

Telaş yok

MOBİL GELİŞTİRİÇİNİN GALAKSİ REHBERİ



Yayınlayan:



Tüm Mobil Platformlar için Hizmetler ve Araçlar Sunar

Enough Software GmbH + Co. KG
Stavendamm 22
28195 Bremen
Almanya
www.enough.de

Geribildirimlerinizi, sorularınızı veya sponsorluk isteklerinizi şuraya gönderebilirsiniz:
mdgg@enough.de
Bizi Twitter'dan takip edin: [@enoughsoftware](https://twitter.com/enoughsoftware)

 **TURKCELL** | Geleceği Yazanlar

**14. Baskı Şubat 2014'te,
Türkçesi Ekim 2014'te yayınlanmıştır**

Bu kitap Creative Commons
Some Rights Reserved License ile lisanslanmıştır.

Türkçe'ye tercüme edilmesine önyak olanlar:
Dr. Görkem Çetin, Ali Işingör, Caner Güral, Cansu Uludağ

Sanat Yönetimi ve Tasarım:

Andrej Balaz
(Enough Software)

Editörler:

Richard Bloor

Marco Tabor
(Enough Software)

Mobil Geliştiricinin Rehberi

İçindekiler

I	Giriş
1	Mobil Galaksisi: Giriş
13	Mobil için Kavramsal Tasarım
25	Android
43	BlackBerry Java Uygulamaları
51	BlackBerry 10
65	Firefox OS
71	iOS
84	Java ME (J2ME)
95	Tizen
100	Windows Phone ve Windows RT
114	Çapraz Platformu Seçmek
132	Mobil Siteler ve Web Teknolojileri
147	Erişilebilirlik
157	Kurumsal Uygulamalar: Strateji ve Geliştirme
169	Mobil Mantıksal Analiz
178	Zengin Medya Desteği Vermek
184	Konum Tabanlı Hizmetlerin Uygulanması
193	Yakın Alan İletişimi (NFC)
201	Dokunsal Titreşim (Haptic Vibration) Özelliği
210	Arttırılmış Gerçekliğin Uygulanması
222	Güvenlik
234	Test Etme
252	Gelir Edinmek
268	Sonuç
270	Yazarlar Hakkında



Giriş

Enough Software'li 2005 yılında kurduğumuzda, ailemiz ve arkadaşlarımız arasında gerçekten ne yaptığımızı anlayan kimse yoktu. Her ne kadar cep telefonları her yerde olsa ve SMS'ler bilinse de, uygulamalar bu kadar yaygın değildi, Uygulama kelimesi bile insanlara çok garip geliyordu; o tarihte bunlara MIDlet veya "mobil uygulamalar" adını koymuştuk. Müşterilerimiz için uygulamalar tasarlamaya ve geliştirmeye devam ettik ve o tarihten bugüne ilginç bir dönem geçirdik. Eski platformlar kayboldu, yeni platformlar ortaya çıktı ve bunlardan bazıları adeta dünyayı ele geçirdi. Genel anlamda, mobil ekosistemi gerçekten çılgın bir deneyimdi.

Mobil Geliştiricilerin Galaksi Rehberi ile bu ekosistemi yakından takip etmeye başladık. Katkıda bulunan yazarlarımız sayesinde bu kılavuz -Symbian veya WebOS gibi platformları çıkarmış olsak da- toplamda sekiz adet mobil platformu kapsamaktadır. Bu arada bu sürüm, Tizen'e değineceğimiz ilk yayındır.

Bu yayın şimdiye kadarki en büyük yayınıımızdır. İlk baskıda 10.000 kopya basılmıştır ve sponsorlarımız olmadan bunu gerçekleştirmek gerçekten mümkün değildi. Bunlardan Paypal (developers.paypal.com adresini ziyaret ederek birçok avantajı görebilirsiniz) ve SAP (mükemmel mobil platformlarını developers.sap.com.com adresinde bulabilirsiniz) firmalarına teşekkür ediyoruz. Ayrıca bu yayın için ilk defa sponsor olan Twilio'ya da hoş geldin diyoruz. Uygulama dağıtımında iki faktörlü kimlik denetiminden uygulamalarınız içerisinde VoIP ve mesaj gönderme özelliklerine kadar bu araçların size nasıl yardımcı olacağını görmek istiyorsanız www.twilio.com sayfasını ziyaret edin.

Gelecek bize heyecan veriyor. Heyecanınızı bizimle Twitter üzerinden @enoughsoftware veya e-posta ile paylaşmanız için sabırsızlanıyoruz.

Yorumlarınızı dört gözle bekliyoruz!

Robert + Marco / Enough Software

Bremen, Şubat 2014





Mobil Galaksisi: Giriş

Mobil geliştirme dünyasına hoş geldiniz. Burası eski devlerin tökezlediği ve yeni yıldızların düzenli olarak doğduğu bir evren.

Bu kitabın odaklandığı nokta, cep telefonu uygulamalarının geliştirilmesidir. Bu odaklanmanın merkezinde şu aşamalar bulunmaktadır: planlama ve açıklama, prototipleme ve tasarım, uygulama, iç test ve kullanıma sokma, bir uygulama dükkânına sunuş, kullanıcılar tarafından keşfedilme, kurulum, kullanım ve geri bildirim. Nihayetinde, kullanıcılarımızın uygulamalarımızı kullanırken iyi vakit geçirmelerini ve diğer kullanıcılara örnek teşkil etmesi için bize olumlu yorumlarda bulunmalarını isteriz.

Ana platformlar için uygulama yöntemleri hakkında bilgi almak için okumaya devam edin. Katılımda bulunmak için bu ilk deneyiminizse, vakit kaybetmemenizi öneriyoruz. Dünya hızlı bir şekilde bilgisayar dünyasındaki uygulama hâkimiyetine geçiş yapıyor ve eğer vakit kaybederseniz, diğerleri sizi çoktan geçmiş olacak.

Mobil uygulamaları geliştirmek, diğer yazılımları geliştirmekle kıyaslandığında birçok ortak nokta bulundursa da belirli özelliklere sahiptir. Bir sonraki bölümde bunlara değineceğiz.

Topoloji: Biçim Katsayıları ve Kullanım Şablonları

Akıllı telefonlar, tabletler ve temel fonksiyona sahip telefonlar arasında bir fark gözetmek zorundasınız. Her bir biçim katsayısı kendi kullanım zorluklarını öne çıkarmaktadır; örnek vermek gerekirse, bir tabletin içerisinde gezinme yöntemi

bir telefonla kıyaslandığında farklı olacaktır. TV sistemleri de mobil geliştiriciler için başka bir biçim katsayısına sahip olmaktadır.

Elbette bir Android uygulamasının kullanım şablonu, iOS'takilerden farklı olacaktır; aynen Windows Phone uygulamalarında ve benzerlerinde olduğu gibi.

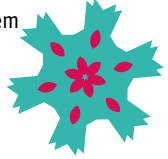
Bu yüzden, bütün olası cihaz boyutlarında (form factor) ve hatta hedeflediğiniz tüm mobil platformlarında birbirinin aynısı bir kullanım tecrübesi sunmaktan kaçınılmalıdır. Aksi takdirde hedeflediğiniz kullanıcı tabanınıza herhangi bir üstün yanı olmayan, vasat bir hizmet sunmuş olursunuz.

Yıldız Oluşumu: Bir Mobil Hizmeti Yaratmak



Bir mobil hizmeti yaratmak için birçok farklı yöntem bulunmaktadır:

- Uygulama
- Web sayfası
- SMS, USSD¹ ve STK²

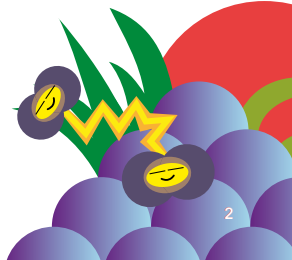


Uygulama

Uygulamalar doğrudan cihaz üzerinde çalışmaktadır. Bunları doğal (native), web tabanlı veya karma uygulama olarak oluşturabilirsiniz.

1 en.wikipedia.org/wiki/USSD

2 en.wikipedia.org/wiki/SIM_Application_Toolkit



Native Uygulamalar

Bir native uygulama, platforma özel API'lar ile belli bir dilde programlanmaktadır. Tipik olarak platforma özel merkezi bir uygulama dükkânından satın alınır, indirilir ve güncellenir. Native uygulamalar, diğer uygulamalara kıyasla genellikle en iyi performansı, en derin entegrasyonu ve en iyi genel kullanıcı deneyimini sunar. Öte yandan, native geliştirme süreci genellikle en karmaşık geliştirme seçeneğidir.

Web Uygulamaları

Bir web tabanlı uygulama HTML5, JavaScript ve CSS bazlıdır ve herhangi bir uygulama dükkânına bağlı değildir. Bir uygulamanın görünümünü ve davranışlarını kopyalamaya çalışan, yerel olarak saklanan mobil bir sitedir.

Web tabanlı ünlü bir uygulama örneği, daha fazla gelir için uygulama dükkânını terk eden Financial Times uygulamasıdır. Bunun aksine, web tabanlı Facebook iOS uygulaması, performansını ve kullanılabilirliğini ciddi anlamda artırmak için bir native uygulamaya dönüştürülmüştür. Web uygulamaların geliştirme sürecini kolaylaştırmak için Phonegap³ gibi farklı uygulama geliştirme çatılarını kullanabilirsiniz.

Karma Uygulamalar

Native veya web tabanlı uygulamaların gelecekte çok popüler olup olmayacağı konusunda birçok tartışma mevcuttur.

Birçok mobil uygulama geliştiricisi için bu tartışmalar, karma uygulamaların ortaya çıkmasıyla beraber azalmaya başlamıştır: bir uygulama, daha iyi bir performans ve platform ile uygulamanın daha iyi uyum sağlaması için native kodu kullanırken, uygulamanın diğer unsurları için HTML5 temelli içerikle beraber bir webview kullanabilir. Ortaya

çıkan uygulamanın bazı kısımları bir native uygulama gibi iş görürken diğer parçalar web teknolojileri ile desteklenmektedir. Web tabanlı parçalar, güncelleme içeriği sunmak için İnternet bağlantısını kullanabilir. Bu bir engel olarak görünse de, web teknolojilerinin kullanımı geliştiricilerin, uygulama dükkânlarına güncelleme sunmadan içeriği ve özellikleri değiştirip güncellemelerine fırsat vermektedir. Buradaki asıl zorluk, gerçekten kullanıcı dostu ve çekici bir uygulama yaratmak için native ve web teknolojilerinin benzersiz özelliklerini bir araya getirmektir.

Web sitesi

Bir web sayfası çoğunlukla ağınız üzerinde yer alır ancak, yerel olarak veri saklamak veya cihazın mevcut yerini öğrenmek amacıyla JavaScript'e sahip cihaz üzerindeki çeşitli telefon özelliklerine de erişimde bulunabilirsiniz. Uygulamaların tersine mobil web siteleri doğaları gereği farklı platformlara yayılırlar. Bir mobil tarayıcının daima WebKit tabanlı olduğunu düşünmemelisiniz; Bkz. Mobil web geliştiricilerinin sitelerini yalnızca WebKit için optimize etmemeleri için Microsoft'un ricası⁴.

SMS, USSD ve STK

SMS, USSD veya STK ile basit hizmetler gerçekleştirilebilir. Herkes SMS'in (Kısa Mesaj Servisi) nasıl çalıştığını bilmektedir ve her telefon SMS'i desteklemektedir ancak diğer karmaşık hizmetler için kullanıcılarınızı metinsel komutları hatırlamaları gerektiği konusunda ikna etmeniz gerekir. Bazı operatörler, yalnızca WiFi cihazlarda çalışan mesaj hizmetleri için API'lar

4 blogs.windows.com/windows_phone/b/wpdev/archive/2012/11/15/adapting-your-webkit-optimized-site-for-internet-explorer-10.aspx

sunmaktadırlar Deutsche Telekom'un ağ API'ları gibi⁵. USSD (Yapısal olmayan Yardımcı Hizmet Verisi), düz metin bazlı menüleri göstermek için kullanılan bir GSM protokolüdür ve becerileri telefon operatörü ve cihaza göre değişmektedir. STK (SIM Uygulama Yazılımı), doğrudan bir telefonun SIM kartı üzerindeki düşük seviyeli ancak etkileşimli uygulamaların devreye sokulmasına fırsat vermektedir.

STK, akıllı telefon uygulamaları üzerinde bu kadar çalışma yapılyorken biraz alakasız kalabilir ancak örnek vermek gerekirse, m-pesa Kenya ve diğer ülkelerdeki yaşam unsurları ve finansal işlemlerde değişiklik yapan bir STK uygulamasıdır.⁶

Mobil İşletim Sistemleri Evreni

Mobil Dünya, bilişim teknolojilerinin diğer alanlarına kıyasla çok daha çeşitlidir. Kişisel bilgisayarlar için yazılım geliştirirken, üç ana işletim sisteminden birini (Windows, Mac OS, Linux) seçersiniz. Konu mobil olduğunda, çok daha fazla alternatif bulunmaktadır. Bu kitap şu anda en fazla kullanılan mobil işletim sistemlerini tanıtmaktadır ancak mobil alandaki değişikliklerin, başka hiçbir alanda olmayan bir hızla hareket ettiğini unutmayın. Gelecek vaat eden birçok teknolojinin, arkalarındaki şirketler ne kadar büyük veya bu şirketlerin geçmiş piyasa alakası ne kadar geniş olursa olsun, ortaya çıktıktan sonra aniden yok olduğuna şahit olduk.

Bu yüzden okumaya devam edin; piyasanın günümüzde nasıl olduğunu görün ve bunu gözlemleyin (en azından, kılavuzumuzun son sürümüne sahip olduğunuzdan emin olun).

⁵ www.developergarden.com/apis

⁶ memeburn.com/2012/03/how-m-pesa-disrupts-entire-economies/

Yıldızsı Gök cisimleri: Android ve iOS

İnsanlar mobil uygulamalar hakkında konuşurken genellikle Android veya iOS'tan bahsetmektedirler. Neden? Piyasa payı söz konusu olduğunda bu iki platform Önemli piyasaların %90'ını ellerinde tutmaktadırlar⁷ (küresel sayılar için aşağıdaki tabloya bakın). Developer Economics 2014 araştırması⁸ ayrıca, iOS ve Android'in geliştirici tüketici bilinci açısından en yüksek seviyede olduklarını göstermektedir. Bu, hangi platformun "temel" olduğuna bakmaksızın geliştiricilerin kullandıkları platform yüzdesidir. Android %71 ile birinci sıradayken, iOS bunu %55 ile takip etmektedir.

Bunun bir diğer anlamı da şu: Uygulamanızı Android veya iOS için geliştirecekseniz, çok fazla rakibiniz olacaktır.

