklı bölgeleri vardır ve o bölgeleri geçmek için bir ya da birden fazla yetenekli pokemona sahip olan bu eğitmenlerin kullanıcı tarafından mağlup edilmesi gerekmektedir. Eğitmeni mağlup eden kullanıcı, haritada ilerleme hakkını elde edebilmesinin yanı sıra tecrübe puanı ve eğitmen tarafından verilen çeşitli eşyalar ile ödüllendirilmektedir.[8]

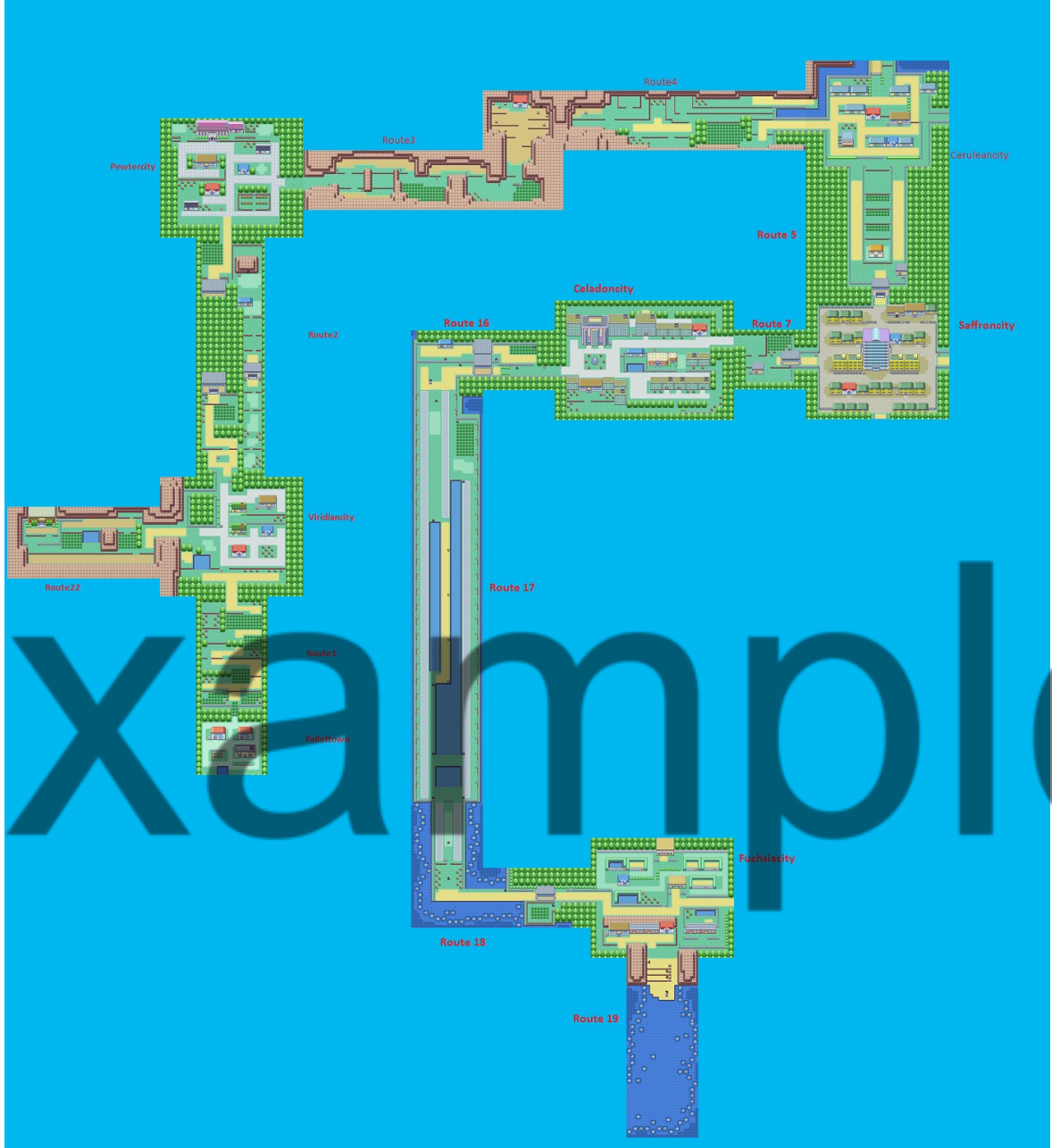


Şekil 2.1: Eğitmenler

Vahşi pokemonlar, kullanıcılar haritada ilerlen kullanıcıları durdurup, onlara saldırı yapan pokemonlardır. Vahşi olarak nitelendirilmelerinin sebebi herhangi bir eğitmenlerinin bulunmamasıdır. Kullanıcı bu vahşi pokemonlardan isterse kaçarak kurtulmaya çalışabilir isterse vahşi pokemonla savaşıp, vahşi pokemonu etkisiz hale getirebilir ya da yakalayıp onu kendi pokemonu haline getirebilir.[12]

Pokemonun hasarlarını iyileştiren eşyalardan, pokemonun evrimleşmesini sağlayan eşyalara kadar oyunda 442 farklı eşya bulunmaktadır.

Harita; oyunda birden çok harita parçasından (şehirler, geçiş bölgeleri) oluşan, kullanıcının oyun serüveni boyunca dolaştığı mekândır. Kullanıcı haritada ilerledikçe karşılaştığı her türlü engellerin de seviyeleri artmaktadır.[7]



Şekil 2.2: Kanto Haritası

Kullanılabilen binalar; pokemon ve eşya alışverişinin yapıldığı market, pokemonların tedavi edilebilip ve sırt çantası ve depo arasında transfer edildiği pokemon merkezi, eğitmen ve eğitmen liderleri ile savaşların yapıldığı spor merkezleridir. [32][30][37]

3. Projenin Kapsamı

Kullanıcılar oyunu internet bağlantısına sahip herhangi bir yerden rahatlıkla oynayabileceklerdir.

4. Projeye İlişkin Tahminler

$$FP = \text{Count Total} * [0.65 + 0.01 * CAF]$$

Tablo 2.1: Tahminler

Ölçü Parametresi	Miktar	Basit	Ortalama	Karışık	Sonuç
Kullanıcı Giriş Miktarı	2	3	4	6	6
Kullanıcı Çıkış Miktarı	2	4	5	7	10
Kullanıcı Sorgulama Miktarı	1	3	4	6	4
Dosya Miktarı	2	7	10	15	20
Ekstra Arayüz Miktarı	0	5	7	10	0
Toplam					40

CAF(Complexity Adjustment Factors):

Tablo 2.2: Faktörler

Faktör	Tahmini Değer (0...5)
1. Yedekleme ve Kurtarma gerekli ?	5
2. Veri Haberleşmesi gerekli ?	3
3. Ayırık Process Fonksiyonları gerekli ?	0
4. Performans Kritik mi ?	5
5. Mevcut Çalışan Çevrede çalışması ?	0
6. Çevrimiçi girişi birden fazla pencereli ?	3
7. Çevrimiçi girişi birden fazla pencereli ?	0
8. Dosya veya Veri Tabanı Güncelleniyor ?	2
9. Giriş, çıkış, dosya ve sorgu işlemleri kompleks ?	5
10. Uygulanan Algoritma Kompleks ?	5
11. Kod tekrar kullanılabilir mi ?	1
12. Yazılımda yükleme var mı ?	0
13. Birden fazla platformda yazılım yüklemesi var mı ?	0
14. Kullanıcı tarafından arayüz değiştirme tasarımı var mı ?	0

Total: 29

CAF 29

Miktar Toplamı 40

$$FP = \text{Count Total} * [0.65 + 0.01 * CAF]$$

$$FP = 40 * [0.65 + 0.01 * 29] = 37.6$$

$LOC = FP * 62 \text{ Loc/FP}$
 (ASP programing lang median)
 $LOC = 2331 \text{ satır kod}$
 $KLOC = 2.3$

$PM = a * KLOC^b * EAF$

$Tdev = c * pm^d$

Tablo 2.3: Özellikler

Faktör	Aralık	Tahmin EAF
<i>Ürün Özellikleri</i>		
1. Yeterli Güvenirlilik	0.75....1.40	1
2. Veri Genişliği	0.95....1.16	1
3. Ürün Kompleksliği	0.70....1.65	1
<i>Bilgisayar Özellikleri</i>		
4. Derleme Süre Kısıtlaması	1.00....1.66	1
5. Ana Depolama Kısıtlaması	1.00....1.56	1
6. Sanal Makine Döngüsü	0.87....1.30	1
7. Bilgisayar Döngü Süresi	0.87....1.15	1
<i>Kişisel Özellikler</i>		
8. Analiz Kapasitesi	1.46....0.70	1
9. Programlama Kapasitesi	1.42....0.70	1
10. Program Tecrübesi	1.29....0.82	1
11. Sanal Makine Tecrübesi	1.21....0.82	1
<i>Proje Özellikleri</i>		
12. Modern Programlama Kullanımı	1.24....0.82	1
13. Yazılım Araçları Kullanımı	1.24....0.83	1
14. Gerekli Yapım Planı	1.23....1.20	1

EAF değerleri çarpımı sonucu=3.675

$a=3.0; b=1.12; c=2.5; d=0.35;$

$PM = a * KLOC^b * EAF$

$Tdev = c * pm^d$

$PM = 3 * (2.3^{1.12}) * 1 = 7.6$

$Tdev = 2.5 * (7.6^{0.35}) = 5$

$7.6/5 = 2$ kişi gerekli

5. Risk Yönetimi

Projede mevcut riskler:

- Kullanıcıların hile yapmasını önleyecek programlama
- Hesap hırsızlığına karşı alınması gereken tedbirler
- Veritabanını efektif şekilde kullanacak programlama
- Sunucuyu efektif kullanacak programlama

6. Zamanlama

2009 PEBU Projesi

Number	Task	Resource	Start	End	Duration	2009						
						January	February	March	April	May	June	July
1	Haritayı oluşturan işleyen ve veritabanına kaydeden program.		1/2/2009	15/3/2009	30							
2	Kaynak sayfa kodunu işleyerek pokemonların özelliklerini çıkartacak ve veritabanına işleyecek program.		1/2/2009	10/3/2009	26							
3	Savaş modülü		3/3/2009	6/6/2009	69							
4	Yapay zeka entegrasi ve harita tasarımı		10/3/2009	10/6/2009	66							
5	Eşyalar,çanta,market,sağlık merkezi,depo		15/5/2009	30/6/2009	32							
6	GYM,kullanıcı,pokedex kayıt		15/6/2009	10/7/2009	19							
7	Proje Test		15/7/2009	30/7/2009	11							

Şekil 2.3: Zaman Diyagramı

7. Proje Kaynakları

Proje yazılımının geliştirilmesi için Microsoft Visual Studio 2008 SP1 aracı kullanılmıştır. Veri tabanı sistemi yönetimi için SQL Server 2008 aracı kullanılmıştır.

8. Proje Modülleri ve Görev Paylaşımı

Proje ekibinin yapısı şekildeki gibidir.

040040215 : Furkan Gözükara

- Genel Site tasarımı
- Hata kontrol ve giderme
- Savaş algoritmalarının ve sayfalarının hazırlanması
- Yapay zeka learning algortiması.
- Kullanıcı Girişi Modülü

040050244 : Osman Ünay

- Harita hazırlama programı tasarımı ve programla haritaların hazırlanması
- Kullanıcının haritada ilerlediği oyun sayfası.
- En kısa yol bulma algoritmasının gerçekleştirilmesi.
- Mesaj kutusu modülü
- Market Modülü
- Pokemon Merkezi Modülü
- Pokedex Modülü
- Kullanıcı Profili Modülü

3.KURAMSAL BİLGİLER

1.Yapay Zekâ Ödül Yöntemi ile Öğrenme (Reinforcement Learning) Algoritması

Projede bilgisayarın yönettiği birimler için öğrenme modülü tasarlanmıştır. Bunun için kullanılan algoritma, ödül yöntemi ile öğrenme (Reinforcement Learning)'dir. Bu algoritma, bilgisayarın yönettiği birimin başarı, kayıp, ödül ve cezasından sonuçlar çıkarmaktadır.[16]

Bu algoritmanın en önemli avantajlarından biri oldukça kısıtlı koşullarda dahi önemli sonuçlar vermesidir. Herhangi bir örneğin verilmediği, herhangi bir eğitmenin olmadığı durumlarda ödül yöntemi ile öğrenme (Reinforcement Learning) algoritması başarıyla çalışabilmektedir. Bu algoritmada bilgisayarın yönettiği birimler deneyerek bir sonuca ve yöneme ulaşır. Değerlendirilen sonucun yarar ya da zarar miktarına göre bu sonuca ödül ya da ceza puanı verilir. Ödül ya da cezaların verilme sıklığı bilgisayarın yönettiği birimin çalışma ortamına göre değişmektedir. Satranç gibi bir oyunda sadece en son kazandı veya kazanmadı gibi bir sonuç geri beslemesi yapılabilirken başka bir oyunda her hamle sonunda doğru veya yanlış gibi bir geri besleme yapılarak ödül veya ceza verilebilir.[16]

Bilgisayarın yönettiği birim, yapacağı yeni hareketlerin sonucunu önceki yaptığı hareketlerin sonuçlarına bakarak tahmin edebilir. Yeni bir hamlesi sonucu başarısız olursa, bu hamlenin de puanı azalır. Ödül ve ceza yöntemli öğrenim fazla hesaplamalar içermemesinden dolayı çok karışık durumlarda dahi oldukça hızlı sonuçlar alabilir.[16]

Ödül ve ceza yöntemli öğrenimde çalışma ortamının bilinmesinin de fazla önemi yoktur, önemli olan geçmişte yapmış olduğu hareket dizileri ve bunların sonuçlarıdır. Bilgisayar tarafından yönetilen birim bu sonuçlardan en iyinin seçilmesine yönelik hareketlerde bulunur. Özellikleri ve işlevi bakımından ödül ve ceza yöntemli öğrenim (Reinforcement Learning) algoritması yapay zekâ biliminin minyatür hali olarak düşünülebilir.[16]

Bu algoritma, projemizde bilgisayarın yönettiği pokemonlar için kullanılmıştır. Projemizde 493 farklı pokemon, 467 farklı saldırı vardır.[14][24] Her pokemon 100 farklı saldırı öğrenebilme yeteneğine sahip olsa da aynı anda en fazla 4 farklı saldırıya sahip olabilmektedir. Bir pokemonun 100 farklı seviyede olabileceği de düşünüldüğünde ortaya çıkabilecek durum sayısı çok fazladır. Ödül ve ceza yöntemi ile öğrenmede saldırı stratejilerinden en iyilerinin seçilmesi hedeflenmiştir. Algoritmanın temelini oluşturan rastgele deneme yöntemi projemizde %50 oran ile belirlenmiştir. Yani bilgisayar %50 oran ile rastgele saldırılar yapacak, %50 oran ile de daha önce yapılmış en iyi savaş stratejisini uygulayacaktır. Rastgele yapılan her savaştaki saldırılar sırası ile veri tabanına kaydedilmektedir. Saldırıların sonuçları birbirlerini etkilediği için saldırı sırası oldukça önemlidir. Ayrıca karşı karşıya gelen pokemonların özellikleri de veri tabanına kaydedilir. Böylece daha önce bu özelliklere sahip pokemonların savaşıp savaşmadığı bilgisi ve savaş sonuçlarına istenildiğinde ulaşılabilir. Projemizde ödül rakip pokemona verilen toplam hasar miktarıdır. Yani rakip pokemona en çok hasar veren saldırı listesi en iyi ödüle sahip olmuş olacak ve bilgisayar tarafından bu savaştaki saldırılar uygulanacaktır. Toplamda daha az hasar vermiş saldırı zinciri cezalandırılacak ve bir sonraki savaşta kullanılmayacaktır. Bu şekilde yapılan savaş sayıları arttıkça, bilgisayarın yönettiği birimlerin hareketleri en mükemmele doğru gider.

