

BLM326 YAZILIM LAB I

Proje 2

Proje: Akıllı Fareler Oyunu

- Proje mobil ortamda çalışacak cihazlar için geliştirilecektir. Bu yüzden mobil cihazlar üzerinde yazılım geliştirmek için kullanılabilen herhangi bir geliştirme platformunda uygun dil kullanılabilir.
- Proje kaynak kodları (source code) CD içerisinde raporla beraber teslim edilecektir. Proje teslimi sırasında raporunuzun çıktısı ve 2 adet cd teslim etmeniz gerekmektedir.

İNTİHAL: İnternette alnan kod parçacıkları mutlaka kod içerisinde belirtilecek ve açıklama satırı ile kaynak gösterilecektir. Aksi durumda kopya olarak değerlendirilecektir. Kopya çektiği ya da kopya verdiği tespit edilen öğrenciler SUNUMA ALINMAYACAKTIR.

ÖNEMLİ : 3 PROJEDEN 2 PROJENİN RAPORUNU TESLİM ETTİĞİ HALDE SUNUMA GELMEYEN ÖĞRENCİLER DEVAMSIZLIKTAN KALACAKTIR.

Projenin Amacı:

Projenin asıl amacı mobil ortamda uygulama geliştirmektir. Mobil ortamda geliştirilen uygulamalar gün geçtikçe yaygınlaşmaktadır ve pek çok alanda, birçok farklı uygulama geliştirilmektedir. Bu yüzden bu alanda da deneyim sahibi olmanız için mobil platformda bir oyun gerçekleştirmeniz istenmektedir.

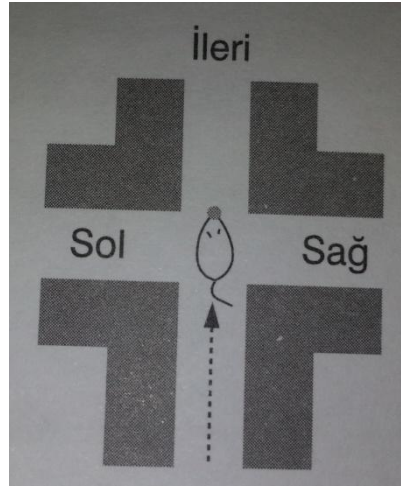
Proje Tanımı ve Yapmanız Gerekenler:

Bir laboratuarda 3 adet akıllı fare eğitilmektedir. Fareler labirent içine bırakılarak hangi yollardan gidecekleri gözlenmektedir. Eğitim sonucunda 3 farenin de birbirine benzeyen yöntemler geliştirdiği gözlenmiştir.

Farelerin hareket özellikleri:

- Fareler bir yol ayrımına geldiklerinde Şekil 1 deki pozisyonlarını ve son hamlelerini akıllarında tutmakta, daha sonra aynı konum ve yöne geri dönebilmektedir. Ardarda geçtikleri ayırım noktalarını birlikte tutabilmekte, geri

dönme durumunda her zaman en son ayırım noktasına, kaldıkları konum ve yöne dönmektedirler.



Şekil 1

- Bir ayırım noktasındaki bütün seçenekleri deneyip çıkış bulamadıklarında daha önceki ayırım noktasında, kaldıkları yerden ve karar verdikleri konumdan devam etmektedirler. Ayırım noktaları Şekil 1 de görüldüğü gibi üç seçenekli olabileceği gibi iki seçenekli (sol - sağ, sol - ileri, sağ - ileri) de olabilir.
- Daha önce geçmiş oldukları yerlerde bıraktıkları izleri hissetmekte, dolayısıyla geçmiş oldukları yollarda duvar varmış gibi davranmaktadırlar. Fareler çıkmaz sokağa gelmedikçe dönmezler. Geldiklerinde de hatırladıkları ve hala denemedikleri seçenek olan son konuma doğrudan zıplarlar.
- Fareler kendi konumlarına göre davranmakta, küresel bir kuzey, güney, batı, doğu ayrımını yapamamaktadırlar. Yani çizime göre aşağı yönde ilerleyen bir fare için ilerisi, aşağıya doğru, sol taraf ise çizime göre sağ yönde olacaktır.
- Bu 3 fare aynı yöntemi öğrenmiş olmakla birlikte ayırım noktasındaki tercihleri birbirinden farklı olarak gelişmiştir. Bir fare önce düz giden yolu denemeyi tercih ederken başkası önce soldaki yolu denemektedir.
- Sonuçta bütün yolları çıkış bulana kadar deneseler de buldukları yollar ve birden fazla çıkış varsa bulabildikleri çıkışlar değişebilmektedir.
- Buna göre farelerin denedikleri yönler aşağıdaki gibidir:

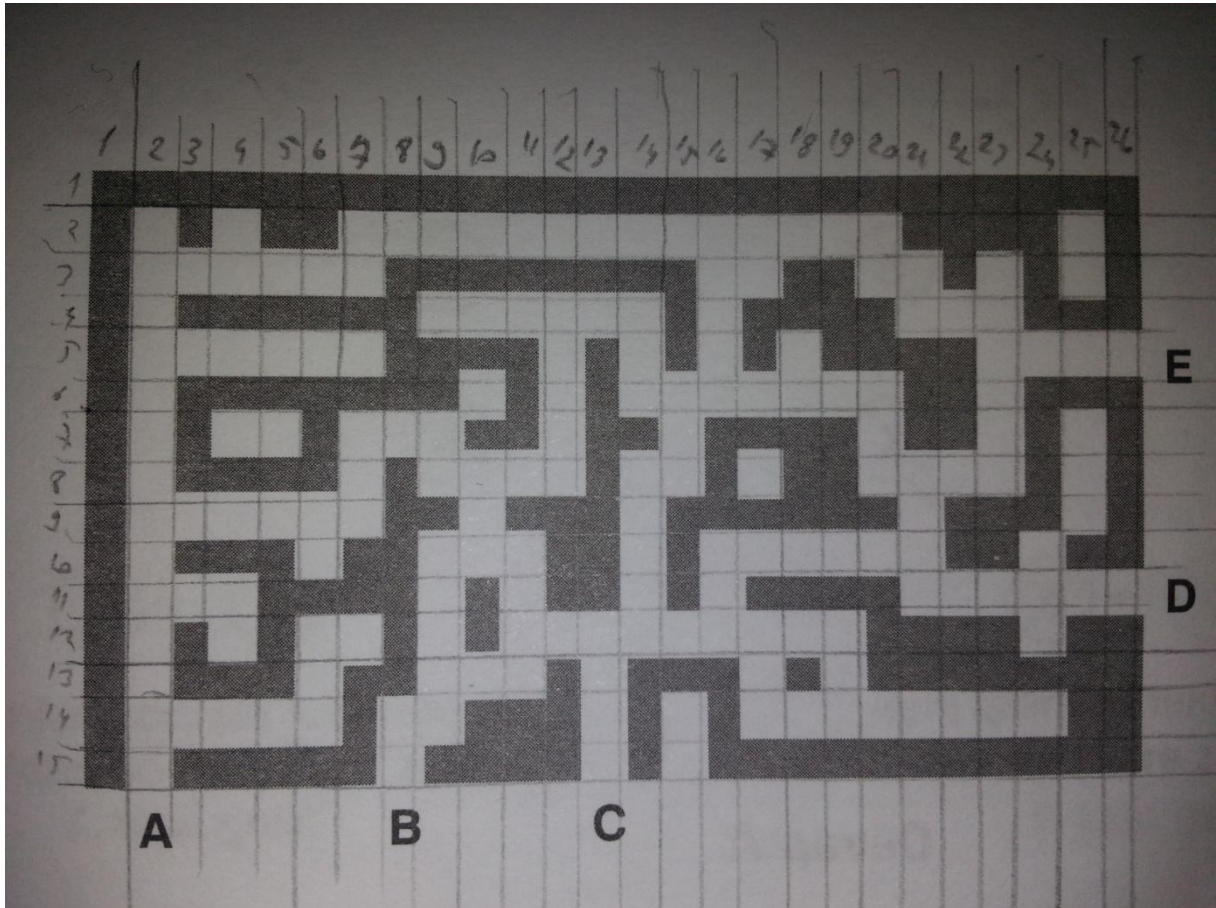
Fare 1: Önce sol, sonra ileri, sonra sağ yönler.

