**BAĞLI LİSTE PROJESİ**

**Kocaeli Üniversitesi**

**Bilgisayar Mühendisliği**

**Enes Kayıklık Emirhan Deniz**

**180201010 180201048**

[**haloagaa0@gmail.com**](mailto:haloagaa0@gmail.com) **emir-han-deniz@hotmail.com**

**GİRİŞ**

Coğrafi Bilgi Sistemi (CBS), dünya üzerindeki karmaşık sosyal, ekonomik, çevresel vb. sorunların çözümüne yönelik mekâna/konuma dayalı karar verme süreçlerinde kullanıcılara yardımcı olmak üzere, büyük hacimli coğrafi verilerin; toplanması, depolanması, işlenmesi, yönetimi, mekânsal analizi, sorgulaması ve sunulması fonksiyonlarını yerine getiren donanım, yazılım, personel, coğrafi veri ve yöntem bütünüdür. Bu proje kapsamında, şehirlerin komşuluk ilişkilerini dikkate alan çok basit bir harita verisi üzerinde bağlı listelerle ilgili temel işlevleri yerine getiren bir uygulama yapılması beklenmektedir. Bu modelde her bir şehrin dört tane özelliğe sahip olduğu bilinmektedir: şehrin ismi, plaka kodu, hangi bölgede olduğu ve bu şehirden direk ulaşılabilen (komşu) şehirler. Bu proje yapımında C dilini kullandık.

**Bağlı Liste**

Bağlı liste herhangi bir tipten node’ların (düğümlerin) yine kendi tiplerinden düğümlere işaret etmesi (point) ile oluşan zincire verilen isimdir. Buna göre her düğümde kendi tipinden bir [pointer](http://bilgisayarkavramlari.com/2007/05/03/pointer-gosterici/) olacak ve bu pointerlar ile düğümler birbirine aşağıdaki şekilde bağlanacaktır. 

Linked List’in avantajı, [hafızayı](http://www.bilgisayarkavramlari.com/2008/11/07/rastgele-erisilebilir-bellek-random-access-memory-ram/) dinamik olarak kullanmasıdır. Buna göre hafızadan silinen bir bilgi için hafıza alanı boşaltılacak veya yeni eklenen bir bilgi için sadece o bilgiyi tutmaya yetecek kadar hafıza alanı ayrılacaktır.

Yukarıdaki figürde görülen bağlı listeye çok benzeyen ve yine çok kullanılan bir bağlı liste uygulaması da çift bağlı liste (doubly linked list) uygulamasıdır.



1. **Giris.java**

Başlangıç ekranıdır. Burada kullanıcının önüne 2 adet buton çıkar eğer kullanıcı PvsC butonunu seçerse Kullanıcı ile Bilgisayarın savaşabileceği GUI ekranı karşısına çıkar. Eğer CvsC butonunu seçerse 2 Bilgisayarın birbiri ile savaşabileceği GUI ekranı önüne çıkar.

1. **PvsC.java**

Bu ekranda Kullanıcı Bilgisayar ile savaşabilir. Ortada 10 adet kart ters bir şekilde üst üste durmaktadır. Kullanıcı ona tıklarsa her oyuncuya 3 er adet kart dağıtılır.

Kullanıcı sol tarafta duran 3 karttan herhangi birini seçer daha sonra Bilgisayar 3 karttan rasgele seçer bu seçilen 2 karttan hangisinin hasar puanı yüksekse o oyuncu 5 puan kazanır. Kartlar bitince ortada kalan 4

karttan her oyuncu 1 er adet çeker. Ortada da kart kalmadığı takdirde oyun biter.

1. **CvsC.java**

Bu ekranda ortada 3 adet buton bulunmaktadır.

**a.Başla**

2 Bilgisayara da 3 er adet kart dağıtılır.

**b.1 Tur Oyna**

Her iki bilgisayar da elinde bulunan desteden rastgele 1 kart seçer ve ortaya koyar.

**c.Tamamen Oyna**

Tüm işlemler otomatik olarak hızlıca gerçekleşir ve oyunun sonucu hemen belirtilir.

**Java Swing**

Swing kütüphanesi Sun Microsystems tarafından piyasaya sürülen resmi Java GUI aracıdır. Swing; %100 java ile yazılmıştır ve Java Foundation Classes(JFC) nin bir parçasıdır. Tam donanımlı masaüstü uygulamaları geliştirmek için kullanılan paketler bütünüdür. JFC; AWT ,Swing ,Ulaşılabilirlik,Java 2D ve Sürükle Bıraktan oluşur. 1997 yılında JDK 1.2 ile birlikte piyasaya sürülmüştür.

**jLabel**

Bir JLabel nesnesi; ekranda yazı, resim veya hem yazı hem resim görüntülemek için kullanılır. JButton, Swing uygulamalarında sıkça kullanılan bir bileşendir. Biz de pokemon kart fotoğraflarını ekranda göstermek için jLabel kullandık. Label’a icon atama işlemi şu şekilde gerçekleştirilir:

**jLabel Api’s**

Label ile işlerimizi gerçekleştirebilmemiz için kullandığımız apiler hakkında bilgi verelim:

1. **getIcon();**

Hangi labelden erişilmiş ise o labeldeki iconu return eder(geriye dönderir).

1. **setIcon();**

Parametre olarak gönderdiğimiz iconu labela atar.