2. En Kısa Yol Bulma Algoritması (A*)

Yol bulma algoritmaları harita üzerinde bir noktadan başka bir noktaya yol bulmaya yarar. En çok kullanıldığı ortamlardan biri gerçek zaman stratejili oyunlardır. Bu oyunlarda oyuncuların bir yere ulaşması için engellerle dolu haritada engellerin etrafından dolaşması gerekebilmektedir. Oyunların ve oyun haritalarının günümüzde daha fazla karmaşık hale gelmesinden dolayı yol bulma algoritmalarının önemi gittikçe artmaktadır.

Projemizde çit, ağaç, su, dağ gibi engellerle dolu haritamızda, kullanıcının istediği bir noktaya en kısa yoldan ulaşmasını sağlamak için yol bulma algoritmalarının en yaygınlarından biri olan A* algoritması gerçekleştirilmiştir. Projede, bu algoritmanın birbirinden farklı yer şekillerine sahip haritalarda efektif bir şekilde uygulanabilmesi için, haritalar önce bölümlere ayrılmıştır, daha sonra bu bölümler koordinatlara ayrılmıştır. Haritada ulaşımın serbest olduğu yollar ve engeller birim karelerden oluşacaktır. Bu kareler algoritmamızda düğümler olarak temsil edilecektir. Kullanıcı eğer etrafında herhangi bir engel yoksa sola, sağa, yukarı ve aşağı olmak üzere 4 yöne hareket edebilmektedir ve her hareketinin maliyeti 1 birimdir. A* algoritmasında bir noktadan hedefe ulaşırken bir sonraki noktanın seçiminde sezgisel yöntem (heuristic) kullanılır. Sezgisel yöntem (heuristic) çeşitli şekillerde tanımlanabilmektedir. Uygulamamızda sezgisel yöntem (heuristic) Manhattan Metodu olarak bilinen şekilde tanımlanmıştır.[6] Bu metotta, ilgili düğümün sezgisel yöntem (heuristic) değeri, o düğümden bitiş düğümüne arada hiç engel olmadığı düşünülerek bulunan uzaklık değeridir.

Algoritmada yol bulurken düğümlerin toplandığı “Açık” ve “Kapalı” olmak üzere iki türlü liste vardır. Açık listede, yol maliyeti en az bir yönden hesaplanan ama maliyetinin hesaplanması henüz tamamlanmamış düğümler tutulur. Kapalı listede ise, yol maliyetinin hesaplanması tamamlanmış düğümler tutulur. Bir düğümün yol maliyeti $F = G + H$ şeklinde hesaplanır. Burada H daha önce de belirtildiği şekilde ilgili düğümün sezgisel yöntem (heuristic) değeridir. G ise ilgili düğümün başlangıç düğümüne olan uzaklık değeridir. Başlangıç düğümünden ilgili düğüme gelirken, ilgili düğümden bir önceki düğüm ilgili düğümün ebeveyn düğümüdür. Dolayısı ile ilgili düğümün G değeri, ebeveyn düğümün G değerine ebeveyn düğümden ilgili düğüme olan hareket maliyeti eklenerek bulunur. Başlangıç düğümünün G değeri 0’dır.

Yol bulunurken öncelikle başlangıç düğümü Kapalı listeye alınır. Daha sonra kapalı listeye alınan bu düğümün G’sine bir birim daha yol maliyeti eklenerek aralarında engel bulunmayan ve kapalı listede olmayan (başlangıçta hiç biri değildir) tüm komşuları için bir G değeri hesaplanır. Komşularından Açık listede olmayanların ebeveyn düğümü bu ilgili düğüm, G’si bu hesaplanan G olarak ayarlanır ve bu komşular Açık listeye eklenir. Komşulardan Açık listede bulunanların önceki G değeri ile yeni hesaplanan G karşılaştırılır. Komşunun yeni hesaplanan G değeri önceki G değerinden küçük ise komşunun G’si yeni hesaplanan G, ebeveyn düğümü de ilgili düğüm olarak güncelleme yapılır. Daha sonra Açık listeden en küçük F değerine sahip düğüm seçilir, Kapalı listeye taşınır, kapalı listeye taşınan düğümün komşuları kontrol edilerek aynı işlemlere hedef düğüme ulaşınca kadar devam edilir. Hedef düğüme ulaşıncaya kadar her düğüm kendisinin ebeveyn düğüm bilgisini tuttuğu için hedeften geri gelinerek yol bulunmuş olur.[4]

Algoritmanın açıklamasında bahsedilen açık ve kapalı listelerden kapalı liste için bir veri yapısı kullanılmamıştır. Çünkü kapalı listeye yerleştirilen düğümün bütün kontrolleri bitmiştir ve bir daha kullanılmayacaktır. Komşuları kontrol edilecek yeni düğüm önceden açık listeye yerleştirilmiş düğümler arasından seçilir. Açık listeden düğüm seçimi

yapılırken açık listedeki düğümlerden yol maliyeti en küçük olan seçilir. Her seçim, en küçük yol maliyetine sahip düğümün bulunacağı bir arama yapmayı gerektirecektir. Bu, işlem maliyetini artıracığı için, listeye her düğüm eklenişinde eklenen düğümün listede düğümlerin F değerine göre bir sıralamada yerleşmesini sağlayan öncelik kuyruğu kullanılmıştır.[2] Bu kuyruğun başında her zaman F değeri en küçük düğüm bulunur. Bu sayede açık listeden bir düğüm alınacağında kuyruğun başındaki düğüm alınır. Bir örnekle A* algoritmasının üzerinde su pokemonu taşımayan bir kullanıcı için çalıştığını düşünerek küçük bir harita parçası üzerinde adım adım çalışmasını incelenirse:[3]

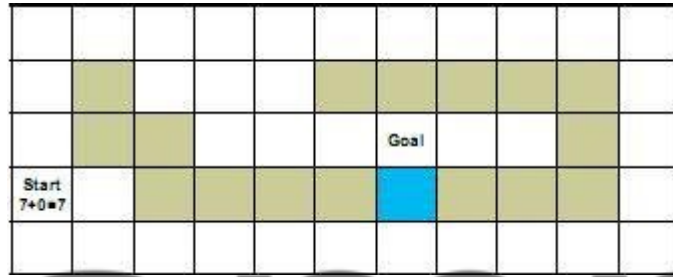
Beyaz kutu açık yolu temsil etmektedir.

Kahverengi kutu kapalı yolu temsil etmektedir.

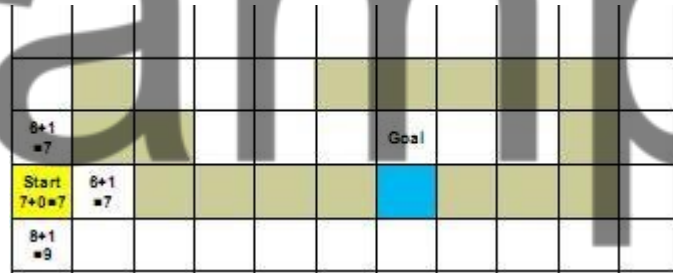
Mavi kutu suyu temsil etmektedir.

Sarı kutular kapalı listeye atılmış bulunan düğümlerdir.

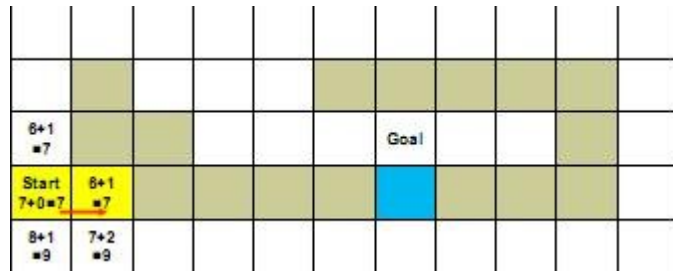
Yeşil kutular algoritma sonucunda bulunan yolu göstermektedir.



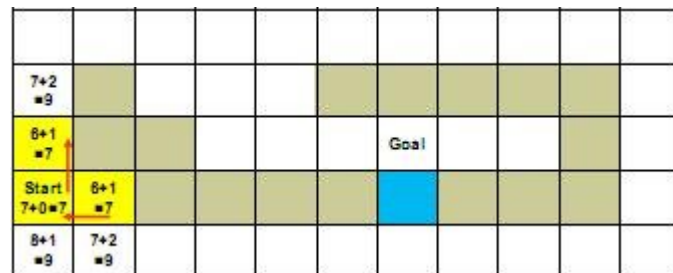
Şekil 3.1: A* Durum 1



Şekil 3.2: A* Durum 2



Şekil 3.3: A* Durum 3



Şekil 3.4: A* Durum 4

8+3 =11										
7+2 =9										
6+1 =7						Goal				
Start 7+0=7	6+1 =7									
8+1 =9	7+2 =9									

Şekil 3.5: A* Durum 5

8+3 =11										
7+2 =9										
6+1 =7						Goal				
Start 7+0=7	6+1 =7									
8+1 =9	7+2 =9									

Şekil 3.6: A* Durum 6

8+3 =11										
7+2 =9										
6+1 =7						Goal				
Start 7+0=7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.7: A* Durum 7

8+3 =11	7+4 =11									
7+2 =9										
6+1 =7						Goal				
Start 7+0=7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.8: A* Durum 8

8+3 =11	7+4 =11	6+5 =11								
7+2 =9										
6+1 =7						Goal				
Start 7+0=7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.9: A* Durum 9

8+3 =11	7+4 =11	6+5 =11	5+6 =11							
7+2 =9		5+6 =11								
6+1 =7						Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.10: A* Durum 10

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11						
7+2 =9		5+6 =11	4+7 =11							
6+1 =7						Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.11: A* Durum 11

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11					
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7						Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.12: A* Durum 12

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11				
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7						Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.13: A* Durum 13

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11	3+10 =13			
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7						Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.14: A* Durum 14

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11	3+10 =13			
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7				2+9 =11		Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.15: A* Durum 15

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11	3+10 =13			
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7			3+10 =13	2+9 =11	1+10 =11	Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.16: A* Durum 16

8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11	3+10 =13			
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7			3+10 =13	2+9 =11	1+10 =11	Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.17: A* Durum 17

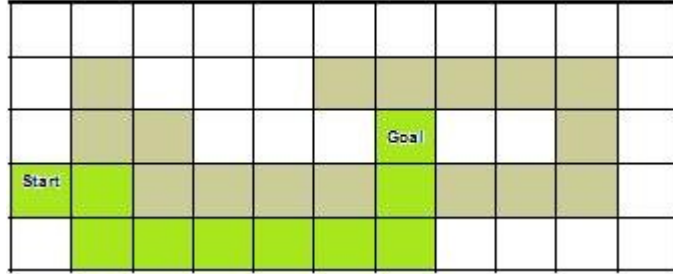
8+3 =11	7+4 =11	6+5 =11	5+6 =11	4+7 =11	3+8 =11	2+9 =11	3+10 =13			
7+2 =9		5+6 =11	4+7 =11	3+8 =11						
6+1 =7			3+10 =13	2+9 =11	1+10 =11	Goal				
Start 7+0 =7	6+1 =7									
8+1 =9	7+2 =9	6+3 =9	5+4 =9	4+5 =9	3+6 =9	2+7 =9	3+8 =11	4+9 =13		

Şekil 3.18: A* Durum 18

Şekil 3.19: A* Durum 19

Görüldüğü gibi A* algoritması kullandığı “heuristic”e bağlı olarak önce haritanın aşağı kısmında yol aramış, engeller dolayısı ile aşağı taraftan yol bulamamış, aramasına haritanın üst tarafından devam ederek, kullanıcıya uygun bir yol bulmuştur. Bulunan yol en kısa yoldur ki A* algoritması eğer bir yol varsa en kısa yolu bulacağını garantilemektedir.

Kullanıcı su pokemonu taşımadığı için haritadaki “su” da kullanıcı için bir engel teşkil etmiştir. Eğer kullanıcı bir su pokemonuna sahip olsaydı, haritadaki “su”yun kullanıcı için normal bir yoldan farkı kalmayacak ve A* algoritması da ona göre çalışıp, kullanıcı için aşağıdaki yolu bulacaktı.



Şekil 3.20: A* Durum 20

Example

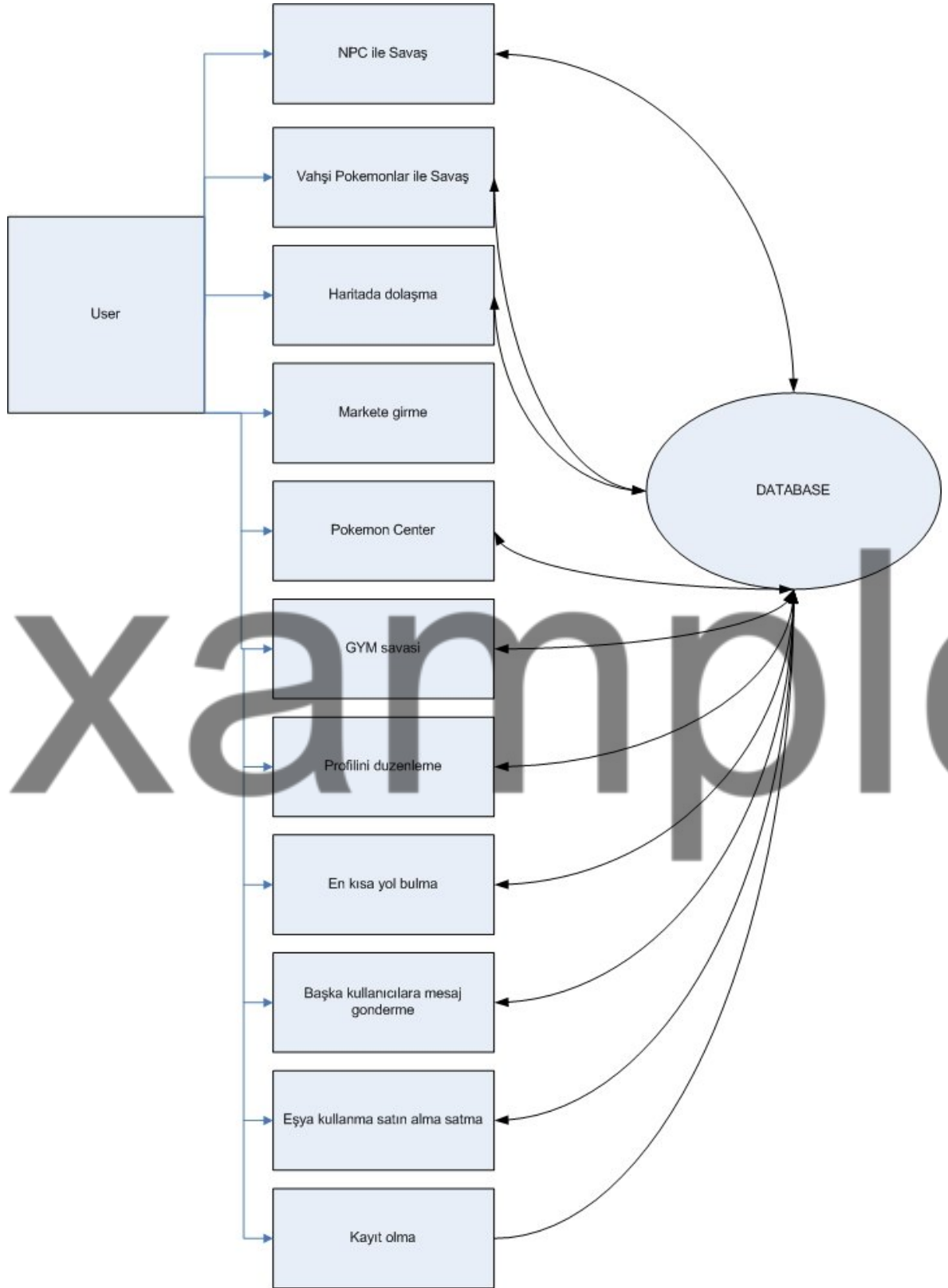
4. ANALİZ VE MODELLEME

1. Analiz

Projemiz kapsamındaki oyun, günümüz popüler web tabanlı birçok oyundan farklı olarak daha büyük bir veritabanı kullanmaktadır, ayrıca yapay zekâ algoritmaları içermesinden dolayı daha fazla işlem gücüne ihtiyaç duymaktadır. Yüzlerce kişinin aynı anda oynayacağı oyunda, sunucunun kullanıcıya kabul edilebilir sürelerde yanıt verebilmesi için algoritmalarda oyuna özel iyileştirmeler yapılma gereği duyulmuştur. Oyunda yapılabilecek hilelere karşı yapılan yoğun kontroller de önemli ölçüde işlem gücü kullanmaktadır. Bu kontrollerin daha efektif kullanılabilip, sunucunun yanıt verme süresini kısaltmak için de çalışmalar yapılmıştır.

