

Önyükleme Eğitmen Oryantasyonu – Isınma Egzersizi

(Bu egzersizi <http://onyukleme1.yakinokul.org/Egitmen-Oryantasyon-Isinma.pdf> adresinde bulabilirsiniz)

plus.google.com adresine gidin, onyukleme.1@gmail.com adresiyle giriş yapın.

we.scheme.org sayfasını açın ve “Log in” düğmesiyle Google bağlantısını onaylayın. Daha sonraki ekranda “Start a new program seçin”

Ekranın alt yarısı “etkileşim penceresi”dir. Fareyi oraya tıklayıp imlecin orada olduğuna emin olduktan sonra:

```
123
(+ 2 2)
(circle 100 "solid" "green")
```

komutlarını yazıp herbirinden sonra enter'a basın. İlkinde bir sayısal değer ifade etmiş, diğer ikisinde birer fonksiyon çalıştırmış (yani değerlendirmiş, değerini hesaplatmış) oldunuz. Bilgisayar her ifadenin değerini bulup gösteriyor. Sayısa ifade için size sayıyı geri gösterir. Fonksiyonlar ise verdiğimiz girdilere göre çıktı üreten makineler gibidir. Çağrının başı ve sonu parantezlerle (...) belirlenir. Parantez içindeki ilk işaret veya kelime fonksiyonun ismidir, çalıştıracığımız fonksiyon sistemin bildiği bir fonksiyon olmalı. Parantezin boşlukla ayrılmış kalan parçaları ise çağırdığımız fonksiyonun ihtiyaç duyduğu parametrelerdir, bunlar sayılar, metinler, görseller olabilir.

“Images” linkini tıklayın, Google Drive'a yüklü görsellerden birini seçip “select” deyin. Sistem etkileşim penceresinde şu fonksiyon çağrısını sizin için oluşturur:

```
(bitmap/url "https://drive.google.com/uc?export=download&id=0B5UV1TeaWsiUSUpMYIU0NWE5Qlk")
```

Enter'a basıp fonksiyonu çalıştırın. Google Drive'daki görselleri dersin ilerleyen kısımlarında bu şekilde kullanacaksınız.

Bir veya birkaç makinenin çıktısını başka bir makinenin girdisi olarak kullanabiliriz. İki örnek:

```
(+ (* 13 7) (/ 23 6))

(overlay (circle 10 "solid" "blue")
  (rectangle 30 8 "solid" "red")
)
```

İkinci örnekte “overlay” fonksiyonuna parametre olarak ihtiyaç duyduğu görseller olarak başka fonksiyonların çıktılarını verdik. Parantezler işlerin birbirine karışmasını engelliyor, açılan her parantez kapatılmalı. Dersler sırasında böyle kombine fonksiyon çağrılarının işleyişini “Değerlendirme Çemberi” kavramıyla öğrencilere aktaracaksınız.

Egzersiz: Google Drive'daki görselin üzerine bir çember yerleştirin.

Bilgisayara yeni şeyler tanımlayarak öğretmeye başladığımızda tam olarak programlamaya girmiş oluyoruz. Tanımlamaları (ve bazen de fonksiyon çağrılarını) ekranın üst yarısını oluşturan “tanımlar penceresi”nde gireceğiz. Oraya fareyle tıklayıp imlecin orada olduğundan emin olun. Şunu girin:

```
(define bir-çember (circle 100 "solid" "green"))
```

“Run” tuşuna basınca sistem yeni tanımlı öğrenmiş olur. Şimdi tekrar “etkileşim penceresi”ne geçip:

bir-çember

yazın. Bilgisayar öğrendiği şeye değerlendirecektir.

Yeni bir fonksiyon öğretmek o fonksiyonun ismini ve parametrelerini belirttikten sonra fonksiyonun üreteceği değeri (yani yeni makinenin işlevini) bilinen işlemler cinsinden tarif etmekle yapılır. Fonksiyon tanımlamasında kullanacağınız sözdizimi şöyledir:

```
(define (fonksiyon-ismi parametre-ismi-1 parametre-ismi-2 ...)  
  ...verilen parametreleri bildik fonksiyonlara besleyerek yeni fonksiyonun sonucunu  
  üretecek fonksiyon çağrıları... )
```

Sonra şu örneği girin:

```
(define (yeşil-çember yarıçap)  
  (circle yarıçap "solid" "green"))
```

Şimdi "Run" deyin. Eğer bir hata (eksik parantez, boşluk, vb.) yapmadıysanız artık sistem sizin tanımladığınız "yeşil-çember" fonksiyonu da öğrenmiş oldu. Şimdi tekrar "etkileşim penceresi"ne geçin ve fonksiyonunuzu çağırın:

```
(yeşil-çember 100)
```

Bu komutu "tanımlar penceresi"nde fonksiyon tanımının altına eklerseniz, "Run" işlemi sadece fonksiyonu tanımlamakla kalmaz bir kez de çağırır ve sonucu görürsünüz.

Egzersiz: Kırmızı bir kuru içinde yeşil çember çizdirecek bir fonksiyon tanımlayın ("square" ve "overlay" komutlarından yararlanabilirsiniz). Tanımlar penceresinde fonksiyonunuzu birkaç farklı parametreyle çağırın.

Şimdi "Save" düğmesini kullanarak yaptığınız tanımları bir program ismi altında saklayabilirsiniz. Daha sonra herhangi bir bilgisayardan tekrar giriş yaptığınızda programınıza ulaşp onu geliştirmeye devam edebilirsiniz.

Animasyonlar

Sistemin saniyede birkaç kez sizin belirlediğiniz bazı fonksiyonları çağırması sonucu animasyon yapılır. Aşağıdaki örneği "tanımlar pencere"sine girip çalıştırın

```
(define (dünyayı-çiz durum)  
  (circle durum "solid" "blue"))
```

```
(define (dünyayı-değiştir durum)  
  (modulo (+ durum 1) 50))
```

```
(big-bang 20  
  (to-draw dünyayı-çiz)  
  (on-tick dünyayı-değiştir))
```

Öğrencilerinize 4. üniteye yaptıracağınız "roket" animasyonu bu sistemin basitleştirilmiş ve özelleştirilmiş bir versiyonunu kullanmaktadır.

Oyunlar

Bu ders boyunca geliştireceğiniz oyun esasen bir şablonla oynayarak gerçekleştirilecek. Şu linkten şablonu açın: <http://www.wescheme.org/openEditor?publicId=RHBJYscAWj> . Programı çalıştırırsanız hiçbir şey olmayacaktır. Programın içinde şu kısmı bulun:

```
(define (update-target x)  
  x)
```

ve aşağıdaki gibi olacak şekilde değiştirin:

```
(define (update-target x)
  (- x 20))
```

Şimdi “Run” diyerek çalıştırın. Bu değişiklik animasyondakine benzer bir etki yaratıyor. Kullanacağınız oyun sistemi arkaplanda bir animasyon sistemi çalıştırmaktadır.

Egzersiz: “(define PLAYER...” diye başlayan satırı bulun ve onun yerine Google Drive'a yüklü görsellerden birini kullanın.

Öğrenciler burada yaptığınız gibi şablonun parçalarını kendi tasarımlarına göre değiştirerek oyunu yazacaklardır.