1. **setVisible();**

Eriştiğimiz labelin görünürlüğünü ayarlar. Parametre olarak “true” gönderirsek label görünür olur , ”false” gönderirsek görünmez olur.

Oyunda toplamda 10 adet Pokemon Kartı olacak ve her oyuncuya 3 er adet dağıtılacaktır. Dağıtmadan sonra ortada 4 adet kart kalacaktır. Kullanıcı ve Bilgisayar dağıtılan 3 karttan birisini seçecek (Bilgisayar rasgele seçmelidir) ve seçilen kartlardan hasar puanı yüksek olan kartı seçen oyuncu 5 puan kazanacaktır. Daha sonra kullanıcı ve bilgisayar ortada olan kartlardan birer tane alacaktır (kartların ne olduğunu bilmeden). Ortadaki ve eldeki kartlar bitene kadar oyun devam edecektir. Oyun sonunda en yüksek puana sahip olan oyuncu oyunu kazanacaktır.

**İsterler**

1. Oyunu ve oyuncuların ellerinde bulunan kartlar görünebilecek ve takip edilebilecek bir arayüz tasarlanacaktır.
2. Destede 10 adet Pokemon kartı bulunacaktır.
3. Oyuncu ve Bilgisayar desteden rasgele 3 kart alır.
4. Oyuncu ve bilgisayar elinde bulunan üç karttan birini seçerek kapalı bir şekilde ortaya koyacaktır. Bilgisayarın seçimi rasgele olacaktır.
5. Kullanılan kartlar tekrar kullanılmayacaktır.
6. Skoru yüksek olan oyuncu oyunu kazanacaktır.



Burada hasar puanı yüksek olan Bilgisayar turu kazanır ve 5 puan eklenir.

**İstenen Sınıflar**

**Pokemon Sınıfı**

Bir Pokemon sınıfı oluşturulacak. Sınıf tanımı şunları içermeli:

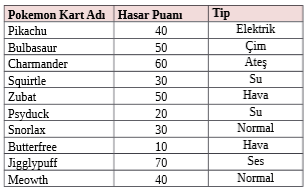
* Constructor metotları yazılacak. (Parametre içeren ve içermeyen olmak üzere en az 2 tane)
* Hasar puanı göstermesi için hasarPuanıGöster() methodu yazılacak.

**Pokemon Türlerine Ait Sınıflar (10 Tane)**

Tablo 1’de verilen pokemonların er biri birer sınıf olarak tanımlanacaktır. Sınflar şu tanımlamaları içermelidir

* Pokemon sınıfından kalıtım alacaktır.
* Constructor metotları yazılacak. (Parametre içeren ve içermeyen olmak üzere en az 2 tane) Pokemon sınıfında bulunan pokemonAdi ve pokemonTipi özelliklerine atama yapmak için süper() kullanılacaktır.
* hasarPuani özelliği bu sınıflarda tutulacaktır ve Pokemon sınıfında bulunan hasarPuaniGöster() methodu override edilecektir.
* Boolean veri tipinde kartKullanildiMi bilgisi tutulmalıdır.

Tablo-1 Pokemon hasar puanları.



**Oyuncu Sınıfı**

Bilgisayar yada kullanıcı olmak üzere oyunu oynayan iki oyuncu olacaktır. Bu iki oyuncunun farklı ve aynı özellikleri bulunmaktadır. Aynı özellikleri temsil etmek için Oyuncu temel sınıfı oluşturulacak ve bu sınıf Abstract sınıf olacaktır.

* oyuncuID , oyuncuAdi ve Skor özellikleri olmalı.
* Constructor metotları yazılacak. (Parametre içeren ve içermeyen olmak üzere en az 2 tane)
* KartListesi özelliği ile oyuncuların elinde bulunan kart listesi tutulacaktır.
* skorGoster() mehodu ile oyuncuların skorları gösterilecek.
* kartSec() methodu olmalıdır.

**Bilgisayar Sınıfı**

* Oyuncu sınıfından extend edilecektir.
* kartSec() methodu override edilecektir.
* Constructor metotları yazılacak. (Parametre içeren ve içermeyen olmak üzere en az 2 tane)

**Kullanici Sınıfı**

* Oyuncu sınıfından extend edilecektir.
* Constructor metotları yazılacak. (Parametre içeren ve içermeyen olmak üzere en az 2 tane)
* kartSec() methodu override edilecektir.

**Kaynakça**

* <https://stackoverflow.com/questions/29641381/java-read-images-from-a-file-into-an-arraylist-and-displaying-onto-a-jpanel>
* <https://www.reddit.com/r/javahelp/comments/32f2wn/add_images_to_an_arraylist/>
* <https://stackoverflow.com/questions/10263861/importing-an-image-file-adding-it-to-an-arraylist-and-then-displaying-the-imag>
* <https://www.javatpoint.com/Pic-Puzzle-Game>
* <https://stackoverflow.com/questions/15400371/adding-images-on-java-jframe-netbeans>
* <https://stackoverflow.com/questions/29452546/how-to-convert-url-to-image-to-image>
* <http://javayaz.com/?page_id=107>
* <https://codereview.stackexchange.com/questions/29630/simple-java-animation-with-swing>
* <https://wmaraci.com/nedir/gui>
* <https://www.javatpoint.com/java-swing>
* Emre Altunbilek , Atıl Samancıoğlu eğitim videoları.

****