Karanlık Madde: Temel Fonksiyonlu Telefon Platformları

Akıllı telefonlar, üzerine en fazla haber yapılan cihazlar olsa da, dünyanın birçok bölgesinde temel fonksiyonlu telefon anlayışı egemenliğini korumaktadır. Küresel olarak 2013'ün ikinci çeyreğinde satılan bütün telefonların %50'i temel fonksiyonlu cep telefonlarıydı⁹ ve kurulum temeli bundan çok daha fazlaydı. En büyük satıcılar Samsung ve Nokia'ydı. Nokia, Nokia mağazası ile büyük bir başarıya sahip olduğunu iddia etmektedir; uygulamalarının indirme sayısının 1 milyonu geçtiği 500'den fazla geliştirici mevcuttu¹⁰. 2011'de yapılan araştırmalar, popüler olmayan platformların geliştiriciler için aslında çok daha büyük bir şans yarattığını göstermiştir: Nokia'nın OVI mağazasındaki temel fonksiyonlu cep telefonu

⁷ www.idc.com/getdoc.jsp?containerId=prUS24442013

⁸ DeveloperEconomics.com

⁹ gartner.com/newsroom/id/2573415

¹⁰ developer.nokia.com/Distribute/Statistics.xhtml

uygulamaları, Apple Uygulama Mağazası'ndaki indirme sayılarına kıyasla 2,5 kat daha fazlaydı¹¹.

Temel fonksiyonlu telefonlar için native uygulamalar geliştirmek mümkündür ve bunun yolu üretici firmalarla yakın ilişkide olmaktan geçer. Bunun dışında kalan tüm diğer yazılımcılar, bu telefonlar için Java ME veya BREW kullanarak uygulama geliştirirler.

Magnetarlar: Windows Phone ve Windows RT

Windows Phone, akıllı telefon evrenindeki 'üçüncü ekosistem'¹² haline geldi; İtalya ve Latin Amerika gibi bölgelerde iPhone'dan daha fazla satış rakamlarına sahiptir. Windows 8.1 ve Windows 8'in pazar payı artık, Net Applications'a göre¹³ bütün Mac OS X sürümlerinin toplamını geçmiştir.

Super Nova'lar: Sailfish OS, Firefox OS, BlackBerry 10 ve Aliyun

Bu platformlar mükemmel başarı hikâyelerine mi dönüşecekler ya da mobil endüstrinin lanetli cihazları olarak tarihe mi gömülecekler? Kimse tam olarak bilmiyor, ancak yorumlamaya açık olarak karmaşık mesajlar bulunuyor.

Fin şirketi Jolla¹⁴, 2013'ün dördüncü çeyreğinde Sailfish OS¹⁵ ile piyasa giriş yaptı. Bu OS çeşitli inceleme puanları aldı, bu yüzden Sailfish'in 2014 yılı içerisinde gelişmesini izlemek çok ilginç olacak.

¹¹ <http://www.research2guidance.com/apps-on-nokias-ovi-store-had-2-5-times-higher-download-numbers-in-q2-2011-compared-to-apps-on-apple-app-store>

¹² kantarworldpanel.com/global/News/news-articles/Apple-iPhone-5S-outsells-5C-three-to-one-in-Great-Britain

¹³ netmarketshare.com/operating-system-market-share.aspx

¹⁴ jolla.com

¹⁵ sailfishos.org

Firefox OS¹⁶, eleştirilenler ve geliştiriciler tarafından çok sevildi; bu yüzden bu kılavuz içerisinde bu platform için özel bir bölüm ayırdık. Küresel olarak farklı yerlerde piyasaya çıktıktan sonra ciddi bir pazar payına sahip olmayı başaramadı. Firefox OS ayrıca tabletler için de çıkacak.

BlackBerry 10'a gösterilen ilk tepkiler şüphe ve heves arasında gidip geldi. Bütün ilgili operatörler BlackBerry 10 cihazını satışa sunsalar da, bu yeni platformun benimsenmesi yavaş oldu. BlackBerry'nin satışları da başarısız olmuş gibi gözükse de, bu yeni işletim sisteminin geleceğini görmek çok ilginç olacak.

Aliyun, Çin'de, bilinmeyen bir pazar payı ile tek cihaz üzerinde piyasaya çıktı. Google'ın, Acer'ı, Open Handset Alliance üyeliğinde Aliyun içeren bir cihaz çıkarmaması yönünde baskı yapması ve Aliyun'un uygulama dükkanında korsan Google Android uygulamaları bulunması sebebiyle¹⁷ platform belli bir seviyede popülerliğe sahip oldu. Aliyun'un Linux temelli olduğu iddia edilse de, kaynak kodu henüz ortaya çıkmadı.

Beyaz Cüce Yıldızlar: Symbian ve Bada

Bugün, Symbian ve Samsung Bada, eski görkemli geçmişlerinin sadece birer gölgesine dönmüş durumdadır. Bada çok kısa ömürlü olsa da, Samsung, Tizen'i yaratırken Bada'nın bazı bileşenlerini kullandı. Symbian bakım moduna alındı ve son büyük patlamasını yarı fotoğraf makinesi yarı telefon olan PureView 808 ile yaptı. Bu platformların önemi ve pazar payı bütün dünyada hızlı bir şekilde azalmaya devam ediyor.

¹⁶ mozilla.org/firefox/os

¹⁷ news.cnet.com/8301-1035_3-57513651-94/alibaba-google-just-plain-wrong-about-our-os

Yeni Doğan Yıldızlar: Tizen ve Ubuntu

2014 yılında çok ilginç yeni platformlar göreceğiz.

Tizen¹⁸ cihazları, 2014'ün ilk çeyreği için duyuruldu. Samsung ve Intel tarafından sessiz sedasız ancak sürekli bir şekilde ön plana çıkartılan Tizen, akıllı telefonların yanı sıra TV'ler, tabletler ve araç içi bilgi-eğlence sistemlerinde kullanılmak üzere geliştirilmiştir. Bu sürümde Tizen hakkında ayrı bir bölüm içermemiz, platformu ciddiye aldığımızı ve Bada'dan daha uzun ömürlü olacağını düşündüğümüzü göstermektedir.

Son olarak, Canonical firması mobil cihazlar için Ubuntu'yu¹⁹ sundu. Buradaki amaç bir bilgisayarın tam gücünü telefona uyarlamaktır. Ubuntu Edge telefonuna fon yaratmak için oluşturulan kitle kaynak çalışması²⁰ hedefine ulaşmadı ancak yine de, Ubuntu pazara girmeyi hedeflemektedir.

Güneş Sistemi: Akıllı telefon OS Pazar Payları

Küresel akıllı telefon pazar paylarına baktığınızda, ortaya çıkan tablo şöyle:

Platform	Pazar Payı 2013 3. Çeyrek	Yıllık Değişim
Android (Google)	%81,3	%+6,3
iOS (Apple)	%13,4	%-2,2
Windows Phone (Microsoft)	%4,1	%+2,0
BlackBerry	%1,0	%-3,3
Diğer	%0,2	%-2,8

(Kaynak: blogs.strategyanalytics.com/WSS/post/2013/10/31/Android-Captures-Record-81-Percent-Share-of-Global-Smartphone-Shipments-in-Q3-2013.aspx)

¹⁸ tizen.org

¹⁹ ubuntu.com/devices/phone

²⁰ indiegogo.com/projects/ubuntu-edge

iOS ve Android dışında diğer platformlarda vakit harcamak isteyen geliştiricilerin boşuna vakit kaybettiğini düşünebilirsiniz. Ancak gerçek bu kadar basit değil. 2013'ün ilk çeyreğinde Dünya çapında ilk kez akıllı telefon satışı temel fonksiyonlu telefon satışlarını geçse de²¹ bir çok bölgede temel fonksiyonlu telefonlar akıllı telefonlara kıyasla hâlâ çok daha fazla miktarda satılmaktadır.

Bu arada, Sailfish OS ve Firefox OS pazar paylarının - küçük miktarda olsa bile - henüz bilinmediğini göz önünde bulundurun. Bu platformlardan herhangi bir tanesi sizin işiniz için en iyi çözüm olabilir - galaksinin merkezinde binlerce rakip uygulama arasından sıyrılmaya uğraşmak yerine evrenin öbür ucunda küçük ve tek bir gezegen olmak, çok daha avantajlı olabilir.

Bunların küresel rakamlar olduğunu unutmayın - bir platform bölgesel pazar payı diğer pazarlardan bambaşka olabilir. Yerelleştirilmiş içeriğin kendi önemini artırdığı bir Dünya'da, hedef piyasanızın ayrıntılarını ve özelliklerini bilmeniz çok önemlidir. Örnek vermek gerekirse, günümüzde Çin 2013'ün üçüncü çeyreğinde dünya çapındaki Android satışlarının %40'dan fazlasından sorumludur²², öte yandan Çin'de üretilen cihazlar Google Play mağazası veya başka bir Google hizmeti olmadan sunulmaktadır.

Hedef bölgenizdeki pazar payı hakkında bilgi almak için comscore²³, StatCounter²⁴, VisionMobile²⁵ veya Gartner²⁶ gibi kaynakları kullanın.

²¹ idc.com/getdoc.jsp?containerId=prUS24085413

²² engadget.com/2013/11/14/android-ios-market-share-gartner-q3-2013/

²³ comscoredata.mine.com/category/mobile

²⁴ gs.statcounter.com

²⁵ visionmobile.com

²⁶ gartner.com

Zaman ve Mekân Hakkında

Geliştiriciler olarak, seçtiğimiz platformlar için büyük bir tutkuya sahibiz. Yine de bunların yalnızca birer teknoloji ürünü olduğunu unutmamalıyız – teknolojiler yalnızca belli bir süre ve mekân açısından önemlidir. Evet, marka tartışmaları eğlencelidir ancak daha geniş bir açıdan bakıldığında daima saçmadır. 80'lerde Atari'ye karşı Amiga kavgası yapanlar elini kaldırsın! Belki de çoğunuz bunu yapmadı ancak ne dediğimizi anlıyorsunuz galiba. FairPhone²⁷ veya IndiePhone²⁸ gibi inisiyatifler gelecekte işletim sistemi veya seçtiğiniz bir üreticiden daha önemli bir hale gelebilir.

Uzayda Kaybolmak

Eğer mobil dünyasının uçsuz bucaksız uzay ortamında kaybolursanız, endişelenmeyin, sakın olun ve okumaya devam edin. Seçeneklerinizi gözden geçirin ve çözmek istediğiniz sorunu ele alın, hedef kitlenizi ve teknik bilginizi gözden geçirin. Hizmetinizi tasarlarken dikkatli bir şekilde çalışın, mevcut soruna odaklanın ve işi basit tutun. "Her şeyi" yapmak yerine bir şeyi düzgün bir şekilde yapmak daha önemlidir. Çözümünüzün tasarımına ve kullanılabilirliğine yatırım yapın. Son olarak, halihazırda başarılı olan bir şeyi kopyalamak yerine doğru bir fikri bulmak çoğunlukla daha iyidir. Bu kılavuz, bilgiye dayalı bir karar vermenize yardımcı olacaktır.

²⁷ fairphone.com

²⁸ indiephone.eu



Mobil için Kavramsal Tasarım

Bir fikri bulduğunuzda bu, mükemmel bir histir. Derhal ne yapacağınızı bilir ve potansiyel kullanıcıların karşı karşıya kalabilecekleri sorunu çözersiniz. Bu ilk aşamadan, nihai uygulama kullanım aşamasına gitmek zorludur. Yalnızca istikrarlı bir uygulama yaratmak istemez, ayrıca yardımcı ve kullanımı kolay olmasını da istersiniz.

Kullanıcı Deneyimi, kullanıcıların uygulama ile etkileşme geçtikleri sırada ve sonrasında uygulamayı algılama şekilleri olarak tanımlanabilir. Tasarımı iyi miydi, kullanımı eğlenceli ve kolay mıydı? Verimli veya eğlenceli bir şekilde kullanıldı mı? Her şey düzgün çalıştı mı? Tasarım ve kod işine girişmeden önce uygulama konseptinizi iyi belirlemenizde fayda vardır. Aşağıda sıraladığımız başlıklar uygulama fikrinizin tasarım ve inşa aşamalarına geçmesinde size yardımcı olacaktır.

Fikri Bulmak

Birkaç cümle kullanarak uygulamanızı açıklayan bir **kavram özeti** yazın. Ekip üyelerinizin dışında farklı kişilere bunu anlatmaya çalışın. Onların bunu nasıl anladıklarını ve buna ilgi duyup duymadıklarını görün.

Kitlenizi tanıyın. Tasarımı ulaştırmak istediğiniz kişiler kimler ve uygulamanızı kullanmadaki amaçları ne? Eğer mümkünse onlarla konuşun ve bunu kendi ağızlarından duyun. Bulgularınızı belgelendirmede yararlı bir teknik, her farklı kullanıcı grubu için bir profil oluşturmaktır.

İçeriğinizi belirleyin. Uygulamanın ana içeriğinin ne olacağını kendinize sorun. Uygulama türüne göre bunlar

fotoğraflar, kullanıcı tarafından oluşturulmuş haberler, özgün veriler (kitaplar, meta veriler, müzik, vb.) olabilir. Görüntülenecek temel bilgileri veya önemli etkileşimi tanıdıktan sonra, tasarım aşamasına doğru bir şekilde odaklanmak daha kolay olacaktır. Örnek olarak, bir e-kitap okuyucu uygulaması yaratıyorsanız, yazı tiplerinin iyi kalitede olduğundan ve temel ekran tasarımınızın okura kitap metnini okuyabilmesi için yeterli alanı sunduğundan emin olmanız gerekecektir.

Temel **işlevi** tanımlayın. Kullanıcılar arayüzle neler yapabilecekler? Bu işlevleri birer fiil gibi düşünüp, listeleyebilirsiniz: dolaş, paylaş, satın al, vb. Bazı faaliyetlerin bununla ilgili olduğunu göreceksiniz. Örnek olarak, eğer uygulamanızın güçlü bir topluluk unsuru mevcutsa, bir grup içerisine alabileceğiniz çeşitli özellikler bulunabilir (paylaşma, yorum yapma, mesaj gönderme, takip etme gibi). Bu başka bir kullanıcı arayüzü (UI) ipucu olabilir. Benzer işlevler birbirine yakın bir şekilde sunulduğunda, kullanıcılara yardımcı olabilir.

Mobil deneyim için tasarım yaparken, uygulamanızın kullanılacağı **bağlamı** düşünmeniz gerekir. Bu hem arayüz kullanılabilirliğini hem de kullanıcıları nasıl etkileyecek? Kullanıcıların pür dikkatine sahip olabilecek misiniz? Ya da onlar hep aynı yerde mi kalacaklar? Uygulamanız tek başına çalışan bir ürün mü? Diğer hizmetlerle ilişkili mi ya da onlara ihtiyaç duyuyor mu? Eğer bir internet bağlantısı yoksa neler olacak? Uygulamanızın kullanıcı arayüzü (UI) bu durum karşısında nasıl bir tepki gösterecek?