Example

2. Sistem Modeli



Şekil 4.1: Sistem Modeli

5.TASARIM, GERÇEKLEME VE TEST

1.Tasarım

Proje tasarımı “Veritabanı Tasarımı” ve “Yazılım Tasarımı” olarak iki alt bölüm altında açıklanacaktır.

1.1. Veritabanı Tasarımı

Projemizde gerçekleştirilen oyun, Pokemon serisinin orijinal senaryosuna mümkün olduğunca bağlı kalınarak hazırlanmasından dolayı, oldukça fazla modül ve bu modüller için oldukça fazlaya veriye ihtiyaç duymaktadır. Veritabanı tabloları ve kısa açıklamaları alfabetik sırada aşağıda verilmiştir.

Tablo 1’deki tabloda eşyaların kullanılmasından sonra kazanılan ek özellikler geçici olarak tutulmaktadır ki özelliklerin gerçekte tutulduğu tablodaki fonksiyon kullanımı sonucu oluşan değişikliklerden etkilenmesinler.

tblAdditionalStats			
	Column Name	Data Type	Allow Nulls
	UsersPokemonsId	int	☐
	MaximumHp	int	☐
	Attack	int	☐
	Defense	int	☐
	SpAttack	int	☐
	SpDefense	int	☐
	Speed	int	☐

Şekil 5.1: Tablo 1

Tablo 2’de öğrenme (Learning) modülünün ihtiyaç duyduğu bilgiler tutulmaktadır.

tblAi			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	BattleInfoId	int	☐
	UserId	int	☐
	CurrentUsersPokemon...	int	☐
	CurrentBattleMoment...	int	☐
	PokemonIdHuman	int	☐
	PokemonIdWild	int	☐
	PokemonLevelHuman	int	☐
	PokemonLevelWild	int	☐
	AttackId1	int	☐
	AttackDamage1	int	☐
	AttackId2	int	☐
	AttackDamage2	int	☐
	AttackId3	int	☐

Şekil 5.2: Tablo 2

Tablo 3’te web sitesinde yapılan duyurular tutulmaktadır.

tblAnnouncements			
	Column Name	Data Type	Allow Nulls
🔑	AnnouncementId	int	☐
	AnnouncementTitle	nvarchar(200)	☐
	AnnouncementBody	nvarchar(4000)	☐
	AnnouncementDate	nvarchar(200)	☐
	AnnouncementMakerId	int	☐

Şekil 5.3: Tablo 3

Tablo 4’te harita parçaları arasında en kısa yol bulunurken, hedefe ulaşmak için gidilmesi gereken bir sonraki harita parçasının bilgisi tutulmaktadır.

tblAStarBetweenRoutes			
	Column Name	Data Type	Allow Nulls
🔑	mapId	int	☐
🔑	startRouteId	int	☐
	nextRouteId	int	☐
🔑	targetRouteId	int	☐
🔑	exitSquareId	int	☐

Şekil 5.4: Tablo 4

Tablo 5'te pokemonların sahip olabilecekleri tüm saldırıların listesi ve bu saldırıların özellikleri tutmaktadır.

tblAttacks			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	AttackName	nvarchar(200)	☐
	TmHm	nvarchar(200)	☐
	AttackType	nvarchar(200)	☐
	AttackCategory	nvarchar(200)	☐
	BasePower	int	☐
	Accuracy	int	☐
	EffectAccuracy	int	☐
	Pp	int	☐
	MaxPp	int	☐
	AttackDefinition	nvarchar(200)	☐
	Affects	nvarchar(200)	☐
	Priority	int	☐
	Sound	nvarchar(200)	☐
	PhysicalContact	nvarchar(200)	☐

Şekil 5.5: Tablo 5

Tablo 6'da vahşi pokemonlar ve eğitmenlerle yapılan savaşların bilgisi tutulmaktadır.

tblBattleInfo			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	UserId	int	☐
	CurrentUsersPokemon...	int	☐
	CurrentBattleMoment...	int	☐
	BattleMomentWildPok...	int	☐
	BattleMomentWildPok...	int	☐
	BattleMomentWildPok...	int	☐
	BattleMomentWildPok...	int	☐
	BattleMomentWildPok...	int	☐
	BattleMomentWildPok...	int	☐
	TurnNumber	int	☐
	UsedUsersPokemonsId1	int	☐
	UsedUsersPokemonsId2	int	☐

Şekil 5.6: Tablo 6

Tablo 7'de savaş sırasında kullanılan bilgiler tutulmaktadır. Pokemona özel tüm bilgiler, savaş boyunca başına gelen tüm olaylar (zehirlenme, donma sağlık puanı gibi...) bu tabloda saklanmaktadır. Her savaş bitiminde o kullanıcıya ait satırlar tablodan silinir.

tblBattleMomentWildPokemon			
	Column Name	Data Type	Allow Nulls
	Id	int	☐
	UserId	int	☐
	PokemonPlace	nvarchar(200)	☐
	PokemonPrice	int	☐
	PokemonId	int	☐
	PokemonLevel	int	☐
	PokemonExp	int	☐
	BattleCount	int	☐
	BattleVictoryCount	int	☐
	MaximumHp	int	☐
	DvPointHp	int	☐
	EvPointHp	int	☐
	CurrentHp	int	☐
	Attack	int	☐
	DvPointAttack	int	☐
	EvPointAttack	int	☐
	Defense	int	☐
	DvPointDefense	int	☐
	EvPointDefense	int	☐
	SpAttack	int	☐
	DvPointSpAttack	int	☐
	EvPointSpAttack	int	☐
	SpDefense	int	☐
	DvPointSpDefense	int	☐
	EvPointSpDefense	int	☐
	Speed	int	☐
	DvPointSpeed	int	☐
	EvPointSpeed	int	☐
	HiddenPower	int	☐
	HiddenPowerType	nvarchar(200)	☐
	OnlyAllowedAttack	nvarchar(200)	☐
	IsResidualDamageTaken	nvarchar(200)	☐
	SandstormLeftTurnCo...	int	☐
	ReflectLeftTurnCount	int	☐
	LightScreenLeftTurnC...	int	☐

Şekil 5.7: Tablo 7

Tablo 8’de kullanıcıların özel mesaj alma konusunda engellediği diğer kullanıcıların listesi tutulur.

tblBlackWhiteList			
	Column Name	Data Type	Allow Nulls
	ListMakerUserId	int	☐
	ListedUserId	int	☐
	ListState	nvarchar(200)	☐
			☐

Şekil 5.8: Tablo 8

Tablo 9’da haritada seyahat ederken karşılaşılan binalardan içine girilebilir olanlar ve bu binaların konumları tutulmaktadır.

tblBuildingsToEnter			
	Column Name	Data Type	Allow Nulls
	buildingId	int	☐
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	buildingType	nvarchar(50)	☐
	buildingName	nvarchar(200)	☐
			☐

Şekil 5.9: Tablo 9

Tablo 10’da kullanıcının en kısa yol bulunmasını istediği binalar ve bu binaların konumları tutulmaktadır.

tblBuildingsToShortPath			
	Column Name	Data Type	Allow Nulls
	buildingId	int	☐
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	buildingName	nvarchar(200)	☐
			☐

Şekil 5.10: Tablo 10

Tablo 11’de bir harita parçasından diğer harita parçasına geçiş noktalarının bilgisi tutulur.

tblCrossingPointsBetweenRoutes			
	Column Name	Data Type	Allow Nulls
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	routeIdNext	int	☐
	squareIdNext	int	☐
			☐

Şekil 5.11: Tablo 11

Tablo 12’de pokemonların seviyelerinin artması için ihtiyacı olan tecrübe puanları tutulmaktadır. Bu puanlar ilk seferde fonksiyon ile bir defaya mahsus olarak hazırlanır, tabloya kaydedilir, daha sonra her seferinde tablodan kullanılır.

tblExpPerLevel			
	Column Name	Data Type	Allow Nulls
🔑	PokemonLevel	int	☐
🔑	Type	nvarchar(200)	☐
	RequiredExp	int	☐
			☐

Şekil 5.12: Tablo 12

Tablo 13’te harita parçalarında kullanıcının karşılaşılabileceği vahşi pokemonların bilgilerini tutmaktadır. Bu bilgiler günün hangi zamanında çıkacağı, hangi tür yer şeklinde çıkacağı, hangi oranda ve hangi seviyelerde çıkacaklarıdır.

tblFoundPokemonsOnRoutes			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	mapId	int	☐
	routeId	int	☐
	type	nvarchar(200)	☐
	dayPeriod	nvarchar(50)	☒
	pokemonId	int	☐
	minLevel	int	☐
	maxLevel	int	☐
	rate	int	☐
			☐

Şekil 5.13: Tablo 13

Tablo 14’te oyundaki eşyaların listesi ve özellikleri tutulmaktadır. Oyun içinde kullanılabilecek bütün eşyalar bu tabloda saklanmaktadır.

tblItems			
	Column Name	Data Type	Allow Nulls
	ItemId	int	☐
	ItemName	nvarchar(200)	☐
	Type	nvarchar(200)	☐
	ItemDescription	nvarchar(4000)	☐
	ItemLocationExplanation	nvarchar(4000)	☐
	MapId	int	☐
	RouteId	int	☐
	SquareId	int	☐
	Dowsing	nvarchar(4000)	☐
	PickUp	nvarchar(4000)	☐
	Fling	int	☐
	NatGift	int	☐
	Price	int	☐
	ImageLocation	nvarchar(4000)	☐
	Move	nvarchar(200)	☐
			☐

Şekil 5.14: Tablo 14

Kullanıcı haritada bazı noktalara geldiğinde, oradan başka noktalara ışınlanmaktadır. Bu noktaların bilgileri Tablo 15’te tutulmaktadır.

tblJumpPointsInRoutes			
	Column Name	Data Type	Allow Nulls
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	routeIdNext	int	☐
	squareIdNext	int	☐
			☐

Şekil 5.15: Tablo 15

Tablo 16’da oyunda kullanıcıya gösterilebilecek bütün yazılar tutulmaktadır. Böyle bir yapı sayesinde oyuna istenildiği kadar dil seçeneği rahatlıkla eklenebilecektir.

tblLanguages			
	Column Name	Data Type	Allow Nulls
	KeyOfSentence	nvarchar(200)	☐
	SentencePageAdress	nvarchar(200)	☐
	TurkishSentence	nvarchar(4000)	☒
	EnglishSentence	nvarchar(4000)	☒
			☐

Şekil 5.16: Tablo 16

Tablo 17’de pokemonların sadece eşyalar ile öğrenebilecekleri saldırılar tutulmaktadır. Bir pokemon, eşya ile 100’den fazla farklı saldırı öğrenebilmektedir.

tblMachineAttacks			
	Column Name	Data Type	Allow Nulls
?	PokemonId	int	☐
	PokemonNational	int	☐
	Name	nvarchar(200)	☐
	Attack25	nvarchar(200)	☐
	Attack25TM	nvarchar(200)	☐
	Attack26	nvarchar(200)	☐
	Attack26TM	nvarchar(200)	☐
	Attack27	nvarchar(200)	☐
	Attack27TM	nvarchar(200)	☐
	Attack28	nvarchar(200)	☐
	Attack28TM	nvarchar(200)	☐
	Attack29	nvarchar(200)	☐
	Attack29TM	nvarchar(200)	☐
	Attack30	nvarchar(200)	☐
	Attack30TM	nvarchar(200)	☐
	Attack31	nvarchar(200)	☐
	Attack31TM	nvarchar(200)	☐
	Attack32	nvarchar(200)	☐
	Attack32TM	nvarchar(200)	☐

Şekil 5.17: Tablo 17

Tablo 18’de haritalar tutulmaktadır.

tblMaps			
	Column Name	Data Type	Allow Nulls
?	mapId	int	☐
	mapName	nvarchar(200)	☐
			☐

Şekil 5.18: Tablo 18

Tablo 19’da harita parçaları ve özellikleri tutulmaktadır.

tblMapSquareFormat			
	Column Name	Data Type	Allow Nulls
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	shapeType	int	☐
	barrierSide	int	☐
	npcId	int	☒
			☐

Şekil 5.19: Tablo 19

Tablo 20’de oyunun çeşitli şehirlerindeki marketlerin her birinde hangi eşyaların satıldığının bilgisi tutulmaktadır.

tblMarketHasItems			
	Column Name	Data Type	Allow Nulls
	id	int	☐
	marketId	int	☐
	itemId	int	☐
			☐

Şekil 5.20: Tablo 20

Tablo 21’de kullanıcıların birbirine gönderdiği mesajlar tutulmaktadır.

tblMessageBox			
	Column Name	Data Type	Allow Nulls
	messageId	int	☐
	messageSenderId	int	☐
	messageReceiverUserId	int	☐
	messageSubject	nvarchar(200)	☐
	messageText	nvarchar(4000)	☒
	messageReceiverRead	int	☐
	messageReceiverDelete	int	☐
	messageSenderDelete	int	☐
	messageSentDate	nvarchar(200)	☐
			☐

Şekil 5.21: Tablo 21

Tablo 22’de pokemonların seviye atlamaları sonucu öğrenecekler saldırılar ve bu saldırıların özellikleri tutulmaktadır.

