GNU Pardus-Linux.Org eDergi

Nisan'10 - Sayı'18

Turhan Selçuk'u Sevgiyle Uğurluyoruz...

İçindekiler

Merhaba **Özgür Yazılım** Dostları,

Türkiye'de İnternet'in delikanlı yaşı olan 18'ine bastığı bu günlerde, dergimiz de 18'ine basmış bulunuyor.

Eminim ki yine sizleri şaşırtmadık ve dergimizi ay başında değil de biraz gecikmeli olarak yayınladık. Ne de olsa Pardus'a gönül veren ve onu destekleyen bir topluluğuz, biraz olsun Pardus'tan çekmiş olmamız garipsenmez sanırım ;)

Türkiye'de İnternet'in delikanlı yaşına bastığından bahsetmiştik. Ancak delikanlı olmasına rağmen, Sayın Mustafa Akgül'ün deyimiyle, İnternet alanında hala Donkişot, Devekuşu ve Harakiri ile özetliyebileceğimiz, adaletsizlik, başını kuma gömme ve kendize zarar vermeye devam ediyoruz. İnternet yasakları konusunda toplumsal tepkimizi daha gür bir şekilde vermemiz gerektiğini düşünüyorum ve umuyorum ki Avrupa İnsan Hakları Mahkemesi'ne uzanan bu konu, olumlu bir şekilde sonuçlanır.

Geçtiğimiz hafta, İstanbul Bilgi Üniversitesi'nde Linux Kullanıcıları Derneği ve ev sahibi üniversite tarafından Özgür Yazılım ve Linux Günleri düzenlendi. Etkinlikten birkaç gün önce bir tanıtım vidyosu hazırlandı. Tanıtım vidyosunda Stallman'ın çok güzel bir cümlesi yer alıyor: "*Free Software is a technical thing. But issue of the Free Software is not a technical issue. It's an ethical, social and political issue.*" Yani, "Özgür Yazılım teknik bir terimdir. Fakat Özgür Yazılım'ın konusu teknik bir konu değildir. Ahlaki, sosyal ve politik bir konudur." Oldukça beğendiğim ve defalarca dinlediğim bu sözler üzerine yorumu size bırakıyorum.

Etkinlikte yerli ve yabancı birçok konuşmacı sunumlar yaptı, çalıştaylar düzenlendi. Fakat arkadaşlarımla konuştuğum kadarıyla etkinlik için seçilen tarih sanki iyi düşünülmemiş gibi durdu. Çünkü duyduğum kadarıyla İstanbul Bilgi Üniversitesi de dahil birçok üniversitenin sınav takvimi ile Açık Öğretim Fakültesi sınavları bu etkinliğin takvimi ile çakışıyormuş. Katılamadım ancak fotoğraflardan gördüğüm kadarıyla güzel bir etkinlik olmuş, tüm LKD ekibini saygıyla selamlıyorum.

Dergimize gelecek olursak, bu sayıda da beğeneceğinizi umduğumuz birçok konuya yer verdik. Diğer sayılara göre tek eksiğimiz ise bu sayı için röportaj hazırlayamamış olmamız.

Bu sayımızda iki farklı oyuna yer verdik. Bunlardan ilki VioletLand ve ikincisi Wesnoth. Sevgili Hamit, bizlere tanıttığı ChromeOS'nin yanında bu iki oyun ile güzel bir iş çıkardı. Bunun yanında Sevgili Burak, Gentoo üzerine bir yazısını bizimle paylaştı ve Sevgili Mehmet Inkscape ile canlı yol efektlerinin nasıl yapılabileceğini anlattı. Sevgili Onur Pardus'ta Scilab kurulumu ve kullanımına giriş yaparken, Muslu Django ile uygulamalar serisine, Armağan ise C dili ile yazılım geliştirme serisine devam etti. Sevgili Kemal Özgür Yazılım, Özgür Toplum kitabının tanıtımını bizlerle paylaşırken, Aydın eğitim üzerinden Özgür Yazılım'a bağladı ve kalemini yine konuşturdu. Bendeniz ise DNS sunucu testleri ve Pidgin Facebook ikilisini sizlerle paylaşmaya çalıştım.

Dergimizle sizleri başbaşa bırakırken, kapakta görmüş oluğunuz **Abdülcanbaz** karakterinin yaratıcısı, büyük karikatürist **Turhan Selçuk**'u saygı ve sevgi ile uğurluyor, dergimizin bu sayısını kendisine ithaf ediyorum.

Saygılarımla.

Erdem Artan
erdem@pardus-linux.org

Giriş

Domain Name System [DNS] (Alan Adı Sistemi), dağıtık yapıda bir veritabanıdır. Bu sistem, makine isimlerini IPv4 (ya da ipv6) adreslere, ya da IPv4 adreslerini makine isimlerine çevirmeye yarar. Ayrıca, bir alan için gerekli e-posta sunucusunun adreslerini, ya da alan adı sunucularının hangi IP adreslerinde bulunduğunu -daha teknik bir ifade ile RR 'ları (Resource Records - Özkaynak Kayıtları)- tutar. Örneğin; pardus-linux.org adresine karşılık gelen IPv4 adresinin 139.179.139.187 olmasının bulunması. [0]

Bu nedenle, İnternet sitelerine bağlanırken, tepki süresini asgari seviyede tutmak için, hızlı DNS sunucuları tercih edilmelidir. Bilindiği üzere, İnternet Servis Sağlayıcı'ların ön tanımlı olarak sunduğu DNS adreslerinin yanında, çeşitli üniversiteler, Google, OpenDNS, UltraDNS gibi oluşumların sunduğu DNS adresleri bulunmaktadır. Peki bunların hangisi en hızlı? Hangisinin erişimi daha mümkün? Hangileri sansür uyguluyor?

İşte, Namebench bu soruların cevabını verebilecek olan bir uygulama. Namebench, ön tanımlı olarak gelen DNS adreslerinin yanı sıra, kullanıcı tarafından eklenen DNS adresleri arasında da çeşitli testler yaparak, sonucu grafiksel bir rapor olarak kullanıcıya sunan bir uygulama.

Uygulama Python ve Tkinter ikilisi kullanılarak yazılmış. Raporu ise bir html dosyası olarak sunuyor.

Kurulum

Uygulamanın kurulumu için, önceki sayılarımızda olduğu gibi, iki farklı yol sunacağız. Bunlardan ilki Kaynak Koddan Kurulum, ikincisi ise Paket Yöneticisi ile Kurulum.

Kaynak Koddan Kurulum

Namebench'in kaynak kodları, http://code.google.com/p/namebench/ adresinden indirilebilir. Kurulumu ise, "setuptools" paketinin kurulu olduğu bir GNU/Linux dağıtımında oldukça kolaydır. Yönetici yetkileri ile şu komutun verilmesi yeterlidir:

```
python setup.py install
```

Yalnız bu noktada küçük bir uyarı yapmakta fayda var. Namebench bu şekilde kurulursa, GNU/Linux dosya hiyeraşisine uygun olmayan şekilde bazı yüklemeler yapabiliyor. Örneğin, `/usr` dizini içinde bir `namebench` dizininin oluşturulduğunu görebilirsiniz. Ayrıca yüklediği üçüncü parti Python modülleri, tarafınızdan daha önce yüklenmiş olan üçüncü parti modüllere zarar verebilir ya da onlarla çakışabilirler.

Paket Yöneticisi ile Kurulum

Namebench'i, Pardus 2009'a, Paket Yöneticisi ile de kurabilmeniz mümkün. Bunun için öncelikle, Pardus-Linux.Org tarafından sunulan "P2009-free" deposunun, Paket Yöneticinize eklenmiş olması gerekmektedir. PiSi Paketi depolarımız hakkında ayrıntılı bilgiyi, forumdan öğrenebilir, paket isteklerinde bulunabilirsiniz.

P2009-free PiSi Paketleri deposu eklenmiş bir bilgisayarda, kurulum, aşağıdaki komutun yönetici yetkileriyle verilmesi durumunda kolayca gerçekleşir:

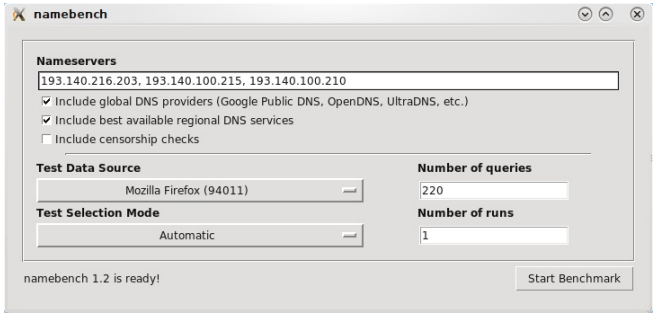
```
pisi it namebench
```

[0] http://www.belgeler.org/howto/djbdns-kurulumu-nasil_dns.html adresinden derlenmiştir.

Bu şekilde kurulumda yaptığımız testler sonucunda, bir sorun ve GNU/Linux dosya hiyeraşisi ile uyuşmayan bir durum bulunmamaktadır.

Kullanım

Namebench'i çalıştırmak için, Çalıştır penceresine ya da komut satırına "`namebench.py`" yazmanız yeterlidir. Paket Yöneticisi ile yükleyenler ise Namebench'i, Çalıştır penceresine veya komut satırına "namebench" yazarak ya da *Uygulamalar > İnternet* yolunu izleyerek çalıştırabilirler.



Namebench arayüzünde, birtakım seçenekler mevcuttur. Bu seçenekler, şu şekilde açıklanabilirler:

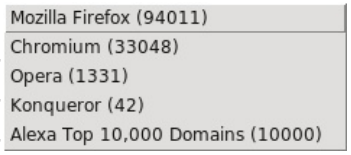
Nameservers: Burada, bilgisayarda öntanımlı olarak kullanılan DNS sunucularının adresleri yer almaktadır. Dilerseniz, başka DNS adreslerini ekleyip, sınanmasını da sağlayabilirsiniz.

Include global DNS Providers: Google Public DNS, OpenDNS, UltraDNS gibi genel DNS sunucularının da sınanmasını sağlar.

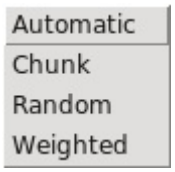
Include best available regional DNS services: Bölgesel olarak, erişimi olabildiğince mümkün olan DNS sunucularının da sınanmasını sağlar.

Include censorship checks: Sınamalara, sansür kontrolünü de dahil eder.

Test Data Source: Tarayıcı geçmişindeki siteleri kullanmak için, bilgisayarda kurulu olan tarayıcılar arasından seçim yapılmasını sağlar.



Test Selection Mode: Otomatik, Chunk, Rastgele veya Ağırlıklı olarak sınamayı seçebilirsiniz. Chunk modunda rastgele bir dizi seçerken, Rastgele (Random) modunda, rastgele sunucular seçer. Ağırlıklı (Weighted) mod ise, tekrarı bulunmayan veriler bulunduğunda kullanılmak üzere yazılmış. Namebench'in kendi sitesinde, Ağırlıklı modun, çoğu zaman en iyi seçenek olduğu vurgulanmış.



Number of queries: Sunuculara kaç tane verinin gönderileceğini belirler. Bu sayının artması, sonucu mümkün olduğunca günlük kullanım değerlerine getirirken, sınama süresini de uzatmaktadır.
Number of runs: Sınamaların kaç kez yapılacağı belirlenir.

Tüm bu seçenekleri kendinize göre ayarladıktan sonra, Start Benchmark düğmesine basarak sınamaları başlatabilirsiniz. Ayarlarınıza bağlı olarak değişen bir süre sonra, Namebench size, html olarak grafikler içeren bir rapor sunacaktır.

Dilerseniz, Namebench'i, komut satırından da kullanabilirsiniz. Komut satırından kullanmak için

```
namebech -x
```

yazmanız yeterlidir. Namebench'in alabileceği bazı parametreler ise şu şekilde:

`-q QUERY_COUNT`: Sunuculara kaç tane sorgu gönderileceğini belirler.

`-r RUN_COUNT`: Sınamanın kaç kez yapılacağını belirler.

`-C` : Sansür kontrolünü etkinleştirir.

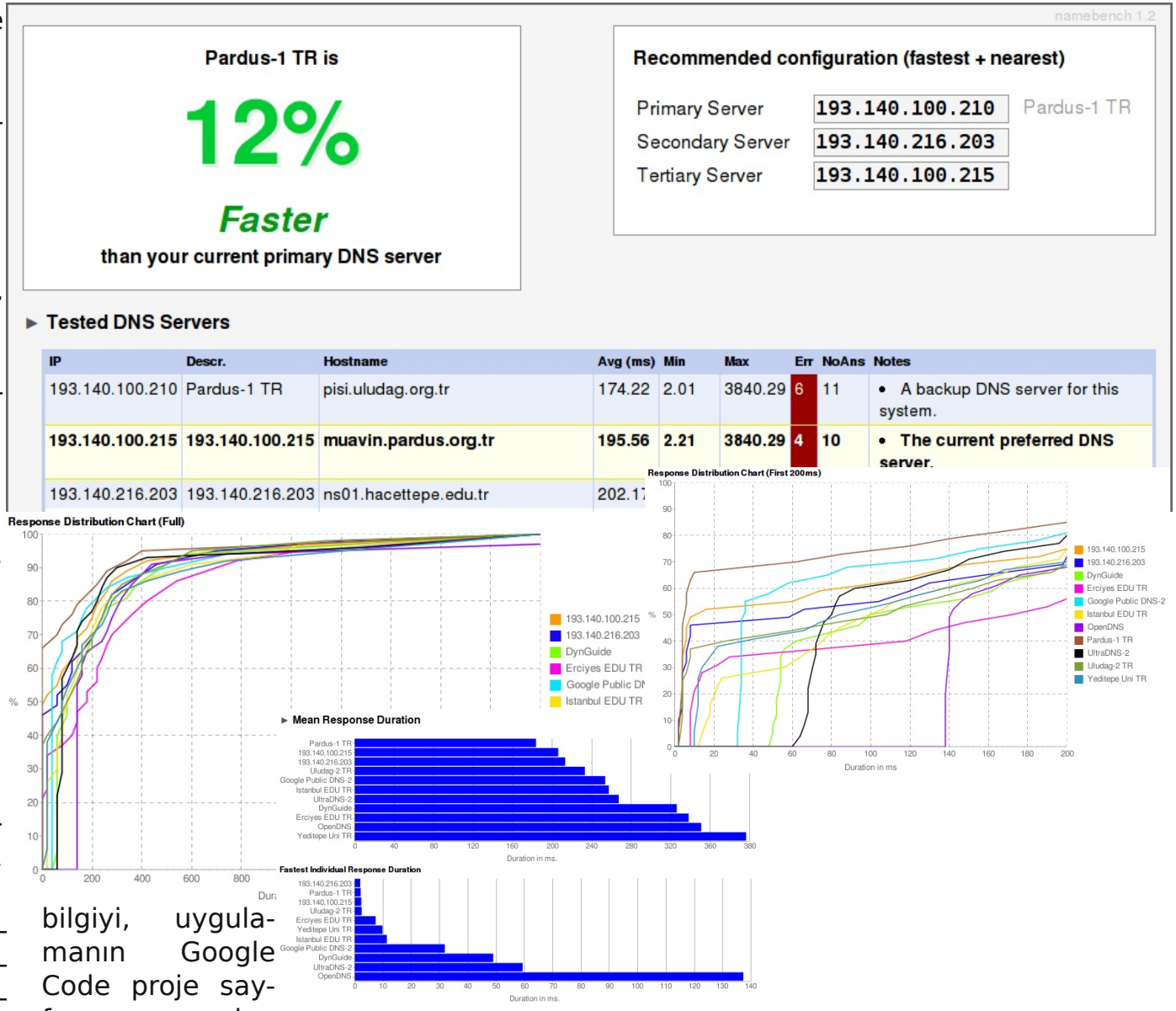
`-m MODE`: Test modunun seçimi. (chunk, random, weighted)

`-i SOURCE`: Veri kaynağının seçimi. (firefox, konqueror, opera)

`-c FILENAME`: Çıktıyı CVS verir.

`-o FILENAME`: HTML çıktısı için isim belirlenir.

`-w` : Çıktıyı tarayıcıda açar.



Son Söz

DNS sunucularını, hızın yanısıra, sansür ve erişilebilirlik yönünden de inceleyebilen bu yazılımı mutlaka denemelisiniz. Yazılımın Tkinter ile yazılmış olması, belki kötü bir görünüm kazandırsa da, bağımlılık açısından düşünüldüğünde, dağıtımlar için oldukça doğru bir seçim olmuş. Namebench hakkında daha geniş bilgiyi, uygulamanın Google Code proje sayfası olan http://code.google.com/p/namebench/ adresinden edinebilirsiniz.

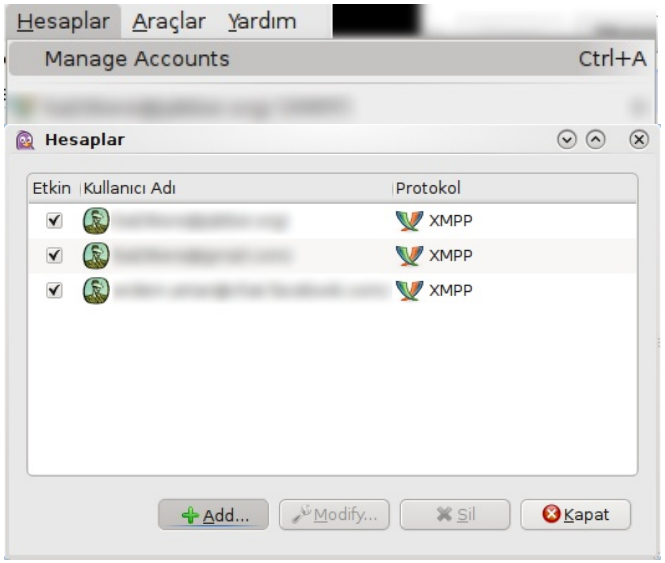
Erdem Artan
erdem@pardus-linux.org

Facebook, kullanıcılarına çok uzun zaman önce sunmaya başladığı sohbet hizmetini, yaklaşık iki ay önce Jabber, Google Talk gibi hizmetlerin kullandığı XMPP'ye taşıyarak, herhangi bir eklenti gerektirmeden masaüstü uygulamalarına taşınmasının önünü açtı.

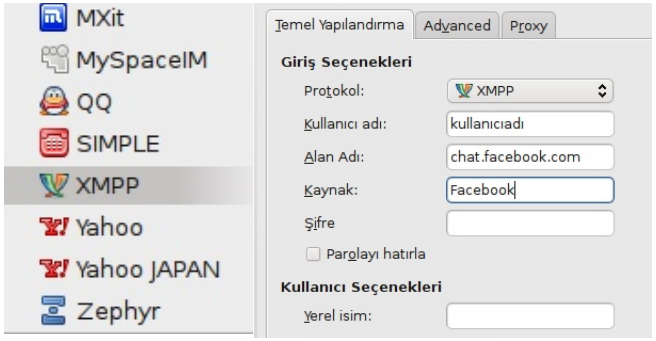
Daha önceleri Pidgin için Facebook eklentisi yardımıyla bağlanılabilen Facebook Sohbet hizmetine, bu yenilik sayesinde sadece bazı kullanıcı hesabı ayarları yapılarak bağlanılabiliyor.

Pidgin'de bu hizmeti kullanabilmek için, Facebook hesabınızda bir kullanıcı adınızın olması gerekiyor. Facebook kullanıcı adını, Facebook.com adresinde oturum açtıktan sonra Hesap > Hesap Ayarları yolunu izleyerek ayarlayabilirsiniz:

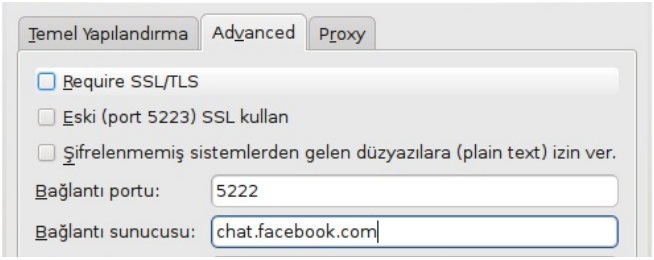
Kullanıcı adının belirlenmesinin ardından, Pidgin ayarlarına geçebilirsiniz. Bunun için Hesaplar (Accounts) menüsünden, Hesapları Yönet'i (Manage Accounts) seçin ve Ekle (Add) düğmesine tıklayın.



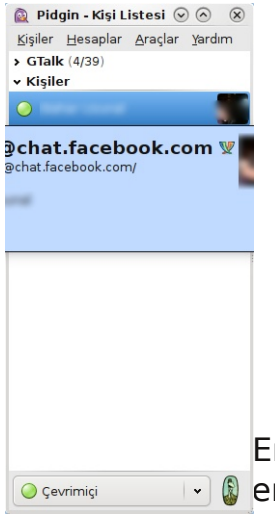
Ardından Hesap Ekle penceresinin Temel Yapılandırma sekmesinde, protokol olarak XMPP seçin. Kullanıcı adı olarak Facebook.com'da ayarlanmış olan kullanıcı adını, Alan Adı olarak bu hizmet için oluşturulmuş olan chat.facebook.com adresini girin. Kaynak kısmına herhangi bir şey yazılması, Facebook Sohbet'teki kişi listesinin ayrı tutulması bakımından tavsiye edilir.



Hesap ekle penceresinin Gelişmiş (Advanced) sekmesinde "SSL/TLS Gerektirir." (Require SSL/TLS) 'in işaretini kaldırılmanız gerekmektedir. Bunun ardından Bağlantı Sunucusu için yine chat.facebook.com adresini girmelisiniz.



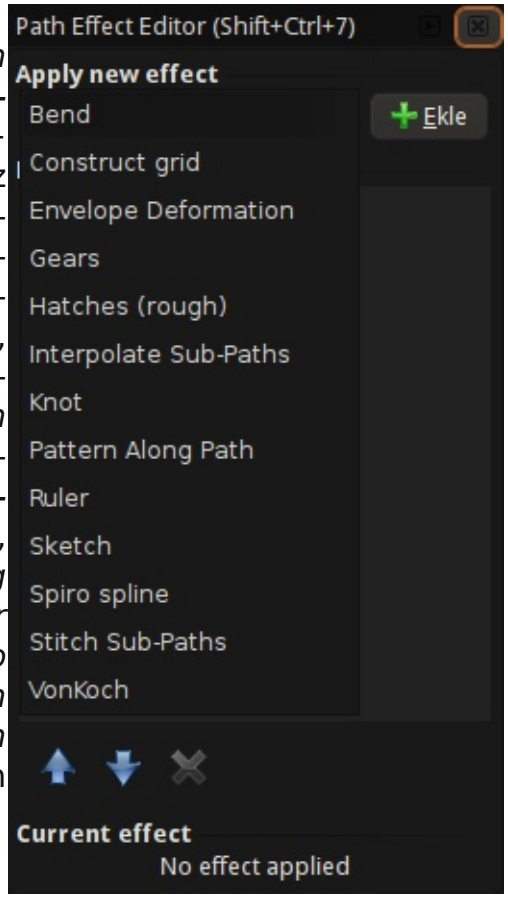
Tüm bu işlemlerin ardından, eğer sunucu ya da istemci tarafında bir hata yoksa, Pidgin rahatça Facebook hesabınıza bağlanacak ve sizi, arkadaşlarınıza Facebook Sohbet üzerinden bağlayacaktır.



Erdem Artan
erdem@pardus-linux.org

Inkscape 0.47 ile gelen önemli yeniliklerden biri de, *canlı yol efektleri*dir (live path effects). Yeni eklenen bu özelliğin en önemli avantajı, çizdiğiniz yolları veya diğer nesnelerin ilk halleri üzerinde (daire, dikdörtgen, spiral, yıldız ama üç boyutlu kutular ve metinler hariç) herhangi bir değişiklik yapmadan, şekillere farklı biçimler vermenize olanak sağlamasıdır.

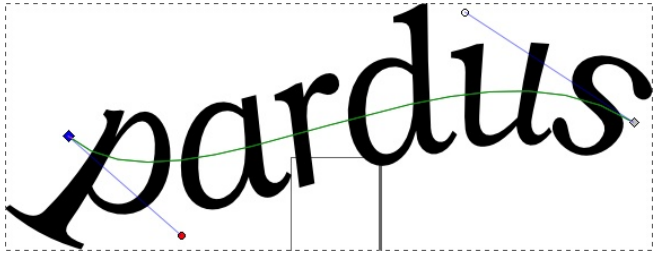
Menü çubuğundan, *Yol > Path Effect Editor... (yol efektleri biçimlendiricisi)* yolu veya *Ctrl+Shift+7* kısayolu ile ulaştığınız iletişim penceresi içerisindeki efektleri, çizdiğiniz nesnelere uygulayabiliyorsunuz. Efektler, listedeki sırası ile *Bend* (bükme) , *Construct Grid* (ızgara oluşturma), *Envelope Deformation* (zarf bozulması), *Gears* (dişliler), *Hatches* (tarama), *Interpolate SubPaths* (araya yol ekle), *Knot* (düğüm), *Pattern Along Path* (yol boyunca desen), *Ruler* (cetvel), *Sketch* (eskiz), *Spiro Spline* (Spiro şeritleri), *Stitch SubPaths* (yolları dik), *VonKoch* (fraktal oluştur) efektlerinden oluşuyor.