Pazar **araştırması** için biraz vakit harcamanız yararlı olacaktır. Sizin için benzer diğer uygulamaları deneyin. Ne yaptıklarını görün: kullanıcıların onlar hakkındaki düşüncelerini öğrenin. Bu, girdiğiniz alanı öğrenmenin iyi bir yoludur.

Birçok soruyu cevapladıktan sonra, yaratmak istediğiniz uygulama hakkında genel bir kavrama gücüne sahip olacaksınız. Fikir geliştirme aşamasında daha da ileriye gittiğinizde bu

soruları sormaya devam edin. Odaklanmak iyidir ve ulaşmak istediğiniz hedefe ne kadar yakın olduğunuzu sorgulayın. Bazen elde etmek istediğiniz şeyler zamanla değişebilir. Belirlenmemiş bir kavram, bir zamanlar ulaşmak istediğiniz hedefin sizi başka fikirlere ve yerlere götürmede yeni bir araç olarak karşınıza çıkabilir.

Kullanıcı Deneyimini Tasarlamak

Uygulamanın genel kullanıcı arayüzünü yakalamak ve geliştirmek için kullanıcı akışları, bilgi mimarisi, etkileşimler, plan yapısı ve görsel tasarım hakkında biraz düşünmeniz gerekir. Ürününüzün çevresi içerisinde bunlar bir arada nasıl çalışacaktır? Her bir kullanım durumunun ayrıntıları nelerdir? Uygulamanızı kullanırken kullanıcıların karşılaşabilecekleri sorunlar nelerdir ve geri bildirim bu konuda nasıl yardımcı olabilir? Ürününüz hazır olduktan sonra, bunun hakkında insanlara nasıl bilgi vereceksiniz ve insanları bunu kullanmaları için nasıl ikna edeceksiniz?

Kullanıcı akışları

Bazı uygulamalar (bir kamera uygulaması gibi) belirli bir görevi gerçekleştirmek için çizgisel bir akışa sahiptir. Bazılarında yinelemeli bir yaklaşım bulunur. Kullanıcının A noktasından çıkıp birkaç adımdan sonra B noktasına ulaştığı “ideal senaryo(lar)ınızı” tanımlayın. İdeal yoldan sapabilecek olan diğer olası yolları araştırın. Akış şemaları çizin veya farklı senaryoları ayrıntılı bir şekilde ortaya çıkarmak için wireframe araçlarını kullanın.

Wireframe

Wireframe, tasarımın T'sine dahi dokunmadan projenin kurgusu ve işlevselliği hakkında hem yazılımcıya hem de müşteriye fikir sahibi olma olanağı sağlayan proje ön hazırlık aşamasıdır. Wireframe aşamasında arayüzünüzün bir "telkafes"i andıran çizgilerle taslağını oluşturursunuz. Wireframe çiziminin amacı, fonksiyonları ve genel arayüz kavramını yakalamaktır. Belli bir ekran için bir wireframe, belli bir senaryoya göre farklı sürümlere/durumlara sahip olacaktır. Bir ağ hatası için aynı ekranın farklı çeşitlerini çizmeniz gerekecektir.

Ayrıntılı planlara dalmadan önce, uygulamayı geliştireceğiniz işletim sisteminin "Temel Tasarım Kılavuzu"nu bilmeniz gerekmektedir. Her bir platform farklı bir ortamdır ve doğru dönüştürmeleri kullanabilmek için bu kılavuzları okumanız gerekir. Aksi şekilde davranmak için iyi bir nedeniniz olmadığı sürece, tavsiye edilen yöntemleri takip edin. Araştırmanızı yapın ve İnternet'teki mevcut desen galerilerinden faydalanın.

"Native" görünümüne yakın olmak size, hızlı kullanılabilirlik faydaları sağlayacaktır. Kullanıcıların, cihazlarındaki diğer uygulamalarda bulunan standart davranışları veya görsel işlemleri hatırlamaları olasıdır. Platforma özel bağlantıları, bu kitabın ilgili bölümlerinde bulacaksınız.

Wireframe çizimi, kalem kağıt yardımı ile yapabilir veya mevcut olan Wireframe araçlarından herhangi birini kullanabilirsiniz. Yeni bir yazılımı öğrenerek vakit kaybetmemek adına kağıt üzerinde çizim yapmak, daha kolay olabilir. Çizimleri değiştirmek ve silip yeniden başlamak, çok daha kolay ve bazen de eğlencelidir. Wireframe aşamasında bu işe özel yazılımları kullanmanın sunduğu en temel avantaj, bu tasarımlar üzerinde başkalarıyla birlikte çalışabilme olanağını ve taslakları tıklanabilir prototiplere dönüştürebilmektir.

Prototiplendirme

Bir prototip, uygulamanızın etkileşimini görselleştirmenin ve değerlendirmenin en iyi yoludur. Büyük bütçeli bir projenizin olması veya hafta sonlarında kişisel bir proje üzerinde çalışmanız fark etmez; uygulamanızın nispeten tamamlanmış bir prototip sürümü olması, kavramınızı diğer kişilere iletmek ve bunun hakkında tartışmak için en iyi yöntemdir. Prototip, nihai kod ve mükemmel çözünürlüğü geliştirmeden önce yapılması gereken bir aşamadır. Kabul edilmiş, tıklanabilir bir gezinti, ekiplerin amaçlarından sapmadan hedeflerine doğru ulaşabilmeleri için yararlı bir referans olacaktır.

Bir prototipi bir araya getirmenin "mükemmel" bir yolu yoktur. Sizin için hangi teknik uygunsa onu kullanabilirsiniz. Kağıt prototiplendirmeden özel oluşturulmuş araçlardan bir tanesini kullanmaya veya tıklanabilir sürümleri bir araya getirmeye kadar (standart sunum araçları gibi) çeşitli yöntemleri kullanabilirsiniz. Eğer kod yazma beceriniz mevcutsa, bir HTML prototipi oluşturmak da iyi bir yöntemdir. Mevcut çerçeveleri ve kütüphaneleri, nihai ürüne benzeyecek ve onun gibi davranacak bir prototip tasarlamak için kullanabilirsiniz.

Kod yazmaya başlamadan önce, bütün prototipi bitirmenize gerek yoktur. İşinizi organize etme şeklinize göre, ilerledikçe uygulamanın çeşitli parçalarına odaklanabilir ve hatta gerçek kodun hızlı iterasyonlarına doğru hareket edebilirsiniz. Yine de, ilk uygulama fikirleri, eksik modelleri geçerli kılmada daha hızlı bir yöntem olacaktır.

Ücretsiz pek çok prototipleştirme yazılımı mevcuttur ve ücretli olanlarının da çoğu deneme sürümleri veya sınırlı sayıda proje için ücretsiz hesap seçeneklerine sahiptir. Gün geçtikçe yeni prototipleştirme uygulamaları da ortaya çıkmaktadır. Deneyebileceğiniz ve seçebileceğiniz bazı uygulamalar şunlardır:

Uygulamaları	İlgili	Durumu
App in seconds appinseconds.com	Web tabanlı, iPhone için prototiplendirme.	(ticari)
Axure axure.com	Masaüstü uygulaması, tel kafesleme ve prototiplendirme.	(ticari)
Balsamiq Mockups balsamiq.com	Wiki'ler ve hata izleme araçları için masaüstü uygulaması veya eklentiler, Wireframe aracı.	(ticari)
Fluid UI fluidui.com	Web tabanlı, iOS, Andorid, Windows ve web projeleri için prototiplendirme.	(ticari)
Mockingbird gomockingbird.com	Web tabanlı wireframe ve prototiplendirme.	(ticari)
OmniGraffle omnigroup.com/products/omnigraffle	Masaüstü uygulaması ve wireframe.	(ticari)
Pencil pencil.evolus.vn	Masaüstü uygulaması veya Firefox eklentisi; özgür GUI prototiplendirme aracı.	(ücretsiz)
POP popapp.in	iPhone ve Android uygulaması, elle çizilen tasarımları ve fotoğrafları tıklanabilir prototiplere dönüştürür.	(ücretsiz)
Proto io proto.io	Web tabanlı, prototiplendirme.	(ticari)

Uygulamaları	İlgili	Durumu
Proty protytype.com	Web tabanlı, hassas wireframe aracı.	(ücretsiz)
UX Pin uxpin.com	Web tabanlı, tasarım ve wireframe.	(ticari)

Görsel tasarım

Görsel olmayan girdileri kullanan bir uygulama yaratmadığınız sürece, uygulama arayüzünüz grafiklere dayanacaktır. Görsel tasarım ayrıntılarına odaklanmanız, uygulamanızın kullanıcı deneyimini artıracak ve diğer uygulamalar karşısında bir adım önde olmasını sağlayacaktır.

Konumlandırma ve görsel hiyerarşi, arayüzünüzün kullanılabilirliğini artırır. Yerleşim, ekrandaki unsurların konumlandırılma ayrıntılarını ve birbirleriyle olan ilişkilerini belirler. Kullanıcılar uygulama arayüzünü bir kere öğrendikten sonra bu, akış boyunca tutarlı kalmalıdır. Örnek olarak, eğer temel eylem düğmeniz ekrandan ekrana renk değiştiriyorsa, bunun kullanıcılar üzerindeki etkisini düşünün; kafaları karışmayacak mı? Değişikliğin önemini anlayabilecekler mi? Eğer renk değişiklikleri kasti ise, bunları iyi ve mantıklı sebepler için yaptığınızdan emin olun.

Düzen tasarımlarına ve wireframe seviyesindeki etkileşimlere benzer olarak, bazı tasarım kararları, belli platform kurallarına göre şekillenebilir. Uygulamanız, tasarlandığı platforma göre farklı görsel öğelere sahip olabilir. Tasarımlarınızın, yazıtipi kullanımı, standart simgeler, düzen kuralları için tavsiye edilen uygulamaları takip ettiğinizden emin olun. Yineliyoruz: daha fazla bilgi ve belirli online kaynaklar için bağlantıları görüntülemek amacıyla bu kılavuzda bulunan platforma özel bölümlere bakınız.

Kullanıcıların uygulamanız ile etkileşime geçebilmeleri için marka kimliğinin kullanıcı arayüzü (UI) içerisinde engelleyici olmayacak bir şekilde bulunması önemlidir. İstenilen görüntüyü ve duyguyu eklemek için arka planı, kontrol renklerini, belki de çeşitli resimleri veya düzen seçeneklerini kullanın. İlave grafikleri kullanabileceğiniz başka bir yer ise (eğer mevcutsa) açılış ekranıdır.

Son olarak, başlatma simgesi, uygulamanızın görüleceği, tanımlanacağı ve değerlendirileceği ilk görsel unsur olacaktır. Bunun güzel görüldüğünden emin olun. Eğer uygulamanızı birden fazla platformda yayınlamak istiyorsanız, kolaylıkla taşınabilir olan bir çizim yaratabilmeniz açısından tasarım gereksinimlerini erkenden kontrol edin.

Birden fazla ekran için tasarım yapmak

Sürekli olarak değişen mobil cihaz pazarında uygulama arayüzünüzün farklı ekran boyutlarında ve görüntü yoğunluklarında nasıl görüneceğini göz önünde bulundurmanız gerekmektedir. Wireframe aşamasında görsel ölçeklenebilirlik-kullanışlılık dengesi üzerine kafa yormak, her ne kadar bu tür ayrıntılarla uğraşmak için biraz erken olsa da; bu size daha sonraki geliştirme ve test süresi süreçlerinde muazzam vakit kazandırabilir. Eğer bu konu sizin için tamamen yeni bir konsept ise, Duyarlı Web Tasarımı'nın (Responsive Web Design) iyi örneklerine bakarak hakkında konu hakkında daha fazla fikir edinebilirsiniz. Web tasarımcıları bu sorunları uzun bir süredir çözmektedirler. Bir kez daha yineleyelim:, bu konu hakkında platforma özel "Tasarım Kılavuzları"nı gözden geçirmek yararlı olabilir.

Kullanıcı testi

Arayüz konseptinizi doğrulamak için en iyi yöntem, kısa bir süre içerisinde bunu gerçek kullanıcıların önüne koymaktır. Bitmiş ve cilalanmış bir ürüne sahip olana kadar beklemenize gerek yoktur. Hatta erkenden test sürecine girmek, hızlı bir şekilde işe yaramayan fikirleri ortaya çıkarması açısından size büyük miktarda vakit kazandırabilir. Tasarımlarınızı geliştirmede ne kadar çok vakit harcarsanız, bunlardan vazgeçmeniz ve yeniden başlamanız o kadar çok zor olacaktır. Neredeyse bittiğini düşündüğünüz bir konsept hakkında geribildirim almak, hızlı bir şekilde güncelleyebileceğiniz tıklanabilir bir prototip hakkında geri bildirim almaya kıyasla daha zordur.

Birkaç kişiden, prototipinizi kullanarak çeşitli görevler yapmalarını isteyin. Eğer tasarladığınız uygulama bir müzikçalar ise, onlardan bir şarkı çalmalarını isteyebilirsiniz. Eğer uygulamanın belirli işlevlerinden emin değilseniz, kullanıcının dikkatini, bir şarkı seçip daha sonra bunun yerine başka bir şarkı seçmek gibi, farklı unsurlara çekebilirsiniz. En dürüst geribildirmeyi alabilmek için kişileri, onlar prototiplerinizi kullanırken yönlendirmeyin.

Ayrıca mevcut uygulamalar üzerinde test oturumları yapabilirsiniz. Diğerlerinin, sizin uygulamanızda olmayan özelliklere ne kadar dikkat ettikleri sizi şaşırtabilir. Testlerden öğrendiğiniz unsurları en mümkün şekilde kendi uygulamanıza dahil edin.