tblNormalAttacks			
	Column Name	Data Type	Allow Nulls
	PokemonId	int	☐
	PokemonNational	int	☐
	Name	nvarchar(200)	☐
	Attack1	nvarchar(200)	☐
	Attack1Level	nvarchar(200)	☐
	Attack2	nvarchar(200)	☐
	Attack2Level	nvarchar(200)	☐
	Attack3	nvarchar(200)	☐
	Attack3Level	nvarchar(200)	☐
	Attack4	nvarchar(200)	☐
	Attack4Level	nvarchar(200)	☐
	Attack5	nvarchar(200)	☐
	Attack5Level	nvarchar(200)	☐
	Attack6	nvarchar(200)	☐
	Attack6Level	nvarchar(200)	☐
	Attack7	nvarchar(200)	☐
	Attack7Level	nvarchar(200)	☐
	Attack8	nvarchar(200)	☐
	Attack8Level	nvarchar(200)	☐
	Attack9	nvarchar(200)	☐
	Attack9Level	nvarchar(200)	☐
	Attack10	nvarchar(200)	☐
	Attack10Level	nvarchar(200)	☐
	Attack11	nvarchar(200)	☐

Şekil 5.22: Tablo 22

Tablo 23'te oyunda bilgisayar tarafından yönetilen oyuncuların(NPC) bilgileri tutulmaktadır.

tblNPC			
	Column Name	Data Type	Allow Nulls
	npcId	int	☐
	npcName	nvarchar(200)	☐
	npcClass	nvarchar(200)	☐
	npcWinningReward	int	☐
	npcWinningRewardItem	nvarchar(200)	☒
	npcPictureName	nvarchar(200)	☒
			☐

Şekil 5.23: Tablo 23

Tablo 24'te bilgisayar tarafından yönetilen oyuncuların(NPC) sahip oldukları pokemonlar ve bu pokemonların özellikleri tutulmaktadır.

tblNPCPokemons			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	npcId	int	☐
	pokemonId	int	☐
	pokemonGender	nvarchar(50)	☐
	pokemonLevel	int	☐
	pokemonGiveEXP	int	☐
	pokemonGiveItemId	int	☒

Şekil 5.24: Tablo 24

Tablo 25’te oyunda aktif olan kullanıcılar tutulmaktadır. Bu tablo ile oyuna aynı anda bağlanabilecek kişi sayısına limit koyulabilir.

tblOnlineUsers			
	Column Name	Data Type	Allow Nulls
🔑	userId	int	☐
	lastMovingTime	nvarchar(50)	☐
	sessionId	nvarchar(50)	☐

Şekil 5.25: Tablo 25

Tablo 26’da oyundaki bütün pokemonlar özellikleri ile birlikte tutulmaktadır.

tblPokedex			
	Column Name	Data Type	Allow Nulls
	PokemonId	int	☐
	PokemonNational	int	☐
	Name	nvarchar(200)	☐
	Type1	nvarchar(200)	☐
	Type2	nvarchar(200)	☐
	GenderMaleRatio	nvarchar(200)	☐
	GenderFemaleRatio	nvarchar(200)	☐
	EggGroup	nvarchar(200)	☐
	Height	nvarchar(200)	☐
	Weight	nvarchar(200)	☐
	Form1	nvarchar(200)	☐
	Form2	nvarchar(200)	☐
	Form3	nvarchar(200)	☐
	Form4	nvarchar(200)	☐
	Form5	nvarchar(200)	☐
	Form6	nvarchar(200)	☐
	Form7	nvarchar(200)	☐
	Form8	nvarchar(200)	☐
	Form1Name	nvarchar(200)	☐
	Form2Name	nvarchar(200)	☐

Şekil 5.26: Tablo 26

Tablo 27’de pokemonların doğalarının özellikleri tutulmaktadır.

tblPokemonNature			
	Column Name	Data Type	Allow Nulls
	Id	int	☐
	NatureName	nvarchar(200)	☐
	IncreasedStat	nvarchar(200)	☐
	DecreasedStat	nvarchar(200)	☐
	FavoriteFlavor	nvarchar(200)	☐
	DislikedFlavor	nvarchar(200)	☐
	Attack	nvarchar(200)	☒
	Defense	nvarchar(200)	☒
	Support	nvarchar(200)	☒
	Change	nvarchar(200)	☒
			☐

Şekil 5.27: Tablo 27

Tablo 28’de harita parçaları ve özellikleri tutulmaktadır.

tblRoutes			
	Column Name	Data Type	Allow Nulls
🔑	mapId	int	☐
🔑	routeId	int	☐
	routeName	nvarchar(200)	☐
	maxColumn	int	☐
	maxRow	int	☐
	defaultStartSquare	int	☒

Şekil 5.28: Tablo 28

Tablo 29’da haritada dolaşırken bulunabilecek eşyalar ve bu eşyaların bulunabileceği bölümler tutulmaktadır.

tblRoutesItems			
	Column Name	Data Type	Allow Nulls
🔑	id	int	☐
	mapId	int	☐
	routeId	int	☐
	itemId	int	☐
	method	nvarchar(200)	☐

Şekil 5.29: Tablo 29

Tablo 30’da haritada bilgisayar tarafından yönetilen oyuncuların, kullanıcıların öğrenebilecekleri saldırıların listesi tutulmaktadır.

tblRoutesMoveTutors			
	Column Name	Data Type	Allow Nulls
🔑	id	int	☐
	mapId	int	☐
	routeId	int	☐
	attackName	nvarchar(200)	☐
	attackType	nvarchar(200)	☐
	attackDefinition	nvarchar(200)	☐
	attackCost	nvarchar(200)	☐

Şekil 5.30: Tablo 30

Tablo 31’de oyundaki yeryüzü şekillerinin listesi tutulmaktadır.

tblShapes			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	Name	nvarchar(200)	☐

Şekil 5.31: Tablo 31

Tablo 32’de savaş sırasında bilgisayar tarafından yönetilen pokemonların saldırılar sonucunda değişime uğrayan yetenekleri tutulmaktadır.

tblStatModifyBattleMomentWildPokemon			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	UserId	int	☐
	BattleMomentWildPokemonId	int	☐
	ModifiedStat	nvarchar(200)	☐
	LeftTurn	int	☐
	MadeTurn	int	☐
	Effect	int	☐
	Affect	nvarchar(200)	☐
			☐

Şekil 5.32: Tablo 32

Tablo 33’te savaş sırasında kullanıcı tarafından yönetilen pokemonların saldırılar veya eşya kullanımı sonucunda değişime uğrayan yetenekleri tutulmaktadır.

tblStatModifyUsersPokemons			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	UserId	int	☐
	UsersPokemonsId	int	☐
	ModifiedStat	nvarchar(200)	☐
	LeftTurn	int	☐
	MadeTurn	int	☐
	Effect	int	☐
	Affect	nvarchar(200)	☐
			☐

Şekil 5.33: Tablo 33

Kullanıcı bir yerden başka yere gitmek için en kısa yolu görmek istediğinde, oyunda gideceği yol çizilir. Bu çizilen yolun bilgileri Tablo 34’te tutulur.

tblUsersAstarInformations			
	Column Name	Data Type	Allow Nulls
🔑	userId	int	☐
🔑	mapId	int	☐
🔑	routeId	int	☐
🔑	squareId	int	☐
	totalSquare	int	☐
			☐

Şekil 5.34: Tablo 34

Tablo 35’te kullanıcının oyunda bulunduğu konum ve o anda savaşta olup olmadığı gibi birtakım bilgiler tutulur.

tblUsersOnMapInformation			
	Column Name	Data Type	Allow Nulls
🔑	userId	int	☐
	mapId	int	☐
	routeId	int	☐
	squareId	int	☐
	wildPokemonId	int	☐
	wildPokemonLevel	int	☐
	footStep	int	☒
	wildPokemonCatchRatio	int	☒
			☐

Şekil 5.35: Tablo 35

Tablo 36’da kullanıcıların sahip olduğu pokemonlar ve bu pokemonların bilgileri tutulmaktadır.

tblUsersPokemons			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	UserId	int	☐
	PokemonPlace	nvarchar(200)	☐
	PokemonPrice	int	☐
	PokemonId	int	☐
	PokemonLevel	int	☐
	PokemonExp	int	☐
	BattleCount	int	☐
	BattleVictoryCount	int	☐
	MaximumHp	int	☐
	DvPointHp	int	☐
	EvPointHp	int	☐
	CurrentHp	int	☐
	Attack	int	☐
	DvPointAttack	int	☐
	EvPointAttack	int	☐
	Defense	int	☐
	DvPointDefense	int	☐
	EvPointDefense	int	☐
	SpAttack	int	☐

Şekil 5.36: Tablo 36

Tablo 37’de kullanıcı bilgileri tutulmaktadır.

tblUsersProfile			
	Column Name	Data Type	Allow Nulls
	userId	int	☐
	userName	nvarchar(200)	☐
	name	nvarchar(200)	☐
	surname	nvarchar(200)	☐
	password	nvarchar(200)	☐
	email	nvarchar(200)	☐
	birthday	nvarchar(200)	☐
	firstPokeId	nvarchar(200)	☐
	selectedLanguage	nvarchar(200)	☐
	gender	nvarchar(200)	☐
	registrationDate	nvarchar(200)	☐
	secretQuestion	nvarchar(200)	☐
	secretQuestionAnswer	nvarchar(200)	☐
	activationCode	nvarchar(200)	☐
	authorityLevel	nvarchar(200)	☐
	charImage	nvarchar(200)	☐
	avatarImageLink	nvarchar(200)	☐
	whoAmI	nvarchar(4000)	☐
	showOthersInMap	nvarchar(200)	☐
	lastEnteredPokemonCenterId	int	☒
	uniquePageSessionId	nvarchar(200)	☒

Şekil 5.37: Tablo 37

Tablo 38’de hangi kullanıcının hangi eğitimcilerle savaştığının bilgisi tutulmaktadır. Savaşıp, yendiği bir eğitimciyle tekrar savaşmamaktadır.

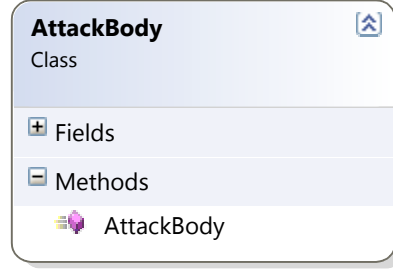
tblUsersVersusNPCs			
	Column Name	Data Type	Allow Nulls
	userId	int	☐
	npcId	int	☐
			☐

Şekil 5.38: Tablo 38

1.2. Yazılım Tasarımı

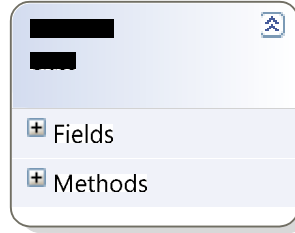
1.2.1. Sınıflar

- 1) **Astar.cs**: Bu Sınıf en kısa yol bulma algoritmasının gerçekleştirildiği sınıftır.
- 2) **AttackBody.cs**: Bu sınıf saldırıların biçimini oluşturan sınıftır.



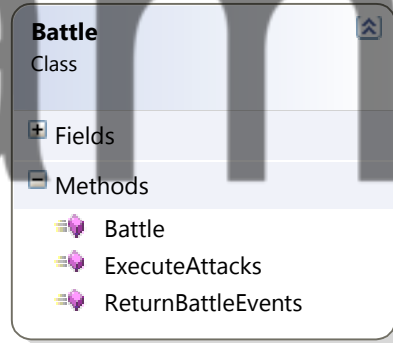
Şekil 5.39: Sınıf Diyagramı

3) Attacks.cs: Bu sınıf her saldırının ek özelliklerinin gerçekleştirildiği sınıftır. Bir saldırı düşmanı zehirliyor işlemleri bu sınıf içinde gerçekleştirilir. Bu yüzden bu sınıfta 480'den fazla metod bulunmaktadır. (resim çok büyük olduğu için metotlar saklandı)



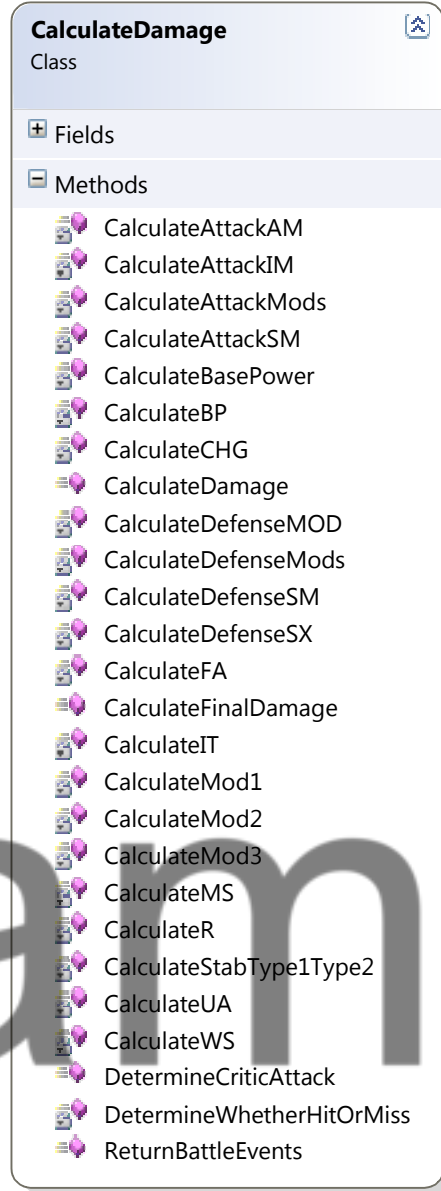
Şekil 5.40: Sınıf Diyagramı

4) Battle.cs: Bu sınıf savaş algoritmasının ana yapısıdır. Bu yapıya göre bilgisayar tarafından yönetilen pokemonlarla savaşlar gerçekleşir. Oyunun mekaniklerine[17] göre dizayn edilmiştir ve çeşitli kurallara göre sıra geçmeli(turn based) tarzda çalışmaktadır.



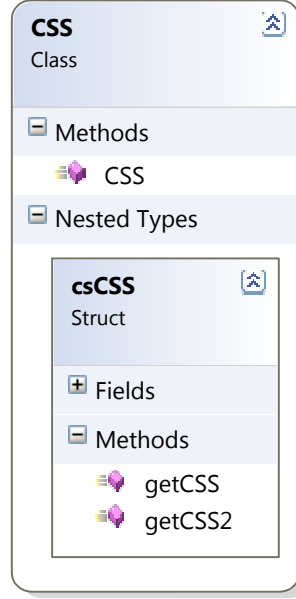
Şekil 5.41: Sınıf Diyagramı

5) CalculateDamage.cs: Bu sınıf, her saldırı sonucunda düşmana verilecek saf hasarın hesaplandığı sınıftır. Oyunun mekaniklerine[17] göre yazılmıştır.



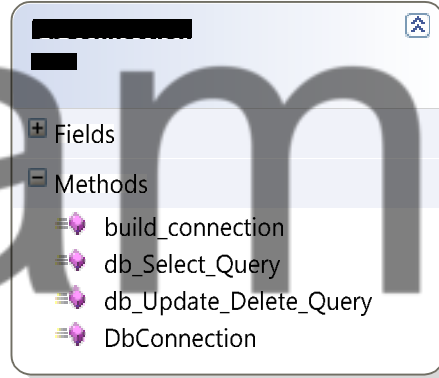
Şekil 5.42: Sınıf Diyagramı

6) CSS.cs: Bu sınıf css yapımız için kullanılmaktadır. Oyunun görselliğini tasarlamada kullanılmaktadır.