Efektler nesnelerin biçimlerini değiştirmede etkili bir araç, ancak geliştirilmesi gereken yönleri var. Örneğin, bir dikdörtgen üzerinde *bükme* (bend) efektini uyguladıktan sonra, efekti kullanarak dikdörtgenin şeklini değiştirirseniz ve sonrasında efekti kaldırırsanız, dikdörtgen olarak yarattığınız şeklin, yola dönüştürüldüğünü görürsünüz ve bunun geri dönüşü yok. Ancak aynı şeyi bir çokgen veya üçgen ile yaparsanız, nesne, yola dönüştürülmeden ilk haline getirilebiliyor.

Bir nesneye birden fazla efekt eklendiğinde, hepsi etkili olacak diye bir kural yok. Bir nesneye eklenen efektlerin sırası ve sayısı, efektlerin vereceği tepkileri belirliyor. Bir nesneye, listedeki efektlerden istediğiniz sayıda ve sırada efekt ekleme özgürlüğü, bir açıdan esneklik sağlarken, diğer yandan da sorun çıkartabiliyor. Herhangi bir şekle, birden fazla efekt ekler ve bu efektlerin ayarları veya liste içerisindeki sıralaması ile oynarsanız; ne demek istediğimi anlayabilirsiniz.

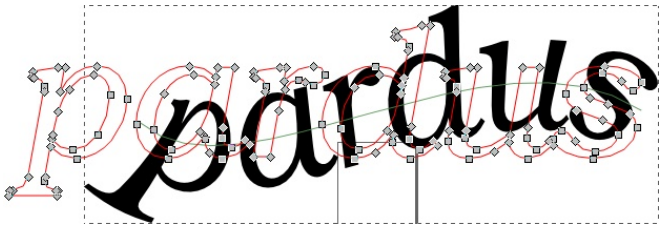
Listedeki ilk efekt *bükme* efekti. Çizilmiş olan herhangi bir şekli alıyor, çizili şekli bir *iskelet* (skeleton) olarak değerlendiriyor ve bu şeklin içerisine boylamasına uzanan bir yol yerleştiriyor. Her bir ucunda bir tane düğüm olan bu yolu kullanarak iskelen olarak adlandırdığımız şekli, tutarlı bir şekilde biçimlendirebiliyorsunuz. Yeşil renkli bu yolu, diğer yolları biçimlendirdiğiniz gibi biçimlendirebiliyor; bu yola düğümler ekleyebiliyorsunuz. Kısacası, iskelet olarak adlandırılan ilk nesnenin biçimini, iki tane düğüm ile kontrol edebiliyorsunuz ve bir yol ile neler yapabiliyorsanız, hepsini yapabiliyorsunuz.



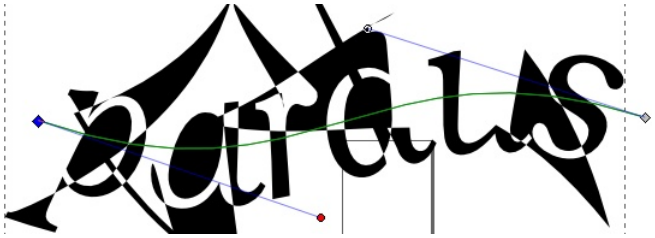
Mesela, tek bir kelime yazın. Yazdığınız kelimeyi seçtikten sonra, *Ctrl+Shift+c* kısayolunu kullanarak kelimeyi metin

türünden yol türüne dönüştürün. Sonrasında *Ctrl+Shift+g* tuşlarına basarak, yola dönüştürülürken oluşturulan grubu çözün. *Ctrl+k* tuşlarına basarak tek bir yol haline getirip, bükme efektini ekleyin. Daha sonra, *Path Effect Editor* iletişim kutusundaki *Edit on-canvas* (tuval üzerinde biçimlendir) seçeneğini seçin. Bu seçeneği seçtiğinizde, yola dönüştürdüğünüz kelimenin şekli ile oynayabileceksiniz. Kelimenin ortasında beliren yeşil çizgiyi veya her iki ucunda belirmiş olan düğümleri ve düğümlerin kulplarını kullanarak, eğilme ve bükülme etkileri yaratabilirsiniz.

Ayrıca, düğümleri kullanarak yolları düzenlemenizi sağlayan aracı seçer (kısayol tuşu *F2*'dir) ve sonrasında bükme efekti uygulamış olduğunuz kelimeyi veya herhangi bir nesneyi, bu araçla seçerseniz, kelimenizin ilk halinin dış hatları, kırmızı olarak belirecektir. Böylece, kelimenin kenarlarını oluşturan düğümleri görebilir ve biçimlendirebilirsiniz.



Orijinal şekilde ne tür değişiklikler veya yaratıcılık denemeleri yaparsanız yapın, *tuval üzerinde biçimlendir* (edit on-canvas) düğmesine tıkladığınızda, şekli, yine yeşil olarak gösterilen yol ile kontrol edebiliyorsunuz.

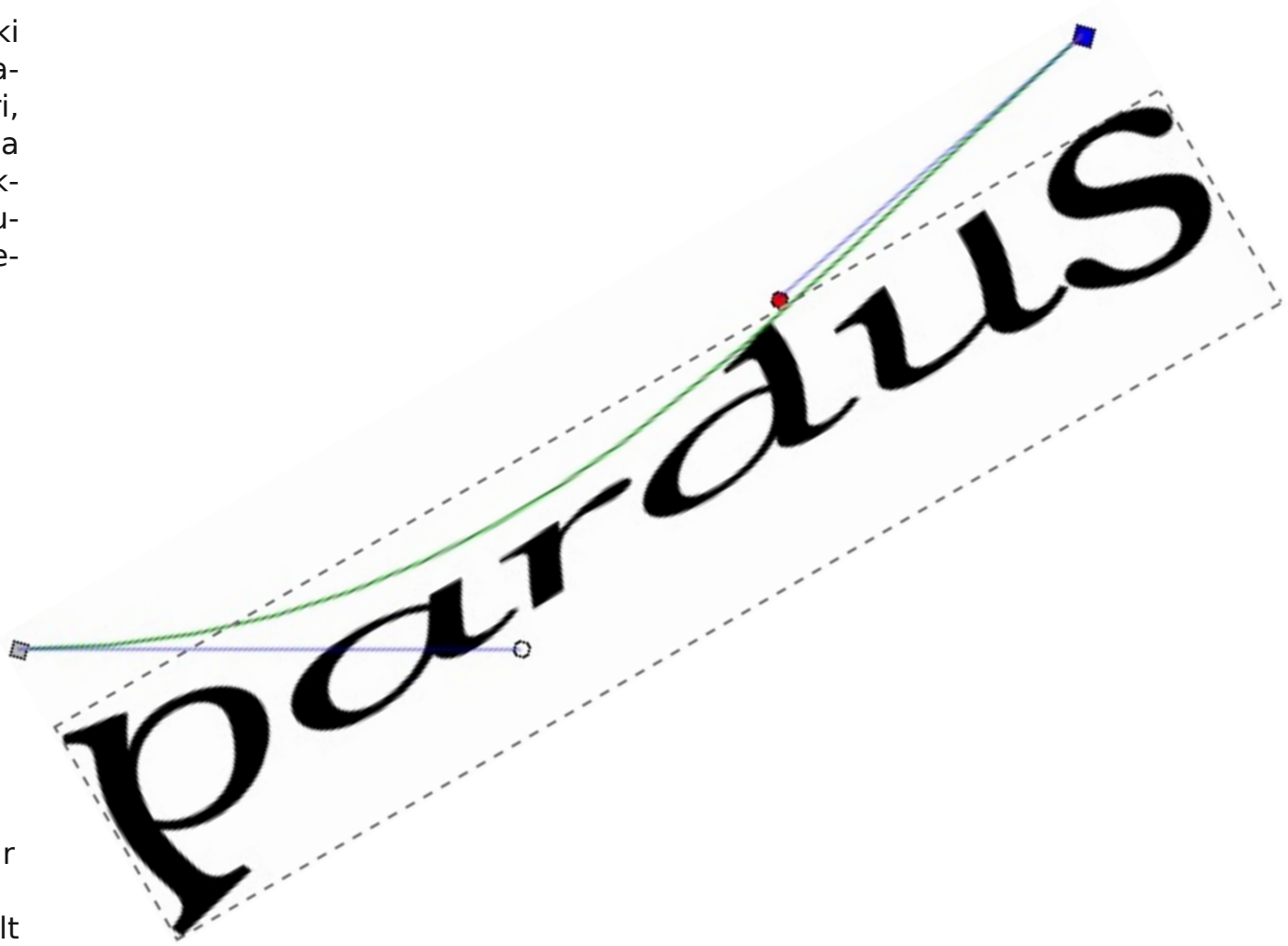


Bükme efektine benzeyen diğer efekt *Envelope Deformation* (zarf bozulması) olarak adlandırılıyor. Zarf bozulmasının bükme efektinden temel farkı, bükmek istediğiniz nesnenin ortası yerine her bir kenarına, nesneyle oynamanıza ve onu biçimlendirmenize izin veren bir yol yerleştirmesi. Bükme efektinde olduğu gibi, canlı yol efektleri iletişim penceresinde efekt seçiliyken görebildiğiniz *tuval üzerinde biçimlendir* (edit on-canvas) düğmesine tıklayarak, nesnenin her bir kenarında yer alan yolları teker teker biçimlendirebiliyorsunuz. Misal, bir metni yola dönüştürdükten sonra (*Ctrl+Shift+c* kısayolunu veya *Yol > Nesneden Yola* yolunu takip ederek) resimdeki gibi üst ve alttan bastırarak farklı etkiler elde edebilirsiniz. Eğer bunun gibi bir şekli, yolu oluşturan bütün ilgili düğümlerin yerlerini değiştirerek elde etmeye çalışsaydınız, en kötü ihtimalle başaramazdınız veya en iyi ihtimalle bu işlem bir iki saatinizi alırdı.

Canlı yol efektlerinin faydalı bir araç olmasının asıl nedenlerinden biri, ortaya çıkan etkiyi beğenmediğinizde işlemi geri almanıza olanak vermesidir. Herhangi bir nesneye uyguladığınız belirli bir efekti kaldırmak için; efekt listesindeki ilgili efekti seçin ve liste kutusunun hemen altında yer alan kırmızı çarpı işaretine tıklayın. Nesneye uyguladığınız bütün efektleri bir seferde kaldırmak için, nesne seçiliyken *Yol>Remove Path Effect* (yol efektini kaldır) yolunu izleyebilirsiniz. Nesne ilk haline, yani üzerinde oynanmamış haline geri dönecektir.

Efektlerden *gears, knot, ruler ve von-koch* dışında kalanlar, işinize epey yarayabilecek araçlar. Biraz kurcalayarak nasıl çalıştıklarını çözebilir ve güzel etkiler elde edebilirsiniz. Yola çıkış amacı;

Extensions (eklentiler) menüsü altındaki özellikleri kanlı-canlı ve etkileşimli olarak kullandırmak olan canlı yol efektleri, şu anda da çok etkili araçlar olmasına karşın, bir kaç ay sonra daha fazla eklentinin listeye eklenmesi ve daha 'tutarlı' hale gelmesiyle Inkscape'in vazgeçilmezlerinden olacaktır.



Mini sözlük:

Apply new effects: Yeni efektler ekle

Effect List: Efekt listesi

Current effect: Şu anki efekt

Bend path: Yolu bük

Width: Genişlik

Original path is vertical: Asıl yol dikeydir

Enable top&bottom paths: Üst ve alt yolları etkinleştir

Enable left&right paths: Sağ ve sol yolları etkinleştir

Edit on-canvas: Tuval üzerinde biçimlendir

Mehmet Gültaş
mehmetgultas@pardus-linux.org
http://martinamca.blogspot.com

Wesnoth'a Dalış yazımızın geçen ayki ilk bölümünde, Wesnoth oyununu tanıyarak, bir giriş yapmış; haritalara başlayacağımızı söyleyerek bırakmıştık. Kaldığımız yerden devam edelim.

Öncelikle, hayal gücümüzü zorlayarak bir hikaye oluşturmamız gerekiyor. Ben "Borucu Çırağı" ismini verdiğim hikayenin kahramanıyım. Hamit'in boru dükkanına gelen yaşlı bir adam, Hamit'i sinirlendirir. Ama maalesef Hamit'in bağrındığı adam, eski bir büyücüdür. Sinirlenen adam, bir büyü yaparak, Hamit'i Wesnoth'un karanlıklarına yollar. Hamit, babasına karşı iktidar savaşı yapan bir iblisin kalesine düşer. Darak isimli bu iblis, babasını yenebilmek için, lanetlenmiş ölümsüz lejyonlardan oluşan bir orduyu emri altına almak istemektedir. Ancak bu lejyonlar, bir anlaşma ile bağlıdırlar ve bu anlaşma, ölümlülerin dünyasında saklıdır. Darak, Hamit'e bu anlaşmayı bulması karşılığında, O'na kendi dünyasına giden yolu göstereceğine söz verir. Hamit'in pek seçeneği yoktur.

Eh, yaratıcılık özürlü olunca bu kadar oluyor. "Ben daha iyisini yaparım" diyorsanız eğer, bu yazı dizisi bitince, geçin klavye başına.

Hikayenin ardından, senaryoları oluşturmanız gerekiyor; yani bir harita üzerinde olan tek görev. Hikayenize, dizi olarak ilerleyen senaryolar yazmanız gerek.

İşin kurgu kısmı geride kaldıktan sonra, bu senaryoların haritalarını oluşturmanız gerekiyor. Bunun için hem **Wesnoth Markup Language (WML)**, hem de Wesnoth harita düzenleyicisi olan Haritacı'yı kullanacağız. Buralar kolay.

İşin zor kısmı, ırk ve karakterlerimizi oluşturacağımız bölüm. Burada, Wesnoth'un kodlama dili olan, WML'yi kullanacağız. WML, yapısı aynen HTML'ye benzeyen, Wesnoth kodlama diline denir. Aslında Wesnoth, -aynen ağ göz atıcısı olan Firefox ya da Opera'nın, HTML'yi okuyup, gerekli dosyaları belirtilen adreslerden çağırarak, size bir web sayfasını göstermesi gibi- WML kodlarını okuyup, belirtilen dosya ve animasyonları size gösteren bir yazılımdır. Elbette Wesnoth, bir ağ göz atıcısından daha fazlası. Ama bu örnek, çalışma mantığını anlatmak için uygun bir örnek. Kahramanlarımızı, savaşçılarımızı ve ırklarımızı oluşturmak için WML'nin yapısını öğrenmek durumundayız. Yani burada birazcık kodlamaya bulaşacağız. WML aynen HTML gibi tag'lardan oluşan bir düzenleme dilidir. Ama gözünüz korkmasın. Bu bir programlama dili değil. Çok karışık bir şey de değil. Sadece biraz uğraş gerektirecek. Ama inanın bu uğraş, en az Wesnoth hikayelerini oynamak kadar zevkli. Oyundaki karakterlerin güçleri, hareket puanları, darbe aldıklarında ya da silahlarının çıkardıkları sesler, saldırı ve savunma animasyonları; hepsi WML ile yazılmış betikler tarafından belirlenir.

Bundan sonraysa, yaptığımız senaryoları sıraya koyup, hikayeyi tamamlayacağız. Daha sonrasındaysa, çalışmalarımızın semeresini göreceğimiz hikayemizi Wesnoth'a entegre edip, test aşamasına geleceğiz. Haydi başlayalım...

Hikaye (Seferberlik)

İlk olarak, -bence- hikayenizi oluşturun. Elbette ki, haritaları yaptıkça çok defa değiştirirsiniz; ama yine de hikayenizi,

genel olarak, başından sonuna tasarlayabilirseniz, haritalarınızın tipini ve sayısını hikayeye göre oluşturmanız çok daha kolay olacaktır. Kim kime karşı olacak, hikayemiz kaç senaryodan oluşacak? Kahraman ne yapacak, sonrasında nereye gidecek? Bu gibi soruların cevaplarını baştan belirlemeniz, işlerinizi çok kolaylaştıracaktır. Hatta, karakterlerinizi önceden belirleyebilirseniz (ve hatta senaryolarda geçecek konuşmalarını bile yazabilirseniz), seferberlik oluşturma aşamasında uğraşmazsınız. Benim gibi yaratıcılık özürlü iseniz, hikaye oluşturma kısmında -varsa- arkadaşlarınızdan yardım alabilirsiniz. Hatta, http://www.wesnoth.org/forum/viewforum.php?f=32 adresinden hikayenizi diğer yazarlarla paylaşabilir ve fikirlerini alabilirsiniz.

Yazılımcıların bir sözü vardır: "*Kodlamaya ne kadar geç başlarsan, o kadar erken biter.*" Programın geneli ve akış diyagramı gibi kodlama öncesi yapılacak işlerle ne kadar çok uğraşılırsa, kodlama o kadar çabuk biter. Zaten eskiler, "on kere ölç; bir kere biç" demişler. Neyse, ben lafı fazla uzatmayayım; zaten yolumuz uzun.

Wesnoth Verileri

İlk olarak, Wesnoth'un verilerini nereden ve nasıl aldığına bir bakalım. Wesnoth, verilerini `/usr/share/wesnoth/data` dizininde saklar. Buraya girip de biraz kurcaladığınız zaman, neyin nerede olduğunu zaten kendiniz de göreceksiniz. Burada bizi, "*campaigns*" ve "*core*" dizinleri ilgilendiriyor. Diğerleri ile bir işimiz yok. "*campaigns*" klasöründe, oyundaki seferberlik hikayelerinin klasörlerini bulacaksınız. Kısaca, burada kullanacağımız verilere ulaşacağımız yollar:

```
wesnoth/data
wesnoth/data/core/units
wesnoth/data/campaigns
wesnoth/images
wesnoth/data/core/images
```

Wesnoth, sizin kullanıcı veri dosyalarınızı ise `~/.wesnoth1.x` dizini altında tutar. Battle For Wesnoth oyunu, şunlar için bu dizine bakar:

`~/.wesnoth1.x/data/campaigns` – seferberliğinizin konfigürasyon dosyaları ve alt dizinlerini içeren bölüm. Mesela benim "Borucu Çırağı" isimli seferberliğim için dizinim ve alt dizinlerim şu şekilde olacak:

`~/.wesnoth1.x/borucuciragi` (seferberlik dizini)

`~/.wesnoth1.x/borucuciragi/_main.cfg` (seferberliğinizin açıklamalarını ve nasıl yükleneceğini belirten metin dos-yası)

Alt klasörler:

```
~/.wesnoth1.x/borucuciragi/scenarios
~/.wesnoth1.x/borucuciragi/units
~/.wesnoth1.x/borucuciragi/images
~/.wesnoth1.x/borucuciragi/music
~/.wesnoth1.x/borucuciragi/sounds
~/.wesnoth1.x/borucuciragi/utils
```

Tekrar hatırlatalım, burada, kendi hikayeniz için kendi oluşturduğunuz resimleri, müzikleri (ogg uzantılı müzikleri kullanabilirsiniz), ünitelerinizi (sadece sizin senaryonuza has üniteler ve kendiniz) buralara koyacağız. Hikayenizde, oyunda mevcut üniteleri kullanacaksanız, bunları, oyun verileri yolunda /core klasörü altında bulabilirsiniz. Yani kullanıcı veri dosyalarımızda, kendi tanımlayacağımız veriler olacak.

`~/.wesnoth1.x/editor/maps` – çoklu oyuncu haritalarınızı (sadece harita verileriniz) içeren bölüm.

Hikayemizi oluştururken burada dikkat edeceğimiz şeyler:

1- Klasörlerin ya da dosyaların isimlerinde Türkçe karakterler kullanmamak. WML, Türkçe desteklemiyor. Ancak sadece isimlerde. Konuşmalarda kullanabilirsiniz.

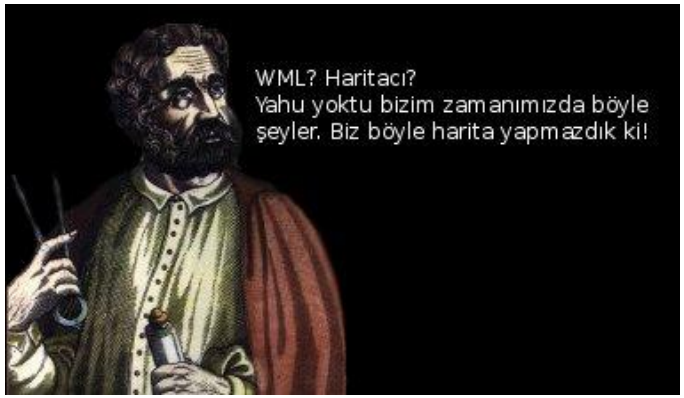
2- Boşluk kullanmamak. Örnek: “Borucu Çırağı” yanlış; “Borucu_Ciragi” doğru.

Seferberlik (oyunda böyle geçtiğinden, hikaye için artık bu terimi kullanacağım) klasörümüz altında, bize kolaylık sağlayacak şekilde, çeşitli klasörler oluşturabiliriz. Bunlar değişebilir, ancak bir “*_main.cfg*” dosyası oluşturmamız gerekiyor. Bu dosya, menüde seferberliğimizin adı, görünecek olan resmi, açıklamaları, zorluk seçimi ve seferberliği yapanlar gibi bilgileri içeren dosya olacak. Seferberlik oluşturma aşamasında, bu dosyanın hazırlanmasından detaylı olarak bahsedeceğiz.

Wesnoth Haritaları

Gelelim, ilk aşamamız olan haritalara. Wesnoth oyununda, harita hazırlamanın en kolay yolu, Wesnoth harita düzenleyicisi olan *Haritacı*'yı kullanmaktır. Harita düzenleyicisini kullanmak oldukça kolay. O yüzden sadece birkaç ayarını anlatacağım. Ancak bir Wesnoth haritasının yapısı ve mantığı hakkında bilgi sahibi olmamız gerekiyor. Çünkü, hikayenize göre, haritalarınızda düzenleyicinin dışında, yapmanız gereken bölgeler olabilir. Mesela, hikayenizde kutsanmış bir yer isteyebilirsiniz. Buraya gelen ünitelerin ya da kahramanınızın gücünün artmasını veya hasar almamasını isteyebilirsiniz. İşte bunlar için devreye yine WML girecek. Onun için harita mantığını kavramadan, harita yapımına başlarsak çuvallarız.

Haritacı, varsayılan olarak; `~/.wesnoth.1.x/editor/maps` dizini altına, harita çalışmalarınız kaydeder. Yine ~/.wesnoth.1.x/data altında çalışmalarınız için açılmış boş klasörler göreceksiniz. Bu klasörleri de kullanabilirsiniz. Erişmenizde kolaylık açısından resimlerinizi, seslerinizi, ayrı ayrı klasörler altında toplamanızı öneririm. Hepsini öncelikle bir yerde toplayın. Seferberlik oluşturma aşamasında, hangilerini nerelere yerleştirip, kodu nasıl ayarlayacağımızı ayrıntılı olarak göreceğiz.

