

# Prolog ile Bilgi Temsili ve Çıkarsama

Umut BENZER 05-06-7670 <http://www.ubenzet.com>

Ege Üniversitesi Bilgisayar Mühendisliđi

Yapay Zeka

**17.10.2011**



## İçindekiler

Problemin Tanımı.....	3
Oyunun Amacı ve Oynanışı.....	3
Oyunu Kazanmak .....	3
Programın Özellikleri.....	3
Predicates & Clauses .....	3
Ekstra Özellikler, Orijinal Yönler ve Diğer Anahtar Noktalar .....	4
Dinamik Veri .....	4
Rastgele Sayı Üretimi .....	4
Kullanıcıdan Veri Almak.....	4
Programı Bitirmek .....	4
Programın İşletimi.....	5
Yararlanılan Kaynaklar .....	6
Tablo .....	6

## Problemin Tanımı

Uygulama klasik bir "Snakes and Ladders" veya Yılanlar ve Merdivenler oyunundan esinlenilerek oluşturulmuş bir zar at-ilerle oyunudur.

Prolog ile yazılmıştır.

## Oyunun Amacı ve Oynanışı

Oyunda amaç, başlangıç karesi olan 0. kareden yola çıkmak ve sırasıyla Bornova, Karşıyaka, Konak, Balçova, Narlıdere ve Urla'yı geçerek Teknokent'e varmaya çalışmaktır.

İki oyuncu ile oynanır. Oyuncular sırayla zar atarlar ve attıkları zar kadar ilerlerler. Geldikleri kare

- Normal bir kare olabilir. Bu karede kalırlar.
- "Olumlu" bir kare olabilir. Oyuncu karedeki değer kadar ileri gider.
- "Olumsuz" bir kare olabilir. Oyuncu karedeki değer kadar geri gider.

Eğer bir oyuncu eksili karelere düşerse, program bu oyuncuyu sıfırdan kareye ilerletir.

## Oyunu Kazanmak

Teknokent'e ilk varan kazanır.

## Programın Özellikleri

Programda temel olarak iki hedef vardır: Oyuncuları AI ve Oyna. Program diğer tüm işleri bu ikisini gerçekleştirebilmek adına yapar.

## Predicates & Clauses

İlerleyen kısımda predicates ve neyi temsil ettikleri/ne amaçla kullanıldıkları anlatılmıştır.

### kare(kareno, kareadi, karepuan)

Oyundaki her bir kareyi (ilerlenecek adım, monopoldeki kareler gibi düşünülebilir) temsil etme için kullanılır. Her karede aşağıdakiler bulunmaktadır:

- Kare numarası, integer, karenin kaçınıcı sırada olduğunu temsil eder.
- Kare adı, string, karenin adı, ne olduğu gibi bilgileri tutar, bunlar oyunda ekranda gösterilir.
- Kare puanı, integer, bu kareye gelenlerin ekstradan kaç kare ileri/geri gideceğini temsil eder.

### oyunculari\_al

Kullanıcıdan oyuncuların isim bilgilerini alır. Kullanıcıdan gelen yeni bilgiler sisteme eklenmesi ve çıkarsamada kullanılan birer gerçektirler. (fact) Bu yüzden gelen bilgiler **asserta** ile sisteme çalışma zamanında dinamik olarak eklenir.

### oyna(oyuncuno)

Bir oyuncunun oynamasını temsil eder. Clause'lar içerisinde her bir oyuncu için iki tane oyna eşitliği bulunur. "Oynamak" için "Zar atmak" ve "İlerlemek" gerekmektedir. Bunlar ve ekrana veri yazdırma ile alakalı başka bazı işlemlerin hepsi gerçekleşmek zorunda olduğu için birbirine and (,) ile bağlanmışlardır.

### oyuncu(oyuncuno, isim)

Oyundaki bir oyuncuyu temsil eder. Bu predicate, kullanıcıdan dinamik olarak alınan verilerin tutulduğu DATABASE kısmındaki xoyuncu ile eşleştirilmiştir. *Doğrudan xoyuncu kullanılamamıştır*

*çünkü, Clauses kısmında Database içinde bulunan Fact'lerin kullanılması derleme hatasına sebep vermektedir. Oyuncu numarası ve oyuncunun isim bilgileri ile temsil edilir.*

### **bulunmaktadır(oyuncuno, kareno)**

Bir kullanıcının bulunduğu bir kare bilgisine karşılık gelir. Kullanıcılar attıkları "zar" doğrultusunda kare değiştirdikçe bulunmaktadır ifadeleri çalışma zamanında otomatik olarak silinir ve güncel haliyle yeniden yaratılır. "bulunmaktadır" "xbulunmaktadır" factine karşılık gelmektedir.