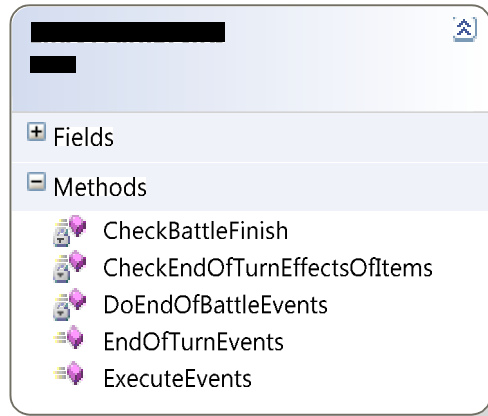
Şekil 5.43: Sınıf Diyagramı

7)DbConnection.cs: Veri tabanı ile yapılan bütün işlemler bu sınıf aracılığı ile yapılmaktadır.



Şekil 5.44: Sınıf Diyagramı

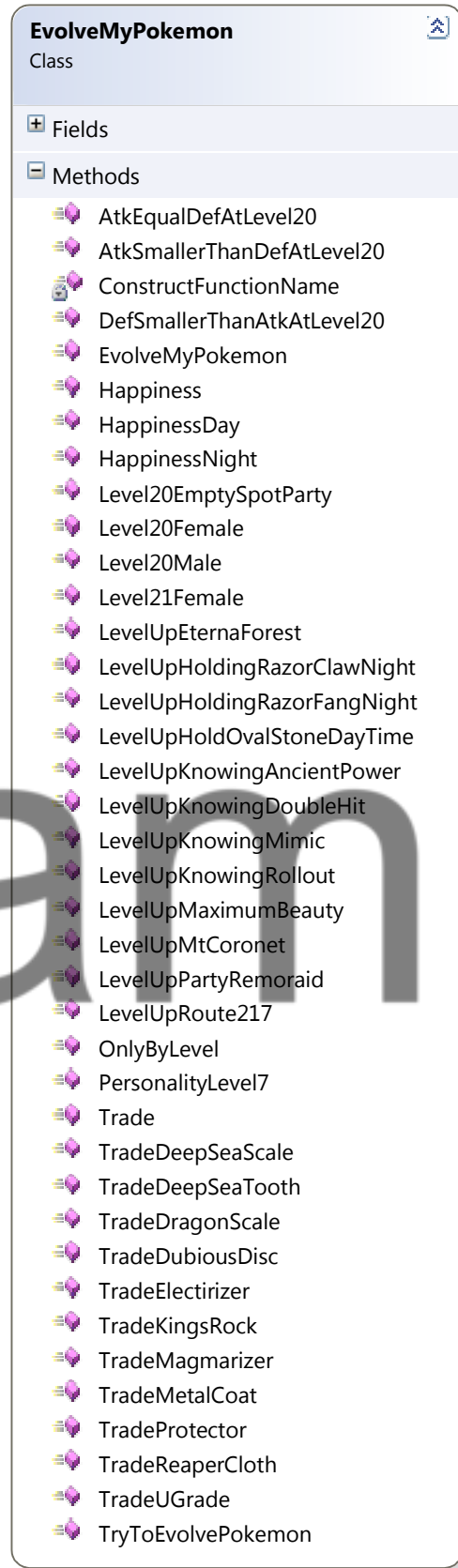
8) EndOfTurnEvents.cs: Savaş mekanizmasında her sıra geçişinden sonra çeşitli olayların yürütülmesi gerekmektedir. Bu olayların yürütüldüğü sınıftır. Oyunun mekaniklerine[17] göre yazılmıştır.



Şekil 5.45: Sınıf Diyagramı

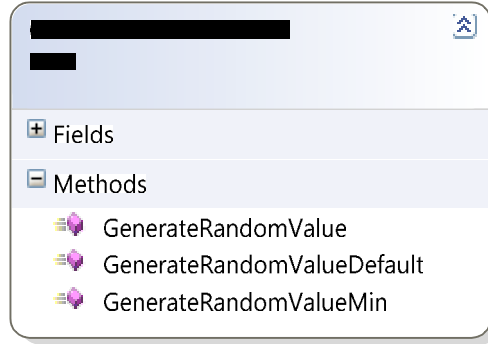
9)EvolveMyPokemon.cs: Pokemonlar 39 farklı yöntem ile evrimleşmektedir. Bu yöntemlerden 36 tanesi bu sınıf içinde yazılmıştır. Bir pokemonun evrimleşme şartının sağlanıp sağlanmadığının tespiti, evrimleşme şartının sağlanması durumunda ilgili evrim işlemlerinin gerçekleştirilmesi bu sınıfta olmaktadır.

Example



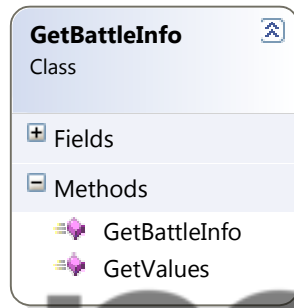
Şekil 5.46: Sınıf Diyagramı

10)GenerateRandomValue.cs: Bu sınıf çok iyi dağılama sahip rastgele sayı üretimini gerçekleştirmektedir.



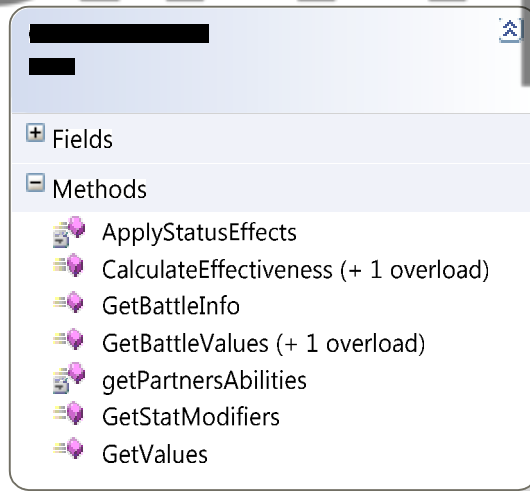
Şekil 5.47: Sınıf Diyagramı

11)GetBattleInfo.cs: Fonksiyonlarda kullanabilmek üzere savaş bilgilerini çeken fonksiyonun bulunduğu sınıftır. Her seferinde tek tek çekmek yerine sınıf tasarlanmıştır.



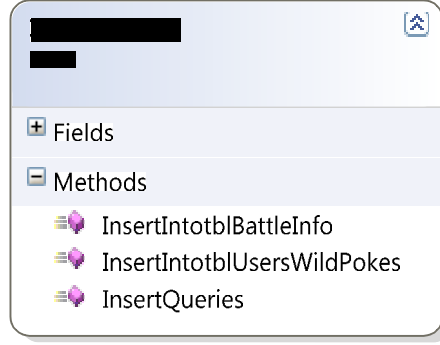
Şekil 5.48: Sınıf Diyagramı

12)GetBattleValues.cs: Tüm savaş verilerinin (pokemonların sağlık puanı gibi) veri tabanından çekildiği sınıftır. Oldukça fazla alan(fields) içermektedir. Ayrıca çeşitli hesaplamalar burada yapılmaktadır.



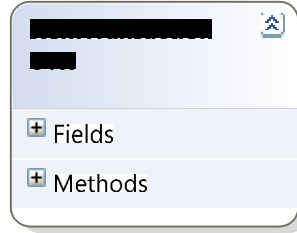
Şekil 5.49: Sınıf Diyagramı

13)InsertQueries.cs:Bazı tablolar oldukça büyük olduğu için bu tablolara veri ekleme işlemi oldukça zor olmaktaydı. Sorgu kodda çok fazla yer kaplamakta, sorgunun yapıldığı tabloda yapılan değişikliklerden sonra bütün sorgular değiştirilmek zorunda kalınmaktaydı. Bu sınıf tasarlanarak sorgu öbekleri bir araya getirilerek bahsedilen sorunlar azaltılmıştır.



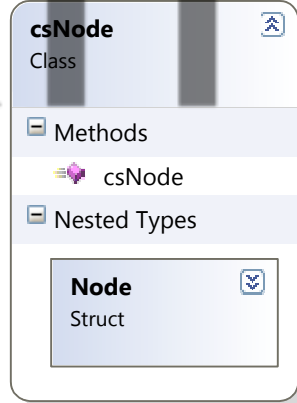
Şekil 5.50: Sınıf Diyagramı

14)ItemTransaction.cs:Bütün eşyaların kullanım metotları bu sınıf içinde yazılmıştır. Oldukça fazla eşya olduğu için metotlar resimde gösterilmemiştir.



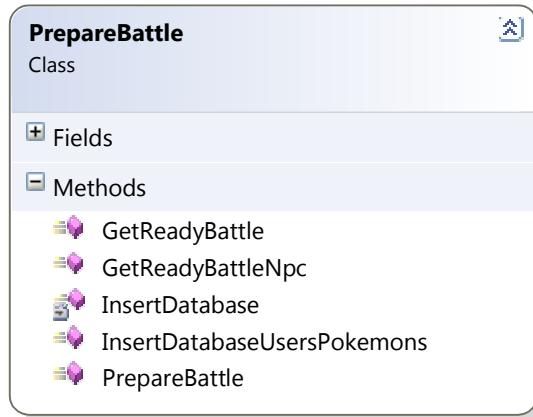
Şekil 5.51: Sınıf Diyagramı

15)Node.cs:İçinde bulunan yapı (structure) ile haritanın bir birim karesi temsil edilmektedir ve yol bulma algoritmasında ve haritada hareket etmede kullanılmaktadır.



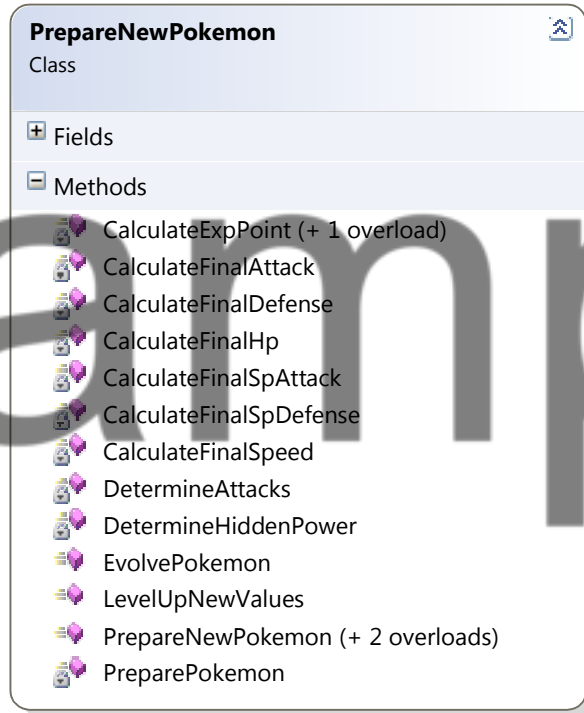
Şekil 5.52: Sınıf Diyagramı

16)PrepareBattle.cs: Bilgisayar tarafından yönetilen bir pokemon ile savaş yapılacağı zaman çeşitli değerlerin hesaplanması ve bu bilgilerin veri tabanına aktarılması bu sınıfta gerçekleşiyor.



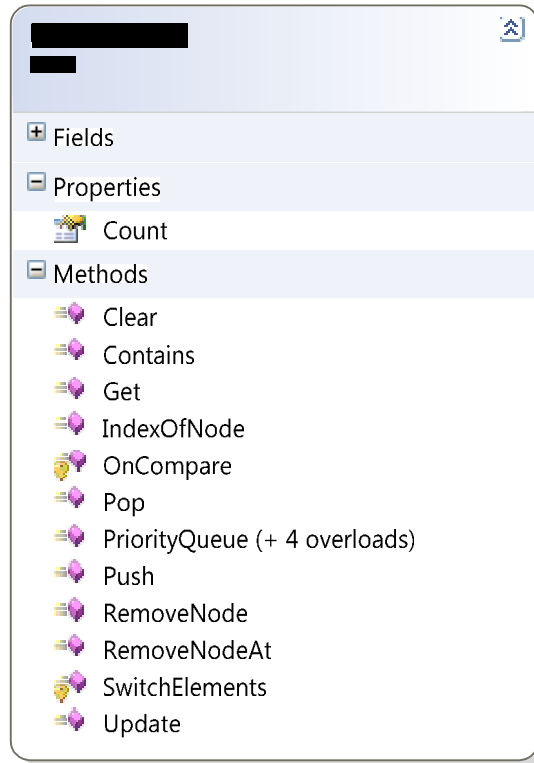
Şekil 5.53: Sınıf Diyagramı

17)PrepareNewPokemon.cs: Veri tabanına herhangi bir pokemon ekleneceği zaman bu pokemonun özelliklerinin oyun mekaniklerine[17] göre hesaplanması bu sınıfta gerçekleştirilmektedir.



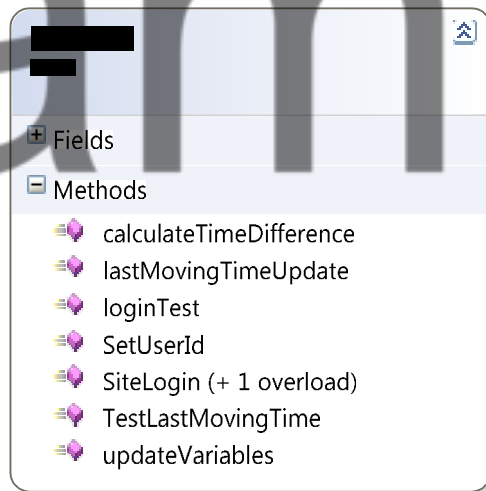
Şekil 5.54: Sınıf Diyagramı

18)PriorityQueue.cs: En kısa yol bulmada açık listeden düğüm seçimi yapılırken açık listedeki düğümlerden yol maliyeti en küçük olan seçilir. Her seçim, en küçük yol maliyetine sahip düğümün bulunacağı bir arama yapmayı gerektirecektir. Bu, işlem maliyetini artıracak için, listeye her düğüm eklenişinde eklenen düğümün listede düğümlerin F değerine göre bir sıralamada yerleşmesini sağlayan öncelik kuyruğu kullanılmıştır.[1]



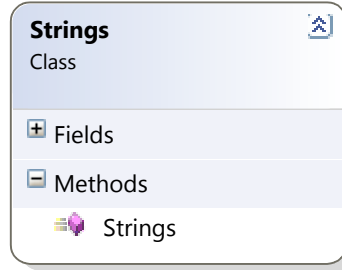
Şekil 5.55: T

19) **SiteLogin.cs**: Siteye giriş yapmak için gerekli bazı işlemlerin yapıldığı sınıftır.



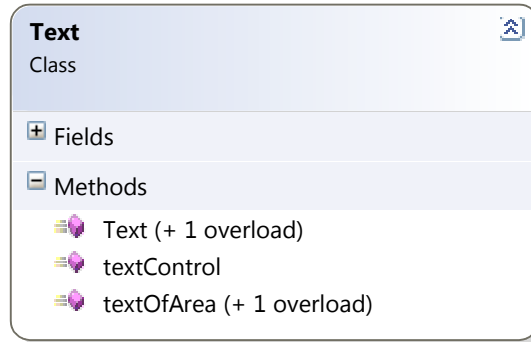
Şekil 5.56: Sınıf Diyagramı

20) **Strings.cs**: Bazı stringler kodda pek çok yerde kullanılmaktadır. Uzunluğu çok fazla olan stringler kullanım kolaylığı ve efektif kodlama için bu sınıf içinde toplanmıştır.



Şekil 5.57: Sınıf Diyagramı

21)Text.cs:Site tasarımıımız birden çok dil desteğine sahip olduğu için ekranda görülen tüm yazılar veri tabanında tutulmaktadır. Bu sınıfta bu yazıları veritabanından çekme metotları bulunmaktadır.