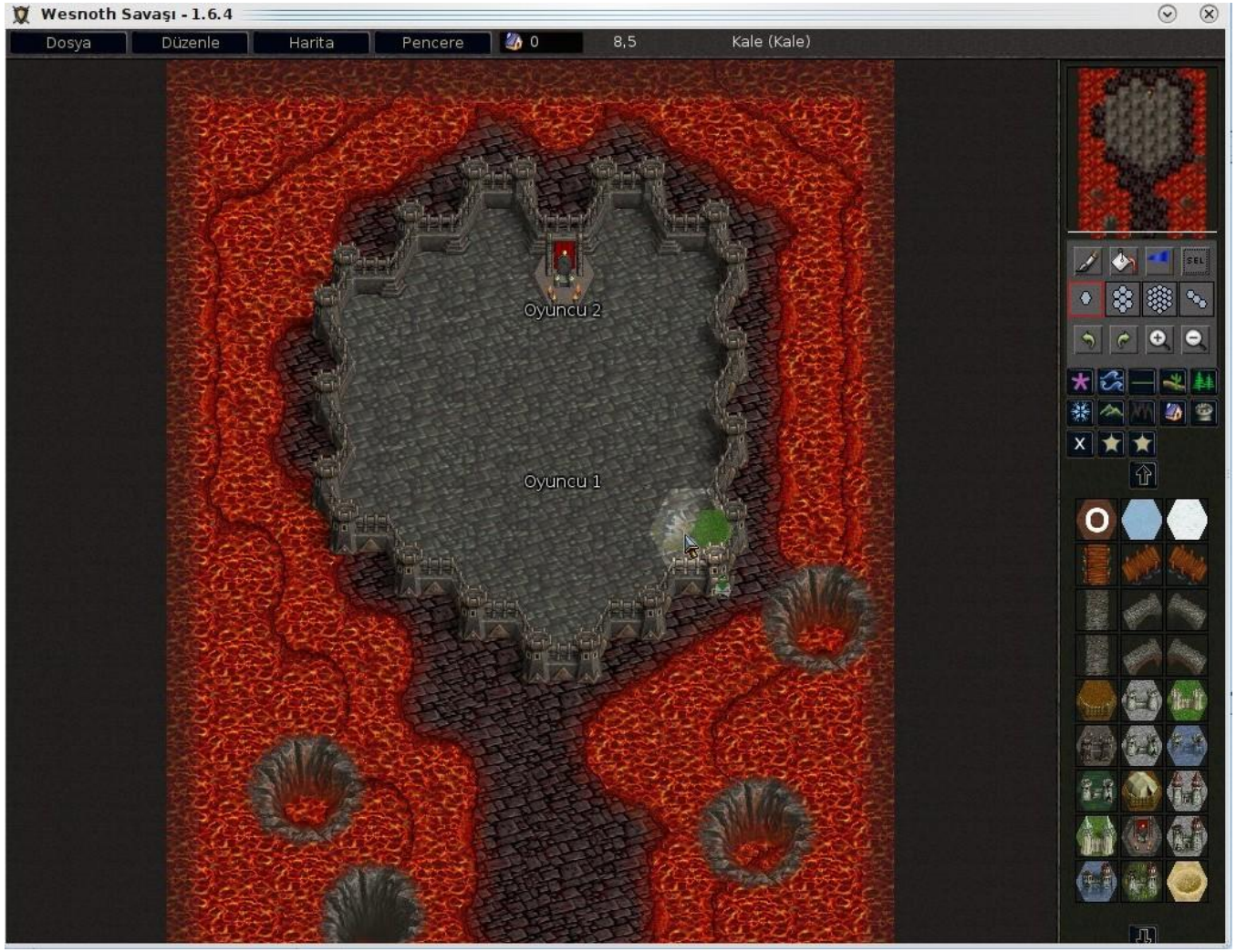


Wesnoth Harita Düzenleyicisi

Battle For Wesnoth oyununu başlatınca, menüdeki “*Haritacı*” düğmesine basarak ya da *Uygulamalar > Oyunlar > Taktik ve Strateji > Harita Düzenleyici* yolunu izleyerek de başlatabilirsiniz.

Bir sonraki sayfada bulunan görüntüde, yazmış olduğum seferberliğin ilk senaryosu olan “*Düşüş*” ün haritasını göreceksiniz.

Harita tasarlamak aslında, hikayeden sonra, oyunun en can alıcı kısmıdır. Dengeli ve eğlenceli bir harita hazırlayabilmek de, en az hikayeyi yazmak kadar, hayal gücü ve yaratıcılık gerektirir. Köyleri dengeli ve adil dağıtmak; geçitleri ve stratejik noktaları ustaca yerleştirmek. Ancak maalesef, henüz bunları



geliştirebilecek bir kod olmadığı için, benim haritalarım bunlardan yoksun olacak.

Haritacı'yı kullanmak, son derece basit. Dosya kısmında, standart "*Harita Yükle*", "*Haritayı Kaydet*", "*Haritayı Adıyla Kaydet (yani Farklı Kaydet)*" gibi seçenekler mevcut. Ancak buradaki "*Düzenleyici Ayarları*" kısmına girdiğinizde, haritanızın gece veya gündüz vakitlerinde alacağı renkleri ayarlayabileceğiniz bir menüye geliyorsunuz. Günün dört farklı zamanında (şafak, gündüz, akşam üzeri ve gece) haritanızın rengini renk tonlarıyla oynayarak belirleyebiliyorsunuz. "*Düzenle*" kısmından da "*Geri al*", "*Tekrarla*" gibi standart komutlar var. Ancak sağ sütundaki kontrol panelinden "*Seçim*" aracını açınca, buraya "*Kes*", "*Kopyala*", "*Yapıştır*" gibi komutlar da eklenmektedir. "*Harita*" kısmından haritayı boyutlandırılması, örtü ekleme gibi işlemleri yapabiliyorsunuz. Ayrıca buradan, düzenleyiciye, rastgele haritalar oluşturtabilirsiniz.

Buradan haritanıza örüntü (delay mask) uygulaması da yapabilirsiniz. Mesela haritanızın sadece belli bir bölgesine savaş sisi uygulaması yapmak istiyorsunuz. O zaman bir örüntü oluşturup, haritanın sadece o kısmına, örüntü üzerinde savaş sisi ekliyorsunuz. Bunu, sanki bir arka zeminin üzerine şeffaf bir katman koyarak, katman üzerine başka şekiller koyduğunuz bir uygulama olarak düşünün.

Aynı anda birden fazla harita üzerinde çalışıyorsanız, "*Pencere*" kısmından bu haritalar arasında geçiş yapabiliyorsunuz. Sağındaki köy sayısı ise, haritada bulunan köy sayısını göstermektedir.

Sağ sütun ise üç bölüme ayrılmaktadır. Sağ en üstte, haritanın mini bir görüntüsü mevcut. Onun altında kontrol paneli bulunmaktadır. Üç sıradan oluşan kontrol panelinden, en üst sırada, fırça, doldurma aracı, oyuncu başlama noktası seçicisi ve seçim aracı bulunmaktadır. Orta sırada, fırça ucu seçimleri vardır. En alt sırada ise, geri al, tekrarla ve yakınlaştırma - uzaklaştırma butonları var. Kontrol panelinin üçüncü bölümü ise, paletimizi (terrain) barındırır. Paletimizden seçtiklerimizi harita üzerine yerleştirerek haritamızı oluşturuyoruz. Sol fare tuşu ile seçtiğiniz paleti harita üzer,ne yine sol tuş ile; sağ tuşla seçtiğiniz paleti ise sağ tuş ile yerleştirebiliyorsunuz. Oldukça basit.

Genel olarak, haritanızın boyutu ile başlayın. Baskın arazi şekli ne olacaksa (ova, kar veya çöl) onunla tüm haritanızı kaplayın. Sonrasında ise haritanızı şekillendirin. Oyuncuların başlangıç noktalarını ise koymayı unutmayın.

Peki haritalarımızda düzenleyicide olmayan bir palet eklemek istersek ya da bir palete farklı özellikler eklemek istersek ne olacak? İşte burada WML işin içine giriyor. Aslında haritalarınızı, sadece KWrite ya da Kate gibi metin düzenleyici ile de oluşturabilirsiniz. Harita da, diğer tüm oyun ögeleri gibi, WML kodlarından oluşmuş bir düz metin (plain/text) dosyasıdır. Bir önceki sayfada görüntüsü olan haritamızın WML kodları şu şekildedir:

```
border_size=1
usage=map

Yl, Yl, Yl, Yl, Qlf, Qlf, Qlf, Qlf, Qlf, Yl, Yl, Yl
Yl, Yl, Qlf, Qlf, Ryd, Ryd, Ryd, Ryd, Ryd, Qlf, Qlf, Yl
Yl, Qlf, Qlf, Ryd, Cud, Cud, 2 Kud, Cud, Cud, Ryd, Qlf, Yl
Yl, Qlf, Ryd, Cud, Cud, Cud, Cud, Cud, Cud, Ryd, Qlf, Yl
Yl, Qlf, Ryd, Cud, Cud, Cud, 1 Cud, Cud, Cud, Ryd, Qlf, Yl
Yl, Qlf, Qlf, Cud, Cud, Cud, Cud, Cud, Cud, Ryd, Qlf, Yl
Yl, Yl, Qlf, Ryd, Ryd, Cud, Cud, Cud, Ryd, Ryd, Ql, Yl
Yl, Yl, Yl, Qlf, Ryd, Ryd, Ryd, Ryd, Qlf, Qlf, Yl, Yl
Yl, Yl, Ql, Qlf, Qlf, Ryd, Ryd, Qlf, Qlf, Yl, Yl, Yl
Yl, Yl, Yl, Yl, Qlf, Ryd, Ryd, Ryd, Qlf, Ql, Yl, Yl
Yl, Yl, Yl, Qxu, Qlf, Ryd, Ryd, Ryd, Qlf, Yl, Yl, Yl
Yl, Yl, Yl, Yl, Yl, Qlf, Qlf, Qlf, Yl, Yl, Yl, Yl
```

Bu kodları alıp, bir metin dosyasına kopyalayıp; Wesnoth Haritacı ile "*Harita Yükle*" dediğinizde, açılacak harita olarak yukarıdaki kodu yapıştırdığınız metin belgesini gösterirseniz; benim "*Düşüş*" adlı haritam açılacaktır.

"`border_size`" değişkeni değerini "`1`" yaparak, haritamızın bir sınırları olduğunu belirtiyoruz. "`usage=map`" ise, Wesnoth'a, bunun bir harita olduğunu söylüyor. Sonrası ise paletlerimizin WML kodları geliyor. Palet kod tablosu:

Değişken	İsim	Grubu
Ai	Buz	Kar
Aa	Kar	Kar
Ww^Bw|	Köprü	Çimenli Arazi, Sığ Su
Ww^Bw/	Köprü	Çimenli Arazi, Sığ Su
Ww^Bw\	Köprü	Çimenli Arazi, Sığ Su
Wo^Bw|	Köprü	Çimenli Arazi, Derin Su
Wo^Bw/	Köprü	Çimenli Arazi, Derin Su
Wo^Bw\	Köprü	Çimenli Arazi, Derin Su
Ss^Bw|	Köprü	Çimenli Arazi, Bataklık
Ss^Bw/	Köprü	Çimenli Arazi, Bataklık
Ss^Bw\	Köprü	Çimenli Arazi, Bataklık
Ce	Kamp	Kale
Ch	Kale	Kale
Cv	Elf Kalesi	Kale
Cud	Cüce Kalesi	Kale
Chr	Yıkıntıu	Kale
Chw	Batık Yıkıntı	Kale, Sığ Su
Chs	Batak Yıkıntı	Kale, Bataklık
Ke	Kamp İçkale	İç Kale
Kh	İçkale	İç Kale
Kv	Elf İçkale	İç Kale
Kud	Cüce İçkale	İç Kale
Khs	Batak İçkale	İç Kale, Bataklık
Dd^Dc	Krater	Çöl
Dd	Çöl	Çöl
Dd^Dr	Moloz	Çöl
Ds	Kum	Çöl
Dd^Do	Vaha	Çöl
Aa^Fpa	Karlı Orman	Orman
Gg^Fet	Ulu Ağaç	Orman
Gs^Fp	Orman	Orman
Gs^Ft	Tropik Orman	Orman
Gg	Çim	Düzlük
Ggf	Çim (çiçekli)	Düzlük
Gs	Bozkır	Düzlük
Ha	Karlı Tepeler	Engebeler
Hd	Kum Tepeleri	Engebeler
Hh	Tepeler	Engebeler
Md	Çöl Dağları	Engebeler
Mm	Dağlar	Engebeler

Değişken	İsim	Grubu
Ql	Lav Çukuru	Mağara
Qlf	Lav	Mağara
Rd	Yol (Çöl)	Çöl
Re	Çamur	Düzlük
Rr	Yol	Düzlük
Rp	Elf Patikası	Düzlük
Re^Gvs	Tarla	Düzlük
Gg^Wm	Yel Değirmeni	Düzlük
Ss	Bataklık	Sulak
Uu	Mağara	Mağara
Uu^Ii	Mağara Hüzmesi	Mağara
Uu^Uf	Mantar Korusu	Mağara
Re^Uf	Mantar Korusu	Orman
Uh	Taşlık Mağara	Mağara
Uh^Ii	Taşlık Mağara Hüzmesi	Mağara
Dd^Vda	Köy	Köy
Dd^Vdt	Köy	Köy
Aa^Vea	Köy	Köy
Gg^Ve	Köy	Köy
Aa^Vha	Köy	Köy
Gg^Vh	Köy	Köy
Hh^Vhh	Köy	Köy
Ha^Vhha	Köy	Köy
Mm^Vhh	Köy	Köy
Gs^Vht	Köy	Köy
Uu^Vu	Köy	Köy
Uu^Vud	Köy	Köy
Ww^Vm	Köy	Köy
Ss^Vhs	Köy	Köy
Ss^Vm	Köy	Köy
Wo	Derin Su	Sulak
Ww	Sığ Su	Sulak
Wwf	Sığ Geçit	Sulak
Wwr	Kayalık Sahil	Shallow Water
Mm^Xm	Aşılmaz Dağlar	Engebeler

Bunlar haritadaki paletlerin WML değerleridir. Haritacı'da, Harita > Arazi Kodlarını Göster kutucuğunu doldurursanız, bu kodları harita üzerinde de görebilirsiniz. Bu kodları verirken uymamız gereken bazı kurallar var. Bunlar:

#Palet kodları 2-4 karakterden oluşabilir.

#İlk harf büyük, sonraki harfler küçük olmalıdır.

#Palet kodları sadece harflerden oluşabilir. Kodlardaki "/", "|" ve "\" işaretleri yön belirten işaretlerdir (köprü yönleri gibi). ^ işareti ise, paletimizin katmanı ile oluşturulduğunu gösterir. Mesela Ww^Bw| kodu, sığ su paleti üzerinde, kuzey - güney doğrultusunda (|) bir köprü katmanı ile oluşturulduğunu gösterir. Yani sığ su üzerinde dikey köprü demektir.

Y,y,Z,z kodları kullanıcının yapacağı palet kodları için ayrılmışlardır. Yani kendi oluşturacağımız paletler için bu kodları kullanabiliriz. Mesela yeni bir palet için Yyz kodu kullanabiliriz.

Yeni Palet Oluşturmak

Paletlerimiz, `/usr/share/wesnoth/data/core/images/terrain` klasörü altında bulunan 72*72 boyutlarındaki *.png dosyalarıdır. Yani yeni bir palet oluşturmak için yapacağımız şey, 72*72 ebatlarında ve transparan arka zeminli bir png dosyası oluşturmak. Bunun için Gimp'i veya kullanmasını bildiğiniz herhangi bir grafik işlemci programını kullanabilirsiniz.

Aslında işin kolayına da kaçabilirsiniz. Palet klasörü altında var olan palet dosyalarından birini açıp, düzenleyip, farklı kaydederek de aynı sonuca ulaşabilirsiniz. Böylesi size avantaj sağlayacaktır. Çünkü oyun haritası altıgenlere bölünmüş durumdadır. Yani oluşturacağınız png resmi içinde, öncelikle, boş bir altıgen oluşturmanız gerekiyor. Paletinizi de o altıgen içerisinde yapmalısınız. `/usr/share/wesnoth/data/core/images/terrain` klasörü altında, boş bir altıgen bulabilirsiniz. Ya da uygun zemine sahip bir palet dosyasını düzenleyerek, farklı kaydedebilirsiniz.

Yeni bir palet oluşturup, Haritacı' ya eklemek için takip edeceğimiz 3 temel adım var:

1. Paletinizi, Gimp veya kullanmasını bildiğiniz bir grafik işlemci programında tasarlayıp; `/usr/share/wesnoth/data/core/images/terrain` klasörü altına kaydetmek.

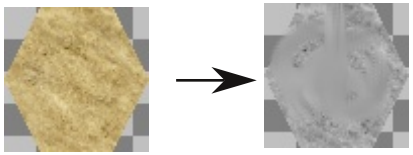
2. `/usr/share/wesnoth/data/core` klasörü altındaki "`terrain.cfg`" dosyasını düzenleyerek paletimizin özelliklerini tanımlamak.

3. `/usr/share/wesnoth/data/core` klasörü altındaki "`terrain-graphics.cfg`" dosyasını düzenleyerek, paletimizi Wesnoth'a tanıtmak.

Paleti Oluşturmak

İlk olarak, paletimizi oluşturmalıyız. Transparan arka zemine sahip bir 72*72 .png resmi olmalıdır. Ancak haritamız altıgenlerden oluştuğu için, paletimiz de bir altıgen olmalıdır. `/usr/share/wesnoth/data/core/images/terrain` altında `alphamask.png` adıyla boş bir altıgen bulabilirsiniz. Bunun üzerinde istediğiniz paleti oluşturabilirsiniz. Ya da hazır bir paleti, düzenleyerek farklı kaydedebilirsiniz. Ben, burada kestirmeden giderek, `desert3.png` paletini yeniden düzenledim. Paletimi "`dark.png`" adıyla

kaydettim. Grafikle aram olmadığı için, sadece renksizleştirip, üzerinde birkaç dağıtma yaptım.



Kabul, baştan savma bir palet tasarımı; ama palet tasarımını anlamak için uygun bir örnek. Sonrasında örneğimizi `/usr/share/wesnoth/data/core/images/terrain` klasörü altına kaydetmek kalıyor. Tabi bu işlemi root yetkileri ile yapabiliyorsunuz.

Siz, kendi tasarımlarınızı yapabileceğiniz gibi, Wesnoth forum sitesinin, "*Artwork*" kısmında[1] pekçok tasarım bulabilir; kendi tasarımlarınızı da paylaşabilirsiniz.

Palet Özelliklerini Tanımlamak

Wesnoth paletlerinin özellikleri, `/usr/share/wesnoth/data/core` klasörü altındaki `terrain.cfg` dosyasında tutulur. Bu dosyayı açtığınız zaman, paletlerin yukarıdaki kod tablosunda en sol sütunda belirtilmiş olan, Haritacı'nın gruplarına göre yapılandırıldığı dikkatinizi çekmiştir. Ancak isimleri İngilizce. Daha önce de belirttiğimiz gibi, WML kodunun dili İngilizce. Türkçe karakter kullanamıyoruz.

Paletimizin özelliklerini belirtmek için WML dilini kullanacağız. Bunun için, aynı HTML'de olduğu gibi taglar kullanacağız. Palet tagı;

```
[terrain]
      ----
      ----
[/terrain]
```

şeklindedir. Paletimizin özelliklerini tanımlayacak olan değişken değerleri, bu taglar arasına yazılır. Değişkenlerimizin bazıları mecburi, bazıları ise isteğimize göre koyabileceğimiz değişkenlerdir. Ancak zorunludan kastım, kullanmazsanız, paletinizin çalışmayacağı değişkenlerdir. İsteğe bağlı değişkenler kullanılmaz ise, paletin özelliklerinin çalışmamasına sebep olur. Değişkenler belirtilmediği takdirde, Wesnoth, o değişken değerini kapalı ya da 0 olarak kabul edecektir. Bu değişkenler:

symbol_image: Zorunlu bir değişkendir. Paletimizin, oyunun küçük haritasında gösterilecek olan resmidir. Doğrudan, `/usr/share/wesnoth/data/core/images/terrain` klasörü altına bakar. Bu klasörde palet için kullandığınız resmin aynını kullanmak isterseniz, bu değeri, resmin adı (uzantısı olmadan) olarak atayabilirsiniz. Bir başka sembol resmi koyacaksanız eğer, resmin bulunduğu klasörün tam yolunu değer olarak girmeniz gerekir. Bizim dark için;

```
[terrain]
      symbol_image=dark
[/terrain]
```

başka bir konumdaysa dosyamız;

```
[terrain]
      symbol_image=/home/hamit/dark.png
[/terrain]
```

şeklinde olacaktır. Ben, dosyanın adını "dark" koyduğum için böyle yazdım. Siz kendi dosyanızın adını yazacaksınız.

editor_image: İsteğe bağlı bir değişkendir. Paletimizin, Wesnoth Haritacı'nın palet tablosunda gösterilecek olan resmidir. Kullanımı aynen *symbol_image* değişkeni gibidir. Eğer belirtilmezse, Wesnoth, *symbol_image* değişken değerini, bu değişken için de kullanacaktır.

[1] http://forums.wesnoth.org/

id: Zorunlu bir değişkendir. Paletimizin kimliğidir. Oyuna paletimizi tanımlarken kullanacağımız veri değerini ifade eder. Kullanımı:

```
[terrain]
      id=dark
[/terrain]
```

şeklindedir.

name: Zorunlu bir değişkendir. Paletimiz, oyunda ve Haritacı'da bu isimle karşımıza çıkacaktır. Değerden önce koyduğumuz "_" işareti, bu değerin çevrilebilir olduğunu gösterir. Burada Tükçe isim kullanabilirsiniz. Kullanımı:

```
[terrain]
      name=_ "Karanlık"
[/terrain]
```

şeklindedir. "_" ile isim tırnakları arasında bir karakter boşluk vermeyi unutmayın.

string: Zorunlu bir değişkendir. Paletimizin, harita ve senaryolardaki WML kodudur. Y,y,Z,z ile başlamalı, 2-4 karakter olmalı ve ilk harf büyük, sonrakiler küçük olmalıdır. Kullanımı:

```
[terrain]
      string=Yyx
[/terrain]
```

şeklindedir. "terrain.cfg" dosyasında, verdiğiniz string değerini aratın. Aynı string değeri, bir başka palette de olmamalı.

unit_height_adjust: İsteğe bağlı bir değişkendir. Ünitenin palet üzerindeyken ne kadar yükselip, alçalacağını belirler. + (yükselme) ve – (alçalma) değerde rakam alır. Kullanımı:

```
[terrain]
      unit_height_adjust=-2
[/terrain]
```

şeklindedir.

submerge: İsteğe bağlı bir değişkendir. 0 ve 1 arası bir değer alabilir. Ünitenin palet üzerinde iken ne kadar batacağını ifade eder. Su ve bataklık paletlerindeki batma efektidir. Mesela, ünite bir su paleti üzerinde yarısına kadar gömülecekse, bu değişken 0.5 değerini alır. Kullanımı:

```
[terrain]
      submerge=-0.4
[/terrain]
```

şeklindedir.

light: İsteğe bağlı bir değişkendir. Sayısal değer alır. Paletin, yerel ışık seviyesidir. Bir değer atarsanız, gece veya gündüz bu palet parlayacaktır. Örnek lavlar. Ancak bunu hüzme katmanı olan ^Li ile karıştırmayın. Uh^Li kodundaki hüzmeli mağara katmanı, kaypak ve adil ünitelerin, saldırı ve savunma özelliklerini nötrlerken; light değişkenimizin bunun üzerinde bir etkisi yoktur. Kullanımı:

```
[terrain]
      light=25
[/terrain]
```

şeklindedir.

heals: İsteğe bağlı bir değişkendir. Sayısal değer alır. Palet üzerine gelen ünitenin, her el başında ne kadar sağlık kazanacağını ya da kaybedeceğini belirler. Eğer değişkene "true" değeri verirseniz, Wesnoth, burayı bir köy gibi algılayarak; buradaki üniteye her el başı 8 YP ekleyecektir. Kullanımı:

```
[terrain]
      heals=8
[/terrain]
```

ya da

```
[terrain]
      heals=true
[/terrain]
```

şeklindedir.

gives_income: İsteğe bağlı bir değişkendir. Eğer "`true`" değeri verirseniz, palet, ele geçiren tarafa her el başı altın kazandıracaktır. Köyler gibi. Kullanımı:

```
[terrain]
      gives_income=true
[/terrain]
```

şeklindedir.

recruit_onto: İsteğe bağlı bir değişkendir. "`true`" değeri verirseniz, bu palet üzerine asker alma ve geri çağırmayı mümkün kılar. Örnek kaleler. Kullanımı:

```
[terrain]
      recruit_onto=true
[/terrain]
```

şeklindedir.

recruit_from: İsteğe bağlı bir değişkendir. "true" değeri verirseniz, asker alabilecek olan ünite bu paletin üzerine geldiğinde, asker alabilir ya da geri çağırabilir. İç kale paletlerinde olması gereken bir değişkendir. Kullanımı:

```
[terrain]
      recruit_from=true
[/terrain]
```

şeklindedir.

aliasof: Zorunlu bir değişkendir. Tanımladığımız bir paletin; üzerindeki ünitelerin saldırı, savunma ve hareket etkinliklerinin hesaplanması için, ne tip bir diğer palet gibi davranılacağını belirten değerdir. Yani, tanımladığınız bir paletin, oyunda özellikleri tanımlanmış bir diğer palet gibi davranmasını bu değişken ile sağlarsınız. Virgül ile ayırarak birden fazla palet de atayabilirsiniz. Kullanımı:

```
[terrain]
      aliasof=Ch,  Ww
[/terrain]
```

şeklindedir. Burada `Chw` kodlu batık kalenin değerini görüyorsunuz. Hem kale hem de sığ su gibi davranmasını belirtiyor.

def_alias: İsteğe bağlı bir değişkendir. Kullanımı, "`aliasof`" değişkeni ile aynıdır. Ancak sadece ünitenin savunma durumundaki hesaplamaları için kullanılır. Saldırı ve hareket durumunu etkilemez.

mvt_alias: İsteğe bağlı bir değişkendir. Ünitenin sadece hareket durumunu etkiler. Savunma ve saldırıyı etkilemez. Kullanımı, "`aliasof`" değişkeni ile aynıdır.

editor_group: Zorunlu değişkendir. Oluşturduğumuz paletin, Haritacı'nın palet tablosunda, hangi palet grubunun altında gösterileceğini belirler. Kullanımı:

```
[terrain]
      editor_group=flat
[/terrain]
```

şeklindedir. Alacağı değerler: `flat, frozen, forest, rough, desert, water, castle, special, bridge, cave village` değerleridir. Eğer yaptığınız paleti, uygun gurubun altına koyarsanız, Haritacı'da bulmanız kolaylaşır.