### **zarat(zarsayisi)**

zarsayisi kısmında gönderilmesi beklenen free variable için uygun bir değer/çözüm olup olmadığını araştırır. Uygun çözüm 1-6 arasında bir değerdir.

### **oyuncuilerlet(oyuncuno, zarsayisi)**

Oyuncu numarası verilen oyuncu ve bulunduğu konum ayrı ayrı çıkarılır. Zar sayısı hesaba katılarak kullanıcının yeni konumu hesaplanır ve oyuncunun yeni karede olduğu gerçeği (fact) sisteme dinamik olarak eklenir.

### **kareilerletmekurali(kareno, nekadar)**

Numarası verilen kare, daha önceden Prolog 'da kayıtlı bilgiler kullanılarak çıkarılır. Buradan, karenin kaç puan olduğu (ilerlemeye etkisi) öğrenilerek bu bilgi sonuç olarak "ne kadar" ile sisteme verilir.

## **Ekstra Özellikler, Orijinal Yönler ve Diğer Anahtar Noktalar**

### **Dinamik Veri**

DATABASE kısmı, asserta(fact) ve retract(fact) ile çalışma zamanında oluşan/alınan yeni bilgiler sisteme eklenerek yapılacak çıkarsamalar değiştirilmiştir.

Kullanıcının bulunduğu konum (oyunun temeli) bu çıkarsamalara (dinamik olarak fact ekleyip çıkarmaya) dayanmaktadır.

Dinamik işlemlere örnek oyuncuilerlet(Oyuncuno, Zarsayisi) :- clauseında bulunabilir.

### **Rastgele Sayı Üretimi**

zarat(Sayı) :- clauseında prolog tarafından hazır olarak sunulan rastgele sayı üretme fonksiyonu random(int, int) kullanılmıştır.

### **Kullanıcıdan Veri Almak**

Programda dinamik atamalar yapılacağından ve program aşama aşama ilerleyeceğinden kullanıcıdan veri alınması gerekmektedir. Bu, Prolog'da hazır bulunan readchar(char) ve readln(string) fonksiyonları ile gerçekleştirilmiştir.

### **Programı Bitirmek**

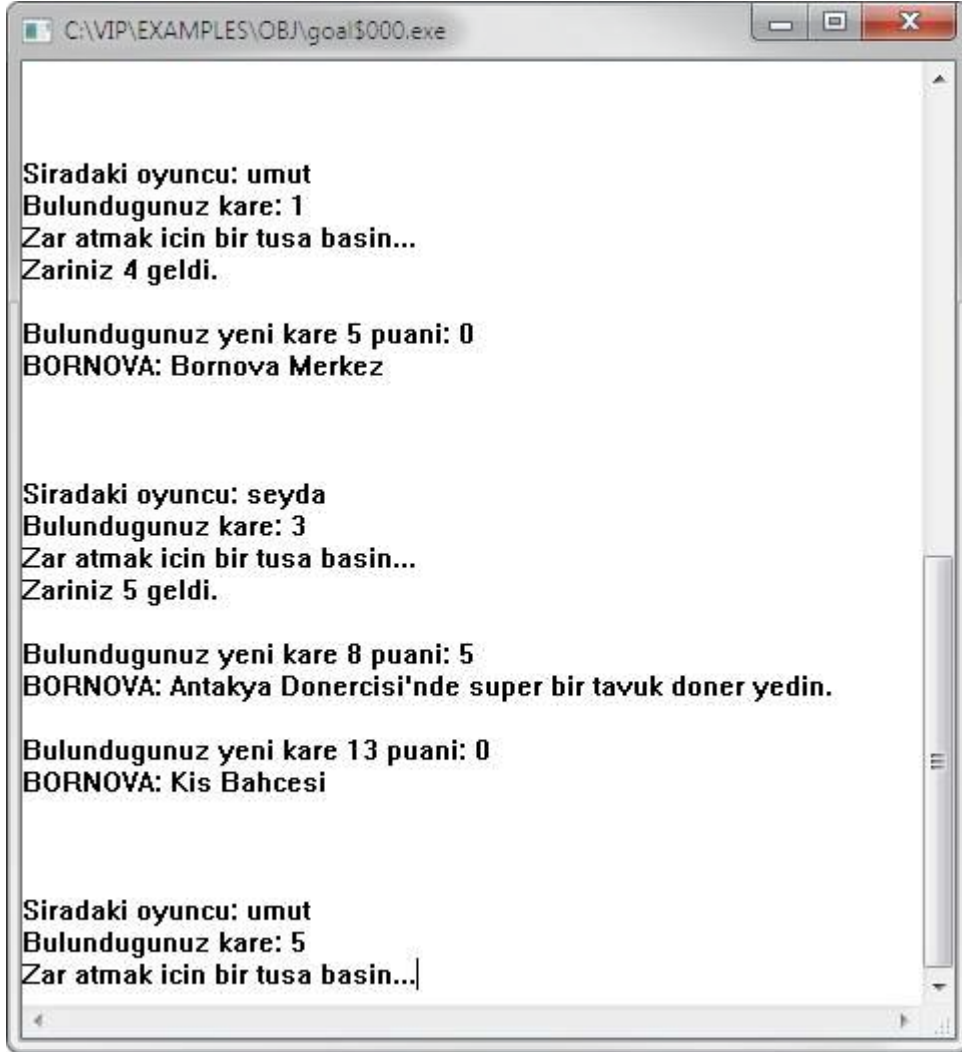
Program doğası gereği, GOAL'ü gerçekleştirmeden kapanması gerekebilir. Bu yüzden exit kullanılmıştır.