Şekil 5.58: Sınıf Diyagramı

1.2.2. Sayfalar

1)BattleLost.aspx:

Kullanıcı yaptığı savaşı kaybettiğinde otomatik olarak bu sayfaya yönlendirilmektedir. Sayfada savaşı kaybettiği haberi verilmektedir.

2) BattleResult.aspx:

Kullanıcı savaşı kazandığında açılan sayfadır. Sayfada kullanıcının savaş sonunda kazandığı tecrübe puanı, kazandığı para gibi bilgiler verilir. Yeni saldırı öğrenen pokemonların saldırısını kaydetme işlemi de bu sayfada yapılmaktadır.

3) BattleWildPokemon.aspx

Savaşın gerçekleştiği ekrandır. Bu ekranda kullanıcı savaştaki pokemonunu yönetir. Ekranda kullanıcının o andaki pokemonunun sağlık, güç vb bilgileri gösterilir. Ayrıca pokemonun yapabileceği saldırılarda bu sayfada gösterilir ve kullanıcı bu sayfadan yapacağı saldırıları seçerek yapar.Pek çok savaş algoritması bu sayfada çağrılmakta ve kullanılmaktadır.

4) SelectWhichPokemon.aspx

Kullanıcı savaşa girmeden önce ve savaş sırasında pokemon değiştirmek istediği zaman bu sayfaya yönlendirilir. Kullanıcı bu sayfada, çantasındaki baygın olmayan pokemonlardan birini seçerek savaşa devam edebilir.

5)error.aspx

Çeşitli yerlerdeki hata kodlarına göre ekrana mesaj bastırmak için kullanılan sayfadır. Bu sayfada hata kodları Session["errorNo"]'dan alınır. Bu değer 1 ise kullanıcı siteye izinsiz veri göndermeye çalışıyordur, 2 ise siteye giriş yapmadan görmemesi gereken sayfalara girmeye çalışıyordur.

6)exit.aspx

Kullanıcının siteden güvenli çıkış yapmasını sağlayan sayfadır. Session[“loginStatus”] bilgisi silinerek sitedeki girişi iptal edilir.

7)forgotpassword.aspx

Şifresini unutan kullanıcıların şifresini öğreneceği sayfadır.

8)forum.aspx

Oyunun formuna yönlendiren sayfadır.

9)gamepage.aspx

Oyunun oynandığı sayfadır. Bu sayfada kullanıcılar, haritalardaki gidilebilir yollarda hareket edebilmektedirler. İstekleri binalara olan en kısa yolu yine bu sayfada öğrenebilmektedirler.

10)help.aspx

Bu sayfada oyuncuların oyun hakkında rehber şeklinde bilgi alması sağlanır.

11)homepage.aspx

Oyunun giriş sayfasıdır. Oyuna giriş mesajı ve online oyuncular gözükür duyurularda buradan yapılır. Oyundan haberler vardır.

12)topranks.aspx

Kullanıcıları çeşitli kategorilere göre sıralayan, bunların diğer oyuncular tarafından gösterilmesini sağlayan ve oyuncuların diğer oyuncuların istatistikî verilerinin görülmesini sağlayan sayfadır. Sıralamadaki kategoriler şunlardır: Pokemon sayısı, savaş sayısı, savaşlarda elde ettiği toplam galibiyet sayısı, salon liderleri ile olan savaş sayısı, salon liderlerine karşı aldığı galibiyet sayısı, ligde yaptıkları toplam savaş sayısı, ligde yaptığı savaşlardaki galibiyet sayısı, pvp savaşı sayısı, pvp kazanma sayısı, ulusal puanı ve son olarak bitirdiği görevler.

13)market.aspx

Kullanıcı bu sayfaya haritalardaki market binalarına giriş yaparak ulaşabilir. Bu sayfada kullanıcı marketten, marketin belirlediği fiyattan, marketin sattığı eşyaları satın alabilir ve markete eşya satabilir.[5] Eşyalarını ve pokemonlarını diğer kullanıcıların satın alabilmesi için istediği fiyattan satılığa çıkarabilir. Başka kullanıcıların satılığa çıkardığı eşya ve pokemonları onların belirlediği fiyattan satın alabilir.

14)messagebox.aspx

Kullanıcılar bu sayfada diğer kullanıcılara özel mesajlar gönderebilir, onlardan aldığı özel mesajları okuyabilir.

15)mybag.aspx

Kullanıcı bu sayfada çantasında taşıdığı pokemonların durumlarını görebilir ve çantasında taşıdığı eşyaları kullanabilir.

16)pokedex.aspx

Pokemonları kullanıcılara tanıtmak amacıyla tasarlanmıştır bir sayfadır. Kullanıcı bu sayfada oyundaki 498 pokemonun bütün bilgilerine ulaşabilir.

17)pokemoncenter.aspx

Kullanıcı bu sayfaya haritalardaki pokemon merkezi binalarına giriş yaparak ulaşabilir. Kullanıcı bu sayfada çantasındaki taşıma limitinden dolayı, çantasındaki pokemonları deposuna, deposundaki pokemonları çantasına aktarabilir. Aynı şekilde çantasındaki eşyaları deposuna, deposundaki eşyaları çantasına aktarabilir. Ek olarak pokemon merkezindeki Hemşire Joy’a müracat ederek yaralı pokemonlarını tedavi ettirebilir.

18)profile.aspx

Kullanıcı bu sayfada oyundaki istatistiklerini görebilir ve kişisel bilgilerini düzenleyebilir.

19)randomImage.aspx

Bu sayfa ile oyunda kayıt olurken ve giriş yaparken güvenlik kodu oluşturulur ve ekrana basılması sağlanır.

20)register.aspx

Bu sayfada kullanıcı oyuna kaydolmak için gerekli bilgileri girer ve oyuna kaydolur.

2. Yardımcı Programlar

2.1. Pokemon Bilgileri Veritabanı Doldurucusu

Projede 493 farklı pokemon, 467 farklı saldırı bulunmakta ve bunların her birinin onlarca farklı özelliği bulunmaktadır. Bu bilgilerin veritabanına elle kaydedilmesi çok uzun zaman alacağı için, referans sitelerimizin[38][39], kaynak kodlarından bu bilgileri otomatik olarak alıp, düzenleyip, üzerinden ek işlemler yaptıktan sonra veritabanına kaydeden programlar tasarlanmıştır. Hazırlanan bu programlarla büyük veri hacmine sahip tblPokedex, tblAttacks, tblNormalAttacks ve tblMachineAttacks veritabanı tabloları doldurulmuştur.

Program temel olarak iki bölümden oluşmaktadır. Birinci bölüm pokemon bilgilerini almakta ve veritabanına eklemektedir. Görünümü şekilde verilmiştir.

Poke...	Name	PokemonNational	pokeType1	pokeType2	pokeGenderMal...	pokeGenderFem...	pokeEggGroup	Height	Weight	pokeForm1
1	Bulbasaur	1	grass	poison	87.5%	12.5%	Monster / Plant	0.7 m	6.9 kg	Basic
2	Ivysaur	2	grass	poison	87.5%	12.5%	Monster / Plant	1 m	13.0 kg	Basic
3	Venusaur	3	grass	poison	87.5%	12.5%	Monster / Plant	2 m	100.0 kg	Basic
4	Charmander	4	fire	null	87.5%	12.5%	Monster / Dragon	0.6 m	8.5 kg	Basic
5	Chameleon	5	fire	null	87.5%	12.5%	Monster / Dragon	1.1 m	19.0 kg	Basic
6	Charizard	6	fire	flying	87.5%	12.5%	Monster / Dragon	1.7 m	90.5 kg	Basic
7	Squirtle	7	water	null	87.5%	12.5%	Monster / Water 1	0.5 m	9.0 kg	Basic
8	Wartortle	8	water	null	87.5%	12.5%	Monster / Water 1	1 m	22.5 kg	Basic
9	Blastoise	9	water	null	87.5%	12.5%	Monster / Water 1	1.6 m	85.5 kg	Basic
10	Caterpie	10	bug	null	50%	50%	Bug	0.3 m	2.9 kg	Basic
11	Metapod	11	bug	null	50%	50%	Bug	0.7 m	9.9 kg	Basic
12	Butterfree	12	bug	flying	50%	50%	Bug	1.1 m	32.0 kg	Basic
13	Weedle	13	bug	poison	50%	50%	Bug	0.3 m	3.2 kg	Basic
14	Kakuna	14	bug	poison	50%	50%	Bug	0.6 m	10.0 kg	Basic
15	Beedrill	15	bug	poison	50%	50%	Bug	1 m	29.5 kg	Basic
16	Pidgey	16	normal	null	50%	50%	Flying	0.3 m	1.8 kg	Basic
17	Pidgeotto	17	normal	null	50%	50%	Flying	1.1 m	30.0 kg	Basic
18	Pidgeot	18	normal	null	50%	50%	Flying	1.5 m	39.5 kg	Basic
19	Rattata	19	normal	null	50%	50%	Ground	0.3 m	3.5 kg	Basic
20	Raticate	20	normal	null	50%	50%	Ground	0.7 m	18.5 kg	Basic
21	Spearow	21	normal	null	50%	50%	Flying	0.3 m	2.0 kg	Basic
22	Fearow	22	normal	null	50%	50%	Flying	1.2 m	38.0 kg	Basic
23	Ekane	23	poison	null	50%	50%	Ground / Dragon	2 m	6.9 kg	Basic
24	Arbok	24	poison	null	50%	50%	Ground / Dragon	3.5 m	65.0 kg	Basic
25	Pikachu	25	electric	null	50%	50%	Ground / Fairy	0.4 m	6.0 kg	Baby/Pre-evo
26	Raichu	26	electric	null	50%	50%	Ground / Fairy	0.8 m	30.0 kg	Baby/Pre-evo
27	Sandslash	27	ground	null	50%	50%	Ground	0.6 m	12.0 kg	Basic
28	Sandslash	28	ground	null	50%	50%	Ground	1 m	29.5 kg	Basic
29	NidoranFemale	29	poison	null	0%	100%	Monster / Ground	0.4 m	7.0 kg	Basic
30	Nidorina	30	poison	null	0%	100%	No Eggs	0.8 m	20.0 kg	Basic
31	Nidoqueen	31	poison	ground	0%	100%	No Eggs	1.3 m	60.0 kg	Basic
32	NidoranMale	32	poison	null	100%	0%	Monster / Ground	0.5 m	9.0 kg	Basic
33	Nidorino	33	poison	null	100%	0%	Monster / Ground	0.9 m	19.5 kg	Basic
34	Nidoking	34	poison	ground	100%	0%	Monster / Ground	1.4 m	62.0 kg	Basic
35	Clefairy	35	normal	null	25%	75%	Fairy	0.6 m	7.5 kg	Baby/Pre-evo
36	Clefable	36	normal	null	25%	75%	Fairy	1.3 m	40.0 kg	Baby/Pre-evo
37	Vulpix	37	fire	null	25%	75%	Ground	0.6 m	9.9 kg	Basic
38	Ninetales	38	fire	null	25%	75%	Ground	1.1 m	19.9 kg	Basic
39	Jigglypuff	39	normal	null	25%	75%	Fairy	0.5 m	5.5 kg	Baby/Pre-evo
40	Wigglytuff	40	normal	null	25%	75%	Fairy	1 m	12.0 kg	Baby/Pre-evo
41	Zubat	41	poison	flying	50%	50%	Flying	0.8 m	7.5 kg	Basic
42	Golbat	42	poison	flying	50%	50%	Flying	1.6 m	55.0 kg	Basic

Şekil 5.59: Pokemon Bilgileri

İkinci bölüm ise saldırıları okumakta, düzenlemekte ve veritabanına kaydetmektedir. Görünümü şekilde verilmiştir.

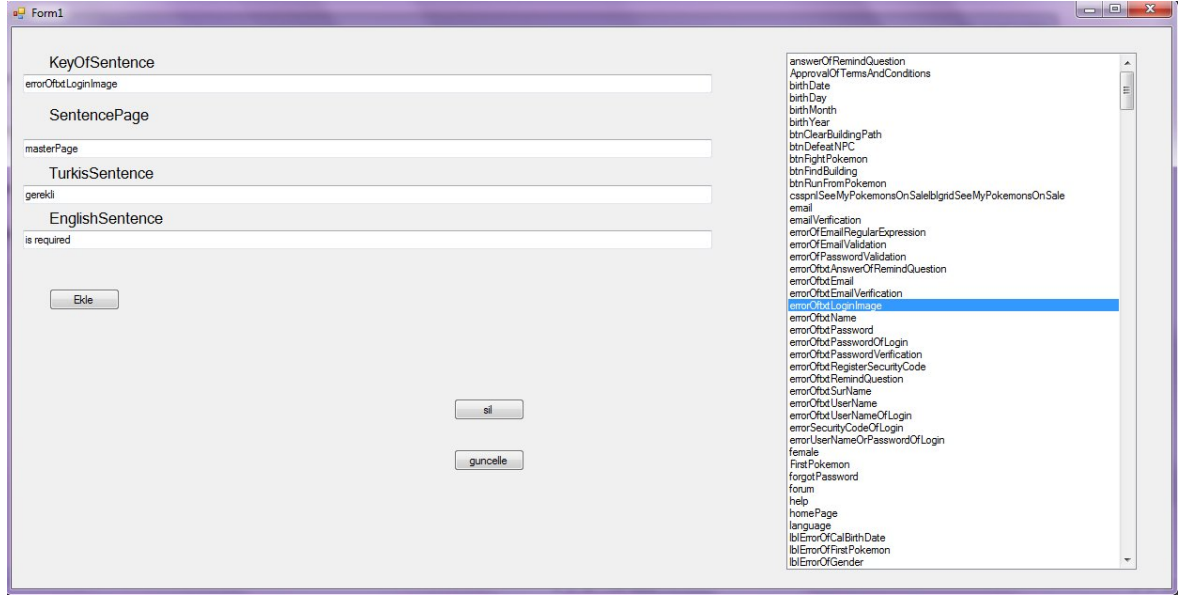
pokemon_numarasi	pokemon_ismi	Attack 1	Attack 1 Level up	Attack 2	Attack 2 Level up	Attack 3	Attack 3 Level up	Attack 4	Attack 4 Level up
1	Bulbasaur	Blissey	Pound	1	Growl	1	Tail Whip	5	Refresh
2	Irysaur	Buneary	Pound	1	Defense Curl	1	Splash	1	Foresight
3	Venusaur	Chansey	Pound	1	Growl	1	Tail Whip	5	Refresh
4	Charmender	Clefairy	Pound	1	Growl	1	Encore	4	Sing
5	Chameleon	Clefia	Pound	1	Charm	1	Encore	4	Sing
6	Charizard	Drowzee	Pound	1	Hypnosis	1	Disable	7	Confusion
7	Squirtle	Exploud	Pound	1	Uproar	1	Astonish	1	Howl
8	Wartortle	Finneon	Pound	1	Water Gun	6	Attract	10	Rain Dance
9	Blastoise	Grimer	Pound	1	Poison Gas	1	Harden	4	Mud-slap
10	Caterpie	Grovyle	Pound	1	Leer	1	Absorb	1	Quick Attack
11	Metapod	Gulpin	Pound	1	Yawn	6	Poison Gas	9	Sludge
12	Butterfree	Happiny	Pound	1	Charm	1	Copycat	5	Refresh
13	Weedle	Hypno	Pound	1	Disable	1	Confusion	1	Hypnosis
14	Kakuna	Igglybuff	Sing	1	Charm	1	Defense Curl	5	Pound
15	Beedrill	Jigglypuff	Sing	1	Defense Curl	5	Pound	9	Disable
16	Pidgey	Jynx	Pound	1	Lick	1	Lovely Kiss	1	Powder Snow
17	Pidgeotto	Lopunny	Pound	1	Defense Curl	1	Splash	1	Foresight
18	Pidgeot	Loudred	Pound	1	Uproar	1	Astonish	1	Howl
19	Rattata	Lumineon	Pound	1	Water Gun	1	Attract	1	Water Gun
20	Raticate	Mew	Pound	1	Transform	1	Mega Punch	10	Metronome
21	Spearow	Muk	Pound	1	Harden	1	Poison Gas	1	Mud-slap
22	Fearow	Nuzleaf	Pound	1	Razor Leaf	1	Harden	3	Growth
23	Ekane	Piplup	Pound	1	Growl	4	Bubble	8	Water Sport
24	Arbok	Sceptile	Pound	1	Leer	1	Absorb	1	Quick Attack
25	Pikachu	Smoochum	Pound	1	Lick	5	Sweet Kiss	8	Powder Snow
26	Raichu	Sunflora	Pound	1	Absorb	1	Growth	1	Mega Drain
27	Sandslrew	Swalot	Pound	1	Sludge	1	Poison Gas	1	Yawn
28	Sandslash	Trecko	Pound	1	Leer	1	Absorb	1	Quick Attack
29	NidoranFemale	Whismur	Pound	1	Uproar	5	Astonish	11	Howl
30	Nidorina	Machop	Leer	1	Low Kick	1	Focus Energy	1	Focus Energy
31	Nidoqueen	Machoke	Leer	1	Low Kick	1	Focus Energy	1	Focus Energy
32	NidoranMale	Machop	Leer	1	Low Kick	1	Focus Energy	7	Karate Chop
33	Nidorino	Mankey	Scratch	1	Leer	1	Low Kick	1	Focus Energy
34	Nidoking	Primeape	Scratch	1	Leer	1	Low Kick	1	Focus Energy
35	Clefairy	Clefable	DoubleSlap	1	Sing	1	Minimize	1	Metronome
36	Clefable	Delcatty	DoubleSlap	1	Sing	1	Attract	1	Fake Out
37	Vulpix	Mime Jr.	Confusion	1	Barrier	1	Tickle	1	Copycat
38	Ninetales	Mr. Mime	Confusion	1	Barrier	1	Magical Leaf	1	Power Swap
39	Jigglypuff	Politoed	DoubleSlap	1	Bubblebeam	1	Hypnosis	1	Perish Song
40	Wigglytuff	Poliwag	Water Sport	1	Bubble	5	Hypnosis	8	Water Gun
41	Zubat	Poliwag	Hypnosis	1	Bubble	1	Water Sport	1	Bubble
42	Golbat	Poliwag	DoubleSlap	1	Bubblebeam	1	Submission	1	Hypnosis