Daha fazlasını öğrenin

Online birçok kaynak bulunmaktadır. Kullanabileceğiniz kaynaklardan bazıları şunlardır:

Online dergiler

- Konigi: konigi.com
- Smashing Magazine (UX tasarımı bölümü): uxdesign.smashingmagazine.com
- UX Magazine: uxmag.com
- UX Matters: uxmatters.com
- UX Mastery: uxmastery.com

Kitaplar

- Susan Weinschenk. **100 Things Every Designer Needs to Know About People.** Teknoloji ile etkileşime girerken insanların neden belirli şekilde davrandıklarını gösteren araştırma bulguları.
- Steve Krug. **Rocket Surgery Made Easy.** Kullanılabilirlik test oturumlarının nasıl yapılacağı yönünde bir kılavuz.
- **This is Service Design Thinking.** Hizmet tasarım iş akışının özellikleri ve teknikleri. thisisservicedesignthinking.com
- **Mobile Developer's Guide To The 5th Dimension.** Bu kılavuzun küçük kardeşi UX/UI tasarımına odaklanmaktadır. wip.org

Diğer kaynaklar

Nielsen Norman Group: nngroup.com

Interaction Design Foundation: interaction-design.org





Android

Ekosistem

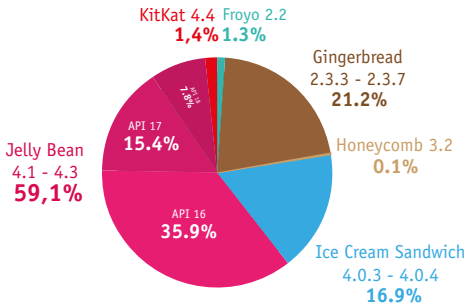
Android platformu, Google'ın liderliğini yaptığı Open Handset Alliance tarafından geliştirilmektedir ve kaynak kodları Kasım 2007'den beri kamuya açıktır. Donanım imalatçılarının çoğu tarafından kullanılması, platformu en hızlı büyüyen akıllı telefon işletim sistemi haline getirdi. 2013'ün dördüncü çeyreğinde satılan akıllı telefonların %91'den fazlası Android tabanlıydı!¹. Mayıs 2013'de düzenlenen Google I/O etkinliğinde Google, şu ana kadar 900 milyon Android cihazının etkinleştirildiğini² ve bunlar arasında tabletler, medya oynatıcıları, set üstü cihazlar, masa üstü telefonları ve taşıt eğlenceli sistemleri olduğunu duyurdu. Google'ın kendi akıllı gözlüğü Google Glass, Android'in hafif bir sürümüyle çalışıyor ve hem web hem de native uygulamaları destekliyor. Bazı Android içermeyen cihazlar da, daha az fonksiyona sahip olsalar da, Android uygulamalarını çalıştırabilmektedir. Bunlar arasında Android runtime desteğine sahip BlackBerry 10 işletim sistemi, yeni açık kaynak kaynak kodlu Sailfish³ ve kitle fonlaması desteğine sahip oyun konsolu Ouya bulunmaktadır.

- 1 www.gsmarena.com/android_worldwide_marketshare_crosses_80_for_the_first_time-news-7171.php
- 2 gigaom.com/2013/05/15/google-io-statshot-900-million-android-devices-activated/
- 3 sailfishos.org

Ocak 2014 itibariyle, Google Play'de 1.000.000'u aşkın mevcut uygulama bulunmaktadır.⁴

Android, platform için yazılmış uygulamaları çalıştırmak için Dalvik sanal makinesini ve önceden kurulu pek çok uygulamayı barındıran bir işletim sistemidir. Platform her altı ayda bir yeni özelliklerle sürekli güncellenerek hızlı bir şekilde büyümektedir; bu kılavuz kaleme alındığı sırada en güncel Android sürümü 4.4 (kod ismi "KitKat") idi. Kendisinden önce gelen sürümlerde olduğu gibi KitKat, getirdiği iyileştirmeler, ince ayarlar ve düzeltmelerle fazla görsele odaklanmadan daha çok arka tarafı ilgilendiren minör bir güncelleme olarak düşünülmüş. Kitkat ile telefonun depolama alanına erişmek için bazı yeni birleşik yollar, yeni bir yazdırma altyapısı ve cihaz sensörleri modu ekleyen Google, geliştiricilerin daha verimli ve tutarlı uygulamalar yazabilmesini hedefliyor. Bu ilavelerin yanı sıra ödeme ve bağliık programları bağlamında kullanılabilecek bir teknoloji olan NFC Host Kartı Emülasyonu eklendi. Son Android sürümüyle kızılötesi alıcı gözlere gelen yerel destek sayesinde, televizyonlar, set üstü ve diğér çeşitli elektronik cihazlar için

4 www.appbrain.com/stats/number-of-android-apps



uzaktan kumanda uygulamaları gibi yaratıcı uygulamaların önü açılıyor.

Android için geliştirme yapılırken en fazla tartışılan konulardan bir tanesi sistemin parçalanmasıdır: Farklı imalatçılar tarafından farklı cihazların fazla miktarda bulunması ve platformun hızlı bir şekilde ilerlemesi, Android uygulamanızın her yerde çalışıp çalışmayacağı konusunda bir belirsizlik yaratmaktadır. Ayrıca, yalnızca çok az sayıda telefon ve tablet modeli (%1,1) en yeni OS sürümünü desteklemektedir. Yaptığınız uygulamanın tüm Android cihazlarına uyumlu olması için yazılım altyapısının Android 2.2 veya yukarısındaki⁵ tüm sürümleri desteklemesini sağlamalısınız.

Android uygulamalarının sağlam bir kullanıcı deneyimine ve tutarlı bir görünüme sahip olmasını teşvik etmek için, Google bir tasarım kılavuzu⁶ oluşturdu. Renk düzenleri, tasarım modelleri ve yaygın kullanım kalıplarının önemine değinen kılavuz, Android ekosistemi için uygulama yaratma süresince faydalanabileceğiniz mükemmel bir yön göstericidir.

Ön Gereksinimler

Android için ana programlama dili Java'dır. Ancak unutmayın, Java kütüphanelerinin yalnızca belirli bir alt setini desteklemektedir ve Android ile çalışmayan platforma özel birçok API'lar bulunmaktadır. "Neler ve Niçin" sorularınıza çevrimiçi olarak yanıt bulmak için Android'in Geliştirici Rehberine⁷ ve "Nasıl" sorularınıza yanıt bulmak için referans belgelerine⁸ bakabilirsiniz. Google ayrıca, yeni geliştiricilerin çeşitli iyi

⁵ developer.android.com/resources/dashboard/platform-versions.html

⁶ developer.android.com/design

⁷ developer.android.com/guide

⁸ developer.android.com/reference

pratikleri öğrenebilmelerine yardımcı olmak için "Android Eğitimleri"⁹ isimli belgelendirme kaynağını sunmaktadır. Bu kaynak, uygulama navigasyonu ve uygulamalar arası iletişim gibi temel konuların yanı sıra, akıllı Bitmap indirmeleri ve daha iyi şarj ömrü için uygulamanızı optimize etmek gibi gelişmiş özellikler hakkında da bilgi edinmenizi sağlayan bir bölüm.

Geliştirmeye başlamak için Windows, Mac OS X ve Linux için mevcut olan Android SDK'ya¹⁰ ihtiyacınız var. SDK içinde uygulamaları oluşturmaya, test etmeye, uygulamadaki hataları ayıklamaya ve analiz etmeye yarayan araçlar bulunuyor. Android Development Tools (ADT)¹¹, IDE'ler ile entegrasyondan sorumludur ve geliştirme sürecinizin olabildiğince rahat olmasını sağlamaya yarar.

IDE desteği

Bugün Google, IntelliJ temelli "Android Studio" ve Eclipse ("Eclipse + ADT Eklentisi olarak bilinir") gibi önceden yapılandırılmış IDE'leri Android Developer Tools ile etkin bir şekilde bir arada sunuyor. Bu araçları kullanmanız, haricen SDK kurulumuna gerek bırakmadığı için vakit de kazandırır ve Android'e daha uygun olan bir deneyim sunar. Örneğin Android Studio doğrudan Gradle desteği ile gelmektedir. Bu yeni IDE ile string ve renkler gibi uygulama kaynaklarını referans numaralarından faydalanarak daha geliştirirken kod satırında görebilirsiniz.

Yakın bir gelecekte Eclipse'in yerine geçmesi hedeflenen yeni geliştirme ortamı Android Studio henüz **beta** aşamasında olduğu için bazı hataları içermektedir; eğer bu sizin için

⁹ developer.android.com/training/index.html

¹⁰ developer.android.com/sdk

¹¹ developer.android.com/tools/sdk/eclipse-adt.html



herhangi bir sorun yaratırsa, Eclipse ile devam etmeniz daha uygun olacaktır.

IDE	Eklenti desteği	Paketlenmiş sürüm
Eclipse	Ayn ADT paketi	Eclipse + ADT Eklentisi
IntelliJ	Ayn Android eklentisi	Android Studio

Daha fazla bilgi ve gerekli indirmeler, Android belgelerindeki "Araçlar"¹² bölümünde bulunabilir.

Native geliştirme

Android NDK¹³, uygulamanız için native bileşenlerin yazılmasını mümkün kılar. NDK, bunu native metotların çağrıları için JNI'dan ve native olmayan metotları callback olarak sunan native alt sınıflardan aldığı güçle sağlar. Daha verimli veri işlemeye ihtiyaç duyan kişiler ve oyun geliştiricileri için önemli bir husustur.

Gerçekleme (İmplementasyon)

Uygulama mimarisi

Android uygulamaları genellikle Activity'ler, Service'ler, BroadcastReceiver'lar ve veri sağlayıcılarının karışımından oluşur ve bunların tümünün uygulamanın manifest dosyasında beyan edilmesi gerekir.

Activity, bağlı olduğu kullanıcı arayüzü ile birlikte

¹² developer.android.com/tools

¹³ developer.android.com/tools/sdk/ndk

uygulama işlevlerinden bir parça sunar. Service, arka planda çalışan görevler için kullanılır ve bu yüzden doğrudan görsel sunumun bir parçası değildir. Broadcast Receiver, sistemin, diğer uygulamaların ya da kendi uygulamanızın yayınladığı mesajları yönetmeye yarar. Provider yani veri sağlayıcısı ise alt kısımlardaki depolama mekanizmalarını (örneğin SQLite) soyutlayarak, bir uygulamanın içeriğine arayüz sağlayan bileşendir.

Bir uygulama çok sayıda bileşenden oluşabilir. Örnek vermek gerekirse, arayüz için bir Activity'yi ve uzun süren görevler içinse bir Service'i kullanabilirsiniz. Bileşenler arasındaki iletişim Intent'ler ya da Android Arayüz Tanım Dili (AIDL) ile yönetilen uzak prosedür çağrılarıyla (RPC) sağlanır.

Intent'ler, bir eylem ile birlikte kullanıcı konumu veya bir URL gibi verileri paket halinde tutar. Bu Intent'ler, platformda belli davranışları tetikleyebilir ve isterseniz Intent'leri uygulamanızın içinde bir mesaj sistemi gibi kullanılabiliyorsunuz. Örnek olarak, bir web sayfasını gösterme Intent'i web tarayıcısını açacaktır. Bu temel ilkenin güçlü yönü, herhangi bir işlevi gerçekleştirme işlemini başka bir uygulamanın üstlenmesini sağlayabilmesidir; Android sistemi, belli bir Intent için daima tercih edilen uygulamayı kullanır. Örnek vermek gerekirse, bir haber okuyucusu uygulaması tarafından tetiklenip açılmış bir web sayfasının "paylaş" Intent'i ile bir e-posta istemcisi açılabilir veya kullanıcının tercihlerine göre ve yüklenen uygulamaya göre bir metin mesajlaşma uygulaması tetiklenebilir. Paylaşma Intent'ini beyan eden herhangi bir uygulama bu durumda kullanılabilir.

Bir uygulamanın kullanıcı arayüzü, Android'e özel XML yerleşim dosyalarıyla Java kodundan ayrı bir yerde yazılır. Farklı ekran boyutları, farklı diller ve cihaz özellikleri için Java koduna dokunmadan farklı yerleşim dosyaları oluşturulabilir. Bu amaç doğrultusunda, yerelleştirilmiş stringler ve görseller, ayrı kaynak klasörlerinde organize edilir. Tabii ki, Java kodu

içerisinde de yerleşim tanımlayıp tasarlayabilirsiniz veya dinamik UI güncellemelerine fırsat vermek için her iki stratejiyi bir arada kullanabilirsiniz.

SDK ve Eklentiler

Geliştirmenize yardımcı olması için SDK içerisinde birçok araç bulunur. Bunlardan en önemlileri:

- **android:** Yeni bir proje oluşturmak veya sanal cihazları ve SDK sürümlerini yönetmek için kullanılır.
- **adb (Android Debug Bridge):** Cihazları sorgulamak, bağlamak ve cihazlara (sanal cihazlar dahil) dosya taşımak, uygulama kurmak ve bunun gibi etkileşimleri gerçekleştirmek için kullanılır.
- **emulator:** Bir sanal cihazın tanımlanmış özelliklerini taklit etmek için kullanılır. Başlaması için biraz beklemek gerekir, bu yüzden her yeni proje derlemesinde başlatmak yerine bir kere başlatıp kullanın.
- **ddms:** Cihazınızın veya emülatörünüzün içerisine bakmak, kayıt mesajlarını izlemek, ağ gecikmesi ve GPS pozisyonu gibi emülatör özelliklerini kontrol etmek için kullanılır. Hafıza tüketimini izlemek ve çalışan görevleri kapatmak için de kullanılabilir. Eğer bu araç çalışıyorsa, isterseniz Eclipse hata ayıklayıcısına emülatörde çalışan bir görevi bağlayabilirsiniz. Bunun ötesinde ddms, 4.0 altındaki Android sürümlerinde ekran resmi yaratmak için (root yetkisi olmadan) tek yoldur.

Buradaki dört araçtan daha fazlasını (çalışan metotların izlerini gösteren, arayüzü didik didik edip her türlü bilgisini veren, rastgele olaylarla uygulamanızı test eden araçlar) SDK'nin tools/ dizininde bulabilirsiniz.

IDE eklentileri, bu dosyaların hepsinin yönetilmesine yardımcı olmak için vardır. IntelliJ'in 11.x sürümü, bir görsel

tasarım editörü içermektedir; böylece arayüz bileşenlerini editöre sürükleyerek hızlı prototipleme yapmak istediğinizde Eclipse ya da IntelliJ arasından istediğinizi seçip kullanabilirsiniz.

Uygulamanızla ilgili fırlayan hatalar (exception) gibi sorunlarla karşılaşıyorsanız, DDMS kayıtlarını kontrol etmeyi veya Logcat mekanizmasını kullanmayı unutmayın. Uygulamanın düzgün çalışması için gerekli bütün izinleri koymayı unutup unutmadığınızı kontrol etmenizi sağlarlar. Örnek vermek gerekirse, uses-permission elementi¹⁴ içerisinde beyan etmeniz gereken `android.permission.INTERNET` iznini unuttuğunuzda bununla ilgili kaydı görebilirsiniz.

Geniş ekranlı cihazlar için Android 2.3'den sonra sunulan Fragment¹⁵ gibi özellikleri kullanıyorsanız, Google'ın sunduğu Android Uyumluluk Destekleme paketini projenize eklediğinizden emin olun. Bu pakete SDK içinden ulaşabilirsiniz ve Android Sanal Cihaz (AVD) Yöneticisi de kullanarak, Android 1.6'dan¹⁶ Android 2.3'e kadar uygulamanızı herhangi bir kullanım sorunu oluşturmada Android 3.0+ özellikleriyle geliştirebilirsiniz. En geniş geriye dönük sürüm desteğini sunmak için uygulamalarınızda v4 destekleme paketlerini kullandığınızdan emin olun. Ayrıca Android 2.1 ve yukarısı için ActionBar kullanmanızı sağlayan, çevrimiçi¹⁷ olarak belgelenmiş, ActionBar desenini gerçekleştirme yolunu sağlayan bir v7 appcompat destekleme kütüphanesi de bulunmaktadır.