## Programın İşletimi

Program öncelikle birinci ve ardından ikinci oyuncunun ismini isteyecektir. İsimleri girdikten sonra sırayla iki oyuncu oynamaya başlar. Oyuncular ilerlemek için tek yapmanız gereken zar atmaktır ve bu da herhangi bir tuşa basılarak yapılabilmektedir.

Şanslı olan Teknokent'e ilk varan olur ve kazanır.

Oyun sona erer.



## Yararlanılan Kaynaklar

Bu projede derste anlatılanlar, dersin slaytlarından ve Visual Prolog ile gelen LANGUAGE.DOC dosyasından fazlasıyla yararlanılmıştır. Bunlar dışında, kod örneği ve Prolog mantığı konusunda yardımcı olarak aşağıdaki kaynaklardan faydalanılmıştır:

- [http://wiki.visual-prolog.com/index.php?title=Programming\\_Pragmatics](http://wiki.visual-prolog.com/index.php?title=Programming_Pragmatics)
- <http://www.ifcomputer.co.jp/en/manuals5.2/uguide/node8.html>
- <http://www.cdf.toronto.edu/~csc384h/fall/simple-prolog-examples.html>
- <http://www.ezy-software.com/ezyprolog/readchar.htm>
- [http://wiki.visual-prolog.com/index.php?title=Visual\\_Prolog\\_7\\_equivalents\\_for\\_Visual\\_Prolog\\_5](http://wiki.visual-prolog.com/index.php?title=Visual_Prolog_7_equivalents_for_Visual_Prolog_5)
- <http://www.anselm.edu/homepage/mmalita/culpro/index.html>
- <http://www.anselm.edu/homepage/mmalita/culpro/arith.html>
- <http://www.anselm.edu/homepage/mmalita/culpro/add0.html>
- [http://en.wikipedia.org/wiki/Prolog#Rules\\_and\\_facts](http://en.wikipedia.org/wiki/Prolog#Rules_and_facts)
- <http://www.fraber.de/bap/bap194.html>

## Tablo

İstenen Özellik	Puan	Var	Açıklama
Orijinal bir kod	3	X	İnternette buna benzer (Snakes and Ladders da dahil) başka kod örneği bulunamadı. Kodların tamamı sıfırdan kendi ürünüdür.
Rapor	3	X	Var.
Recursion Kullanımı	1	X	Var. oyna ve oyuncuilerlet ifadeleri birbirlerini sürekli çağırırlar.
Liste Kullanımı	1		Yok.
Ekstra Özellikler	2	X	Var. Veri alma ve dinamik olarak kural eklemek.
50 satırdan az kod	-1		~170 satır, 100'ü bilgi amaçlı (kare clauses), kalan ~70'i kod sayılabilir.
<3 farklı kalıp	-1		8 tane.