Fare 2: Önce ileri, sonra sol, sonra sağ yönler.

Fare 3: Önce sağ, sonra sol, sonra ileri yönler.

Farelerin ilerleyeceği labirent:

- Geliştireceğiniz uygulamada kullanacağınız labirent ve kapı (A, B, C, D, E) örneği Şekil 2 de verilmiştir. 0 ve 1 rakamlarını kullanarak (Örneğin duvarlar 0, ile gidilecek yollar yani boşluklar 1 ile gösterilip labirent oluşturulur. Şekil 2 deki labirent 15X26 lık matris şeklindedir. Bazı koyu renk kareler (duvarlar) içinde ince beyaz alanlar kalmıştır, beyaz alanları dikkate almayınız bu değerler '0' yani duvardır (örn:(2,5), (12,20), (5,10), (7,10).. vb.)) kendi labirentinizi oluşturup (matris boyutu ve kapı sayısı size bağlı, Şekil 2 deki labirentten bağımsız olup, kapılar ve duvarlar farklı yerlerde olabilir) **benioku.txt** belgesine kaydetmelisiniz.
- Mobil uygulamanızda bu labirent bilgisi **benioku.txt** belgesinden okutturulacak ve fareler bu labirentte gezinme işlemlerini yapacaklardır.



Şekil 2

Verilen bu bilgiler doğrultusunda;