Android 3.1+ için uygulamanızı geliştirirken, ana ekran widget'larınızın boyutlarını değiştirebilmenize, USB üzerinden dijital kameralar, oyun çubukları ve diğer cihazlara

¹⁴ developer.android.com/reference/android/Manifest.permission.html

¹⁵ developer.android.com/guide/topics/fundamentals/fragments.html

¹⁶ android-developers.blogspot.com/2011/03/fragments-for-all.html

¹⁷ developer.android.com/tools/support-library/features.html

bağlanabilmenize olanak sağlayacaktır. Android 4.x sürümleri, genişletilebilen bildirimler, kilit ekranı widget'leri ve yüz algılamasına sahip bir kamera gibi ilginç özellikleri beraberinde getiriyor. Native hesaplama (computing) altyapısı olan Renderscript (3.1'de tanıtıldı), büyük ölçüde değiştirilmiştir ve artık doğrudan grafik işleme becerileri sunmamaktadır ancak bunun yerine ağır veri işleme işlemleri için kullanılabilir.

Daha eski Android sürümlerine sahip cihazlara yönelik geriye dönük uyumluluk sağlamak için Google, Play Store üzerinden güncellenen ve Google Maps gibi en yeni kütüphaneleri ekleyen Google Play Services altyapısını¹⁸ kullanmaya başladı. Eğer kullanıcı kimliğini doğrulamayı amaçlıyorsanız, uygulamanıza gerçek kullanıcı verilerinin faydalarını sunacak Google+ Sign'in yeteneklerine bir göz atmak isteyebilirsiniz. Bu işlevsellik kullanıcı adına Google hesabının kullanımına izin veren OAuth 2.0 erişim jetonları aracılığıyla yönetiliyor.

Test Etme

Bir uygulamayı test etmenin ilk adımı, onu bir emülatör veya gerçek cihaz üzerinde çalıştırmaktır. Daha sonra gerekirse DDMS aracıyla hata ayıklama yapabilirsiniz.

Android işletim sisteminin bütün sürümleri, herhangi bir modifikasyon olmadan cihazlar üzerinde çalışacak şekilde yaratılmış olsa da bazı donanım üreticileri, platformun parçalarını değiştirmiş olabilirler. Bu yüzden, çeşitli cihazlar üzerinde test etmek çok önemlidir. Hangi cihazların en popüler olduğunu görmek için AppBrain'in listesine¹⁹ bakın.

Testi otomatik hale getirmek için Android SDK, kullanışlı

¹⁸ developer.android.com/google/play-services/

¹⁹ www.appbrain.com/stats/top-android-phones

ve yetenekli test enstrüman²⁰ araçları ile gelmektedir. Testler, SDK'da bulunan Android mock nesneleri kullanılarak standart JUnit biçimi kullanılarak yazılabilir.

Instrumentation sınıfları UI'ı gözlemleyebilir ve tuşa basmak gibi sistem olaylarını gönderebilir. Test metodlarınız bu olaylar gerçekleştiikten sonra uygulamanızın durumunu kontrol edebilir. MonkeyRunner²¹, bütün uygulamayı test etmek için kullanılan güçlü ve genişletilebilir bir test otomasyon aracıdır. Bu testler hem sanal, hem de fiziksel cihazlarda çalıştırılabilir.

SDK'nin 21. sürümünde Google, Android Jelly Bean ve yukarısında fonksiyonel UI testleri yapmaya yarayan, daha verimli bir UI test otomasyon altyapısını²² sunmuştur.

Bu araç, `uiautomatorviewer` komutuyla kabukta (shell) çalıştırılabilir ve gösterilen View'lar hakkında bazı bilgiler de dahil olmak üzere o anki arayüzü yakalayıp size gösterir.

Testlerin gerçekleştirilmesi nispeten kolaydır: Testlerinizi yazdıktan sonra Ant inşa sistemi yardımıyla bir JAR dosyası olarak testinizi oluşturursunuz. Daha sonra bu dosyanın cihazınıza yüklenmesi ve `adb shell uiautomator runtest` komutuyla çalıştırılması gerekecektir.

Ekim 2013'te Espresso²³ adında yeni bir araç Google tarafından piyasaya sürüldü. Bu araç kullanıcı arayüzünüzü test etmek için hızlıca prosedürel testler yazmanızı sağlayan, kullanımı kolay bir API sunmaktadır.

Robotium²⁴ gibi açık kaynaklı test altyapıları, diğer otomatikleştirilmiş testlerinizi tamamlar. Hatta Robotium, uygulamanın kaynak kodu mevcut olmasa bile APK dosyaları

²⁰ developer.android.com/guide/topics/testing/testing_android.html

²¹ developer.android.com/guide/developing/tools/monkeyrunner_concepts.html

²² android-developers.blogspot.de/2012/11/android-sdk-tools-revision-21.html

²³ googletesting.blogspot.de/2013/10/espresso-for-android-is-here.html

²⁴ code.google.com/p/robotium

üzerinden test edilmesine olanak sağlar. Roboelectric²⁵, standart/masaüstü JVM içerisindeki IDE'nizde testleri doğrudan gerçekleştirebilen başka bir mükemmel araçtır.

Otomatikleştirilmiş testleriniz, Jenkins veya Hudson gibi sürekli entegrasyon sunucuları üzerinde çalıştırılabilir. Roboelectric, standart bir JVM üzerinde çalışır ve Android'in çalışma zamanı (runtime) ortamına ortamına ihtiyaç duymaz. Robotium dahil diğer otomatikleştirilmiş test altyapıları, Android'in Instrumentation sınıflarını baz alır ve Dalvik JVM içerisinde çalıştırılmaları gereklidir. Android Emulator Plugin²⁶ gibi eklentiler, bu testlerin Hudson ve Jenkins içerisinde yapılandırılmasına ve çalıştırılabilmesine fırsat vermektedir.

İnşa Etme

Seçtiğiniz IDE içerisinde uygulamanızı doğrudan inşa etmeniz yanı sıra Android uygulamaları oluşturmak için daha rahat yöntemler bulunmaktadır. Gradle²⁷ artık, Android için resmi olarak desteklenen bir inşa otomasyon aracıdır. Bunun yanı sıra, topluluk tarafından desteklenen bir maven eklentisi de²⁸ bulunmaktadır. Her iki araç da, farklı Maven depolarından (Örn: Maven Central Repository²⁹) gelen bağımlılıkları kullanabilir.

Google, kütüphaneleri Gradle için Android arşiv (.aar) dosyası şeklinde taşımaktadır. Bu dosyalar Android SDK Yöneticisi tarafından elde edilebilir. Gradle için "android-library" eklentisini kullanarak kendi kütüphanelerinizi veya SDK'larınızı paketleyebilme fırsatına sahipsiniz. Gradle dostu

²⁵ robolectric.org/

²⁶ wiki.hudson-ci.org/display/HUDSON/Android+Emulator+Plugin

²⁷ tools.android.com/tech-docs/new-build-system

²⁸ code.google.com/p/maven-android-plugin/

²⁹ www.maven.org

Android kütüphanelerini bulmak için mükemmel bir kaynak olan "Gradle, please"³⁰ sayfasına bakabilirsiniz.

İmzalamak

Uygulamalarınız daima inşa süreci (build) sırasında, debug ya da release imzası ile imzalanır. İmzalama ücretlerinden (ve güvenliğinden) kurtulmak için kendi imzalama mekanizmanızı kullanabilirsiniz.

Aynı imza, uygulamanızın güncellemeleri için de kullanılmak zorundadır – bu yüzden keystore dosyanızı veya şifrenizi kesinlikle kaybetmemelisiniz. Unutmayın: bütün uygulamalarınız için aynı anahtarı kullanabilirsiniz veya her yeni uygulama için yeni bir tane oluşturabilirsiniz.

Dağıtım

En yeni ve mükemmel uygulamayı yarattıktan ve test ettikten sonra, onu "Play" ismindeki Android'in uygulama mağazasına koymanız gerekir. Bu, müşterilere ulaşmak ve uygulamalarınızı satmak için iyi bir yerdir. Android 1.6 ve üst sürümleri, Google Wallet vasıtasıyla uygulama içi satışları da desteklemektedir. Bu, ekstra içerik, özellik setleri ve benzerlerini Android Play³¹ yapısını kullanarak uygulama içinden satabilmenizi sağlar. Ayrıca uygulama metadata'sı için diğer uygulama portalları tarafından kullanılan bir kaynak olarak kullanılmaktadır. Uygulamanızı Android Play'e yüklemek için play.google.com/apps/publish/ adresine gidin.

Google Alışveriş hesabınız ile hizmete üye olmanız ve 25 dolarlık aylık ücreti ödemeniz gereklidir. Üyeliliğiniz

³⁰ gradleplease.appspot.com

³¹ developer.android.com/guide/google/play/billing/

onaylandıktan sonra uygulamanızı yükleyebilir, ekran görüntüleri ve tanımlar ekleyebilir ve daha sonra uygulamayı yayınlatabilirsiniz.

Uygulamanızın `AndroidManifest.xml` dosyası içerisine `versionName`, `versionCode`, uygulama simgesi ve bir uygulama adı etiketi eklediğinizden emin olun. Manifest dosyası içinde beyan edilen özellikler (uses-feature kısımları), farklı cihazlar için uygulamaları filtrelemek için kullanılmaktadır.

Google Play mağazasında yapılan en yeni değişikliklerden bir tanesi de ürünün aşamalı olarak görücüye çıkmasına ilave olarak alfa ve beta testleridir. Bu sayede uygulamayı bütün kullanıcılara dağıtmadan önce dostça bir kullanıcı testi yapmanız mümkün olur. Ayrıca, Geliştirici Konsolunda doğru seçenekleri ayarlayarak sadece belirli cihazları hedefleyebilir ve kullanıcı tabanınızı anlamanıza yardımcı olacak ayrıntılı istatistikleri çıkartabilirsiniz.

Yerleşik yerelleştirme hizmetini kullanarak küçük bir ücret karşılığında uygulamanıza yeni dilleri kolaylıkla ekleyebilirsiniz – bu konunun önemi hakkında ayrıntılı bilgi için Yerelleştirme Kontrol Listesini³² kontrol ettiğinizden emin olun.

Android Play’de birbirine rakip olan birçok uygulama bulunmaktadır. Alternatif uygulama mağazalarını³³ denemek isteyebilirsiniz. Farklı ödeme yöntemleri sunarlar ve belirli tüketici gruplarını hedefleyebilirler. Bu mağazalardan bir tanesi, Kindle Fire tablet ailesi üzerinde yüklü gelen Amazon Appstore’dur.

³² developer.android.com/distribute/googleplay/publish/localizing.html

³³ onpfp.org/appstores/

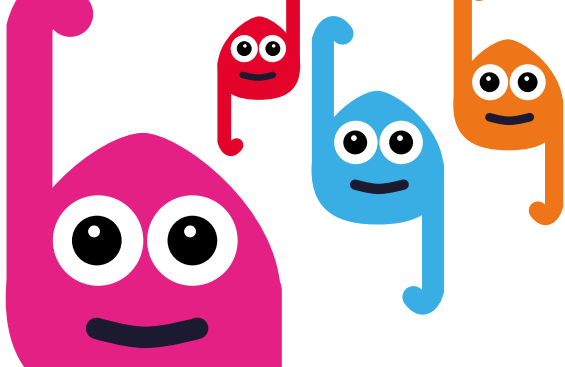
Uyum

Android'in uyumu arttıkça satıcıya özel ekosistem de kendi SDK'larıyla, tamamen özelleştirilmiş Android sürümleriyle ve alfa ve beta testi gibi unsurları içeren araçlarıyla gelişti. Bunun bir avantajı, kullanıcılar için mükemmel bir deneyim sunan çok sıkı bir entegrasyon oldu ancak dezavantajı da, ekosistemin artan ölçüde bölünmesidir. Satıcıya özel mağazalar genellikle, kendilerinden başka araçlar kullanan genel uygulamaların yüklenmesine izin vermemektedirler.

Bunların bir örneği, temel olarak Android'in özelleştirilmiş bir türevi olan ve Android tableti en büyük pazar payı ile temsil eden Amazon'un Kindle Fire ekosistemidir: Uygulama için satın alımları veya haritaları etkinleştirmek için Google'ın Play hizmetlerini kullanmak yerine benzer işlevsellik sunan Amazon'un kendi kütüphanelerini kullanmanız gerekir. Bunun arkasında mantık basittir: Kindle cihazları, Google'ın hizmetlerini çalıştırmak için gerekli olan kütüphanelere sahip değildir. Amazon ayrıca, kendi kitlenizi hedeflemenize yardımcı olan kendi reklam ve oyun hizmetlerini sunmaktadır. Dört farklı cihaz için (1. Jen, 2. Jen, HD 7" ve HD 8.9") emülatörler sunan Amazon, gerçekçi bir sanallaştırma ortamı sunarak uygulamanızı mükemmel hale getirmenize yardımcı olur. Kendi geliştirici toplulukları için sunduğu test fırsatının yanı sıra Amazon, kendi Appstore'larına yüklenen herhangi bir uygulamaya inceleme yapabiliyor.

Doğru kaynakları bulmanıza yardımcı olabilecek birkaç unsur aşağıda görülebilir:





Satıcı	Belgelendirme
Amazon	developer.amazon.com/sdk/fire.html
HTC	htcdev.com
LG	developer.lge.com
Motorola	developer.motorolasolutions.com/ community/android
Samsung	developer.samsung.com/android
Sony	developer.sonymobile.com

İlginç bir şekilde, gitgide daha fazla satıcı (Samsung ve HTC gibi) "Google Play Edition" ismindeki cihazlarının eklentisiz Android sürümlerini sunmaya başladılar. Bu cihazlar normal modellere kıyasla aynı donanımı kullanırlar ancak herhangi bir yazılım kişiselleştirmesi ile sunulmamaktadırlar. Bu cihazlar doğrudan Google'ın Play Store'u aracılığı ile dağıtılırlar ve Google deneyimine bağlı kalmak isteyen kullanıcılara en modern cihazları sunarlar.

Gelir Kaynağı Oluşturma

Bir uygulamayı, mevcut uygulama mağazalarından birinde satmanın yanı sıra, üzerinden gelir kaynağı oluşturmak için birçok farklı yol vardır. Bunlardan bir tanesi, tıklamaya veya görüntülenmeye dayalı reklam olabilir ve bu düzenli bir gelir sağlayabilir. Bundan başka, Google Play Store'u kullanan Google'ın kendi hizmeti³⁴ veya PayPal'ın Mobile SDK³⁵ ve Mobil Ödeme Kütüphanesi³⁶ gibi Uygulama içi Faturalandırma olanakları bulunmaktadır. Çoğu hizmetler, işlem bazlı ücretler ve sundukları paket üyelik olanakları, paralel ödemeler veya önceden onaylanmış ödemelerle ilgili farklılık gösterebilir. Eğer uygulamanız için oldukça etkili şeyler kullanmak istiyorsanız, kamera etkileşimli kredi kartı tarama için card.io'nun SDK'sını³⁷ kullanmak isteyebilirsiniz.