Create Attacks % 99,79 Add Data To Database Delete Data From Database

Şekil 5.60: Pokemon Bilgi Düzenleme

2.2. Dil Ekleme Programı

Projede birden fazla dil desteği sunulabilmesi için, kullanıcı ile iletişimde kullanılan bütün yazıların, destek verilen dillerdeki hali veritabanında tutulmuştur. Bu yazıların destek verilen dillerdeki halini veritabanına eklemek, güncellemek ve silmek için şekilde verilen program hazırlanmıştır.

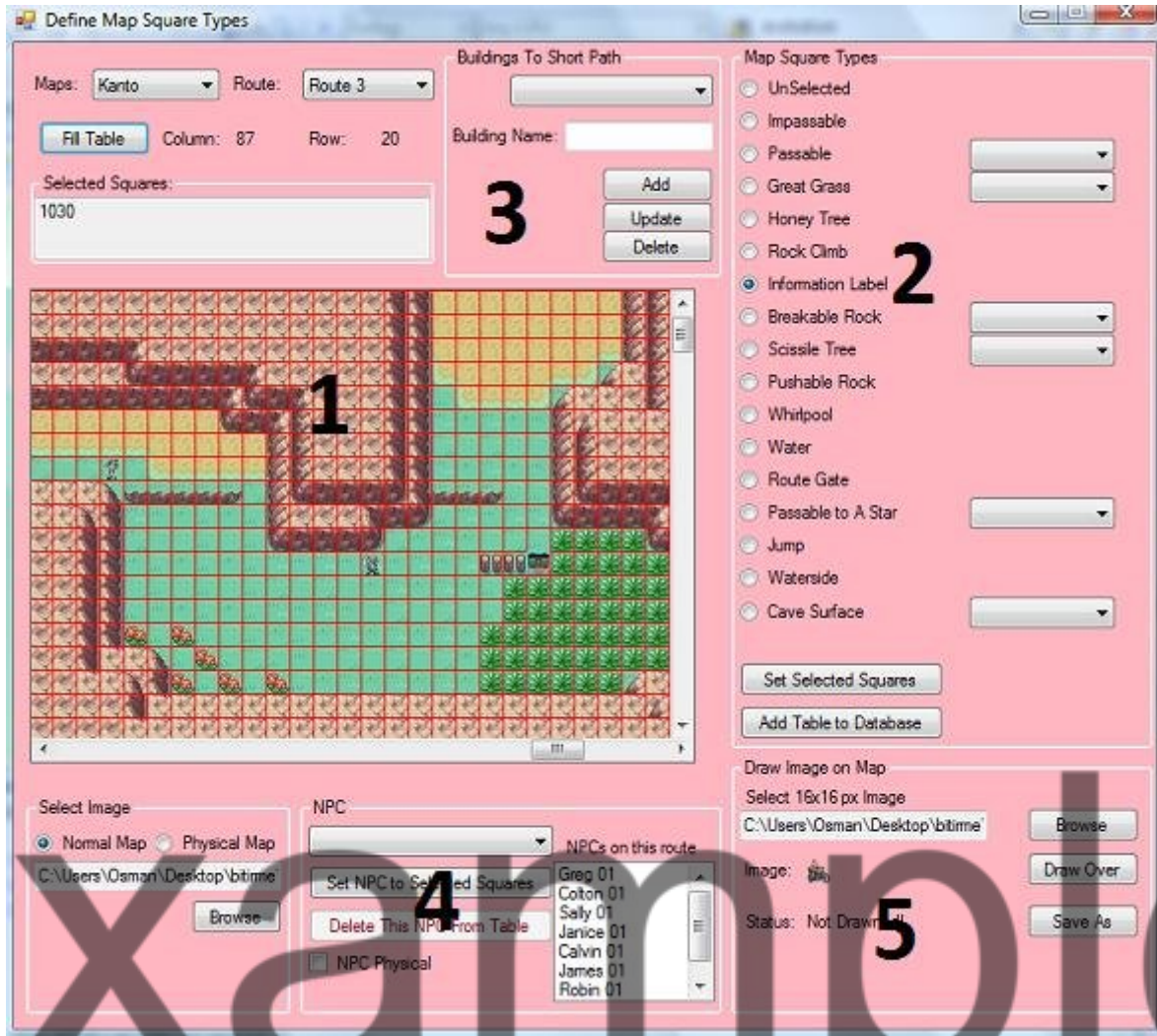


Şekil 5.61: Dil Ekleme Programı

2.3. Harita Hazırlama Programı

Oyunun webde oynamaya hazır gelebilmesi, oyunun bütün verilerinin veritabanına daha hızlı ve doğru bir şekilde kaydedilebilmesi için masaüstü uygulaması olarak yardımcı programlar hazırlanmıştır. Bunlardan biri de harita hazırlama programıdır. Programın modülleri aşağıda tanıtılmıştır.

Example



Şekil 5.62: Harita Şekilleri Tanımlama Programı

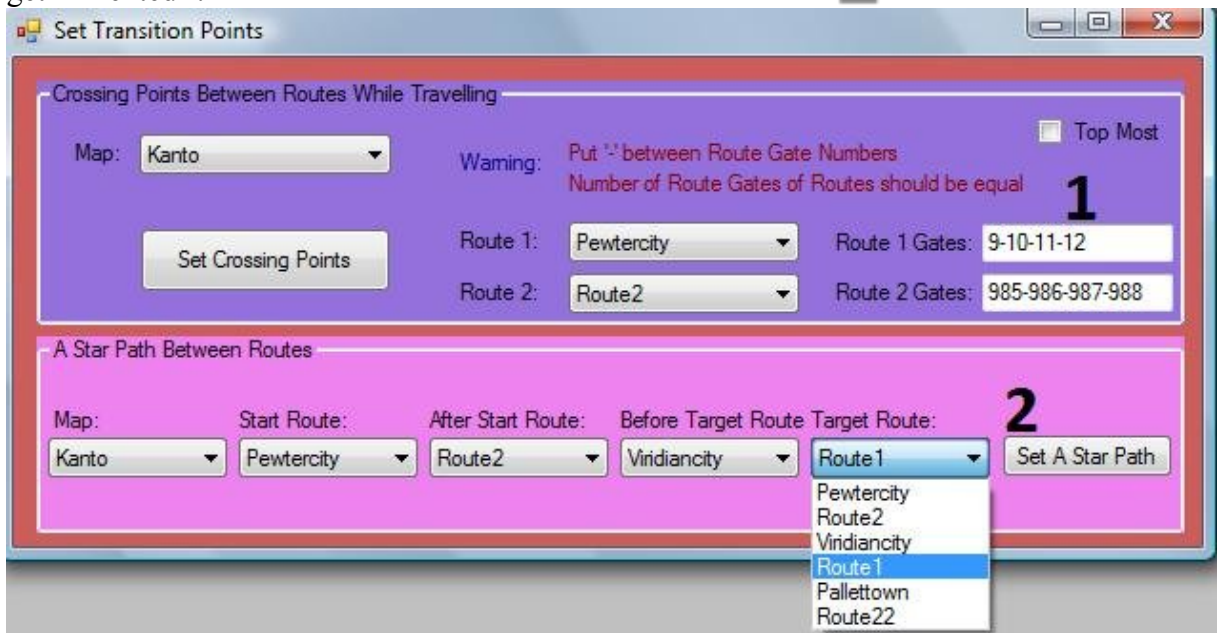
Haritalarda birbirinden farklı yer şekilleri bulunmaktadır. Hepsinin farklı görevleri vardır. Örnek olarak oyuncular haritada gezerken yürüyebildikleri yerler vardır, yürüyemedikleri yerler vardır. Vahşi pokemonlarla karşılaşabilecekleri yerler vardır, bina girişleri, bilgilendirilme levhaları vardır. Yukarıdaki program oyunda kullanılacak haritaların sisteme daha hızlı ve doğru bir şekilde tanıtılabilmesi amacıyla hazırlanmıştır. Önce üst bölmeden sisteme tanıtılacak olan haritanın ismi seçilir. Daha sonra alt bölmeden haritanın resmi açılır. “1” numaralı alanda açılan harita resmi, şekillerin doğru bir şekilde seçilebildiğini göstermek amacıyla birim karelere ayrılmıştır. Önce fare yardımıyla aynı türden olan alanlar seçilir, seçilen alanın şekli ise “2” numaralı alanda bulunan yer şekilleri içinden seçilir ve ilgili butona basılarak haritanın o kısmında ne tür yer şekli bulunduğunun bilgisi veritabanına girilerek harita sisteme tanıtılır. Kullanıcının oyunda çeşitli ihtiyaçlarını giderdiği binalar vardır. Büyük bir haritada bu binaların yerini bilemeyebilirler. Oyunun kullanıcılara sağladığı en önemli kolaylıklardan biri, kullanıcının gitmek istediği yere en kısa yoldan nerden gidildiğini istediği an rahatlıkla öğrenebilmesidir. Bu yüzden bu binaların konumları da sisteme tanıtılmalıdır. “3” numaralı bölüm de bunun için hazırlanmıştır. Kullanıcı oyunda ilerlerken zaman zaman önünü bazı eğitimciler keserler. Kullanıcının oradan geçebilmesi için bu eğitimcilerle savaşıp onları yenebilmesi gerekmektedir. “4” numaralı bölümde bu haritadaki eğitimcilerin haritanın hangi bölgelerinde bulunacağı ve nerelerden geçen kullanıcıları yakalayacağı sisteme

tanıtılır. Eğitimcilerin yeri sabit olacağı için harita üzerinde eğitimcinin bulunduğu yere eğitimcinin küçük resminin çizilmesi gerekmektedir. Haritada eğitimcinin resminin çizileceği nokta seçildikten sonra, “5” numaralı bölüm yardımıyla eğitimci resmi seçilir ve harita üzerine çizilir.



Şekil 5.63: Harita Boyutlarını Düzenleme Programı

Bazı harita resimlerinin boyutlarında uyumsuzluk olabilmektedir. Bu resimlerin boyutlarının oyuna uygun hale getirebilmesi için yukarıdaki program hazırlanmıştır. “1” numaralı bölümde birim kare boyutundaki oyuncu resminin, haritanın birim karelerinin tam üzerine gelip gelmediği kontrol edilmektedir. “2” numaralı bölümdeki yön butonları ile haritada gezdirilen oyuncu ile haritanın boyutlarında sorun olup olmadığı kontrol edilmektedir. Haritanın boyutlarında sorun varsa “3” numaralı bölüm yardımıyla haritanın istenen bölgelerinden istenen miktarda kesmeler yapılarak harita oyuna uygun hale getirilmektedir.



Şekil 5.64: Harita Parçaları Geçiş Bölgelerini Ayarlama Programı

Haritalar parça parça bölümler halindedir ve bu bölümler arasında geçiş noktaları vardır. Yukarıdaki program bu geçiş noktalarının sisteme tanıtılabilmesi için hazırlanmıştır. “1” numaralı bölümde kullanıcının bir bölümden diğer bölüme geçerken kullanacağı geçiş noktaları tanıtılmıştır. Bölümler arasında geçiş bölgelerinin, en kısa yol bulma algoritmasının sağlıklı bir şekilde çalışabilmesi için de sisteme tanıtılması gerekmektedir. Bunun için de “2” numaralı bölüm hazırlanmıştır.

NPC Registration

NPC Name: Add NPC

NPC Class: 1 Update NPC

NPC Winning Reward: Delete NPC

NPC Id	NPC Name	NPC Class	NPC Reward
1	Danny 01	Schoolboy	992
2	Quinn 01	Cooltrainer	1824

☐ Form remains top most

NPC Pokemon Registration

NPC Name: Danny 01 Pokemon Level: 5 Add NPC Pokemon

Pokemon Name: Pikachu Experience: 25 Update NPC Pokemon

Pokemon Gender: ☐ Male ☒ Female Pokemon Give Item: Electirizer Delete NPC Pokemon

NPC Id	NPC Name	Pokemon	P. Gender	P. Level	P. Give EXP	P. Give Item
1	Danny 01	Jynx	female	31	910	null
1	Danny 01	Electabuzz	male	31	1036	null
1	Danny 01	Magmar	male	31	1109	null
2	Quinn 01	Ivysaur	male	38	1148	null
2	Quinn 01	Starmie	male	38	1685	null

Şekil 5.65: Eğitimci Bilgileri Düzenleme Programı

Oyunda ilerleyen kullanıcının geçiş bölgelerini kesen ve onları kendisiyle savaşmaya zorlayan eğitimcilerden ve bu eğitimcilerin haritanın hangi bölgelerinde bulunacağını sisteme tanıtan programdan daha önce bahsedilmişti. Yukarıdaki program ise bu eğitimcilerin özelliklerini sisteme tanıtmak amacıyla hazırlanmıştır. “1” numaralı bölümde sisteme yeni bir eğitimci eklenebilmektedir. “2” numaralı bölümde ise sisteme eklenen eğitimcinin sahip olduğu pokemonlar ve özellikleri ile bu eğitimcinin mağlup edilmesi durumunda kazanılacak ödüller sisteme tanıtılmaktadır.