Fare 1-> sol, ileri, sağ

Fare 2-> ileri, sol, sağ

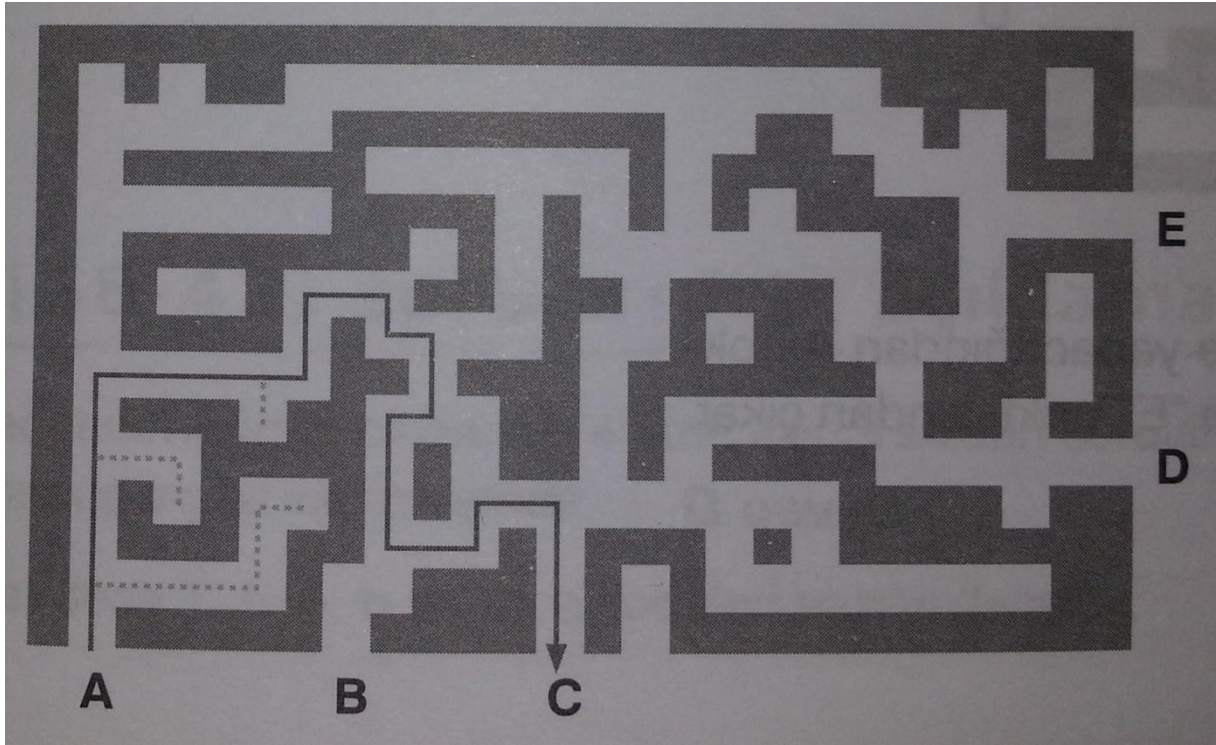
Fare 3-> sağ, sol, ileri

şeklinde ilerler. Yani yol ayırımına geldiklerinde, örneğin 1. fare; önce sola gitmeyi dener, yol kapalıysa ileri gitmeyi dener ve orası da kapalıysa sağa gider. Her 3 farede çıkmaz sokağa girerlerse geri dönerler ve diğer yolu tercih ederler. Aynı zamanda daha önce geçtikleri yerlerde kendi kokularını algılayıp çıkmaz sokakmış gibi geri döner ve diğer yolu tercih ederler.

Örnek çözümler:

Örnek 1: 3 numaralı fare A kapısından bırakılırsa hangi çıkışı bulduğuna bakalım;

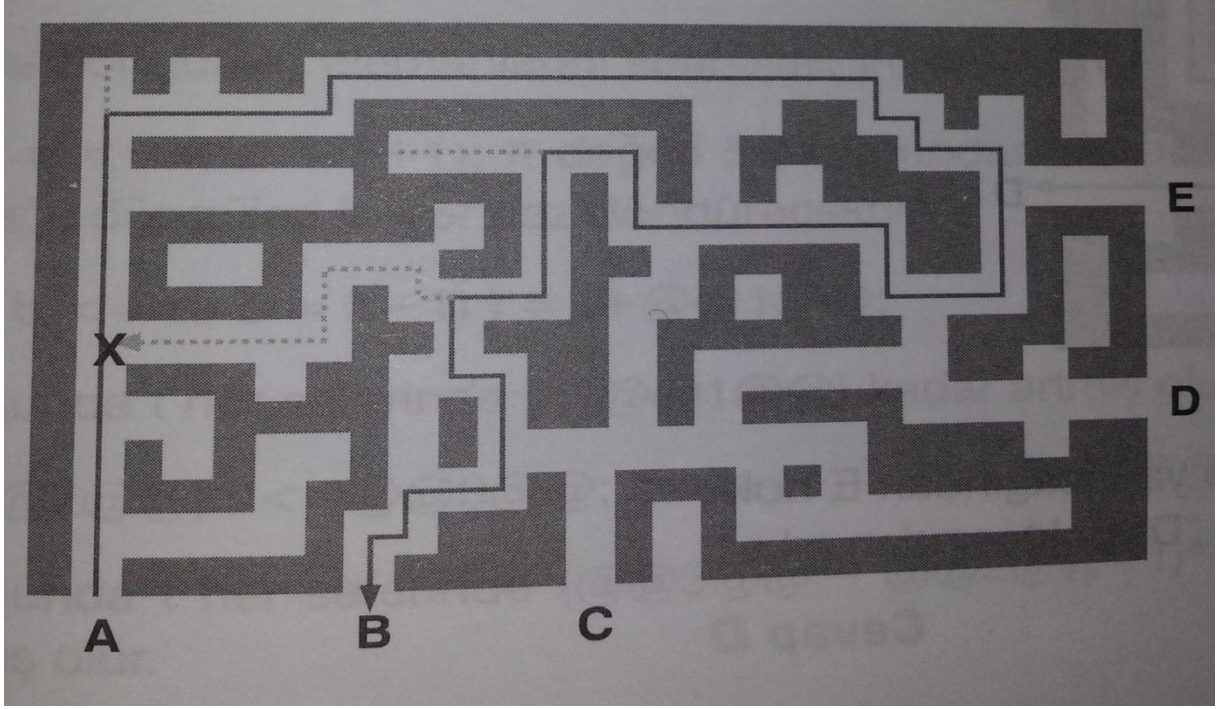
Fare 3 yol tercihini sağ-sol-ileri şeklinde yapar. A noktasından çıkar. Yol ayırımına geldiğinde yol tercihine göre ilerler ilk tercihinde yol kapalıysa geri dönüp ikinci tercihini uygular. Şekil 3 de verilen yolu izler ve C kapısından çıkar.