Paleti Wesnoth'a Tanımlamak

Paletimizi yaptık. Özelliklerini de oluşturduk. Peki, nedir bu palet? Düzlük mü,

orman mı, yoksa sadece süs amaçlı bir katman (overlay) mı?

Bunun tanımını ise, /usr/share/wesnoth/data/core klasörü altındaki "terrain-graphics.cfg" dosyasını düzenleyerek yapacağız.

Not: *Bu dosyaya, sadece kendi paletlerinizin tanımlarını ekleyin. Eğer, %100 ne yaptığınızdan emin değilseniz, oyunun orijinal paletlerinin ayarlarını değiştirmeyin. Yoksa oyun çalışmayabilir.*

Bu dosya; Katman, Köy Binaları, Palet Tabanı ve Geçiş bölümlerinden oluşur. Katman kısmında, sadece görsel amaçlı olup da oyuna bir etkisi olmayan nesneler tanımlanır. "*Köy Binaları*" kısmında köyler, "*Palet Tabanı*" kısmında arazi paletleri, "*Geçiş*" bölümünde ise paletler arası geçişler tanımlanır.

Bizim, arazi paletlerimiz için kullanacağımız kısım, "*Palet Tabanı*" kısmı. Burada

```
TERRAIN_BASE
```

komutunu kullanacağız. Bu komut, paletimizin id değişkeninde tanımladığımız değer ile birlikte kullanılır. Kullanımı:

```
{TERRAIN_BASE       palet_id_değeri         palet_resmi}
```

şeklindedir.

Komutu, "terrain-graphics.cfg" dosyasında, ilgili alanın altına ekleyip kaydediyoruz.
TERRAIN_BASE, bazı parametreler de alabiliyor. Bunlar _P (olasılık) ve _RANDOM (rastgele) parametreleri.

Eğer isterseniz, aynı paletin birden fazla çeşidini oluşturabilirsiniz. Her resim için ayrı ayrı palet tanımlamanıza gerek yok. Aynı palet için, dokuz farklı resim kullanabilirsiniz. Bunları, /usr/share/wesnoth/data/core/images/terrain klasörü altına, resim.png, resim2.png,...resim9.png şeklinde resimler oluşturabilirsiniz.

_P parametresi, bu resimlerin hangilerinin, ne olasılıkla yerleştirileceğini belirler. Yani, Haritacı'da tek bir palet vardır. Ancak, kaydedip, bu parametre ile belirtmiş olduğunuz resimler, belirttiğiniz olasılık sıklığında haritaya yerleştirilir. Bu parametrenin varsayılan değeri 100'dür. Olasılık miktarı kısmı 0 ile 100 arası bir değer alabilir. Kullanımı:

```
{TERRAIN_BASE_P    palet_id_değeri  olasılık_miktarı   palet_resmi}
{TERRAIN_BASE_P    palet_id_değeri  olasılık_miktarı   palet_resmi2}
{TERRAIN_BASE_P    palet_id_değeri  olasılık_miktarı   palet_resmi3}
.
.
.
```

şeklindedir.Örnek:

```
{TERRAIN_BASE_P      Yyx          25       dark}
{TERRAIN_BASE_P      Yyx          35       dark2}
{TERRAIN_BASE_P      Yyx          65       dark3}
```

_RANDOM parametresi ise, _P parametresinin otomatikleştirilmiş hali olarak da tanımlanabilir. Bu parametre, tek satır komutla, oluşturmuş olduğunuz palet resimlerinin eşit olasılıkla hari

taya yerleştirilmesini sağlar. Kullanımı:

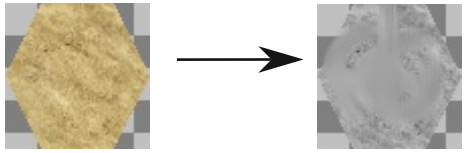
```
{TERRAIN_BASE_RANDOM     palet_id_değeri        palet_resmi}
```

şeklindedir.

Not: *Burada kullanacağınız resimler, aynı isimde ancak seri olarak numaralandırılmış olmalı. dark.png, dark2.png, dark3.png... gibi.*

Şimdi yukarıda anlattıklarımızı örneklendirelim. `dark.png` ile oluşturduğumuz paleti haritaya yerleştirelim. Bu paletimiz, lanetlenmiş bir alan olsun. Üzerine gelen üniteler, saldırı ve savunmada %20 etkinlikle savaşırken, her el başında da 2 YP kaybetsin.

1- `desert3.png` dosyasını, değiştirerek `/usr/share/wesnoth/data/core/images/terrain` klasörü altına `dark.png` olarak kaydettim.



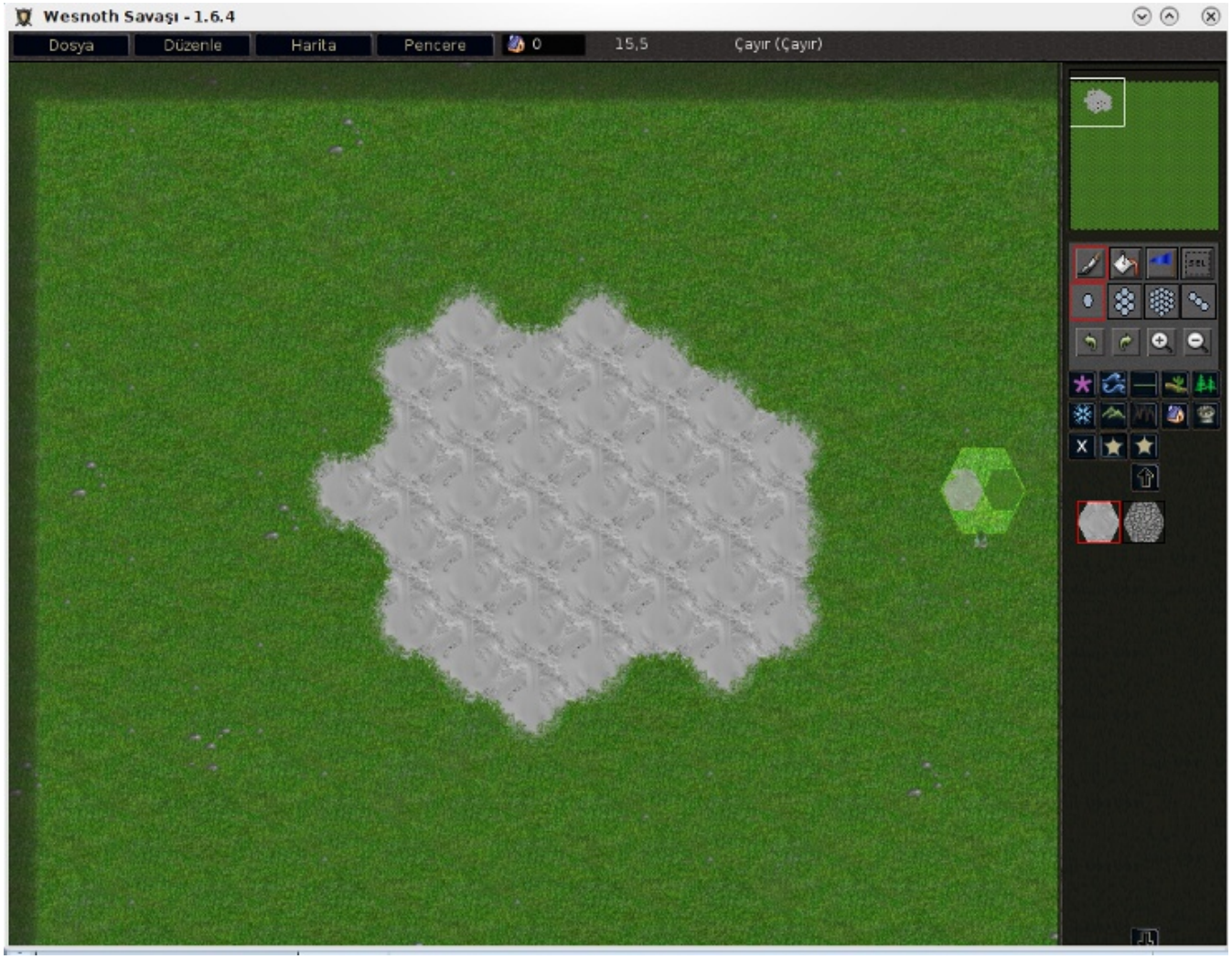
2- `/usr/share/wesnoth/data/core` klasörü altındaki `terrain.cfg` dosyasını düzenleyerek paletimizin özelliklerini tanımlıyoruz. Şu kodu dosyaya ekliyoruz:

```
[terrain]
      symbol_image=dark       (sembol resmi)
      id=dark                 (palet kimliği)
      name=_ “Karanlık”       (palet adı)
      string=Yyx              (palet kodu)
      aliasof=Ww              (sığ su; bu paletin üzerinde
                              bulunan karakterlerin
                              etkinlikleri %20'ye;
                              hareket puanları da yarıya
                              düşecek.)
      unit_height_adjust=-2   (üniteleri 2 birim alçalt)
      heals=-2                (-2 sağlık ver (2 YP azalt))
      editor_group=flat       (paleti Haritacı'da düzlük
                              grubu altına at)
[/terrain]
```

3- `/usr/share/wesnoth/data/core` klasörü altındaki “`terrain-graphics.cfg`” dosyasını düzenleyerek, paleti tanımlıyoruz. Sadece tek bir resim oluşturduğum için herhangi bir parametre kullanmama gerek yok. Ancak siz, birden fazla resim veya katman oluşturarak, görselliği güzelleştirebilirsiniz. Şu kodu, dosyanın “`BASE TERRAINS`” kısmına ekliyoruz.

```
{TERRAIN_BASE    Yyx      dark}
```

Sonrasında Haritacı'yı açıp, tasarla-dığımız haritanın istediğimiz yerine yeni paletimizi koyuyoruz. İşte lanetli bölgemiz hazır. Gri bölgede bulunan üniteler, %20 etkinlikle savaşacak, hareket puanları yarıya inecek ve her el başında 2 YP kaybedecekler.



Böylece harita yapısının temelini tanımış olduk. Önümüzdeki ay, biraz daha çetrefilli bir konu olan, karakter oluşturma konusu ile devam edeceğiz. Bir sonraki ay görüşmek üzere.

Hamit Giray Nart
hamit@pardus-linux.org

Giriş

1979 yılında, yani konsol oyunlarının altın çağı olan yıllarda, Atari Inc. “Asteroids” adlı bir oyun çıkarmıştı. Vektör grafiklere sahip olan oyunda, ortada duran üçgen şeklinde bir uzay gemisini yönetip, sağdan soldan geçen gök taşlarını vurarak ortalığı temizlemeye çalışırdık.

VioletLand oyununu oynamaya başladığım anda, Asteroids adlı bu oyun aklıma geldi. Asteroids oyununu bilenler, VioletLand oyununun türünü anlamışlardır sanırım. Oyun, “kuşbakışı hepsini vur”(top down shoot'em up) tarzında, platform bağımsız bir oyun. GNU/Linux, Mac, FreeBSD ve Windows sürümleri mevcut. Açık kaynaklı olan oyunumuz GPL ve CCL ile lisanslanmış.

Senin gibi bir kızın, bunun gibi bir yerde ne işi var? Güzel bir soru. Ancak cevabı yok. Hikaye falan yok. Sadece Violet adında bir kızcağız ile zombi ve dev örümceklerden oluşan bir canavar ordusu. Canavarların istilasına uğramış topraklarda, ölüme mahkum olmuş Violet'i yönetiyorsunuz. Amacınız ise, ölmeden önce, mümkün olduğu kadar fazla zombi ve örümcek vurarak, deneyim puanı toplamak.

Kurulum

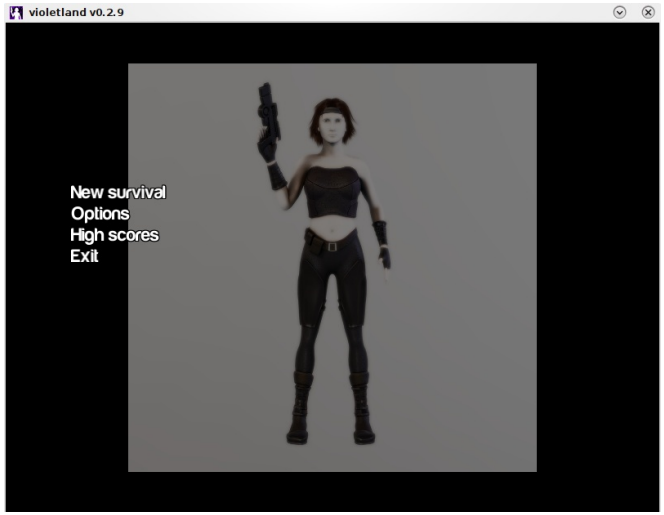
VioletLand oyununun kaynak kodları, http://code.google.com/p/violetland/ adresinden indirilip derlenebileceği gibi; PiSi paketi Pardus-Linux.Org tarafından hazırlanan P2009-free deposundan edinilebilir. P2009-free deposu hakkında ayrıntılı bilgiyi `http://forum.pardus-linux.org/viewtopic.php?f=329&t=21222` adresinden edinebilirsiniz.

P2009-free deposu eklenmiş bir sistemde yönetici yetkileri ile verilecek olan “`pisi it violetland`” komutu ile oyunu kurabilirsiniz.

Oynanış

Oyun kuruluduğunda, oyunu başlatan kısayolu Uygulamar menüsündeki Oyunlar bölümüne yerleşiyor.

Oyunu başlattığınızda aşağıdaki ekran karşınıza gelecek. “New Survival” ile yeni oyuna başlıyoruz. “Options” ise oyunun seçeneklerine girmemizi sağlıyor. Buradan ses, görüntü ayarları dışında oyunun genel ayarlarını da yapabiliyoruz.



Oyun, CrimsonLand oyununun bir kopyası. Ancak ondan farklı olarak, dinamik

gece-gündüz özelliği var. Oyunda, Violet bir canavarlar ordusuna karşı savaşıyor. Amacınız ise, ölmeden, olabildiğince fazla deneyim puanı toplayabilmek. Bunu ise altı değişik silah kullanarak yapabiliyorsunuz.

Silahlarınız:

PM Tabanca: Başlangıç silahınız. Sekiz mermi alan yarı otomatik tabanca.

Remington508: Bu tip oyunların olmazsa olmaz silahı pompalı tüfek.

*Uzi:*Hızlı bir hafif makineli olan Uzi.

AK-47: AK47 piyade tüfeği. Uzi'den daha güçlü ve isabetli ama daha yavaş bir silah.

Lazer Tüfeği: Yeşil bir lazer ışını silahı. En önemli özelliği, doğrultusundaki tüm yaratıklara hasar vermesi. İkinci özelliği ise, zehirli mermi geliştirmesinden etkilenmemesi.

M32 Bomba Atar: Oyundaki en etkili silah. Tüfek bombası atan bu silah, toplu katliam için birebir.

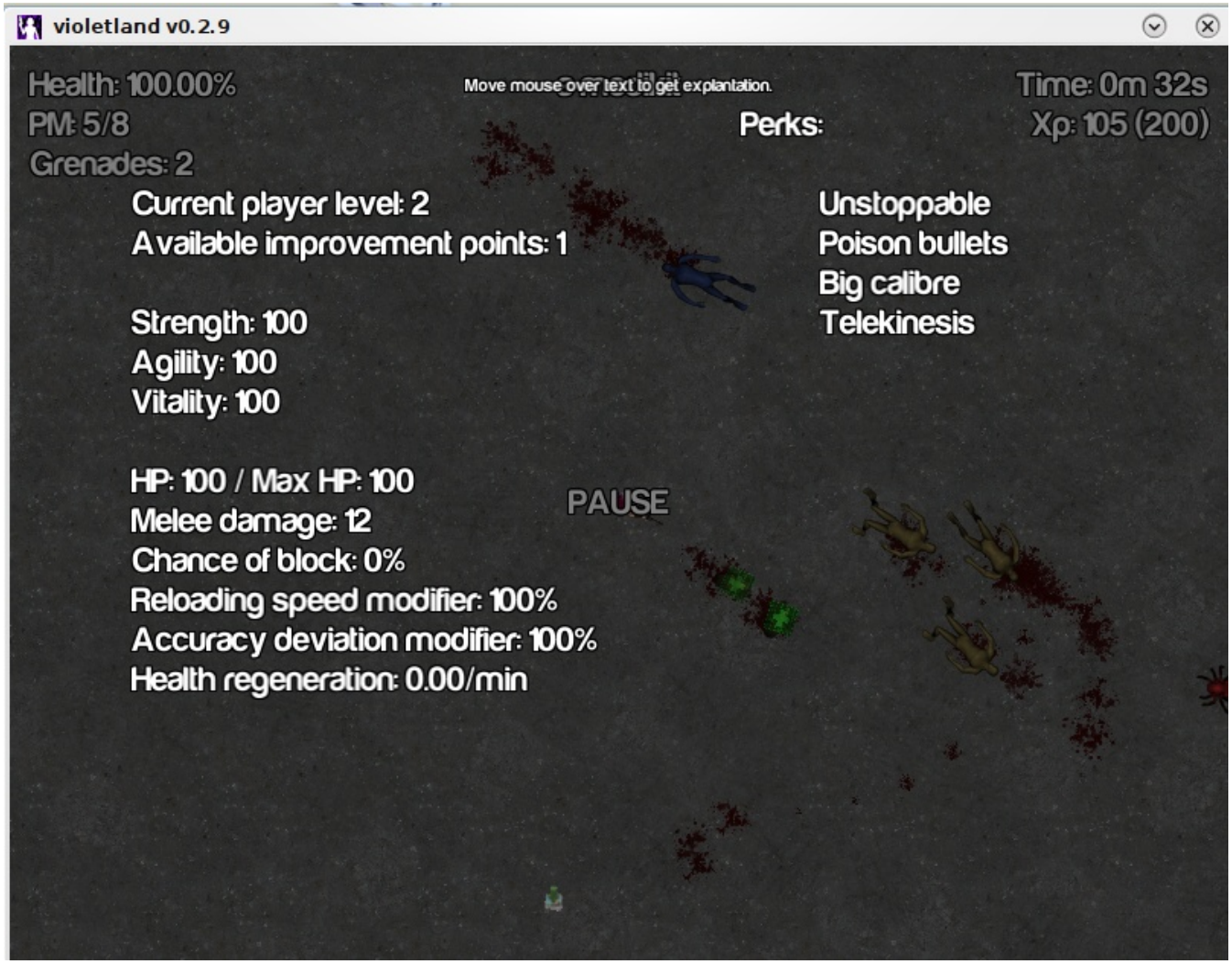
Bu silahları, öldürdüğünüz düşmanlardan alabiliyorsunuz. Ancak silahı alınca, o silahla oynuyorsunuz. Yani tek bir silah taşıyorsunuz. Silah değiştirmek istiyorsanız, bir düşmandan düştüğünde, yerden almak zorundasınız.

Yararlı nesneler de aynı şekilde düşmanlardan düşüyor. Bunlar sağlık çantaları, mermi güçlendirme ve çevrenizdeki düşmanları geçici süre dondurma gibi nesneler.

Oyunun ekranına bakacak olursak, sol yukarıda sağlık seviyemizi görebiliyoruz. Altında kullandığımız silah ve mermi miktarı var. Soldaki rakam, mevcut mermi sayısını gösterirken; sağdaki maksimum kapasiteyi gösteriyor. Merminiz sınırlı değil. Şarjör boşalınca otomatik dolduruyor. Ya da boşalmasını beklemeden sağ fare tuşu ile de silahı doldurabiliyorsunuz. “g” tuşu ile nişan lazerini, “f” tuşu ile de feneri açıp kapatabiliyorsunuz.

Violet'i WASD tuşları ile hareket ettirirken, fare ile ateş edeceğimiz yöne dönüyoruz. Sol fare tuşu ile ateş ediyoruz. Boşluk tuşu ise el bombası atmamızı sağlıyor.

Sağ üstte ise süreyi, altında da tecrübe puanımızı görüyoruz. Soldaki sayı, mevcut puanımızı gösterirken; sağdaki sayı ise bir sonraki kademeye geçeceğimiz tecrübe puanını gösteriyor. Violet, bir



üst kademeye geçince gücü doluyor ve bir geliştirme puanı kazanıyor. “c” tuşu ile karakter ekranına giriyoruz.
Bu geliştirme puanlarımızı harcayacağımız özellikler ise:

Fiziksel Güç (Strength): Silahsız hasar (Melee damage) özelliğini arttırırken, nişan alma becerisini düşürür. Canavarlar size değdikleri zaman, enerjinizi azaltırlar. Ancak siz de onlarınkini azaltırsınız. Silahsız hasar, sizin ne kadar enerji azaltabileceğinizi gösterir.

Çeviklik (Agility): Canavarların saldırılarını savuşturma şansını (Chance of block) arttırırken; yeniden doldurma hızınızı azaltır.
Yaşam Gücü (Vitality): Yaşam puanı (HP) ve sağlık toparlanma (Health regeneration)özelliklerini arttırır.

Bunların dışında ek yetenekler de (Perks) kazanabilirsiniz. Bunlar:

Durdurulmazlık (Unstoptable): Canavarlar sizin yolunuzu bloke edemezler, ancak hala size zarar verebilirler.

Zehirli Mermiler (Poison bullets): Merminizi yiyen canavar zehirlenir ve ölene kadar yavaş yavaş enerji kaybeder. Sadece Uzi ve pompalı tüfek için.

Geniş Çaplı Cephane (Big calibre): Mermileriniz aynı doğrultudaki birden fazla düşmana hasar verebilir.

Telekinezi (Telekinesis): Sağlık çantaları, silahlar gibi düşmanlardan düşen yararlı nesneler, yavaş yavaş size doğru hareket ederler.

Sonsöz

Oyunun herhangi bir sonu yok. Canavarlar her yerden ve hiç durmadan; gittikçe kalabalıklaşan bir şekilde geliyorlar. Siz ise ateş ede ede, devamlı kaçıyorsunuz.

Grafik ve sesler etkileyici. Özellikle de Violet'in ölürken attığı çığlıklar. :-)

Oyunun arka zemini değişkenlik gösteriyor. Ancak kahverengi toprak olan zemin, zombilerin görünmesini güçleştiriyor. Bu olmadık yerlerde sorun olabiliyor. Müzik ise, düştüğünüz duruma göre (canavarlar çok kalabalıklaşınca ya da yaşam gücünüz kritik derecede azaldığında) değişiyor.

Harita boş. Bence yıkık duvarlar, ağaçlar veya bazı binalar gibi engeller koysalar iyi olurdu.

VioletLand, oyuncuların alışageldiği oyunlardan biraz farklı. Bir gelişim çizgisi, yani bir senaryo,seçenekler ve bir sonu yok. Ancak zaman geçirmek isteyenler için oldukça güzel bir seçenek. Yalnız hırs yapıp da, rekor kırmak gibi bir gaflete düşerseniz, zombi filmlerindeki o klasik sözü, bu oyunda da sık sık söylersiniz: “LANET OLSUN! HERYERDELER.”

Kalın sağlıcakla...

Hamit Giray Nart
hamit@pardus-linux.org

Kütüphane Dosyaları İle Çalışma

Bu yazıda sizlere, C dilinin standart kütüphanelerini kullanabilmek için gerekli konfigürasyonları anlatacağım. Ayrıca, konsol uygulamalarımızda giriş ve çıkış işlemlerini yapabilmemizi sağlayan fonksiyonları barından "curses" kütüphanesinden bahsedeceğim.