Buildings To Enter

Map Name: Kanto Route Name: Viridiacity Square Id: 1005 Building Type: market Building Name: Viridian City - Market

Add Update Delete

Map Name	Route Name	Square Id	Building Type	Building Name
Kanto	Pewtercity	893	market	Pewter City - Market
Kanto	Viridiacity	949	market	Viridian City - Market

1

Market Items For Sale

Map Name: Kanto Route Name: Viridiacity Market Name: Viridian City - M Item Name: Pearl

Add Update Delete

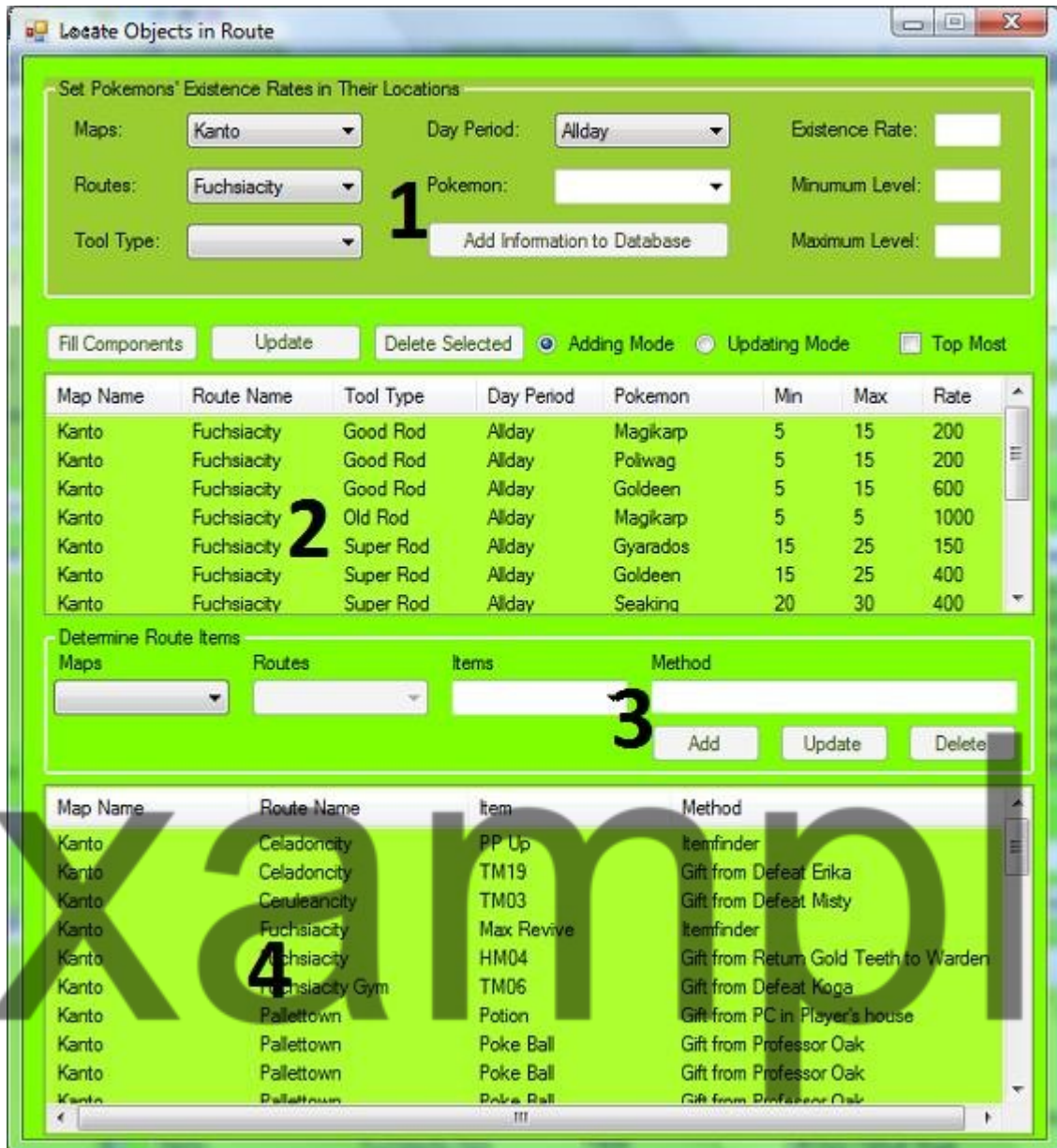
Map	Route	Market	Item
Kanto	Pewtercity	Pewter City - Market	Poke Ball
Kanto	Pewtercity	Pewter City - Market	Antidote
Kanto	Pewtercity	Pewter City - Market	Awakening
Kanto	Pewtercity	Pewter City - Market	Burn Heal
Kanto	Pewtercity	Pewter City - Market	Paralyz Heal
Kanto	Pewtercity	Pewter City - Market	Potion
Kanto	Pewtercity	Pewter City - Market	Escape Rope
Kanto	Pewtercity	Pewter City - Market	Repel
Kanto	Pewtercity	Pewter City - Market	Great Ball

2

☐ Form remains top most

Şekil 5.66: Harita Bina Bilgileri Düzenleme Programı

Oyunda kullanıcının çeşitli ihtiyaçlarını giderdiği binalar vardır. Bu binaların bilgilerinin sisteme eklenmesi gerekmektedir. “1” numaralı bölümde bu binalar, binaların tipi ve oyundaki yeri sisteme tanıtılmaktadır. Oyunda kullanıcının kullandığı eşyalar vardır. Bu eşyalar haritada ilerken kazanılacağı gibi bazı eşyalar da marketlerden satın alınabilmektedir fakat sistemdeki her markette satılan eşyalar farklıdır. Hangi markette hangi eşyaların satıldığı bilgisinin sisteme aktarılabilmesi için “2” numaralı bölüm hazırlanmıştır.



Şekil 5.67: Harita Nesne Bilgileri Düzenleme Programı

Kullanıcı oyunda ilerlerken çeşitli bölgelerde vahşi pokemonlarla karşılaşabilmektedir. Vahşi pokemonlar bir eğitmeni bulunmayan pokemonlardır. Kullanıcılar bu pokemonları ele geçirmek ya da onları yenip, kendi pokemonlarının özellikleri geliştirmek, artırmak amacıyla isterse onlarla savaşılabılır. Haritanın hangi bölgelerinde hangi vahşi pokemonlar çıkabileceğini sisteme tanıtmak amacıyla yukarıdaki program hazırlanmıştır. “1” numaralı bölümde önce vahşi pokemonla karşılaşılacak bölge seçilir. Bu bölgeler su, büyük çim veya su kenarı olabilir. Su kenarında pokemon yakalamak için olta kullanılır. Eski olta, iyi olta ve süper olta olmak üzere üç çeşit olta vardır ve bu oltaların yeteneği oranında daha güçlü pokemonlar yakalanır.[12] Daha sonra bu vahşi pokemonlar ile günün hangi zamanlarında karşılaşılabilceği bilgisi seçilir, karşılaşılacak vahşi pokemon seçilir, bu pokemonla karşılaşma oranı seçilir, en son olarak da karşılaşılacak bu pokemonun en az ve en üst seviyeleri seçilir. “2” numaralı bölümde ise sisteme eklenmiş vahşi pokemonlar bütün bilgileri ile liste halinde gösterilmiştir. Kullanıcı oyunda ilerlerken çeşitli bölgelerde çeşitli şekillerde eşyalar elde edebilmektedir. “3” numaralı bölümde bu eşyalar, bu

eşyaların bölgesi ve bu eşyaların elde ediliş yönetimi sisteme eklenebilmektedir. “4” numaralı bölümde ise sisteme eklenen bu bilgiler listelenmektedir.

Example

6.DENEYSEL SONUÇLAR

Proje kiralanan bir sunucuya yüklenmiştir. Yükleme aşamasından önce veritabanının yükleme aşamasının zahmetli olacağı tahmin edilmiştir. Fakat sunucunun Microsoft Sql Publish Wizard'ı desteklemesi sayesinde veritabanı uzun sürede ama başarılı bir şekilde yüklenmiştir. Yayınlanan proje dosyaları FTP ile uzak sunucuya aktarılıp, DbConnection sınıfındaki bağlantı değişkenin değerleri değiştirildikten sonra proje internet üzerinde çalışır hale gelmiştir.

Example

7.SONUÇ VE ÖNERİLER

Sınırlı işgücü ve zaman kısıtlaması göz önüne alındığında takımlar arkadaşları arasındaki uyum sayesinde oldukça başarılı bir proje gerçekleştirilmiştir.

Çok kullanıcıli web tabanlı oyunlarda en önemli kısıtlardan biri kaynak sıkıntısıdır. Sunucuların kullanıcılara kabul edilebilir süre içerisinde yanıt verebilmesi için efektif kodlama gereklidir. Yine çok kullanıcı oyunlarda yapılabilecek olası hilelere tedbir almak oldukça önemlidir.

Example

8.KAYNAKLAR

- [1] Techprone, "Top 5 Benefits of Online Games", 2009,
<http://www.techprone.com/top-5-benefits-of-online-games/>
- [2] Codeproject, "Priority Queue", 2004,
<http://www.codeproject.com/KB/recipes/priorityqueue.aspx>
- [3] Dr. Sanem Saniel Talay, Artificial Intelligence (BLG435E) @ Istanbul Technical University, 2009
- [4] Stuart J. Russell and Peter Norvig, Artificial Intelligence A Modern Approach, Prentice-Hall 1995, pp. 118-123
- [5] Serebii, "Market Items", 2009,
<http://www.serebii.net/pokearth/kanto/ceruleancity.shtml>
- [6] Gamedev, "Path Scoring", 2005,
<http://www.gamedev.net/reference/programming/features/astar/page2.asp>
- [7] Pokearth, "Kanto", 2009,
<http://www.serebii.net/pokearth/kanto/index.shtml>
- [8] Bulbapedia, "Trainers", 2009,
http://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon_trainer
- [9] Bulbapedia, "History Of Pokemon(Pokemon Efsanesinin Doğuşu)", 2009
http://bulbapedia.bulbagarden.net/wiki/History_of_Pok%C3%A9mon
- [10] Bulbapedia, "Experience(Pokemonların her savaşta kazandıkları tecrübe puanı hesaplama algoritması)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Experience>
- [11] Bulbapedia, "Level(Pokemonlardaki seviye sistemi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Level>
- [12] Bulbapedia, "Catch Rate(Pokemon Yakalama Algoritmasının Mekaniği)", 2008
http://bulbapedia.bulbagarden.net/wiki/Catch_rate
- [13] Bulbapedia, "Type Chart(Pokemonların birbirlerine karşı olan üstünlükleri tablosu)", 2009
http://bulbapedia.bulbagarden.net/wiki/Type_chart
- [14] Psypoke, " POKEMON DIAMOND, PEARL AND PLATINUM ITEMS LIST (Pokemon oyunlarındaki eşya listesi)",2009
<http://www.psypokes.com/dp/items.php>
- [15] Bulbapedia, "Trainer(Oyundaki eğitmenler hakkında bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Trainers>
- [16] Stuart J. Russell and Peter Norvig, Artificial Intelligence : A Modern Approach, pp.615-642 , 1995 by Prentice-Hall, Inc.A Simon & Schuster CompanyEnglewood Cliffs, New Jersey 07632
- [17] Bulbapedia, " Game mechanics(Oyunda kullanılan bütün oyun mekaniklerinin listesi)", 2009
http://bulbapedia.bulbagarden.net/wiki/Category:Game_mechanics

[18] Smogon, " The Complete Damage Formula for Diamond & Pearl(Savaş yapısındaki hasar hesaplama mekaniklerinin kaynağı) ", 2008
http://www.smogon.com/dp/articles/damage_formula

[19] Psypoke, " POKEMON STATUS EFFECTS (Pokemonların durum ve durum efektlerinin mekaniklerinin anlatıldığı sayfa)"
<http://www.psypokes.com/lab/status.php>

[20] Psypoke, "POKEMONS STATS GUIDE (Pokemonların temel güç öğelerinin mekanikleri)", 2009
<http://www.psypokes.com/lab/stats.php>

[21] Bulbapedia, " Damage modification (Kritik vuruş yapma mekaniği)", 2009
http://bulbapedia.bulbagarden.net/wiki/Critical_hit#Critical_hit

[22] Bulbapedia, " Moves with a high critical-hit ratio (Yüksek kritik vuruş yapma ihtimali olan saldırılar)", 2009
http://bulbapedia.bulbagarden.net/wiki/Category:Moves_with_a_high_critical-hit_ratio

[23] Bulbapedia, "Type(Pokemon tipleri)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Type>

[24] Psypoke, " POKEMON RED, BLUE AND YELLOW ATTACKS LISTING(Tüm saldırılar listesi)", 2009
<http://www.psypokes.com/rby/attacks.php>

[25] Serebii, "pokeearth/kanto(Projede kullanılan harita resimleri kaynakları)", 2009
<http://www.serebii.net/pokeearth/kanto/>

[26] Bulbapedia, " Special tiles (Yer yüzü şekilleri listesi)", 2009
http://bulbapedia.bulbagarden.net/wiki/Category:Special_tiles

[27] Bulbapedia, "Route 1(Harita özelliklerine bir örnek)", 2009
http://bulbapedia.bulbagarden.net/wiki/Route_1

[28] Bulbapedia, "Fishing(Balık avlama oyun mekaniği ile ilgili bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Fishing>

[29] Bulbapedia, " Poké Ball (Oyundaki pokemon topu eşyası mekanikleri hakkında bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Pokeball>

[30] Bulbapedia, " Pokémon Center(Pokemon merkezleri ile ilgili bilgi)", 2009
http://bulbapedia.bulbagarden.net/wiki/Pokemon_Center

[31] Bulbapedia, "Evolution(Pokemon evrimleşme oyun mekaniği hakkında bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Evolve>

[32] Bulbapedia, " Poké Mart (Oyundaki marketler hakkında bilgi)", 2009

http://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon_Mart

[33] Bulbapedia, " Pokédex(Oyundaki pokedex hakkında bilgi) ", 2009
<http://bulbapedia.bulbagarden.net/wiki/Pokedex>

[34] Bulbapedia, "Bag(Oyundaki çantam hakkında bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Bag>

[35] Bulbapedia, " Pokémon storage system(Oyundaki depo sistemi hakkında bilgi) ", 2009
http://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon_storage_system

[36] Legendary Pokémon, " IV Pokédex ", 2009,
<http://www.legandarypokemon.net/pokedex>

[37] Bulbapedia, "Gym(Oyundaki spor salonlarıyla ilgili bilgi)", 2009
<http://bulbapedia.bulbagarden.net/wiki/Gym>

[38] Legendary Pokémon, " IV Pokédex (Yardımcı program için kaynak kodlarının alındığı kaynak)", 2009
<http://www.legandarypokemon.net/pokedex>

[39] Legendary Pokémon, " Attack Database (Yardımcı program için kaynak kodlarının alındığı kaynak)", 2009
<http://www.legandarypokemon.net/attacks>