Şekil 3

Örnek 2: 2 numaralı fare A kapısından bırakılırsa hangi çıkış kapısında olacağına bakalım;

Fare 2 yol tercihini ileri, sol, sağ şeklinde yapar. A kapısından çıktığında yol tercihlerini sırasıyla uygulayarak ilerler fakat 'x' noktasına geldiğinde kendi kokusunu fark ettiği için bir önceki yol ayırımına geri dönüp ikinci yol tercihini uygular. Bu nedenle B kapısından çıkar. (Bknz. Şekil 4)



Şekil 4

Projeden Beklenenler:

- Mobil uygulamanızda kullanıcı verilen kapılardan birini giriş seçecek ve Fare 1, Fare 2, Fare 3 için hangi kapılardan çıkış yapılacağı uygulamanız tarafından bulunacaktır.
- Sırasıyla her fare için , bu farelerin hangi çıkış kapısından çıkacağı bulunduğundan sonra izlenen yol ekrana çizdirilecektir. (Bknz. Şekil 3 ve Şekil 4)
- Kullanıcı tarafından seçilen giriş kapısı için **en kısa yol algoritması** (Dijkstra, Bellman Ford ve Floyd algoritmaları en çok bilinen en kısa yol bulma algoritmalarıdır. Bu algoritmalardan birini yada farklı bir en kısa yol bulma algoritması kullanabilirsiniz. Algoritma seçimi size bırakılmıştır.) kullanılarak minimum yol maliyetli en kısa çıkış kapısı belirlenecektir. Kullanıcının seçtiği bu giriş kapısından ayrı ayrı giren Fare 1 , Fare 2 ve Fare 3 farelerinin hangi kapılardan çıktığı ve bu kapılara giderken toplam yol maliyetleri ayrı ayrı hesaplanacaktır. 1., 2. ve 3. farelerin çıkış kapılarına giderken hesaplanan toplam yol maliyetleri; en kısa çıkışın toplam yol maliyetiyle kıyaslanıp listelenecektir. (Her birim karenin maliyeti 1 birim olarak alınacaktır. Toplam maliyet birim cinsinden bulunacaktır.)

Demo Günü Teslim Etmeniz Gerekenler :

Kodlama ve uygulamayı yaptığınız ortam (IDE) hakkında sorulacak olan sorulara hazırlıklı olmanız beklenmektedir.

Demo günü teslim edilecekler:

2 adet Cd, raporunuzun çıktısı (hard-copy).

CD içinde olması gereken dökümanlar:

- Projenizin kaynak kodları
- Proje'nin rapor çıktısı (soft-copy).
- CD içinde kodunuz, programınızın çalıştırabilmesi için gerekli olan instructionlar (readme.txt) ve raporunuz (Daha önce verilen rapor formatına göre hazırlamalısınız).
- Proje raporunuzu proje dokümanlarının içinde yer alan rapor formatına uygun olarak hazırlamanız gerekmektedir.

Son Teslim tarihi: 11.11.2013 Pazartesi günüdür. Projeyi kime teslim edeceğiniz ve sunum tarihleriniz daha sonra duyurulacaktır.