GCC derleyicisi ile derleme işlemi yapılacak uygulamada kütüphane dosyaları kullanılmışsa, derlenmeden önce bunun bir şekilde bildirilmesi gerekir. Bu bildirimi yapmanın yolu; derleme işlemi için gereken komuta –l parametresini dahil etmektir.

```
gcc ornek.c –o yeni –lncurses
```

Yukarıdaki gibi bir derleme işleminde, kaynak kod içerisinde kullanılan `ncurses.h` kütüphanesi kaynak koda eklenmeyecek, sadece ilgili kütüphaneye, programın çalışması esnasında erişilecektir. Bu parametre, kullanım amacına göre farklı olabilir. Örneğin; matematiksel fonksiyonların bulunduğu kütüphane olan `math.h` kütüphanesini eklemek istersek;

```
-lm
```

parametresini girmemiz yeterli olacaktır. Bazı kütüphaneler için gereken parametreler aşağıda verilmiştir:

MySQL Kütüphanesi	-lmysql
Matematik İçin	-lm
Şifreleme İçin	-lcrypt
PostgreSQL İçin	-lpq
POSIX İçin	-lpthread
Glib İçin	-lglib
Vga İçin	-lvga

Bazen, kullandığımız kütüphanelerin başlık dosyaları, derleyicinin varsayılan olarak baktığı dizinde olmayabilir. Bu durumda –l parametresi ile birlikte, ilgili başlık dosyasının bulunduğu dizin de belirtilmelidir.

```
gcc ornek.c –o yeni -I/home/ornek/kutuphaneler –lncurses
```

Eğer kütüphanenin nesne dosyası başka bir dizinde ise, bunu da –L parametresi ile belirtmemiz gerekecektir.

```
gcc ornek.c –o yeni -I/home/ornek/kutuphaneler
            -L/usr/local/lib  –lncurses
```

-static parametresi

Kaynak kodumuz içerisinde, `#include <time.h>` gibi bir bildirim yapmış olalım. GCC derleyicisi ile derleme yaparken `-static` parametresini kullandığımızda, time.h kütüphanesi kaynak kodumuz içerisine eklenecektir. Elbette bu işlem, uygulamamızın boyutunu arttıracaktır.

curses kütüphanesi

GNU/Linux sistemi üzerinde, konsol uygulamalarında ekran işlemleri yapmak için kullanılan en yaygın kütüphane; "curses" kütüphanesidir. "curses" kütüphanesi oluşturulduktan bir müddet sonra iyileştirilerek, "ncurses" yani "newcurses" adını almıştır. Klasik "curses" kütüphanesinin başlık dosyası, curses.h; yeni "curses" kütüphanesinin başlık dosyası ise ncurses.h biçimindedir. Bu kütüphaneyi kullanabilmek için, derleme işleminde, –lncurses parametresinin girilmesi gerekir.

"curses" kütüphanesinde, standart C fonksiyonları ile aynı işlemleri yapan bir çok fonksiyon mevcuttur. Örneğin; ekrana bir şeyler yazdırmak için kullanılan "printf()" fonksiyonu ile aynı işlemi yapan "printw() ncurses" fonksiyonu mevcuttur. Benzer olarak, girilen veriyi almamızı sağlayan "scanf()" fonksiyonu yerine, "scanw() ncurses" fonksiyonu vardır.

"curses" fonksiyonları kullanılmadan önce, "initscr()" fonksiyonu kullanılmalıdır. İşlemimiz bittikten sonra ise, "endwin()" fonksiyonu çağrılmalıdır.

Şimdi sizlere bir örnek yapıp açıklayacağım.

```
#include <ncurses.h>
#include <stdio.h>

//değişkenlerin tanımlanması
int main(){
signed int dizi[5];
int sayac1,sayac2,gecici;
//değişkenlerin tanımlanması

//dizinin doldurulması
for(sayac1 = 0 ; sayac1 < 5 ; sayac1++){
initscr();
scanw("%d",&dizi[sayac1]);
endwin();
}
//dizinin doldurulması

//sıralama ve çıkış
for(sayac1=0;sayac1 < 5;sayac1++){

for(sayac2 = 0;sayac2 < 4;sayac2++){
      if(dizi[sayac2] < dizi[sayac2 + 1]){
            gecici = dizi[sayac2];
            dizi[sayac2] = dizi[sayac2 + 1];
            dizi[sayac2 + 1] = gecici;
      }
}
      printf("%d | ",dizi[4 - sayac]);
}
//sıralama ve çıkış

return 0;
}
```

Program, kullanıcıdan beş adet tamsayı alıyor ve bunları küçükten büyüğe doğru sıralıyor. Kodlara bakarsak, program, 3 bölümden oluşuyor.

Değişkenlerin Tanımlanması: Burada, programımızda kullanacağımız değişkenler tanımlanıyor.

Dizinin Doldurulması: Tanımladığımız "`dizi[5]`" adlı tamsayı türünden dizimizi, kullanıcıdan aldığımız sayılar ile doldurmaktayız. Bu işlemi, beş defa dönecek bir "for" döngüsü ile yapıyoruz.

Sıralama ve Çıkış: Bu kısımda ise, diziye doldurduğumuz sayıları, küçükten büyüğe doğru sıralatıyor ve ekrana çıkış veriyoruz. Bu işlemi ise iki "for" döngüsü ile yapmaktayız. İlk "for" döngüsü, beş defa dönecek; ilkinin her dönüşünde, ikinci for döngüsü ise dört defa dönmüş olacaktır. Sıralama işlemi, ikinci "for" döngüsünde şu mantıkla gerçekleşmektedir: Döngünün ilk turunda, dizinin 0. elemanı ile 1. elemanı karşılaştırılır. Eğer ilk eleman, ikinci elemandan küçükse, bu değerler yer değiştirilir. Ve döngüden çıkıldığında, en küçük rakam, dizinin en büyük indexine (burda `dizi[4]`' tür bu) yerleşmiş olur. Döngünün her turunda bu olay, karşılaştırılan dizinin +1. elemanı için tekrarlanır. Ekranda çıkarma şekline göre, diziyi, büyükten küçüğe veya küçükten büyüğe sıralayabilirsiniz.

Aslında yukarıdaki sıralama mantığı, "`bubble sort`" algoritmasıdır.

Algoritma, bir problemin çözümüne yönelik yapılacak işlemlerin tamamıdır. Bir de akış diyagramı vardır. Akış diyagramı ise, algoritmanın standartlaştırılmış bazı özel sembollerle gösterilerek ifade edilmiş halidir. Büyük çaplı projelerde, algoritma ve akış diyagramı tasarlamak, projeyi kodlarken çıkması muhtemel bir çok karışıklığı önleyecektir. Bu uygulamalar, projenin güncellenmesi veya hataların çözülmesi gibi işlemlerde de çok önemli rol oynarlar. Büyük projelerde temel kural, işi parçalamak ve bir yerden bu parçaları bütünleştirip yönetmektir.

Bu yazımızın da sonuna geldik. Herkese sağlık ve mutluluklar dilerim...

Armağan Can
armagan@pardus-linux.org

Bu dersimiz de Django'da Fotoğraf galeri oluşturmaktan bahsedeceğiz Daha önce ki derslerimizin hatırlatmalarını da yapacağız. Bu konu diğerlerine göre biraz daha kısa olacak çünkü diğer uygulamalar gibi oluşturulacak bu yüzden html dosyaları üzerinde daha fazla duracağız.

Geçen sayımızdaki projemiz üzerinden devam edeceğimiz için `settings.py` dosyamızı düzenlememiz gerekiyor.

```
# -*- coding: utf-8-*-
import os,sys
DIRNAME = os.path.dirname(__file__)
DEBUG = True
TEMPLATE_DEBUG = DEBUG
ADMINS = (('Muslu', 'musluyuksektepe@gmail.com'),)
MANAGERS = ADMINS
DATABASE_ENGINE = 'sqlite3'           # 'postgresql_psycopg2',
'postgresql', 'mysql', 'sqlite3' or 'oracle'.
DATABASE_NAME = '/home/muslu/dj ango/blogproje/deneme.db'
DATABASE_USER = ''
DATABASE_PASSWORD = ''
DATABASE_HOST = ''
DATABASE_PORT = ''
ROOT_URLCONF = 'blogproje.urls'
TIME_ZONE = 'Europe/Istanbul'
TIME_FORMAT='H:i'
DATETIME_FORMAT='d F Y l - H:i'
DATE_FORMAT='d F Y l'

LANGUAGE_CODE = 'tr-TR'
USE_I18N = True
MEDIA_ROOT = os.path.join(DIRNAME, 'static/')
MEDIA_URL = "/static/"
STATIC_DOC_ROOT = "/static/"
ADMIN_MEDIA_PREFIX = '/media/'

DEFAULT_CHARSET = 'utf-8'
gettext_noop = lambda s: s
LANGUAGES = (('tr', gettext_noop('Türkçe')),)

SITE_ID = 1

SECRET_KEY ='lp6x_4-7d)=p=lhpt-
9gxjpwsovfx9y!ag)8j_&$891r8n93vp'
TEMPLATE_LOADERS = (

'django.template.loaders.filesystem.load_template_source',

'django.template.loaders.app_directories.load_template_source
',
#     'django.template.loaders.eggs.load_template_source',
)
TEMPLATE_CONTEXT_PROCESSORS = (
    "django.core.context_processors.auth",
    "django.core.context_processors.i18n",
    "django.core.context_processors.debug",
    "django.core.context_processors.media",
    "django.core.context_processors.request",

)
MIDDLEWARE_CLASSES = (
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
  'django.contrib.auth.middleware.AuthenticationMiddleware',
```

```
    'django.middleware.doc.XViewMiddleware',
    'django.middleware.locale.LocaleMiddleware',

'django.contrib.flatpages.middleware.FlatpageFallbackMiddlewar
e',
)

TEMPLATE_DIRS = (os.path.join(DIRNAME, "templates"))
AUTH_PROFILE_MODULE = 'accounts.UserProfile'

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.flatpages',
    'django.contrib.markup',
    'django.contrib.humanize',
    'django.contrib.admin',
    'blogproje.blog',
    'blogproje.fotoalbum',
    'sorl.thumbnail',
    )
```

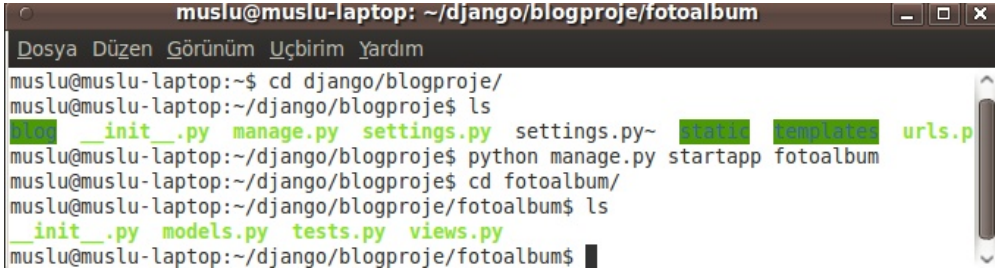
Kullanılacak uygulamalar kısmına blogproje projemiz altında blog ve fotoalbum adında uygulamalarımızı ekledik. Ayrıca Sorl tarafından hazırlanmış küçük fotoğraflaştırma eklentisinide kullanacağız. Bu eklenti PIL (Python Image Library) kullanmaktadır. Açıklamları ve detaylı kullanım alanları için burayı tıklayabilirsiniz. Bura

Ayar dosyamızı düzenledikten sonra olacak dediğimiz uygulamamızı hazırlamaya başlayalım.

Komut sistemimize geçip;

```
cd /home/muslu/django/blogproje/
python manage.py startapp fotoalbum
```

yazıyoruz.



Evet, Fotoalbum uygulamamız oluşturuldu. Şimdi sırası ile düzenlemeye başlayalım.

Önce models.py dosyamızı düzenleyip veritabanına bilgi girmemiz gerekiyor.

Hani alana ne bilgisi girilecek?
Kısıtlamalar nelerdir?
Yardımcı açıklamalar
Karakter sayıları...

`models.py` dosyamızı açıyoruz

```
#-*- coding: utf-8-*-
#Türkçe karakter kullanılması için
from django.db import models
```

```
#django nun kendi modelleri
from datetime import datetime
#tarih kaydetmek
from sorl.thumbnail.fields import ImageWithThumbnailsField
#Sorlun küçük resim uygulaması

class Fotoalbum(models.Model):
    baslik = models.CharField(max_length=300, help_text='Max:
300 Karakter')
    slug = models.SlugField(max_length=300, help_text='Max:
300 Karakter')
    tarih = models.DateTimeField(default=datetime.now)
    guncelleme = models.DateTimeField(auto_now=True)
    aciklama = models.CharField(max_length=1000,blank=True,
help_text='Max: 1000 Karakter')

    def __unicode__(self):
        return self.baslik
#baslik hücresine utf8 özelliği veriyoruz. Yani bu alana utf8
karakterleri gelecek diye önemli bilgi veriyoruz

    class Meta:
        verbose_name_plural = "Albümler"
# Meta sınıfına kişileştirme yaparak yönetim panelindeki
uygulama adını kendimiz belirliyoruz

class Fotolar(models.Model):
    fotoalbum = models.ForeignKey(Fotoalbum)
#fotoalbum listemizi foreignkey özelliği ile üst tablodan
bilgi çekiyoruz
    fotoaciklama = models.CharField(max_length=200,blank=True,
help_text='Max: 200 Karakter')
    image = ImageWithThumbnailsField(blank=True,
upload_to='fotoalbum/%Y/%m/%d', thumbnail={'size': (150,
150)})
#küçük resimlerin kayıt yapılacağı klasöre yıl  ay gün
klasörleri oluşturup iç içe kayıt ettiriyoruz

    class Meta:
        verbose_name_plural = "Fotoğraflar"
# bu metanın adı da fotoğraflar olsun
#NOT: Burada kullandığımız foreignkey özelliği; üst tablodan
aldığımız bilgilere sınırsız resim eklemek için kullandık.
Kullandığınız yönetim paneli temasına göre bu özellik
değişecektir.
```

Veritabanımıza istediğimiz özelliklerde ki alanları yazdık. Şimdi `views.py` ile bu bilgileri yönlendirme yapacağız.

`views.py` dosyamızı açıyoruz

```
#-*- coding: utf-8-*-
import re
from datetime import datetime
from django.shortcuts import render_to_response
from blogproje.fotoalbum.models import Fotoalbum
# blogproje projemizden fotoalbum modelsinden
Fotoalbumtablosunuı yükledik

def fotoalbumonsayfa(request):
    gonderiler = Fotoalbum.objects.all().order_by('-
tarih')[:5]
    sayfayagidenbilgiler = {'gonderi_listesi':gonderiler,}
```

```
    return
render_to_response('fotoalbum/fotoalbumliste.html',sayfayagide
nbilgiler)
#blog uygulamamızdaki gibi
# gonderiler aında bir liste ve bu listeye Fotoalbum
tablosundaki verilerin tümünü tarihe göre ters sırala ve son 5
kaydı göster dedik. Bu gonderi_listesine eşitle ve templates
klasöründe ki fotoalbum klasör altında fotoalbumliste.html
dosyasına gönder.


def fotoalbumdetay(request,slug):
    gonderiler = Fotoalbum.objects.all()
    sayfayagidenbilgiler = {'gonderi_listesi':gonderiler,}
    fotoalbumdetay = gonderiler.get(slug=slug,)

sayfayagidenbilgiler.update({'fotoalbumdetay':fotoalbumdetay})

    return
render_to_response('fotoalbum/fotoalbumtekliste.html',sayfayag
idenbilgiler)
# detay linki tıklandığında sluga göre dizileme yap gelen
bilgileri güncelle ve templates klasörü altında ki fotoalbum
klasörünün içinde fotoalbumtekliste.html dosyasına gönder.
```

Evet. Veritabanımıza bilgileri girdik bu bilgileri çekip html dosyalarımıza gönderdik. Şimdi `admin.py` dosyamızı düzenlememiz gerekiyor.

`admin.py` adında bir dosya oluşturuyoruz.

```
# -*- coding: utf-8-*-
from blogproje.fotoalbum.models import Fotoalbum, Fotolar
from django.contrib import admin

class FotolarInline(admin.TabularInline):
    model = Fotolar
    extra = 1
#FotolarInline sınıfına Tabözelliği veriyoruzve modelleri
Fotolardan alıp teker teker çoğalt diyoruz.
#Hani demiştim ya yönetim panelinize göre bu kullanım şekli
değişebilir. Bazı temalarda stil olarak eklenen inline
özelliği + linki getirir siz her tıkladığınız da alta doğru
bir tab açılır ve siz sınırsız sayıda giriş yapabilirsiniz.

class Fotoalbum_Admin(admin.ModelAdmin):
    prepopulated_fields = {"slug": ("baslik",)}
    list_display = ('baslik', 'tarih', 'aciklama')
    list_filter = ['tarih']
    date_hierarchy = 'tarih'
    ordering = ('baslik',)
    search_fields = ('baslik',)
    inlines = [FotolarInline]
#Özellikler blog admin.py de kigib biz sadece inlines olarak
Fotolarinline sı da ekle dedik

admin.site.register(Fotoalbum, Fotoalbum_Admin)
#kayıt ettirip yönetim panelinde bunu göstermesini söyledik
```

Yönetim paneli özelleştirmesinide yaptıktan sonra sıra da `urls.py` dosyamızı düzenlememiz kaldı.

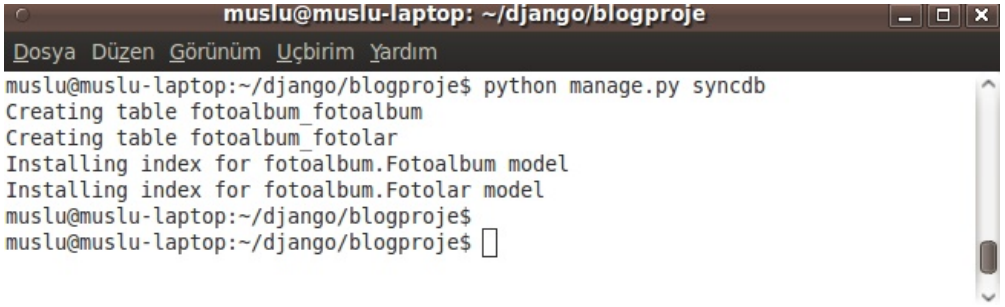
`urls.py` dosyamızı açıyoruz.

```
from django.conf.urls.defaults import *
from blogproje.blog.views import *
from blogproje.fotoalbum.views import *
#fotoalbum uygulamasında ki views sınıflarını import ettik
from django.contrib import admin
admin.autodiscover()
import os,sys

KlasorYolu = os.path.dirname(__file__)

urlpatterns = patterns('',

(r'^$','django.views.generic.simple.direct_to_template',{'temp
late':'index.htm'},'index'),
    (r'^static/(?P<path>.*)$', 'django.views.static.serve',
{'document_root': os.path.join(KlasorYolu, "static/")}),
    (r'^blog/$',blogonsayfa),
    (r'^blog/([\w\-]+)/$',blogdetay),
    (r'^fotoalbum/$', fotoalbumonsayfa),
    (r'^fotoalbum/([\w\-]+)/$', fotoalbumdetay),
#fotoalbum uygulamasının sınıflarını /fotoalbum/ linki gelirse
çalıştır dedik
    (r'^admin/', include(admin.site.urls)),
)
```

Artık uygulamamızı veritabanına ekleyebilir ve denememizi yapabiliriz. Komut satırına geçip

```
python manage.py syncdb
```

komutunu veriyoruz.



Herhangi bir hata yapmadıysak bu ekranı görmemiz gerekiyor.

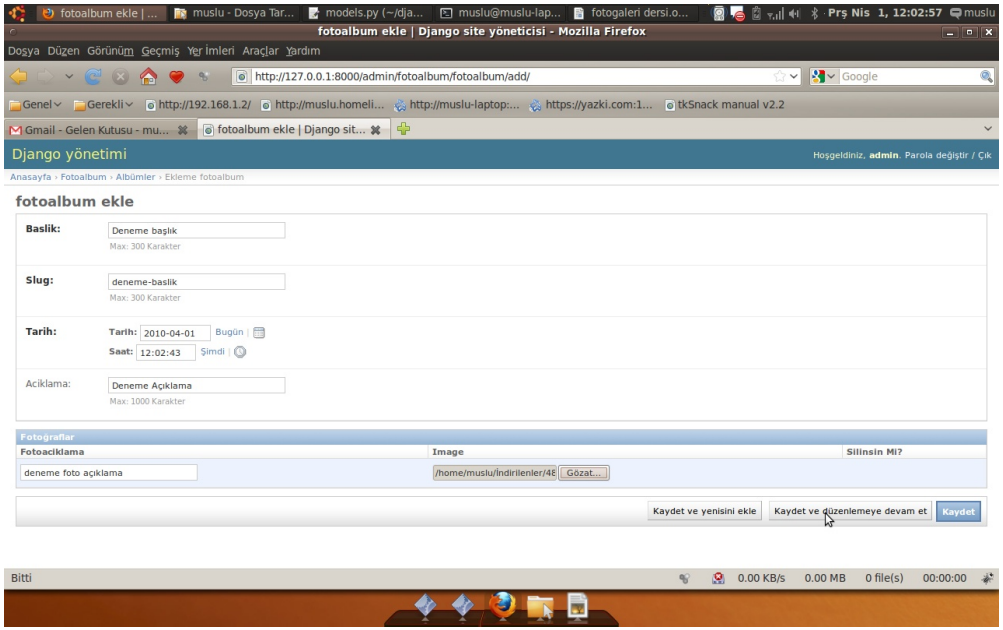
Şimdi `python manage.py runserver` komutu ile denemeye başlayabiliriz.

`index.htm` dosyamızı açıp fotoalbum için bir link koyalım. Ayrıca efektler kullanmak içinde lightbox eklentisini de indirip static klasörümüze kopyalıyoruz.

```
{% load markup %}
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<title>{%block title%}{%endblock title%} | Yazki.com</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="/static/default.css" rel="stylesheet"
type="text/css" />
```

```
<script type="text/javascript"
src="/static/lightbox/js/jquery-1.3.2.min.js"></script>
<script type="text/javascript"
src="/static/lightbox/js/jquery.lightbox.min.js"></script>

</head>
<body>
<a href="/">Anasayfa</a>
<a href="/blog/">Blog</a>
<a href="/fotoalbum/">Fotoalbum</a>
{%block main%}{%endblock main%}

</body>
</html>
```

Admin panelimize giriş yapıp kayıt girelim ve bu kayıtları html dosyalarımızda gösterelim.



Kaydet ve düzenlemeye devam et 'yi tıklayarak aynı kayda istediğimiz kadar resim ekleyebiliriz.
(Harici bir yönetim paneli teması kullanmıyorsanız)

İstediğiniz kadar kayıt ve resim ekleyebilirsiniz.

Artık html dosyalarımıza geçelim.

Fotoalbum adında bir klasör oluşturuyoruz ve `fotoalbumliste.html` adında bir dosya oluşturuyoruz. Bu dosya `/fotoalbum/` olarak link geldiğinde çalışacak ve `fotoalbumonsayfa`'daki verileri gösterecek.

`fotoalbumliste.html` adında bir dosya oluşturuyoruz:

```
{% extends "index.htm" %}
<title>{%block title%}Fotoğraflar{%endblock title%}</title>
{% load markup %}
{% load thumbnail %}
// sorlun hazırladığı thumbnail eklentisini de ekliyoruz.
{% block main %}
  {% for fotooku in gonderi_listesi %}
<h1><a href="/fotoalbum/{{ fotooku.slug
}}/">{{fotooku.baslik}}</a></h1>
<p>{{fotooku.aciklama}}</p>
{{ fotooku.tarih|date }}
{% endfor %}
{% endblock %}
```

Başlıklara slug linkini veriyoruz ve `urls.py`'den detay sayfasına gönderiyoruz.

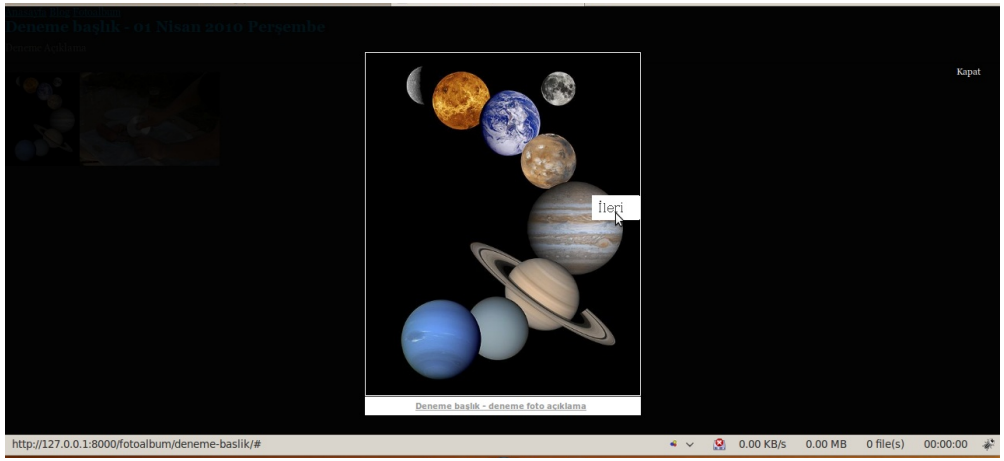
Detayları göstermek için `views.py` de yazdığımız `fotoalbumdetay` fonksiyonu ile `fotoalbumtekliste.html` dosyasına yönlendiriyoruz.