Samsung'un Apps'i veya Amazon'un Appstore'u gibi satıcıya özel ekosistemler için optimize edilmiş gelir kaynağı fırsatlarından yararlanmak amacıyla onların SDK'larını kullanmanız gerekebilir.

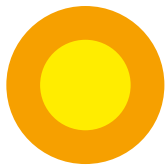
Ödeme yönteminizin, uygulamanızı yayınlamak istediğiniz farklı mağazaların şartları ve koşulları ile uyumluluk gösterdiğinden emin olun. Özellikle, farklı kuralların mevcut olduğu dijital indirmelere bakmak isteyebilirsiniz.

³⁴ <http://developer.android.com/google/play/billing/>

³⁵ github.com/paypal/PayPal-Android-SDK

³⁶ developer.paypal.com/webapps/developer/docs/classic/mobile/gs_MPL/

³⁷ card.io





BlackBerry Java Uygulamaları

Ekosistem

BlackBerry platformu, Kanadalı şirket Research In Motion (günümüzdeki adıyla BlackBerry Ltd.) tarafından geliştirildi ve 1999'da piyasaya sunuldu. BlackBerry cihazları, rahat metin girişi için tam klavye ile sunulduğu, uzun şarj ömrüne, özgün mobil sosyal ağı olan BlackBerry Messenger'a (BBM), e-posta ve güçlü bir anlık bildirim hizmetine (push notification) sahip olduğu için hızlı bir şekilde popüler hale geldi.

Öte yandan son 5 yıl içinde birkaç özel ülke pazarı dışında (Nijerya, Endonezya ve Güney Afrika), BlackBerry telefonlarının genel piyasa payı son yıllarda sürekli olarak geriledi. 2013 yılının 3. çeyreğinde BlackBerry cihazları, küresel akıllı telefon sevkiyatının yalnızca %2'sini oluşturmaktaydı¹. Kaybettiği pazar payını geri alabilmek için RIM, 2012 yılında tamamen yeni bir işletim sistemini tanıttı: BlackBerry 10 (BB10). 2013 yılından itibaren de üretilen tüm yeni cihazlarda BB10'i kullandılar. Bu yeni işletim sistemi hakkında daha fazla bilgi almak için kitapçımızdaki BB10 bölümüne göz atabilirsiniz.

Bu bölüm, 2013 yılından önce piyasaya sürülen ve Java tabanlı olan BlackBerry cihazları için uygulama geliştirmeye odaklanmıştır. Gelecekte aşamalı olarak kaldırılacak olsa da, BlackBerry Java API, eski BlackBerry cihazları için uygulama geliştirmede kullanılan en yaygın yöntemdir. Bu yüzden bu bölüm buna odaklanmıştır.

1 bgr.com/2013/10/30/blackberry-market-share-q3-2013/

Gerekli Araçlar

İlk önce Java SDK'sını² indirin. Daha sonra Eclipse'e ve BlackBerry eklentisine³ ihtiyacınız olacak. Bunlar ayrı ayrı ya da son yayınlanan BlackBerry OS için gerekli olan SDK ve simülatörleri içeren uygun bir paket olarak indirilebilir.

Kitabımızın bu kısmı kaleme alındığı sırada Java tabanlı BlackBerry OS'un son yayınlanan sürümü 7.1 idi. BlackBerry OS 6 ve daha eski cihazlar için SDK'ları indirme talimatları indirme sayfasında mevcuttur. Ekstra cihaz simülatörleri RIM'in web sayfasında⁴ bulunabilir.

Uygulama paketinizi bir cihaz üzerinde test etmek için BlackBerry Masaüstü Yöneticisi'ni⁵ indirip kurmanız gerekir. Ayrıca daha hızlı kullanım için, JDE ile beraber gelen javaloader adlı aracı kullanabilirsiniz.

Uygulama

BlackBerry JDE, kısmen JAVA ME ve onun bazı popüler JSR uzantılarını baz alır. Bu, BlackBerry uygulamalarının yalnızca Java ME teknolojileri kullanarak oluşturulabildiği anlamına gelmektedir. Diğer bir seçenek ise, platformu bütünüyle kullanmanıza fırsat veren BlackBerry'nin tescilli uzantılarını ve arayüz geliştirme çatılarını (framework) kullanmaktır. Her iki API'nı da JavaDoc'u online⁶ mevcuttur.

2 oracle.com/technetwork/java

3 us.blackberry.com/developers/javaappdev/javaplugin.jsp

4 us.blackberry.com/sites/developers/resources/simulators.html

5 us.blackberry.com/apps-software/desktop/

6 blackberry.com/developers/docs/7.0.0api/index.html



Native UI bileşenleri belli bir dereceye kadar biçimlendirilebilir ancak görünümlerini mevcut temadan alırlar. Bu, her bileşen için kod içerisinde `Field.applyTheme()` methodunu geçersiz kılarak önlenebilir.

OpenGL-ES'den ana ekran etkileşimi ve kriptografiye kadar BlackBerry API'ları, ilgi uyandıran uygulamalar yaratmanız için ihtiyaç duyduğunuz her şeyi size sağlayacaktır. Resmi BlackBerry araçlarına ek olarak, uygulamalarınızı geliştirmenize fırsat veren üçüncü parti uzantıları bulunmaktadır. Bunlar arasında J2ME Polish⁷ veya CSS kullanarak kullanıcı arayüzünüzü (UI) tasarlayıp canlandırmanızı sağlayan Glaze⁸ bulunur.

Hizmetler

BlackBerry, reklam, harita, ödeme ve anlık bildirim (push notification) hizmetleri⁹ dahil olmak üzere uygulamalarınızı geliştirmek için yararlı olabilecek birçok hizmet sunmaktadır.

Anlık bildirim (push notification) hizmeti¹⁰, temel olarak posta, mesaj ve haber uygulamalarında kullanışlıdır. Ana faydası, cihaz sürekli olarak ağı yoklamak yerine, ağın güncellemeleri anlık bildirim olarak iletmesini bekler. Bu ağ trafiğini, pil sarfiyatını ve maliyeti azaltır. BlackBerry anlık bildirim hizmetini (push notification) ücretsiz ama sınırlı bir hizmet olarak sunmaktadır; limiti aşan sayıda anlık bildirim (push notification) göndermek isterseniz ücretli altyapı hizmetinden faydalanabilirsiniz.

⁷ j2mepolish.org

⁸ glaze-ui.org

⁹ developer.blackberry.com/services/#platform

¹⁰ us.blackberry.com/developers/platform/pushapi.jsp

Uygulama Taşıma

BlackBerry cihazları arasında uygulamaları taşımak çok kolaydır çünkü işletim sistemi (OS), parçalanma sorunlarını en aza indirmek için dikkatle çalışan tek bir şirket tarafından yaratılmıştır. Ancak bu, ortaya çıkabilecek tüm zorlukları ortadan kaldıramamaktadır:

- Bazı sınıflar (FilePicker gibi) ve fonksiyonlar yalnızca belirli işletim sistemi (OS) sürümlerinde mevcuttur.
- Farklı ekran çözünürlüklerini ve görünüm modlarını (enine-boyuna görünüm) desteklemeniz gerekir.
- Dokunmatik ve dokunmatik olmayan cihazları desteklemeniz ve hedef cihaza göre değişen çeşitli özellikleri (Storm'un fiziksel olarak tıklanabilir özel dokunmatik ekranı gibi) desteklemeniz gerekebilir.

Java ME ve Android gibi diğer Java platformlarına uygulamalar taşımak, BlackBerry UI'sini taşımak mümkün olmadığı için karmaşıktır. Platformlar arası taşınabilirlik, uygulamanızın BlackBerry bileşenlerini ve API'larını ne sıklıkla kullandığına göre değişir. Örneğin, BlackBerry'nin anlık bildirim (push notification) sınıflarını diğer platformlarda kullanmak mümkün değildir.

Test Süreçleri

BlackBerry, çeşitli telefonlar için simülatörler sunmaktadır. BlackBerry'nin hata ayıklama ve test araçları, Android ve iOS gibi diğer modern platformlarla eşdeğerdir: simülatörler, geliştiricilerin çok çeşitli olayları (gelen aramalar, GPS koordinatlarındaki değişiklikler, ağ durumlarındaki değişiklikler, vb.) cihaz üzerinde hata ayıklayarak simüle etmelerini

sağlayarak, kodlarını gerçek donanımlarda test etmelerini kolaylaştırır.

Buna ek olarak, otomatik test de mümkündür ancak sınırlı ve karmaşıktır. Olayları bilgisayarınızdan aktarmak için paket FledgeController aracını¹¹, cihaz (veya simülatör) üzerinde çalışan bir BlackBerry uygulamasından aktarmak içinse EventInjector class'ı¹² kullanabilirsiniz. Bu konu hakkında çok az sayıda kaynak olduğundan, üzerinde bolca düşünmeniz ve korsanlık yapmanız gerekebilir.

İmzalama

Platformun (ağ veya dosya API'ları) birçok ciddi güvenlik sınırı ve özelliği, uygulamanın yayıncısının tanınacağı bir şekilde imzalanmasını gerektirmektedir. Bunu yapabilmek için BlackBerry'den¹³ bir imza anahtarı edinmeniz gerekir. İmzalama işlemi, uygulamayı paketleyebilen rapc aracı kullanılarak yapılır.

Dağıtım

BlackBerry'nin kendi dağıtım kanalı, uygulamalarınızı yayınlatabileceğiniz AppWorld'tür¹⁴. Paralı uygulamalar için %70 gelir payı alırsınız. Ayrıca GetJar da¹⁵, BlackBerry

¹¹ docs.blackberry.com/en/developers/deliverables/15476/Using_the_BB_Smrtphn_Smltr_programmatically_607582_11.jsp

¹² www.blackberry.com/developers/docs/4.1api/net/rim/device/api/system/EventInjector.html

¹³ <https://www.blackberry.com/SignedKeys/codesigning.html>

¹⁴ appworld.blackberry.com

¹⁵ getjar.com

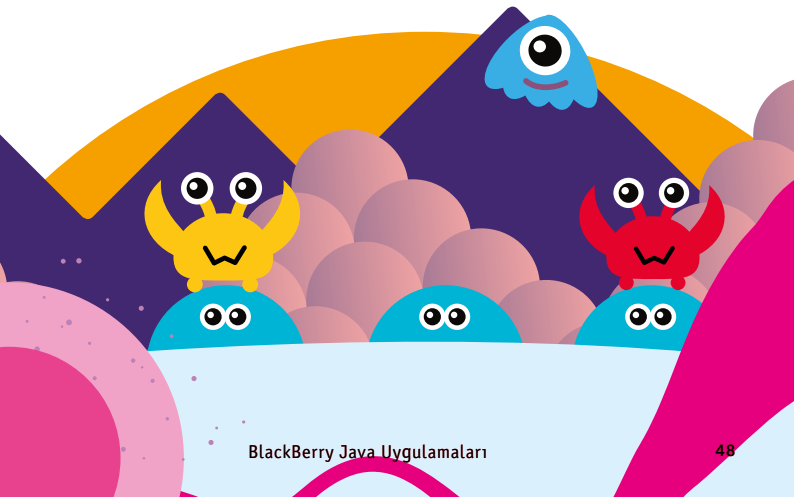
uygulamalarını yayınlayan bağımsız ve popüler bir web sitesidir.

Daha fazla bilgi

Eğer BlackBerry Java geliştiriciliği hakkında daha fazla bilgi almak istiyorsanız, aşağıdaki kaynaklar size yardımcı olabilir.

Paketli örnek uygulamalar

SDK'lar, mükemmel örnek uygulamalar ile beraber gelmektedirler. Bunlar arasında "Yeni Başlayanlar" uygulamalarından karmaşık konum belirleme ve çokluortam uygulamalarına kadar her şey bulunmaktadır.



Online

Mevcut birkaç önemli online kaynak:

- Resmi BlackBerry belgelendirme sitesi¹⁶
- BlackBerry geliştirici forumları¹⁷

İnternet'te BlackBerry OS ile ilgili çeşitli sorunlara ve konulara resmi kaynaklardan çok daha iyi şekilde değinen tonlarca bilgi bulunmaktadır. Arama motorları sizin dostunuzdur!

Kitaplar

BlackBerry Java geliştiriciliği ile ilgili kitaplar:

- **BlackBerry Development Fundamentals**¹⁸ by John Wargo
- **Beginning BlackBerry 7 Development 2nd Edition** by Anthony Rizk
- **Advanced BlackBerry 6 Development 2nd Edition** by Chris King

¹⁶ developer.blackberry.com/java/documentation/

¹⁷ supportforums.blackberry.com/t5/Java-Development/bd-p/java_dev

¹⁸ bbdevfundamentals.com/



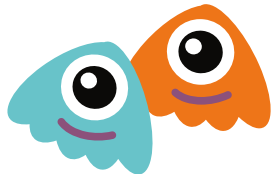
BlackBerry 10

Ekosistem

BlackBerry 10 platformu (BB10), BlackBerry'nin (önceden RIM ismiyle bilinen şirket) iddialı ve modern bir işletim sistemiyle yeniden piyasaya çıkışını sağladı. BlackBerry bu çıkışı, rakip mobil işletim sistemleri ile aynı seviyeye gelebilmek için gerçekleştirdi.

BB10 cihazları piyasaya 2013 yılının ilk çeyreğinde çıkmıştır. Eski nesil BlackBerry OS telefonlar için (OS 6 ve 7.x) herhangi bir güncelleme planı bulunmamaktadır. Şu anda satışta beş farklı BB10 tabanlı telefona ulaşmak mümkün. Yeni amiral gemisi Z30'dur. Bunun yanı sıra Z10 ve fiziksel klavyeye sahip Q5, Q10, Q30 bulunmaktadır. Kitapçığının bu kısmı kaleme alınırken, farklı boyut ve biçimiyle öne çıkan BlackBerry Passport ve BlackBerry Classic de piyasaya çıkmaya hazırlanıyordu.

Önceki bölümde özetlendiği gibi, BlackBerry büyük bir piyasa baskısı altındadır. Daha fazla pay kaybetmemek için BB10'u çok başarılı bir hale getirmek zorundalar; bu yüzden yeniden piyasaya çıkış yolunda çok daha fazla yatırım yapmaları gerekiyor. Bunun anlamı, yeni platform için uygulama geliştirmek isteyen geliştiriciler için yeni ve ilginç fırsatlar yaratmaktır. İşletim sistemi tamamen yeni olsa da, gömülü cihazlar için gerçek zamanlı bir çekirdeğe sahip olan bir Unix türevi olan QNX'i temel almaktadır. BlackBerry World isimli mağaza veya push hizmeti gibi diğer BlackBerry ekosistem



unsurları deęiřmemiřtir. Kuruluřlarda BlackBerry iin byk bir kaynak olacak olan unsur, BlackBerry Enterprise Server (BES) isimli mobil ynetim yazılımı'dır.

En yeni BlackBerry SDK aracıysa 2014 yılının ikinci yarısında duyurulan 10.3 srmdr.

Geliřtirme

BB10 ile uygulamalar, ok eřitli yazılım teknolojileri kullanılarak geliřtirilebilir:

- C Native SDK
- C++ Cascades SDK
- HTML5 (WebWorks SDK)
- Adobe Air
- Android Runtime
- BlackBerry App Generator

BlackBerry firması geliřtiricileri yeni platforma ekmek iin bir simlatr, GitHub¹ zerinde birok rnek projeler ve srekli olarak gncellenen belgelendirme² de dahil olmak zere pek ok kaynak sunmaktadır.

Mevcut Java API'nı artık desteklenmemesi sebebiyle BlackBerry zc bir řekilde yoęun tepkiyle karřılařtı. Bu, daha eski, BB10 olmayan BlackBerry cihazları iin kod yazan Java geliřtiricilerinin yukarıda belirtilen teknolojileri yeniden ęrenmeleri anlamına geliyordu. Geliřtiricilerin tm bunu yapmak istemedięi iin birok geliřtiricinin "gemiye terk edip" rakip platformlar iin alıřacaęı konusunda endiře

¹ github.com/blackberry

² developer.blackberry.com/platforms/bb10



bulunmaktaydı. Öte yandan, mevcut nesil uygulamalar için herhangi bir geçiş yolu olmadığından geliştiricilerin yeni platform için bunları yeniden yazmaları gerekmektedir. Yeni BB10 platformu gerçek zamanlı gömülü bir işletim sistemi olan QNX³'i temel aldığı için bu gerekliydi. Diğer taraftan, yeni platform kendi uygulamalarını kolaylıkla taşıyabilecek olan web geliştiricileri ve özellikle Android geliştiricileri için yeni fırsatlar sunmaktadır.

C Native SDK

BlackBerry Native SDK, geliştiricilerin kendi mevcut uygulamalarını platforma sunabilmeleri için bir çok açık standardı desteklemektedir. Başlangıç olarak bir Native Dev Site⁴ bulunmaktadır. Native SDK ile kodunuzu yazmak, uygulamanızın donanımınıza en yakın şekilde olmasını sağlar. BlackBerry 10 native SDK'sı, BlackBerry 10 OS altında çalışan programları geliştirmek için ihtiyaç duyacağınız her şeyi içermektedir: bir derleyici, bağlayıcı, kütüphaneler ve geniş bir Entegre Geliştirme Ortamı (IDE). Windows, Mac ve Linux için mevcuttur.

Temel geliştirme adımları şunlardır:

- Bir imza hesabı ve anahtarı talebi
- Native SDK'nın⁵ kurulumu
- Simulator'ün⁶ kurulumu ve yapılandırılması
- Geliştirme ve kullanım için ortamınızın yapılandırılması
- İlk projenizin oluşturulması
- Örnek uygulamaların çalıştırılması

³ www.qnx.com

⁴ developer.blackberry.com/native/beta

⁵ developer.blackberry.com/native/download

⁶ developer.blackberry.com/native/download

Yeni bir özellik olarak BlackBerry, NDK desteğine Scoreloop'u⁷ eklemiştir. Scoreloop, mobil sosyal oyunları mümkün kılan bir teknolojidir. Geliştiricilerin kendi oyunlarına sosyal özellikleri ekleyebilmelerini sağlar ve oyunun kendine has görünümünü ve hissini korur. Şu anda mevcut olan özelliklerden bazıları şunlardır:

- Kullanıcı profili
- Skor tahtası
- Meydan okumalar
- Ödüller ve başarılar

C++ Cascades SDK

C++ ve Cascades ile geliştirmek başka bir seçenektir. Cascades, geliştiricilerin kolay UI uyarlaması için güçlü bir desteğe sahip bir BlackBerry native uygulamasını inşa etmelerine fırsat vermek için tasarlanmıştır. Cascades çerçevesi, uygulama mantığını kullanıcı arayüzü oluşturma motorundan ayırmaktadır. Bir uygulama içerisinde belirtilen kullanıcı arayüzü kontrolleri, bunların özellikleri ve davranışları, QML⁸ adı verilen bir işaretleme dili içerisinde tanımlanmıştır. Uygulamanız çalıştığında, UI oluşturma motoru kullanıcı arayüzü kontrollerini görüntüler ve belirlenmiş olan herhangi bir geçişi veya efekti uygular. Cascades SDK aşağıdaki özellikleri sunmaktadır:

- Cascades UI'si ve platform API'ları

⁷ developer.blackberry.com/native/documentation/bb10/com.qnx.doc.scoreloop.lib_ref/topic/overview.html

⁸ en.wikipedia.org/wiki/QML

- UI'nizi C++, Qt Modelleme Dili (QML) veya her ikisi içerisinde geliştirmek için araçlar
- Temel UI kontrollerinin avantajlarını kullanmak ve yeni kontroller yaratmak
- Mobil ve Wi-Fi ağları üzerinde iletişim
- Medya dosyalarının kaydı ve oynatılması
- Verilerin depolanması ve alınması
- Sertifika yönetimi ve kriptografi araçları

Cascades uygulama geliştirme çerçevesi, Qt arayüz geliştirme teknolojileri kullanılarak inşa edilmiştir. Bu mimari yapı Cascades'in Qt nesne modelini, olay modelini ve sarma modelini güçlendirmesine fırsat vermektedir. Qt içerisindeki yuva ve sinyal mekanizması, güçlü ve esnek nesneler arası bir iletişime olanak tanır. Cascades çerçevesi, temel Qt sınıflarını (QtCore, QtNetwork, QtXml ve QSql ve diğerleri) kullanır ve bunları temel alır. Bu da, geliştirilen çeşitli unsurları programlamak yerine bunları tanımlamalarını sağlar; başka bir deyişle, bir animasyonu programlamak yerine onun süresini ve tipini tanımlamaları yeterli olacaktır. Bu yaklaşım, Core Animation ile iOS'taki işleyişe benzerdir. JSON benzeri biçimlendirmesi sebebiyle QML, tecrübeli JavaScript geliştiricileri tarafından bile yazılabilir.

Geliştiricilere bu yeni UI inşası yaklaşımında yardımcı olmak amacıyla, Cascades Builder isimli bir araç bulunmaktadır. QNX Momentics IDE'sine'dahil edilmiştir ve geliştiricilerin görsel bir arayüz kullanarak bir UI tasarlayabilmelerine fırsat verir. Bir kod üzerinde değişiklik yaptığınızda, bunun etkilerini tasarım görünümünde hemen görebilirsiniz. Geliştiricinin herhangi bir kontrol programlamasına gerek yoktur, basit bir şekilde taşı ve bırak yaklaşımını kullanabilir.

Eğer bir tasarımcıysanız, Cascades Exporter⁹ sizin için idealdir. Bu Adobe Photoshop eklentisi, resimlerinizi bölüp yeniden boyutlandırır ve bir tmz dosyasına paketler (sıkıştırılmış, bölünmüş ve metadata destekli resim kaynakları). Bu kaynak dosyaları, QNX Momentics IDE'si ile bir geliştirici tarafından kolaylıkla kullanılabilir.

Daha fazla bilgi için bir Cascades Dev Sitesi¹⁰ mevcuttur.

HTML5 WebWorks

Eğer bir Web/JavaScript geliştiriciyseniz, mevcut becerilerinizi BlackBerry uygulamaları yazmak için kullanabilirsiniz. Kullanabileceğiniz iki önemli araç bulunmaktadır:

Bunlardan ilki WebWorks SDK¹¹. Diğer özelliklerinin yanı sıra, normal web sayfalarını yazmanıza ve bunları kolaylıkla native BlackBerry uygulamaları olarak paketleyebilmenize fırsat verir. Webworks'ün yeni 2.0 sürümü, PhoneGap olarak da bilinen Apache Cordova yazılım geliştirme çerçevesine sıkı bir şekilde uymaktadır. BlackBerry, bütün webworks-apis'leri çapraz platform Cordova aracı eklentileri olarak yayınlamıştır. Eğer HTML içerisinde BlackBerry-UI stilini taklit etmek istiyorsanız, GitHub içerisinde size yardımcı olabilecek bir proje bulunur. Bunun adı BBUI.js'dir¹² ve normal web sayfalarınızı native BlackBerry-UI uygulaması olarak görünecek şekle sokabilmek için kapsamlı bir CSS sunar. Bu yaklaşım için HTML'i geliştirmek amacıyla çeşitli veri özelliklerini kullanırsınız. Bbui.js'ye alternatif olarak, BB10 temasına sahip jQueryMobile için destek

⁹ developer.blackberry.com/cascades/documentation/design/cascades_exporter

¹⁰ developer.blackberry.com/cascades

¹¹ developer.blackberry.com/html5/download/sdk

¹² github.com/blackberry/bbUI.js

de bulunmaktadır. SenchaTouch çerçevesi de¹³ ayrıca BB10'u desteklemektedir.

İkinci araç ise Ripple Emulator'dür¹⁴. WebWorks uygulamaları için bir BB10 cihaz simülatörü görevi gören bu emülatör, bir Chrome tarayıcısı uzantısıdır. İvmeölçer ve GPS sensörü gibi donanıma özel çeşitli unsurları taklit etmektedir. Komut satırını kullanmadan bile uygulamınızı paketleyip kullanmanızı sağlar.

BlackBerry 10 işletim sisteminin donanım hızlandırmasına sahip WebGL desteği sunduğunu bilmek yararlı olabilir ve dâhili bir özellik olan WebInspector vasıtasıyla mobil cihaz üzerinde hata ayıklama ve profillemeye işlemlerini gerçekleştirebilirsiniz.

¹³ www.sencha.com/products/touch

¹⁴ developer.blackberry.com/html5/download/ripple



WebWorks ile geliştirme hakkında daha fazla bilgi almak için HTML5 Dev mikro sitesi¹⁵ bulunmaktadır.

Adobe Air

Eğer bir Adobe AIR geliştiricisiyseniz, yeni bir dağıtım kanalı olarak BB10'u ekleyebilirsiniz. BlackBerry cihazları için uygulamalar yaratmak amacıyla Adobe AIR için BlackBerry 10 SDK'yı kullanacaksınız.

BlackBerry uygulamalarını yaratmak/taşımak için Adobe ActionScript ve Adobe Flex API'ları olan SDK'yı kullanabilirsiniz. Bu API'lar benzersiz çeşitli UI bileşenleri ve önceden belirlenmiş görünümünün yanı sıra, BlackBerry cihazlarına özel olaylar için dinleyiciler sunmaktadır. Adobe Flash Builder API'larının kullanımı ile uygulamanız ayrıca, ivmeölçer ve konum belirleme bilgileri gibi mobil cihazlara özel uygulamalara erişim sağlayabilir.

Adobe AIR uygulamanızı geliştirmeye başlamak için:

- Windows için VMware Player veya Mac için VMware Fusion'ı indirin ve kurun
- BlackBerry 10 simülatörünü indirin
- Adobe AIR için BlackBerry 10 SDK'yı indirin
- Adobe Flash Builder, Powerflasher FDT veya komut satırı araçlarıyla geliştirmeye başlayın

Daha fazla bilgi için özel web sitesini¹⁶ ziyaret edin.

¹⁵ developer.blackberry.com/html5

¹⁶ developer.blackberry.com/air/

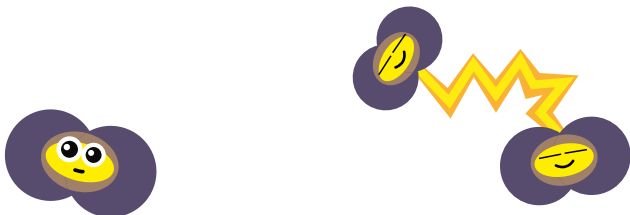
Android Runtime

BlackBerry 10.2.1 ve üzeri sürümlerde Android Jelly Bean 4.2.2 platform uygulamalarını çalıştırmak amacıyla Android uygulamaları için BlackBerry Runtime'ı kullanabilirsiniz. Runtime'ı kullanmak için Android uygulamalarınızı BAR dosyası biçiminde yeniden paketlemeniz gerekmektedir.

Bir geliştirici olarak, uygulamanızı yeniden paketlemek için aşağıdaki araçlara ihtiyacınız olacaktır. Bu araçlar ayrıca, Android SDK'sındaki bazı API'lar desteklenemeyeceği veya kısmen yalnızca BlackBerry platformları üzerinde olabileceği için uygulamanızın BlackBerry 10 üzerinde çalışma uyumluluğunu kontrol edecektir.

— Eclipse için eklentinin yeniden paketlenmesi

Bu aracın temel avantajı, Eclipse'den ayrılmadan BlackBerry PlayBook, BlackBerry Tablet Simulator, BlackBerry 10 Dev Alpha Simulator ve BlackBerry 10 cihazları üzerinde uygulamaların uyumluluğunu, yeniden paketlemesini kontrol etmek, hata ayıklama ve çalıştırma becerisidir. Ayrıca bu eklenti, uygulamanız dağıtılmadan imzalamak için kullanabilirsiniz. Eğer uygulamanızı imzalamadan test etmek istiyorsanız, hedef cihaz veya simulator üzerinde bir hata ayıklama belirteci yaratmak ve kurmak için eklenti kullanabilirsiniz.



— Çevrimiçi paketleyici

Android uygulamaları için BlackBerry paketleyicisinin temel avantajı, yalnızca tarayıcınızı kullanarak Android uygulamanızı hızlı bir şekilde yeniden paketleyebilmek için kullanabilmenizdir. Uygulamayı, uyumluluk açısından test edebilir, BlackBerry Tablet OS veya BlackBerry 10 uyumlu BAR dosyası olarak yeniden paketleyebilir ve daha sonra BlackBerry World mağazası üzerinden dağıtılabilmesi için imzalayabilirsiniz.

— Komut satırı yeniden paketleme araçları

Android uygulamaları için BlackBerry SDK'sının temel avantajlarından bir tanesi bunu, APK dosyası formatından BAR dosyası formatına kadar çeşitli Android uygulamalarını yeniden paketleyebilmek için kullanabilmenizdir. Ayrıca, Android uygulamalarınızın uyumluluğunu kontrol etmek, uygulamalar imzalamak, hata ayıklama belirteçleri yaratmak ve geliştirici sertifikalarını oluşturmak için satır komutu araçlarını kullanabilirsiniz.