`fotoalbumtekliste.html` adında bir dosya oluşturuyoruz:

```
{% extends "index.htm" %}
<title>{%block title%}{{fotoalbumdetay.baslik}}{%endblock
title%}</title>
{% load markup %}
{% load thumbnail %}

{% block main %}
<h2>{{fotoalbumdetay.baslik}}</h2>
 <p>{{fotoalbumdetay.aciklama}}</p>
{% for fotooku in fotoalbumdetay.fotolar_set.all %}
<a rel="lightboxtour" href="{% thumbnail fotooku.image
1024x768 %}" title="{{fotoalbumdetay.baslik}} - {{
fotooku.fotoaciklama }}">
<img src="{% thumbnail fotooku.image 200x120 %}"
alt="{{fotoalbumdetay.baslik}} - {{ fotooku.fotoaciklama }}"
border="0"></img></a>
{% endfor %}
<p>{{ fotoalbumdetay.tarih|date }}</p>
{% endblock %}
```

`{% thumbnail fotooku.image 200x120 %}` yazarak küçük resim boyutlarını belirtiyoruz. Hangisi uygun ise ona göre küçültme yapar ve resim kalitesini bozmadan uygun boyutta sayfada gösterir. Farklı boyutlar deneyebilirsiniz.

Herhangi bir resmi tıkladığınızda bu şekilde resimleri sıralatabiliriz.



Kullandığınız hazır temalar var ise bu kodları aralara koyarak istediğiniz gibi efektler uygulayabilir ve görsellik katabilirsiniz.

Muslu YÜKSEKTEPE
musluyuksektepe@gmail.com
muslu@pardus-linux.org
www.yazki.com

Giriş

Geleneksel Linux dağıtımları, programları sıkıştırılmış paketler halinde dağıtırlar. Paketler geliştirici ekibin belirlediği ve yapı için en uygun olan kıvamda, hemen hemen tüm sistemlerde çalışmaları için belli başlı mimarilerde belirli seçeneklerle derlenirler. Aynı durum çekirdek paketleri için de geçerlidir ve performans açısından çok daha vahim bir unsur içerir. Çekirdek paketi dağıtımın kurulduğu tüm sistemlerde çalışmak üzere hazırlanır ve alakalı alakasız pek çok sürücüyü ve opsiyonu içerir. Bu ise hantallık anlamına gelir. Örneğin, bir masaüstü sistem kullanıcısı iseniz, dizüstü cihazlara ait sizi, hiç de ilgilendirmeyen sürücüleri de açılışta yüklemek zorunda kalırsınız.

Bu program paketlerinde, benim bohça demeyi tercih ettiğim bir üst uzantı bulunur. Örneğin Debian'daki “deb” uzantısını ele alalım. Bir deb paketi tam manasıyla bir bohçaya benzer, içinde bazı sıkıştırılmış dosyalar ve ufak tefek metin dosyaları bulunur. Sıkıştırılmış dosyalardan biri programa ait verileri tutar, diğeri ise programı sisteme tanıtacak bilgileri saklar. Diğer dağıtımlar da genellikle bu konseptin dışına çıkmazlar.

Yaklaşım

Gentoo yukarı bahsettiklerim gibi sıradan bir dağıtım değildir. Her kullanıcı kendine has bir dağıtım oluşturur. Sistem, çekirdekten tutun da kullanılan programlara dek kullanıcının isteklerine ve donanımına göre şekillendirilebilir. Bu da belirli oranda performans artışı sağlar. Ayrıca kullanıcı, Gentoo ile diğer dağıtımlarda edinemeyeceği kadar bilgi edinebilir. Çünkü Gentoo, kullanıcısını sistemin derinliklerine inmeye ve araştırmaya zorlar. Depo yapısı da alışılmışın dışındadır. Normal dağıtımlarda paket oluşturma betikleri kullanıcının bilgisayarında bulunmaz ve paketler kullanıcı bilgisayarında derlenmez. Gentoo'da ise durum tamamen farklıdır, `/usr/portage` dizini altında tutulan paket oluşturma betikleri ya da efsaneleşmiş adıyla `ebuild` dosyaları, paket yönetim sistemi Portage'a programları, `make.conf` dosyası içinde belirtilen işlemci mimarisine göre derleyip sisteme kurması için gerekli bilgileri verir. Bu anında derleme yaklaşımı, Gentoo'nun hemen hemen tüm işlemci mimarileri ile geliştirici ekip pek yorulmaksızın çalışmasına müsaade eder.

Konsept hakkında konuşurken `USE FLAG` kavramından bahsetmezsek olmaz. Bunlar aslında, diğer dağıtımlarda paketçilerin inşa betiklerini yazarken kullandıkları derleme opsiyonlarından başka bir şey değildir. `USE FLAG`'ler yardımıyla bir programdan herhangi bir özelliği çıkarabilir ya da ekleyebilirsiniz. Örneğin programın kullandığı veritabanı türü ile oynayabilir ya da bir dosya türünü desteklememesini sağlayabilirsiniz. Bunlar tek tek değerlendirildiğinde ufak ayrıntılar olarak görülebilir. Ama bir bütün halinde bakıldığında sistem üzerinde geniş bir hâkimiyet sağlar ve istediğiniz gibi bir sistem ortaya çıkarabilirsiniz. Bu kavramı daha iyi anlamak için Gparted paketini inceleyebiliriz, aşağıda bu paketin kullandığı `USE FLAG`'ler görülüyor:

```
sys-block/gparted-0.4.5      USE="-debug
-dmraid -fat -gnome -hfs -jfs -kde -ntfs
-reiser4 -reiserfs -xfce -xfs"
```

Önünde “-”işareti olan flag'ler desteklenmeyecek. Yani hfs, jfs, ntfs, reiser4, reiserfs ve xfs dosya sistemleri Gparted tarafından desteklenmeyecek. Diğer flagler ise başka özellikleri açıp kapatmaya yarıyor. Program üzerindeki hâkimiyetinizi anlayabiliyor musunuz?

Tüm bu özelliklerinden dolayı Gentoo, ileri seviyedeki kullanıcılara hitap eder ve meta dağıtım olarak adlandırılır.

Kurulum

Sisteme yaklaşım farklıdır, kurulumun da bundan altta kalır yanı yoktur. Diğer dağıtımlardaki gibi klasik bir “kurulum CD'si” kavramı gelişmemiştir. Kurulum yapabilmek için kuluçka sistemler kullanılabilir. Bu kuluçka sistem herhangi bir dağıtım da olabilir, Gentoo'nun resmi LiveCD'leri ya da minimal ISO'ları olabilir. Yeter ki elinizin altında standart ve güncel bir GNU/Linux dağıtımınız olsun.

Yansılardan indireceğiniz stage-* dosyalarını kurulum yapacağınız dosya sistemine açıp kurulum kitabındaki yönergeleri takip ederseniz tecrübenize, işlemcinizin gücüne ve en önemlisi sabrınızla orantılı bir sürede Gentoo'ya kavuşabilirsiniz. Hatırlatmakta fayda var, Gentoo Handbook'un en güncel versiyonu yaklaşık 100 sayfa tutuyor, bu dokümanın kurulum için gerekli tüm bilgiler ve ilk kullanım bilgilerini içerdiğini belirtelim. Resmi Gentoo dokümanlarının tamamı bundan çok daha fazlasıdır. Tabii ki tüm bu dokümanları satır satır okumak zorunda değilsiniz, kimse bunu yapmıyor da zaten. Yapmanız gereken şey, kurulum ve temel kullanım belgelerine göz atmak ve forumlardan yardım istemek. Tabii bu noktada İngilizce bilgisi şart. Çünkü Gentoo'nun Türkçe bir kullanıcı topluluğu bulunmuyor.

Uzun lafın kısası, Gentoo ne yaptığını iyi bilen ve sistem üzerinde hâkimiyet kurmak isteyen kullanıcıların Linux dağıtımıdır. Program kurarken biraz beklemeniz gerekse de, sistem üzerinde kurduğunuz hâkimiyet ve sistem yönetimi hakkında edindiğiniz bilgiler paha biçilmez olacaktır.

Burak Sezer
bburaksezer@gmail.com

Özgür yazılım hızla gelişirken bilimsel programlar giderek artmakta. Özgür yazılım, bilimsel çalışmalar için çok önemli. Programların çok pahalı olması, kapalı kodlu olması ve bizim tarafımızdan geliştirilememesi kadar gereksiz bir şey olamaz. Bilim yapmak isterken, önüne bilimsel olmayan engeller çıkması hiç de güzel değil. Ama bilim adamları bunun da çözümünü buldu: GNUPlot ve Maxima'nın yanında güzel bir matematik yazılımı da Scilab. Scilab diğer yazılımlar gibi sadece komut vererek grafik çizme ya da denklem çözme yazılımı değil; aynı zamanda bir dil. Scilab dilini öğrenmek zorunda da değilsiniz. Çünkü, C ve Fortran da destekliyor.

Prorgamla yapabilecekleriniz şu şekilde:

*Lineer cebir
*Polinomlar ve rasyonel fonksiyonlar
*Yaklaşımlar
*Lineer ve non-lineer optimizasyon
*Düzenli diferansiyel denklemler ve diferansiyel cebirsel denklemler
*Lineer matrix eşitsizlik optimizasyonu
*Diferansiyel ve non-diferansiyel optimizasyon
*Sinyal işleme
*İstatistik

Scilab, pardus.org.tr ve pardus-linux.org depolarında bulunmayan bir yazılımdır. Bu yüzden biraz farklı kuracağız. İstersek direk olarak çalışan kodu 32 bit Pardus için [0] adresinden, 64 bit Pardus için [1] adresinden indirebilirsiniz. Herhangi bir dizindeyken `scilab-5.2.1/bin/scilab` yoluna gidin. Scilab programını çalıştırdığınızda kullanabilirsiniz. Ancak, sürekli bu yola gitmek istemezsiniz. Herhangi bir dizindeyken Pardus tuşuna sağ tıklayın. Menü düzenleyiciyi açın. Kaydetmek istediğiniz bölüm üzerine sağ tıklayarak, Yeni öğeyi seçin. Komut kısmındaki klasöre tıklayarak, kaydettiğiniz herhangi bir dizini seçin. Menüden de girerek kullanabilirsiniz.

Bu programları indirdiğimizde direk kul-

[0]http://www.scilab.org/download/5.2.1/scilab-5.2.1.bin.linux-i686.tar.gz
[1]http://www.scilab.org/download/5.2.1/scilab-5.2.1.bin.linux-x86_64.tar.gz
[2]http://www.scilab.org/download/5.2.1/prerequirements-scilab-5.2.1-src.tar.gz
[3]http://www.scilab.org/download/5.2.1/prerequirements-scilab-5.2.1-x86_64-src.tar.gz
[4]http://www.scilab.org/download/5.2.1/scilab-5.2.1-src.tar.gz

lanabiliyoruz. Ancak bazı gereksinimleri var. Bu gereksinimleri 32 bit Pardus için [2] adresinden, 64 bit Pardus için [3] adresinden indirrebilirsiniz.

Kaynak kodundan kurmak isterseniz de yine gereksinimler inmiş olmalı. İndirmezseniz çok uzun bir süre gerekli yazılımları indirmek zorunda olacaksınız. Kaynak kodu [4] adresinden indirebilirsiniz. Paketi istediğiniz bir dizine çıkarın. Çıkardığınız dizini açın ve F4'e basın ya da konsoldan da girebilirsiniz. Dizine girince `./configure` komutunu verin. Hata almazsınız konfigürasyon sorunsuz demektir; kurulum da sorunsuz olacaktır. `make` komutunu verin, hata almazsanız su komutuyla root olun ve `make install` komutunu verin. Hata almazsanız kurulum tamamlandı.

Artık, ödevlerimizi yapabiliriz. Konsolu kullanmaya başlayacağız. Scilab, GNU/Linux kullanıcılarına kolay gelecektir. Help komutunu vererek başlayalım. Ayrıca, bir konu ile ilgili yardım almak için help komutundan sonra konuyu yazabilirsiniz. Örneğin help bytecode komutuyla bytecode ile ilgili yardım alabilirsiniz. Mail listelerine üye de olabilirsiniz. users@lists.scilab.org listesi bütün kullanıcılar içindir. Bu listeye üye olmak için users-subscribe@lists.scilab.org adresine mail atmalısınız. Geliştiricilerin mail listesiyse dev@lists.scilab.org adresidir. Geliştirici mail listesine üye olmak için dev-subscribe@lists.scilab.org adresine mail atmalısınız.

Ekrana çıktı vererek dili öğrenmeye başlayalım. “Hello world!” çıktısı vereceğiz. Konsoldan şöyle bir komut verebilirsiniz:

```
-->Hello World !
```

Bu komutu verince hemen şöyle bir çıktı verecektir:

```
s =
Hello World !
```

Atadığımız sabitlerin çıktısını almak istersek de şöyle bir komut vermeliyiz:

```
-->disp (s)
```

Bu komutun çıktısı şu şekilde olacaktır:

```
Hello World !
```

Scilab'ta editörü de kullanabiliriz. Böylece betiklerimizi ekleyebiliriz. Editörü açmak için şöyle bir komut vermeliyiz:

```
-->editor()
```

Editörde en çok kullanılan özellikler Execute menüsündedir. Execute menüsündeki özsellikler şöyle:

Load Into Scilab: Kaydettiğimiz betiği konsolda çalıştırır. Şöyle bir betik yazalım:

```
s = “Hello world!”
disp(s)
```

Bu betiği test.sce şeklinde kaydedelim. Load into Scilab'a tıkladığımızda konsolda çalışacak.

Evaluate Selection: Betikte seçili kodları konsolda çalıştırır.

Execute File Into Scilab: Çalıştırdığı betiği yazarak çalıştırır.

Onur Tuna
onur@pardus-linux.org

Standford Üniversitesi'nde doktora yapan iki öğrenci olan Lawrence Page ve Sergey Brin, tez konusu olarak, BackRub adlı bir arama motoru projesi üzerinde çalışıyorlardı. Bu proje, sadece anahtar kelime ve metadata içerisinde değil; aynı zamanda arama ve kontrol sıklığına da dayanan bir arama şekli (PageRank) kullanıyordu.

Başta Stanford Üniversitesi'nin çatısı altında faaliyet gösteren Google, 1998 senesinde, bağımsızlığını kazandı.

1998'den bu yana, Google'ın aldığı yolu tarife, sanırım GoogleMaps uygulaması bile yetmez. :-) Arama motorundan başlayan Google, bugün, İnternet üzerinden hizmet veren pekçok uygulamayı bize sunuyor. GoogleAnalytics, GoogleMaps, GoogleEarth, GoogleChrome, GoogleDocs vs... Üstelik bunlar bize ücretsiz geliyor. Google'ın açık kaynak ve mobil dünyasına katkısı ve desteği de cabası.
Google'ın son hamlesi ise, İnternet üzerinden sunduğu tüm hizmetlerini, tek bir işletim sistemi altından kontrol edilmesini sağlayacak bir proje olan, Google ChromeOS.

ChromeOS hakkında pek çok şey yazıldı, çizildi. Kimi bir Linux türevi olacağını

söyledi. Kimisi bağımsız bir işletim sisteminden bahsetti. Hatta bir çekirdek ve birkaç modül derleyerek yaptıkları GNU/Linux dağıtımlarını, “ChromeOS Beta” şeklinde yutturmaya çalışanlar bile oldu. Hiçbir açıklama yapmayan Google, tüm bu söylenenlere belki kıs kıs güldü; belki de ilham aldı. 2009 senesinde Google, ChromeOS'un kodlarını İnternet'e vererek, tüm bu tartışmalara noktayı koydu.
ChromeOS henüz çıkmış değil. Ancak kodlarını indirip derleyerek, nasıl bir proje olacağı hakkında bir fikir sahibi olabiliriz. ChromeOS hakkında bilgilere ve kaynak kodlarına [0] resmi sayfasından ulaşabilirsiniz.

ChromeOS Derlemek

Kaynaktan derlenen her yazılım gibi, ChromeOS'un da derlenmek için bazı gereksinimlere ihtiyacı var. Bunlar:

64 Bit'ilk bir Linux dağıtımı (Ben, ChromeOs derlemesini 64 bitlik Ubuntu Karmic Koala üzerinde yaptım. Derleme için 64 bit bir ortama ihityacınız var. Derlemede kullanılan “chroot” ortamı, 64 bitlik bir ortam. X86 tabanlı sistemlerde çalışmıyor. Ubuntu kullanmamdaki amaç ise, -sitesinde öyle denmesinin yanında :)- belirtilen paket ihtiyaçlarının, Ubuntu deposundan kolaylıkla kurulabilmesi).

Paket ihtiyaçları:
ChromeOS derlemesinde gerekli paket listesi; aynen babamın pazar listesi gibi.

```
Python >= 2.4
python2.5-dev
Perl >= 5.x
gcc/g++ >= 4.2
g++-multilib >=4.2
bison >= 2.3
flex >= 2.5.34
gperf >= 3.0.4
pkg-config >= 0.20
```

[0] http://dev.chromium.org/chromium-os

```
libnss3-dev >= 3.12
libasound2-dev
libgconf2-dev
libglib2.0-dev
libgtk2.0-dev
libnspr4-0d >= 4.7.1+1.9-0ubuntu0.8.04.5
libnspr4-dev>= 4.7.1+1.9-0ubuntu0.8.04.5
freetype-dev
libcairo2-dev
libdbus-1-dev
libbz2-dev
libjpeg62-dev
libpam0g-dev
libexpat-dev
libbzip2-dev
mesa-common-dev
libgl1-mesa-dev
libglu1-mesa-dev
libxss-dev
```

Bunları tek seferde kurmak istiyorsanız vereceğiniz komut:

```
sudo apt-get install subversion pkg-config python perl g++ g++-multilib bison flex gperf libnss3-dev libgtk2.0-dev libnspr4-0d libasound2-dev libnspr4-dev msttcorefonts libgconf2-dev libcairo2-dev libdbus-1-dev libbz2-dev libjpeg62-dev libpam0g-dev libglu1-mesa-dev libxss-dev python2.5-dev
```

Ubuntu, bunları birkaç dakika içinde indirip kuruyor.

ChromeOS'u kurmanın iki, yöntemi var: Birinci yöntem, geliştirici sayfasında açıkladığı şekilde chroot derleme ortamı içinde derlemek. İkinci yöntem ise, kaynak kodunu doğrudan indirerek derleme yapmak.

İkinci yöntem, diğer derleyicilerin kullandığı ve ilk yönteme göre daha kolay görünen bir yol. Ancak ben, bir türlü ikinci yöntem ile derlemeyi başaramadım. Her adımda farklı farklı hatalar verdi ve sonunda pes ettim. O yüzden ilk yöntemi, yani geliştirici sayfasında uygulanan yöntemi kullandım.

İlk olarak ev dizininiz altında `chromiumos` adlı bir dizin oluşturun. Daha sonra git ve takip sürümlerini indirmemiz gerekiyor. Bunun için:

```
sudo apt-get install git-core subversion
```

komutunu kullanıyoruz.

Sonra, [1] adresinden, ChromeOS deposunu bilgisayarımıza indirecek ve gerekirse güncellemeleri yapabilecek olan depot-tools gerecini indirmemiz gerekiyor. İndirdiğimiz .tar.gz dosyasını, `chromiumos` dizinimizin altına açıyoruz. Ancak depot-tools klasörünün kendisini değil; içeriği chromiumos dizini altında olmalı. Sonra sırasıyla şu komutları veriyoruz:

```
./gclient config http://src.chromium.org/git/chromiumos.git
./gclient sync
```

İlk komut, kodların alınacağı adresi belirtirken; ikinci komut, bilgisayarımızdaki ChromeOS deposu ile kaynak deposunu kıyaslayarak, gücelleme yapıyor. Ancak bizde depo olmadığı için, haliyle, tüm depoyu indiriyor. Deponun indirilip kurulması, İnternet ve sistem hızınıza bağlı olarak değişmekle beraber, bende 1 saat 15 dakika civarı bir süre aldı. “gclient” ChromeOS çekirdeğini indirirken, bir yirmi dakika kadar konsol hareket etmedi.

Depo indirildikten sonra,
chromiumos dizini altındaki src/scripts klasörü altına giderek;

```
./make_chroot.sh
```

komutu ile “chroot” ortamını oluşturu-

[1] http://dev.chromium.org/developers/how-tos/install-gclient

yoruz.

“gclient”, third_party altındaki “shflags” dizinini indirmekte sorun yaşayabiliyor. Eğer, chroot ortamını oluştururken;

```
./common.sh: Line 78: /home/kullanıcı_adınız/chromiumos/src/third_party/shflags/files/src/shflags: No such file or directory
```

şeklinde bir hata ile karşılaşırsanız, yukarıdaki adımda çalıştırdığımız gclient, bu klasörü toparlayamamış demektir. [2] adresinden dizin içeriğini indirip, shflags dizininin altına yerleştirebilirsiniz.

Oluşturduğumuz chroot ortamına girmek için;

```
./enter_chroot.sh
```

komutunu veriyoruz. “`Mounting chroot environment`” diye yanıt veriyor konsolumuz. Artık ChromeOS derleme ortamı içindeyiz.

Seçeneksel olarak burada Chrome gözatıcıyı, ChromeOS içerisine yerleştirebilirsiniz. Ancak bunu yapmanız şart değil. İnşa işlemi sırasında bu otomatikman yapılacaktır.

Sonrasında, ChromeOS kurulumu için bir kullanıcı adı ve parola belirlememiz gerekecek. Kullanıcı belirlerken iki seçeneğimiz var. İlk olarak:

```
./enable_localaccount.sh USERNAME
```

komutu ile, ChromeOS başlangıcındaki giriş ekranını iptal ederek, Google hesabı olmadan giriş yapılmasını sağlayabilirsiniz. USERNAME seçeneğini herhangi bir isim ile de değiştirebilirsiniz. İkinci olaraksa:

```
rm chromeos_pam_localaccount.sh
```

komutu ile `chromeos_pam_localaccount.sh` dosyasını silerek, her kullanıcının Google hesabının kullanıcı adı ve şifresi ile giriş yapabilmesini sağlayabilirsiniz.

Şifreyi ise;

```
./set_shared_user_password.sh
```

komutu ile ayarlıyoruz. Burada belirlediğimiz şifre, ChromeOS kurulumu aşamasında sorulacak olan şifre olacak. Dosyayı çalıştırınca, konsol, sizden bir şifre belirlemenizi isteyecek. Şifreniz, `/src/scripts/shared_user_passwd.txt` dosyasına kaydedilecektir.

Sıra geldi paketleri inşaya. Burada kuracağımız iki paket takımı var. Platform paketleri ve çekirdek. Bunları ayrı ayrı;

```
./build_platform_packages.sh
./build_kernel.sh
```

komutları ile kurabileceğimiz gibi; tümünü

```
./build_all.sh
```

komutu ile de kurabiliriz. Artık sona yaklaşıyoruz. İnşa edilmiş ChromeOS' umuzun “image” dosyasını oluşturabiliriz artık. Bunun için:

```
 ./build_image.sh
```

komutunu vereceğiz. Eğer Chrome göz atıcısı ile ilgili olarak,

```
Downloading http://chrome-web/buildbot/snapshots/chromium-rel-linux-chromiumos/LATEST
--2010-03-21 16:47:37-- http://chrome-web/buildbot/snapshots/chromium-rel-linux-chromiumos/LATEST
```

[2] http://code.google.com./p/shflags/downloads/list

```
Resolving chrome-web... failed: Name or service not known.
wget: unable to resolve host address `chrome-web'
make: *** [build-stamp] Error 1
dpkg-buildpackage: failure: debian/rules build gave error exit status 2
```

gibi bir hata ile karşılaşırsanız, "src/platform/chrome/copy_chrome_zip.sh" dosyasını açarak, indirme adresini yeniden ayarlamanız gerekecek demektir. Dosyayı metin düzenleyici ile açarak,

```
BASE_FROM="http://chrome-web/buildbot/snapshots/chromium-rel-linux-chromiumos"
```

satırını bulun. Şu şekilde değiştirin:

```
BASE_FROM="http://build.chromium.org/buildbot/continuous/linux"
```

"image" dosyasının inşası bittikten sonra, image, /src/build/images klasörü altında oluşturulacaktır. build_image.sh dosyası, bu dizin altında, sıralı rakamlardan oluşan bir alt dizin oluşturacaktır. Bu alt dizin, mbr.images ve rootfs.images bilgilerini barındırır.

Sonrasında,

```
exit
```

komutu ile "chroot" ortamından çıkıyoruz. Konsol, "Unmount chroot environment" şeklinde yanıt veriyor. Şimdi sırada ChromeOS image dosyamızı, USB belleğe aktarmaya geldi.

***Not*: *Bu işlemi yapmadan önce "chroot" ortamından ayrıldığınıza emin olun. İşlemi yarıda kesmeyin ve bu işlem esnasında, bilgisayarınızı başka bir işlem için kullanmayın.*

En az 4GB'lık bir USB belleğe ihtiyacınız olacak. USB belleği bilgisayarınıza takın. Konumunu öğrenmek için,

```
sudo fdisk -l
```

komutunu kullanabilirsiniz.

İmage dosyasını USB belleğe yazdırmak için;

```
./image_to_usb.sh --from=~chromiumos/src/build/images/ALTDİZİN --to=/dev/USBBELLEK
```

şeklinde konsola komut veriyoruz. ALTDİZİN, "build_image.sh" komutunun oluşturduğu dizin olurken; USBBELLEK de USB belleğimizin adresi olack. Örnek:

```
/image_to_usb.sh --from=~chromiumos/src/build/images/999.9999.40207.042865-a1  --to=/dev/sdc
```

Burada USB belleğin temel adresi olan /dev/sdc kullanın. Bölüm belirten, /dev/sdc1 adresini değil (bende USB bellek sdc olarak bağlandı. Bu sizde değişebilir. sdb olabilir.)

Cihazdaki tüm bilgiler silinecek ve yaklaşık 15 dakikalık bir süre sonra USB bellek hazır olacaktır.

ChroemOS Kurulumu

Bilgisayarı yeniden açarak, ChroemOS image dosyamızı yazdığımız USB bellekten sistemi başlatıyoruz. Giriş yaptıktan sonra, Ctrl+Alt+T tuşları ile konsolu açarak, şu komutu çalıştırın:

```
/usr/sbin/chromeos install
```

****Not:* *Tüm diskiniz silinecektir. ChromeOS size; "diski böleyim, yok başına mı kurayım, sonuna mı kurayım, ön yükleyici yükleyeyim" şeklinde seçe-*

nekler sunmuyor. Doğrudan tüm diski kullanıyor. Diz üstü bilgisayarlarda genelde, sistem kurtarma şeklinde bir bölüm olur. Buraya sürücü ve işletim sisteminin bir image dosyası konularak, tek tuşla sistem geri yükleme gerçekleştirilir. Eğer, ChromeOS yüklemeye çalıştığınız bir diz üstü olacaksa, bu bölümler kaybolacaktır. Dikkat derim.

Kurulum tamamlanınca, USB belleği çıkartarak, sistemi yeniden başlatın.
Tüm bu işlemler, hatalar için uğraştım zamanları düşersek, yaklaşık 4 saate yakın sürdü.

Veee.. ChromeOS (ChromiumOS) Karşınızda

İlk izlenim: 8-9 saniye gibi rekor bir sürede açılıyor. :-)

Giriş ekranından Google kullanıcı adı ve şifreniz ile giriş yapıyorsunuz. Ne var ki, Türkçe klavye desteği mevcut değil (varsa da ben bulamadım). O yüzden İngilizce Q klavye kullanıyoruz.

Google'ın Chrome web tarayıcısını, açık kaynak camiasında olup da duymayan, hatta görmeyen kalmamıştır herhalde. İşte ChromeOS, Google Chrome tarayıcısı temelli çalışan bir tür İnternet terminal sistemi.

Bence ChromeOS'un,şimdiki haliyle, bir işletim sisteminden ziyade, bir terminal sistemi olduğunu söyleyebiliriz. Ancak, henüz tamamlanmış değil.

Webkit motoru kullanan Chrome göz atıcısını çalıştıracak şekilde derlenmiş ve gereksiz tüm modül ve fonksiyonlar çıkartılmış bir çekirdekten ibaret. Aslında oldukça orjinal bir fikir.

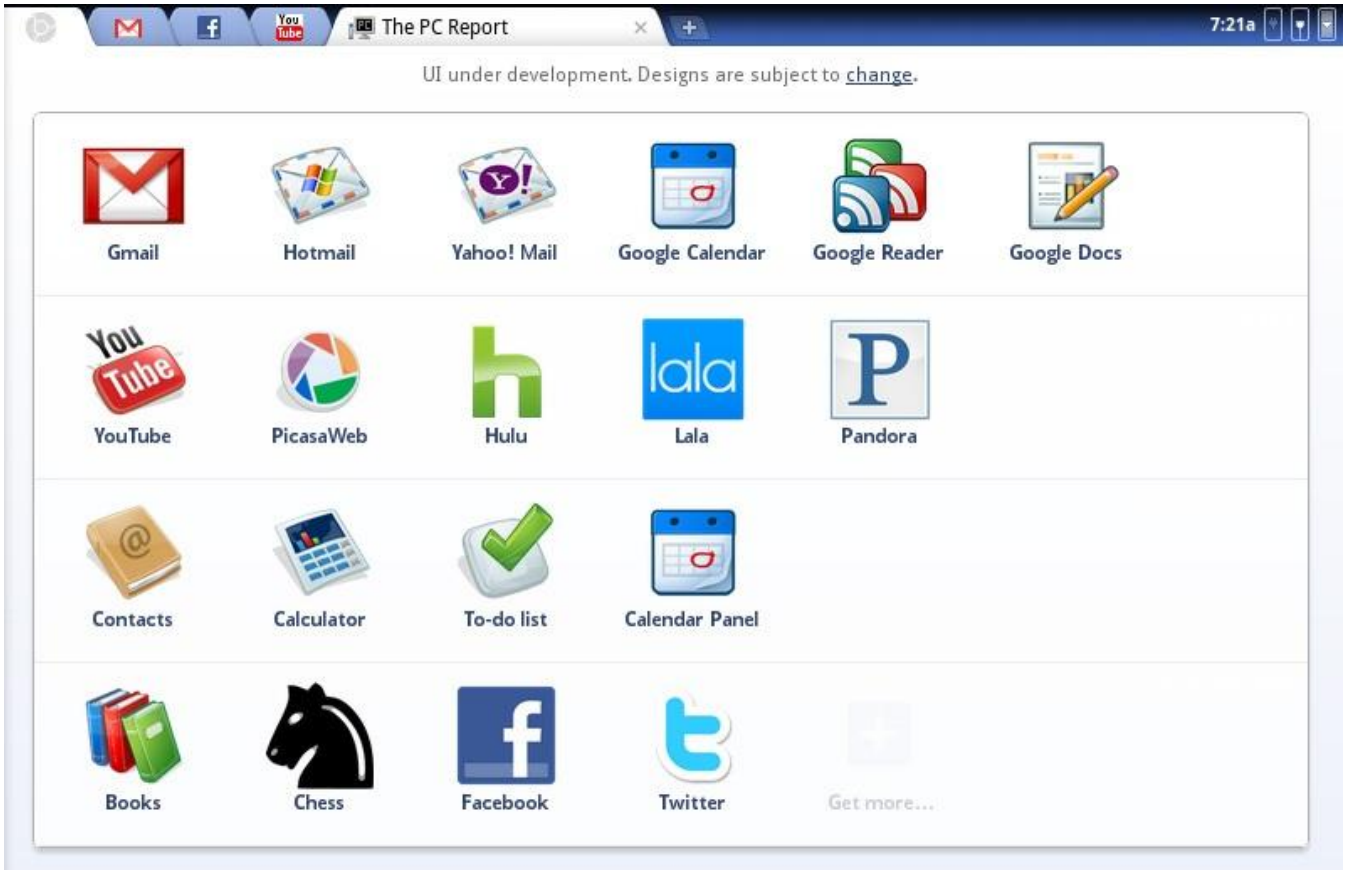
Giriş yaptığımızda, tüm Google hizmetlerine de giriş yapmış oluyoruz. Bir başka işletim sisteminde değil de; sanki tam ekran Google Chrome kullanıyormuş gibi hissediyorsunuz.

Google'ın tabiri ile, "*bilgisayarını sadece İnternet için kullanan kullanıcılara hitap edecek bir işletim sistemi*". ChromeOS

ile İnternet üzerinden müzik dinleyebilir, film/canlı yayın akışı izleyebilir, e-posta gönderebilirsiniz. Yani kısaca, ağ üzerinde yapacağınız herşeyi yapabilirsiniz. ChromeOS'un bir bölgesel depolama desteği yok. Yani sabit diskinize erişiminiz yok. Bir dosya tarayıcısı yok. Bunlara gerek de yok. Mevcut İnternet uygulamalarının hepsi ;ChromeOS'da çalıştığı gibi, ChromeOS uygulamlarının hepsi de birer İnternet uygulaması. Sadece SSD desteği var ki; o da bilgisayarınıza kendisini kurması için mecburi olan bir destek.

Ağda olan, ağda kalır. :-) ChromeOS üzerinde çalışan tüm uygulamalar, ağ üzerinde çalışan İnternet uygulamalarıdır. Yani bilgisayarınıza herhangi birşey kurmaz. Tüm bilgiler "Cloud"(Bulut) üzerinde toplanır.

ChromeOS'un, hızı da çok iyi. Siteler arası geçiş, sanki televizyon kanalları arası geçiş gibi. Hiçbir eklenti indirip kurmadan; Facebook, Twitter gibi sosyal uygulamalarda ve bankacılık işlemlerinde hiçbir sorunla karşılaşmadım. Sadece TRT'nin sitesinde canlı yayınları izleyemedim. Onun dışında hiçbir sorunla karşılaşmadım.
ChromeOS, asgari düzeyde donanım

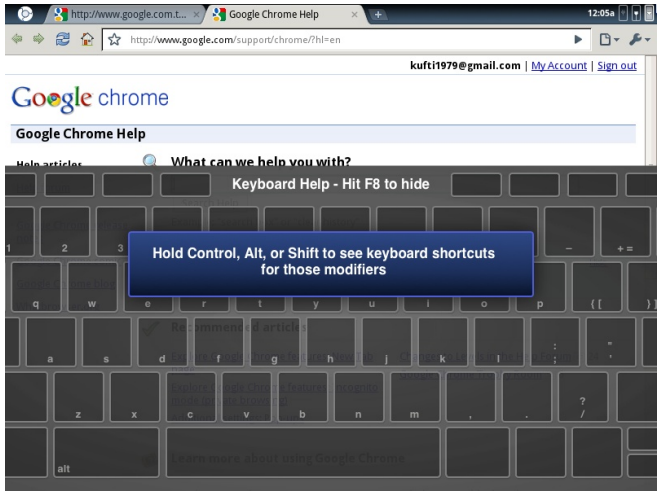


koşturacak seviyede. Çünkü pekçok modül yok. Ancak -doğası gereği- ağ donanım desteği tam. Öyle de olmak zorunda zaten. İnternet olmadan, ChromeOS bir hiç. Kurduğum diz üstü bilgisayar olan Acer Aspire AS7736'nın kablolu, kablosuz tüm ağ aygıtlarını hiçbir ayar yapmadan tanıdı. Sadece ağ yöneticisinden (sağ üstteki üç düğmeden ortada olanı) "EnableWi-Fi" seçeneğini seçip, modemime bağlanarak, WPA2 şifresini girerek, kablosuz bağlantıyı sağladım.

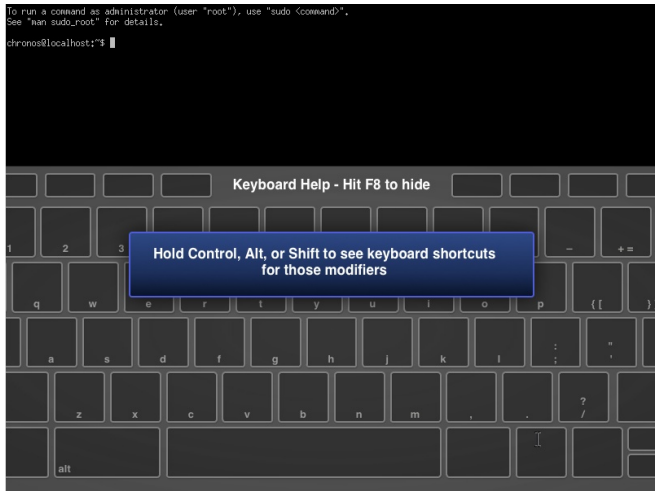
ChromeOS, kullanıcıların, Chrome göz atıcı ayarlarının yanında, sadece birkaç basit ayar yapmalarına müsaade ediyor. Yine de bazı kısayollar mevcut. Bunlar:

F8 – Klavye kısayollarını gösteren bir yardım ekranı açılır. Burada Ctrl, Alt ve Shift tuşlarına tek tek veya beraber ba-

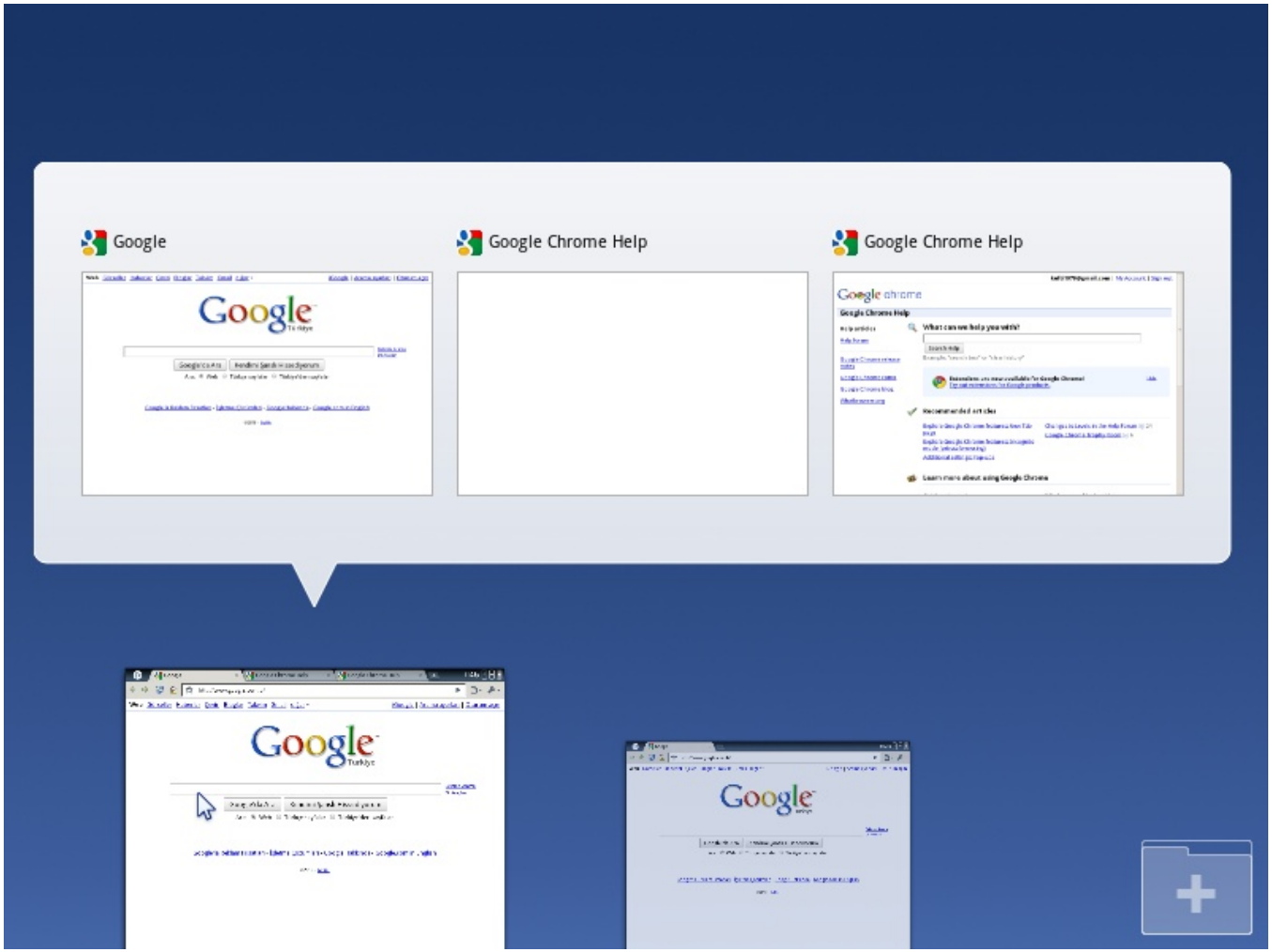
sarak, diğer tuş kombinasyonlarıyla kullanıldığında yapacakları görevleri gösteren bir ekran açılır. Yine F8 ile, bu ekranı kapatabilirsiniz.



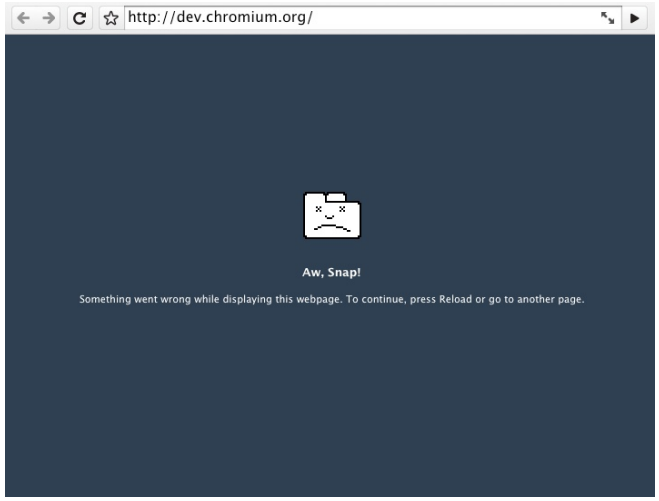
Ctrl+Alt+t tuşları ise ChromeOS konsolunu açar. Burada çekirdek Linux komutlarını kullanabilirsiniz. Konsoldan masaüstüne dönmek içinse "exit" komutunu kullanıyoruz.



ChromeOS'ta da çoklu masaüstü özelliği mevcut. Bunun için F12 tuşuna basıyoruz. Mevcut masaüstü, ekranın altına küçülüyor. Sağ alttaki + işareti ile yeni masaüstü oluşturuyoruz ve fare tıklamasıyla masaüstleri arasında geçiş yapıyoruz.



Sonsöz (mü acaba?)



Google'ın İnternet üzerinden yapmış olduğu ChromeOS, açık kaynak ve kullanım ile de ilgili pekçok soru ile birlikte geliyor.

Teknik olarak, bilgilerinizin hepsi İnternet'te Cloud üzerinde olacak. Pek güvenlik ne olacak? Bu bilgilerin gizliliği nasıl sağlanacak? USB belleklere erişebiliyoruz; peki kendi verilerimizi, kendi sabit disklerimizde depolayamayacak mıyız? Bilgisayarlar kişisel bilgisayar olmaktan çıkıp, ChromeOS kullanan birer İnternet terminaline mi dönüşecekler? Ya EyeOs, alternatif olabilir mi? ChromeOS açık kaynak kodlu bir uygulamadır ve açık kaynak uygulamaları çalıştırıyor. Açık kaynak felsefesi ve kullanım ile ilgili olarak ise endişeler mevcut. Google, "*İnternet'in Microsoft'u*" olarak nitelendiriliyor. İkinci bir MS, tam da ihtiyacımız olan şeydi(!) Google'ın yaptığı şeyler ortada. Açık kaynak camiasına kazandırdıkları ve desteği tartışılmaz. Ama kişisel bilgilerinizi Google'a emanet eder misiniz?

Bu soruların cevapları için sayfalarca yazı yazılabilir. Ancak, ChromeOS henüz tamamlanmış değil. Derleme sırasında, pekçok dosya güncellenerek yeri ve adı değiştirildiği için, birçok hatayla karşılaştım. Ya yeni konumu .sh dosyasına yazarak, ya da dosya içeriklerini elle indirerek sorunu halledebildim. ChromeOS tayfası, saatler içinde güncellemeler yaparak, son hızla ChromeOS'u geliştirmeye devam ediyor. Google, ChromeOS çıkışını 2010 senesinin ikinci yarısı olarak açıklıyor. Microsoft ise; Bind arama motoru, Cloud Computing ve WebApps ile Google'a yanıt vermeye çalışıyor. Google, İnternet üzerinde, MS'nin tekeline son verebilir mi? Peki, kendi tekel olur mu?"Dinsizin hakkından imansız gelir" diyebilir miyiz? Ya sonra, imansızın hakkından kim gelir peki?

Bir uygulama çıkaracağı zaman çok öncesinden davullar çalmaya başlayan Microsoft'un aksine Google, işlerini sessiz sedasız yapıyor. Yukarıdaki sorunların cevabını verecek tek şey, şu an için, zaman gibi görünüyor.

Hamit Giray Nart
hamit@pardus-linux.org

Eğitim, Bilişim ve Piyasa Gerçeğinde Yazılım Tekeli

Giriş

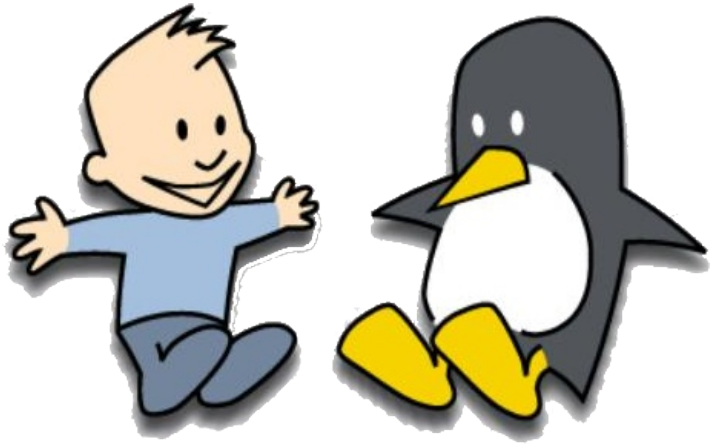
Eğitim kuşku yok ki, bir toplumun nerede olacağını gösteren en önemli etkendir; diğer çevre faktörleri ve sermaye değerleri ise sonra gelir. Toplumların eğitime verdiği önem bu yüzdendir ki son derece önemlidir.

Ülkemizde ise iki farklı düşünce yapısına sahip kuşakların yetişmesinin önüne E-ğitim ve Öğretimin Birleştirilmesi Yasası ile geçilmiş ve eğitim sistemi laik, bilimsel doğrultuda yapılmaya başlanmıştır.

Çocuklar artık bizim kuşağımız diyebileceğimiz 80'li yılların çocuklarndan farklılar. Artık tüm bilgiler eğitim ve öğretimin örgün olarak işlendiği okullarda ve günlük yaşamlarında hazır olarak önlerine sunulmakta. Bu sunuş çoğu kez de etkileşimli olduğu gibi başlarında rehber biri olmaksızın çocukların genç dimağlarının bilginin kendilerine göre ölçüsüne bakılmaksınız maruz kalmakta. Seks, cinsellik, uyuşturucu, ölüm, savaş, acı ve diğer binlerce zamanı gelince öğrenilmesi gereken bilgiler çocuklarca karşı karşıya kalındığında anında özümsenmekte.

Bilinen bir gerçek insanın yaşamına çocukken edindiği bilgi ve yönlendirmelerin son derece etkili olduğudur. Bugün tüketim dünyası ise bu durumu çok iyi bildiğinden tüketim çılgınlığı ise daha alt yaşlara doğru çekilmekte. Öyleki ailecek gidilen bir sinemada film öncesi maruz kalınan reklamlarda alalade bir şekilde kandom reklamları, cinsellik unsuru yüksek parfüm reklamları yayınlanmakta ve bu doğal karşılandığı gibi etkisi çocuklardan sakındarılamamakta. Burada bilgi hedef kitleden ziyade algısı doğru olmayan kitleye de; çocuklara; ulaşmakta ve yanlış etkileşimlere yol açmaktadır.

Pazar payının büyütülmesi ve daha alt yaş seviyelerinde tüketimin arttırılması için dev tekeller ve üreticiler hem bu kontrolsüz bilgi salınımını yapıp pazarlama çalışmalarını ve reklamlarını bu yönde yaparken diğer yandan sosyal sorumluluk çalışmaları altında gençleri ve çocukları bazı zararlı maddelerden, davranışlardan korumak için eylemlerde bulunurlar ki, bu bilinen bir şeydir.



Ancak yapılan bu çalışmalar verilen mesajın adresine teslim edildikten sonra değiştirilmesinde etkili olmadıkları gibi, alınan mesajın doğrulunun sınanması ve yanlışlanması uzun zaman almaktadır. Bu süre içinde ise firma amacına ulaşmaktadır: satış ve paralelinde kar çoklaması!

Yazılanların bilişim üzerindeki etkisine gelince

Bugün ilk öğretimden üniversitelere, halk eğitim merkezlerinden kişisel beceri kullanımına dek yapılan bilişim eğitimlerinde genel olarak piyasa hakimi firmaların kaynak kodu kapalı, erişilmez, yasak, yazılım ve bilişim özgürlüğüne olduğu gibi bilginin de paylaşımına engel olan yazılım ve işletim sistemleriyle yapılmaktadır.

Piyasayı denetim altına alan bu firmalar (ki eşyaya adını koymak kapitalist sistemin doğası gereğidir; bahsi geçen firmalar ise Microsoft, Apple...) eğitimin kendi pazar payları için ne kadar önemli olduğunu bildiklerinden bağışlarla, donanım destekleriyle, öğrencilere ve eğitmenlere yönelik indirimli satış kampanyalarıyla, sosyal yardımlarla körpe beyinlerde kendi ürünlerinin mutlak doğruluğunu yerleştirdiler: Bilgisayar eşittir Windows.

Özellikle de çocukların tüketim konusunda ailelerini yönlendirmeleri ve istediklerini dayatmaları, hele ki bu çağda olanların kolaya kaçmalarındandır ki, işaret ettikleri bilgisayarın alınmasında beraberinde ön yüklü olarak gelen kaynak kodu kapalı işletim sistemi kullanılacaktır. Bu ise eşitliğin; genç beyin sorgulama yolları kapanmadan seçeneklerin mevcut oluşunu öğrenene dek; bozulması ise çok zordur.

Okullarda, kurs yerlerinde, halk eğitim merkezlerinde ücreti ne olursa olsun verilen bilgisayar eğitimi değil sadece bir işletim sisteminin öğretilmesi, dahası ezberletilmesidir. Acı gerçek ise bunun yüksek ögretimde de sürmesidir.

Edinilen bir alışkanlığın yerle bir edilmesi zamanla kısmen mümkün olur ki, boş yere söylenmiş bir söz değildir: alışmış kudurmuştan beterdir. Küçük yaşta edinilen alışkanlığın değiştirilmesi hatta farklı seçeneklerin olduğu bilgisinin eğitim süreci içinde öğrenciye aktarılması son derece önemlidir. Şimdi eğitim sisteminde yapılan farklılıkların dışlanması ve seçeneksiz bir eğitimdir; dahası Microsoft, Apple ve diğer kaynak kodu kapalı yazılımların dayatılmasıdır. Bu süreç içinde Özgür Yazılıma ise yer verilmediği bariz bir şekilde görülecektir. Kaldı ki, doğru olan tüm seçeneklerin belirtilmesidir. Doğru ya da yanlış olanı anlamak zaten aklın ve bilimin ışığında doğru olan bilgiyi yanlışlayarak doğruya ulaşan öğrenci olacaktır.

Yazılanların piyasa üzerindeki etkisine gelince

Diğer bir gerçek ise piyasada yer alan bilgisayar ürün ve bileşenlerini satan teknoloji mağazalarının durumudur. Bu mağazalar tam anlamı ile tekelci firmaların pazarlama çalışmalarına teslim olmuşlardır; üstelik sattıkları ürünlerde hiçbir şekelde değişiklik ve yönlendirme yapma hakları dahi bulunmamaktadır. Kaldı ki, bu mağazaların satış elemanlarına şirket için eğitim ve çeşitli ürün bilgileri, kurslar verdikleri bilinen bir ger-çek; verilen eğitim ise yine Microsoft, Apple odaklı.

Ülkemizde eğitim ve teknoloji pazarında Özgür Yazılım tamamı ile dışlanmış durumda. Hazır satın alınan bilgisayarlar, kurulan sistemlerde sadece satın alınması yapılmış Microsoft ürünleri kurulu olarak hizmete sunulmakta. Bu ise kursiyerlerin, eğitim alanların ve toplumun hak ve özgürlüklerinin kısıtlanması demek. Örneğin piyasaya hakim olan Microsoft Windows işletim sistemi ile yazılım ve donanım üreticileri ürünlerini satabilmek için tam uyumlu çalışmakta; buna da mecburlar, çünkü piyasada Özgür Yazılım kendilerine yeterince değil hiç kar getirmiyor. Dolayısıyla üretim.

Verimlilik, zaman ve plan yönetici yazılımların çoğu Microsoft Windows işletim sistimine uygun olmak zorunda; haliyle firmaların aradığı çalışan personelin de bu sistemi kullanmayı bilmesi beklentiler arasında. İş bulup çalışmak isteyen insanlar da piyasanın dayatmasına boyun eğmek zorunda kalıyorlar.

Bugün ticari otomasyon, muhasebe, basit sözlük ve metin, hesap araçları, personel, stok, hasta, araç kayıt uygulamaları ülkemizde yazıldığı gibi bu yazılımları üreten beyinler farklı bir işletim sistemi mevcut olduğunu bilseler, Özgür Yazılımdan haberdar olsalar neden kodlama ve üretim safhasını Özgür Yazılıma, GNU/Linux dağıtımlarına göre planlamasın, üretmesin?

Aydın Bez
aydinbez@pardus-linux.org
http://www.ozguryazilimsendikasi.org

[1] GNU Genel Kamu Lisansı sürüm 3'ün gayri resmi Türkçe çevirisi için Pardus Viki ekibine teşekkürler: http://tr.pardus-wiki.org/GNU_GPL_(Genel_Kamu_Lisansı)_Sürüm_3_Gayrıresmî_Çevirisi

[2] Creative Commons (by-nc-sa): Bu lisansa sahip eseri kopyalayabilirsiniz, üzerinde değişiklik yapıp yenisini üretebilirsiniz. Sağlanması gereken üç şart var. İlki, eserin tüm kopyalarında da eserin ilk sahibinin belirtilmesi. İkincisi, eserin hiçbir kopyası ya da eserden üretilmiş yeni eserlerin hiçbirisinin ticari ortamda kullanılmaması. Üçüncüsü, eserin tüm kopyalarında ya da eserden üretilmiş yeni eserlerde de aynı lisansın kullanılmaya devam edilmesi. http://tr.wikipedia.org/wiki/Creative_Commons

Kitap Tanıtımı: Özgür Yazılım, Özgür Toplum

Kemal Karataş

GNU/Linux camiası çok zor zamanlar atlattı. Son zamanlarda, varolan tartışmalarına bakıldığında, Richard M. Stallman ve Özgür yazılım Vakfı'nın neyi, neden yaptığını araştırmak ve ona göre fikirleri belirlemek gerekli. Özgür yazılım, Özgür toplum kitabı, RMS'in seçme yazılarını dilimize kazandırıp, bize sunan değerli bir eser olmuş cidden. Farklı zamanlarda yapılan konuşmalar ve söyleşilerin düzenli biçimde toplanması zor bir iştir, ama TMMOB bu zor işi hakkını vererek yapmış.

GNU/Linux dünyasını tanımak ve makul fikirlere sahip olmak isteyenlerin mutlaka okuması gereken bir eser bu.. Linux yerine neden GNU/Linux dendiği, GNU felsefesini araştırma imkanını kaçırmamanız lazım. Kitapı okumadan evvel, ilginizi çekebilecek başlıklar;

***GNU Projesi ve Özgür Yazılım**
***Telif Hakkı, Copyleft ve Patentler**
***Özgürlük, Toplum ve Yazılım**

Kitap, sadece bir biçimde düzenlenmiş. Beyaz kapağın üzerine, GNU'nun simgesi yerleştirilmiş. Esprili bir kapak dizaynı olmuş bence. İngilizce aslından çevirenler: ***Serkan Çapkan, İzlem Gözükeleş, Tahir Emre Kalaycı, Çiğdem Özşar, Birkan Sarıfakoğlu***. İngilizce 1. Baskı – 2002, Türkçe 1. Baskı Kasım 2009 yılında yayımlanmış durumda.

Kitabın boyutu en-boy oranında geniş düzenlenmiş. Özellikle, kalın bir kitap olmaması için, ince bir kitap görünümü verilmiş. 273 sayfalık kitabı, bir solukta okuyacaksınız. Bazı kitaplar, başlangıçta ilgi çeker; ama teknik detaylarla okuyucuyu sıkar. Bu kitap ise, düzenlemesi ve yazıların sıralanması ile okuyucuyu sıkmıyor. Hem bilgilendiren hem de teknik-GNU felsefesi hakkında bilgilendiren bir kitabı bulmak önemlidir.

Bu bilgilendirici ve başucu kitabı yapılacak kitap sizi bekliyor. Daha böyle kaliteli eserlerin tanıtımında görüşmek ümidiyle.

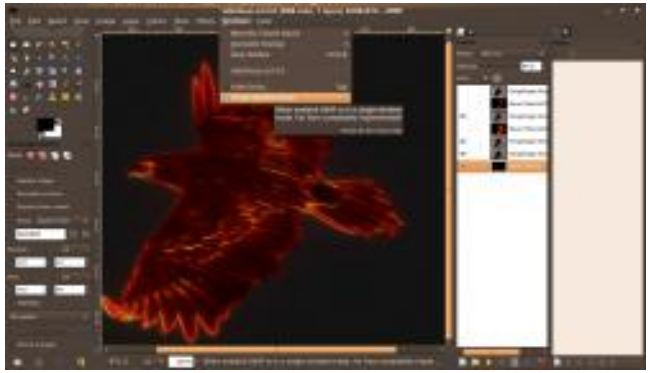
Kemal Karataş
kemal@pardus-linux.org

T 2.8'de Yeni Özellikler

1: Yeni “Tek Pencere Modu” yeni kullanım özellikleri sunuyor.

Tek Pencere Modu, İsteğe Bağlı bir özellik. Yani çalışırken, çoklu pencere modunu da kullanabilirsiniz.

Araç kutusu, diyalog kutusu gibi pencereler artık tek bir ana pencerede görünecek. Artık üst üste binen pencereler ve buna benzer sıkıntı yaratacak durumlar son bulacak.



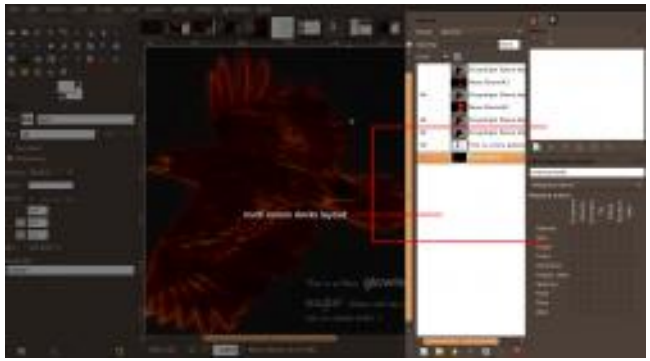
2: Yeni açık pencere gezinti seçeneği

Artık resimler arasında gezinti yapmak için görev çubuğunu kullanmak zorunda kalmayacaksınız. Açılmış bir resmin üstündeki gezinti bölümü ile tüm açık resimler arasında kolayca gezinti yapabileceksiniz.

3: Gelişmiş dock özellikleri!

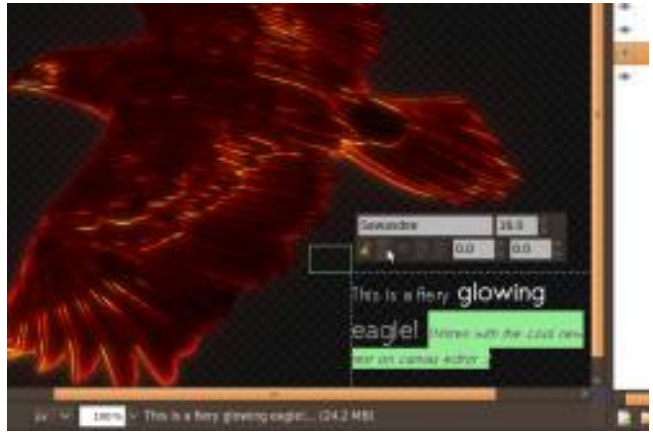
Diyaloglarınızın kenetlenmesi önemli derecede geliştirilmiş ve GIMP arayüzünü yönetmek için daha dinamik bir yol sunar olmuş.

Çok sütunlu docklar bir diğer yenilik. Sürükle bırak yöntemiyle artık arayüz kişiselleştirmeleri oldukça kolaylaştırılmış durumda.



4: Kanvas metinlerini doğrudan düzenleyebilme!

Artık kanvas metinlerini direkt olarak düzenlemek mümkün. Yani ayrı bir metin penceresinin açılmasına gerek yok. Böylece belli bir metin kutusunun içindeki yazıyı değiştirebilir, kalın veya eğik yazabilir, boyutuyla oynayabilirsiniz.



5: Değiştirilmiş dosya ihracı

Değiştirilmiş dosya ihracı önemli derecede daha hızlı çalışma sunuyor.

“Kaydet” ve “Farklı Kaydet” fonksiyonları artık sadece GIMP’in kendi dosya türü olan XCF türünde kaydetmek için kullanılıyor. Png ya da Jpg biçiminde kaydetme seçeneği artık burada değil.

Eğer açık bir dosyanın png sürümünü

yapmak isterseniz, onu yeni bir “İHRAÇ” girdisiyle ihraç etmek zorundasınız. Bu GIMP’in önceden çalışabildiği tüm dosya türleri için geçerli.

Yine çalışma akışı hızlanmış, çünkü xcf olmayan bir resmi açtıktan sonra, basit bir “ÜZERİNE YAZ” girdisi mevcut. Bu sayede resmi üzerine aynı özellikleri (örneğin JPEG kalitesi 98) kullanarak yazabilirsiniz. XCF dosyası ihraç edildikten sonra, mesela PNG dosyası halinde, siz aynı dosya adı ve özellikleri kullanarak tekrar ihraç etme şansına sahip olacaksınız.

6: Katman Setleri / Katman Grupları

Artık katmak grup/setleri oluşturabilir ve katmanları bu bir resimde çalışmakta olduğunuz katmanların çokluğundan sıkılmamak için setlere taşıyabilirsiniz.

Özellikle sitelerini GIMP ile tasarlayan site geliştiricileri için hoş bir özellik. Setleri saklayabilir, gösterebilir, kopyalayabilir ve tüm katmanları aynı anda taşıyabilirsiniz.

7: Büyük ölçüde geliştirilmiş kaynak yönetimi

GIMP’in fırçaları, gradyanları ve modelleri artık seçilebiliyor ve kendilerini açıklayan sözcüklerle etiketlenebiliyor. Etiketledikten sonra, bu etiketlere göre filtreleme yapabilme olanağı mevcut oluyor.

Bu çok sayıda fırça seti yüklenmiş olanlar için oldukça kullanışlı bir özellik.

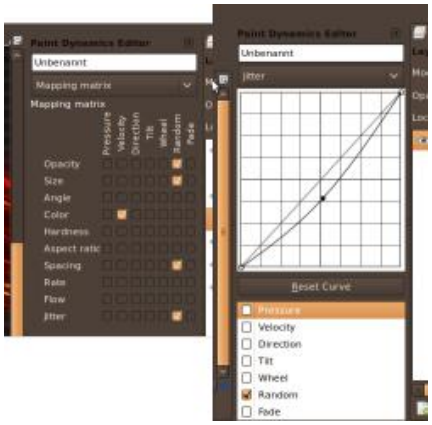
8: Döndürülebilir fırçalar

Fırçalar artık döndürülebiliyor! Her türlü boyama işleri için harika!

9: Gelişmiş boyama dinamikleri

Boyama dinamikleri, boyama araçları özelliklerinden kaldırıldı. Bunlar artık ke-

netlenebilir diyaloglarda mevcut.

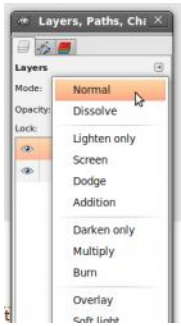


10: Yeni “Pikselleri Kilitle” özelliği

Yeni “Pikselleri Kilitle” özelliği, katmanlar diyalogunda seçildiğinde hatalı katmanlardaki boyamalardan koruyor.

11: Resim boyutlarında hesaplamalar mümkün

Resim boyutlandırması yapılırken, artık matematik işlemleri kullanılabiliyor. Örneğin (200+20+20)*2/3 girip 173 sonucunu elde edebilirsiniz.

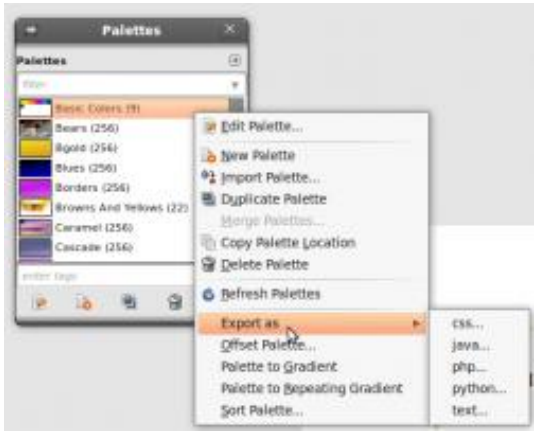


12: Katman modları sıralanıyor

Katman modları daha karanlık/ daha açık resim gibi belli bir kategoride sıralanıyor.

13: İmaj Paleti: CSS, PHP, Java, Python ya da düz metine ihraç etme

Renk değerlerini artık birçok dosya türüne ihraç edebilirsiniz. Eğer PHP, Java ya da Python’u seçerseniz, renkler birer değişken olarak tanımlanacak. CSS için ise, isimsiz sınıflar oluşturulacak ve düz metin dosyaları sadece Hex kod listesi içerecek.



14: Araç kutusu hizalamaları ve özellikleri

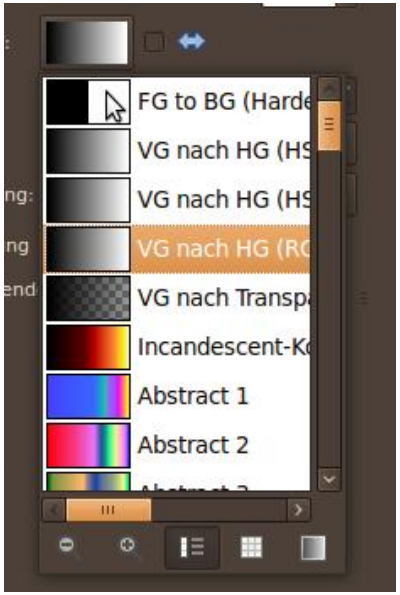
Araçlar penceresinin düzenlenmesi artık Düzenle / Seçenekler yolu ile mümkün. Araçlar önem sırasına göre sıralanabilir ya da tamamen kaldırılabilir.

15: Yeni / Değiştirilmiş Klavye Kısayolları

Yeni: Alt(+Shift)+Tıklama = Katmandan alfaya
Yeni: Ctrl+Shift+E = Dosya / İhraç Et
Yeni: Ctrl+E: hızlı ihraç et
Değiştirilmiş: Ctrl+Shift+R = Kanvas boyutunu pencere boyutuna göre ayarla
Değiştirilmiş: Ctrl+R = Pencere boyutunu kanvas boyutuna göre ayarla

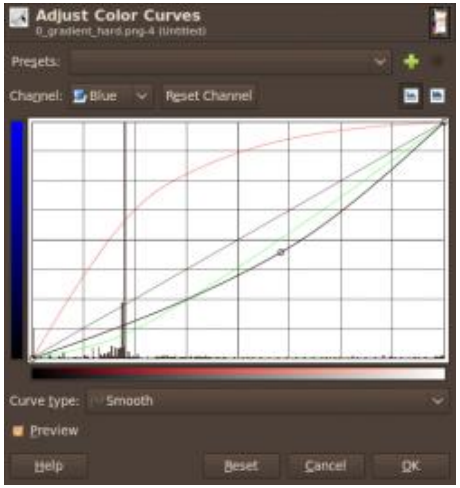
16: Yeni basit gradyan

Arka alan resmi ile ön alan resmini birbiri içine geçmeden, keskin bir biçimde ayıran basit bir gradyan eklendi.



17: Renk Eğrileri

Renk eğrileri RGB (kırmızı, yeşil, mavi) kanallarını daha iyi ince ayar için arka planda gösteriyor.



18: Bazı küçük değişiklikler

- Kanvas kenarları artık “manyetik”
- JPEG2000 desteği
- 16 bit RAW ithal etme desteği

19: GIMP ve GEGL

GEGL tamamen GIMP’e entegre edilmiş durumda. Yani resmin son hali artık GEGL’in işi.

20: GIMP 2.8 Lesser GPL v3 ile lisanslanacak.

Not: Bu yazı aşağıda adresi ve lisansı belirtilen aslından Türkçeye tercüme edilerek yayınlanmıştır. Yazı, GIMP 2.7.1 kullanılarak hazırlanmış olduğundan, GIMP 2.8 için kesin sonuçları içermemektedir.

Kaynak : http://www.gimpusers.com/tutorials/gimp-2-8-features-preview-april-2010.html

Türkiye'nin İnternet'i Delikanlı Yaşında!

12 nisanda Türkiye İnterneti 18. yaşına başlıyor. Bilişim Sivil Toplum Kuruluşları bunu tüm ülkede İnternet Haftası olarak kutluyor. İnternet Haftası etkinlikleri 5-18 Nisanda tüm ülkeyi saran bir İnternet Şenliğine, Bilgi Toplumu, e-dönüşüm, e-türkiye ve e-devlet kavramlarının geniş kitlelerle tanıştırıldığı bir İnternet ve Bilişim Fırtınasına döndürmeyi hedefliyor.

Özgür Yazılım ve Linux Günleri Düzenlendi

Linux Kullanıcıları Derneği ve İstanbul Bilgi Üniversitesi tarafından, 2-3 Nisan tarihlerinde İstanbul Bilgi Üniversitesi Dolapdere Kampüsünde Özgür Yazılım ve Linux Günleri etkinliği gerçekleştirildi.

Etkinlik takviminin bazı üniversite sınavları ile çakışmasından dolayı bazı hoşnutsuzlukların olmasına rağmen, etkinliğin özellikle 3 Nisan Cumartesi günü olan kısmının büyük ilgi gördüğü gözlendi.

Etkinlikte çekilen fotoğraflar http://galeri.linux.org.tr/ adresinde...

Pardus @ Chemnitzer Linux Tage 2010

13-14 Mart 2010 tarihlerinde Almanya'nın Chemnitzer Teknik Üniversitesi'nde Chemnitzer Linux Tage yani Chemnitzer Linux Günleri kapsamında Pardus tanıtıldı.

Basın bildirisine göre yaklaşık 2600 ziyaretçinin katıldığı Chemnitzer Linux Günlerinde, yaklaşık 80 kişi ile Pardus standında aktif konuşmalar gerçekleştirildi.

Etkinlik fotoğraflarını http://galerie.pardususer.de/thumbnails.php?album=27 adresinden görebilirsiniz.

Nisan Ayı Linux ve Özgür Yazılım Etkinlikleri

13 Nisan, Yakın Doğu Üniversitesi'nde I. Kıbrıs Uluslararası Özgür Yazılım Konferansı düzenlenecek. Konuşmacı olarak Mustafa Akgül, Ufuk Çağlayan, Mustafa Karakaplan, Necdet Yücel, Onur Tolga Şehitoğlu, Devrim Seral, Ali Erdinç Köroğlu, Sami Arbak ve Ulrich Weigand katılacak.

14 Nisan, Sakarya Üniversitesi'nde İnternet Haftası etkinliği kapsamında Linux Semineri verilecek. Konuşmacı olarak Enver Altın katılacak.

15 Nisan, ODTÜ'de EMO Genç ve İvme Genç tarafından "Pencereleri Kapatın" Özgür Yazılım Etkinliği düzenlenecek. Konuşmacı olarak Doruk Fişek, Burak Oğuz, İzlem Gözükeleş, İlker Kalaycı katılacak.

17-18 Nisan, İstanbul Kültür Üniversitesi'nde Hosting Zirvesi kapsamında Necdet Yücel tarafından IPv6 ve 64-bit Pardus üzerine sunum yapılacak.

Bu Sayıda Emeği Geçenler

Armağan Can
Aydın Bez
Erdem Artan
Gürhan Şüküroğlu
Hamit Giray Nart
Kemal Karataş
Mehmet Gültaş
Melike İlteralp
Muslu Yüksektepe
Onur Tuna
Uğur Çaylık

Pardus-eDergi,
Pardus Kullanıcıları Derneği
hizmetlerinden olan Pardus-Linux.Org topluluğu
tarafından hazırlanmaktadır ve
Pardus-eDergi.Org adresi
üzerinden yayımlanmaktadır.

Pardus-eDergi'nin tasarımında,
hazır alınan resimler hariç tutulursa,
özgür yazılımı destekleyen
bir topluluk olmanın verdiği sorumluluk ile
sadece özgür yazılımlar kullanılmıştır.

Pardus-eDergi'ye ulaşmak için
Pardus-eDergi.Org sitesinin iletişim formunu,
Pardus-Linux.Org forumlarını,
Freenode üzerindeki #pardus-destek ve
#parduslinuxorg kanallarını,
dergi@pardus-linux.org elektronik posta adresini
kullanabilirsiniz.

Pardus-eDergi'ye katkıda bulunmak için
Pardus-Linux.Org forumlarını, katkıda bulunarak neler
kazanabileceğinizi görmek için ise
Pardus-eDergi.Org adresini ziyaret edebilirsiniz.