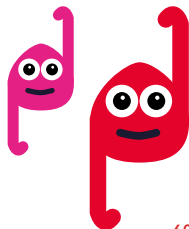
Eğer BB10 üzerinde Android uygulamalarını kullanmaya ilişkin daha fazla bilgi almak istiyorsanız, lütfen özel web sitesini¹⁷ ziyaret edin.

BlackBerry App Generator

Eğer bir geliştirici değilseniz, BlackBerry BlackBerry App Generator¹⁸ ile BB10 için basit bir uygulama yaratma aracı sunmaktadır. Bu araç;

¹⁷ developer.blackberry.com/android

¹⁸ blackberryappgenerator.com/blackberry/



- Web sayfası,
- RSS beslemeleri
- Tumbler
- Facebook
- YouTube
- flickr

ve daha fazlası gibi çeşitli girdi kaynaklarını temel alan uygulamalar oluşturabilmenizi sağlar. Bir logo ve renk seçimine sahip olacak şekilde özelleştirilebilen ayrıntılı bir uygulama yaratır. Basit bir haber uygulaması için bu yaklaşım tamamen yeterlidir ancak "CNN" benzeri bir şaheser yaratmayı bekleme-
meyin.

Test Süreçleri

BlackBerry, aynı bir indirme¹⁹ seçeneği olarak BB10 telefonları için bir simülatör sunmaya devam etmektedir. Bu simülatör, uygulamanızı gerçek bir BlackBerry cihazında çalıştırma yönteminize benzer şekilde bir PC/Mac/Linux üzerinde çalıştırabilmenize fırsat vermektedir. Teste yardımcı olması için simülatör, controller ismindeki küçük bir uygulama ile gelmektedir. Bu araç, şarj seviyesi, GPS konumu, NFC veya cihazın yatırılması gibi ayarları simüle etmenize fırsat verir. Böylece uygulamanızın gerçek durumlarda nasıl çalışacağını önceden görebilirsiniz.

¹⁹ developer.blackberry.com/devzone/develop/simulator/

İmzalama

Platformun güvenlik açısından önemli birçok sınıflı ve özelliği (ağ oluşturma veya dosya API'leri gibi), dağıtımcının belirlenebilmesi için uygulamanın imzalanmasını gerektirmektedir. BlackBerry için bir uygulama geliştirme işlemindeki bir son adım genellikle sancılı bir durumdur.

Eğer imzalanmamış uygulamanızı fiziksel bir cihaz üzerinde test etmek istiyorsanız, hata ayıklama belirteci isimli bir dosyayı talep etmeniz gerekmektedir. Bu belirteç, belli bir BB10 cihazının imzalanmamış uygulamaları çalıştırmasına izin verir. Bu kurulum prosedürü için BlackBerry Anahtar Sipariş Formu²⁰ üzerinden bir imzalama dosyası (client-PBDT-xxxxx.csj) talep etmeniz gerekir. E-posta aracılığıyla dosyayı aldıktan sonra, komut satır araçlarıyla bir hata ayıklama belirtecini yükleyebilirsiniz. Bu kurulumdan sonra cihazınızda imzalanmamış uygulamaları çalıştırabilirsiniz. Bu işlemin her bir cihaz için ayrıca yapılması gerektiğini unutmayın.

Eğer uygulamanızı BlackBerry World'de yayınlamak istiyorsanız, BlackBerry Anahtar Sipariş Formu²¹ üzerinden sipariş verilmiş olan bir imzalama anahtarına da gerek duyacaksınız. Bu kurulum sürecinde size yardımcı olmak için BlackBerry, işlemi adım adım anlatan bir web sayfası²² sunmaktadır.

²⁰ www.blackberry.com/SignedKeys/codesigning.html

²¹ www.blackberry.com/SignedKeys/codesigning.html

²² developer.blackberry.com/CodeSigningHelp/codesignhelp.html

Dağıtım

Önceki bütün işletim sistemi sürümlerinde olduğu gibi BB10 uygulamaları BlackBerry World²³ üzerinden dağıtılmaktadır. Gerekli satıcı hesabı, Vendor Portal for BlackBerry World²⁴ adresinden oluşturulabilir.

Ücretli uygulamalar için geliştiriciler, gelir payının %70'ini almaktadırlar.

İkinci seçenek ise Kurumsal Uygulama Dağıtımı yöntemidir. Bu, herhangi bir kullanıcı için dağıtıma sunmak yerine organizasyon ya da firmanız içerisinde bir dâhili uygulamayı kullanmanıza olanak sağlar. Bu yöntem, B2B uygulamaları için son derece uygundur.

Kurumsal Uygulama Dağıtımı hakkında daha fazla bilgi almak için, web sitesini²⁵ ziyaret edebilirsiniz.



²³ appworld.blackberry.com

²⁴ appworld.blackberry.com/isvportal

²⁵ developer.blackberry.com/distribute/enterprise_application_distribution.html



Firefox OS

Ekosistem

Başka bir mobil işletim sistemine ihtiyacımız var mı? Mozilla Foundation ihtiyacımız olduğunu düşündü ve daha düşük fiyatlı akıllı telefonları hedefleyen, Linux tabanlı açık kaynaklı bir işletim sistemi olan Firefox OS'yi¹ geliştirdi. En başından beri güncellemeler için ortaya konulan sürüm döngüleri fazla iddialıydı: Şubat 2013'teki 1.0 sürümünün çıkışından sonra sürüm 1.1 Ekim ayında, 1.2 de aynı yılın Aralık ayında piyasaya çıkartıldı.

Kitle piyasası için çıkartılan ilk Firefox OS cihazı, 80 Dolara satılan ve yükselen piyasaları hedef alan ZTE Open'dı. Almanya'da Alcatel'in One Touch Fire cihazının piyasaya çıkışı ile Firefox OS resmi olarak Ekim 2013'te Avrupa piyasasına giriş yaptı. Telefonun başlangıç fiyatı 90 Avro'ydu.

Firefox OS'de uygulamalar HTML tabanlıdır. Phonegap gibi araçlarla HTML5 web uygulamalarını paketleyebiliyor olsanız da Firefox OS'lerde ana geliştirme dili olarak HTML/JavaScript/CSS kullanılmaktadır. Bunun anlamı bir web geliştiricisi, sisteme native uygulama yazmaya kolaylıkla başlayabilir. Birazcık Firefox OS tarafından sunulan JavaScript API üzerine

¹ mozilla.org/firefox/os



ve uygulamaların paketlenmesi üzerine bilgisini artırması gerekecektir.

Firefox OS temel olarak üç temel bileşenden oluşmaktadır:

- **Gonk:** Düşük seviye Linux Kernel ve donanım soyutlama katmanıdır (HAL). Teoride donanım satıcısının donanımını Firefox OS uyumlu yapması için Gonk'u cihaza uygun hale getirmesi yeterlidir.
- **Gecko:** Uygulamanın çalışma ortamı (runtime). Gecko, HTML'yi, JavaScript'i ve CSS'yi ayrıştırır, çalıştırır ve işler. Uygulamaların işlevselliğini sağlayabilmesi için gerekli tüm donanım erişimleri bu çalışma zamanı altyapısınınca sağlanır. Bir ağ katmanı, grafik katmanı, yerleşim (layout) motoru, Javascript için sanal makinesi ve cihazla Gecko'yu bağlamaya yarayan bir porting katmanı vardır.
- **Gaia²:** HTML, CSS ve JavaScript'de yazılmış olan kullanıcı arayüzü (UI). Gaia, standart diyaloglar için gerekli olan bütün kullanıcı arayüzü (UI) unsurlarını sunmaktadır. Open Web API'leri ile işletim sistemi arayüzünü oluşturur.

Geliştirme

Firefox OS için bir uygulama yaratmanın iki yolu bulunur: sunucudaki uygulamalar ve paketlenmiş uygulamalar. Her iki durumda da kodu HTML, CSS ve JavaScript ile yazarsınız. Sunucudaki uygulamalar temel olarak bir web sitesidir. Kolaylıkla güncellenebilirler ancak web API'sine sınırlı bir erişim sunarlar ve bir internet bağlantısına ihtiyaç duyarlar. Paketlenmiş

² github.com/mozilla-b2g/gaia github.com/mozilla-b2g/gaia github.com/mozilla-b2g/gaia

uygulamalar cihazın içinde, yerelde çalışır ve temel olarak uygulamanın bütün varlıklarını içeren bir zip dosyasıdır.

Normal web uygulamalarından farklı olarak Firefox OS uygulamaları bir manifestoya³ ihtiyaç duyarlar. Bu manifesto, uygulamanızın adını, tanımını, simgelerini ve diğer bilgilerini belirten bir meta-verisidir.

Minimum düzeydeki bir manifesto şuna benzer:

```
{
  "name": "Hello World",
  "description": "Yet another...",
  "launch_path": "/index.html",
  "icons": {
    "128": "icon.png"
  },
  "developer": {
    "name": "Your name",
    "url": "http://..."
  },
  "default_locale": "en"
}
```

Firefox WebAPI⁴ titreşim, konum belirleme, şarj durumu, alarm, IndexedDB, yakınlık sensörü, ortam ışığı sensörü ve arşiv gibi özelliklere erişim sunar. API'leri kullanarak, örneğin, JavaScript içerisinde `navigator.battery.level` komutu üzerinden şarj durumuna erişim sağlayabilirsiniz.

Eğer WebAPI'sinin sunduğundan daha fazla özelliğe

³ developer.mozilla.org/en-US/docs/Web/Apps/Manifest

⁴ wiki.mozilla.org/WebAPI



ihtiyacınız varsa, Activity'leri kullanabilirsiniz. Mozilla, Android Intent'lerine benzer şekilde MozActivity nesnesini kullanır. Kullanıcıya belirli bir görev için hangi uygulamayı kullanmak istediği sorulacaktır.

Kısa bir mesaj yaratma örneği şöyle gösterilebilir:

```
var sms = new MozActivity({  
  name: "new",  
  data: {  
    type: "websms/sms",  
    number: "+46777888999"  
  }  
});
```

Bu örnekte de resim galerisine (bir görseli seçmek için) erişimde bulunmayı göstermektedir:

```
var pick = new MozActivity({  
  name: "pick",  
  data: {  
    type:  
  }  
});
```



Simülatör ve Test Etme

Mozilla, tarayıcı eklentisi olarak Firefox OS'i için indirilebilir bir simülatör⁵ sunmaktadır. Firefox OS 1.2, Uygulama Yöneticisi⁶ ile birlikte gelir. Bu yeni geliştirici aracı, kodlar üzerinde uzaktan hata ayıklama yapılabilmesini sağlar ve manifest edi-

⁵ addons.mozilla.org/en-us/firefox/addon/firefox-os-simulator/

⁶ developer.mozilla.org/en-US/Firefox_OS/Using_the_App_Manager

tor gibi grafik arayüzü (GUI) olan yardımcı araçlar sunar. Ancak simülatöre körü körüne güvenmemeniz tavsiye edilmektedir: Gerçek Firefox cihazlarına kıyasla simülatörlerde çok daha fazla RAM bulunmaktadır.

Dağıtım

Mozilla'nın, Marketplace⁷ isminde oluşturduğu küresel bir uygulama mağazası var. Mozilla'nın kılavuzlarına⁸ göre incelenen uygulamanız Marketplace'te yayınlandıktan sonra, elde ettiğiniz gelirin %70'i size kalıyor.

Daha Fazlasını Öğrenin

Firefox OS uygulamanızı geliştirmek ile ilgili olarak daha fazla bilgi almak için kullanacağınız ilk kaynak, Mozilla Geliştirici ağıdır⁹. Video dersler de dahil olmak üzere hızlı bir başlangıç, marketplace.firefox.com/developers/docs/quick_start adresinde bulunabilir. Ayrıca Andre Fiedler de kendi Sideshare.net sunumunda (Doing mobile web Apps for Firefox OS - the right way¹⁰) yararlı bilgiler sunmaktadır. Son olarak, Mozilla'nın geliştirici destekçisi Chris Heilmann kendi blogunda¹¹ bir çok bilgi sunmaktadır.



⁷ marketplace.firefox.com

⁸ http://developer.mozilla.org/en-US/docs/Web/Apps/Publishing/Marketplace_review_criteria

⁹ developer.mozilla.org

¹⁰ slideshare.net/andrefiedler1/doing-mobile-web-apps-for-firefox-os-the-right-way

¹¹ hacks.mozilla.org/author/cheilmann



iOS

Ekosistem

iOS'un Kısa Tarihçesi

Apple iOS'u (o zamanki adıyla iPhone OS) 29 Haziran 2007'deki MacWorld 2007 etkinliğinde piyasaya sürülen ilk iPhone ile beraber tanıttı. O tarihten itibaren Apple, her sene Haziran ve Ekim arasındaki belli tarihlerde yeni nesil bir iPhone cihazı ile yeni bir iOS sürümünü tanıtmaya başladı. Ekim 2013 tarihinde, en sonuncu sürüm olan 7.0'ı, 2013 Apple Worldwide Developer Conference'da (WWDC) ilk defa görücüye çıkardı. Orijinal iOS'un çıkışından sonra ilk defa bir yeni iOS sürümü, kullanıcı arayüzüne ve estetiğe ilişkin genel bir yenileşme sunmaktaydı. Birçok kullanıcı için iOS 7'deki değişiklikler farklı bir görünümü ifade etse de, önceki işletim sistemlerinin tasarım kalıplarını içerisinde geliştirilen uygulamalar yeni sürümde çok farklı bir görüntüde olacağı için geliştiricilerin daha fazla çalışmalarını gerektirdi.

iOS ile Çalışan Cihazlar

Şu anda Apple, iOS ile çalışan şu farklı cihazları (farklı konfigürasyonlarda) satıyor:

- iPhone
- iPod touch
- iPad
- Apple TV

Apple TV haricinde, bütün bu cihazlar App Store içermekte ve üçüncü parti uygulamaları çalıştırabilmektedir.

Cihazların çoğu piyasaya çıkmalarının üzerinden geçecek en az 2 yıl sürede, en güncel iOS sürümüyle uyumlu çalışabilecek şekilde destekleniyor. Bu nedenle geliştiriciler bir uygulama geliştirmeyi planladıklarında bu durumu da hesaba katmalılar. Eski donanımı kullanmak, genellikle daha az donanım kaynağı anlamına gelir ve bazen de farklı ekran boyutları ve/veya farklı ekran çözünürlüklükleri demektir.

iOS cihazlarının ayrıntılı listesi, özellikleri ve desteklenen iOS sürümleri, Wikipedia'da¹ bulunabilir.

Cihaz ve Uygulama Satışları

Apple'dan alınan bilgiye göre, ki bunlar genellikle özel medya etkinliklerinde duyurulan kilometre taşlarıdır, 700 milyondan fazla iOS cihazı satılmıştır². Makaledeki grafiğe göre, bu cihazların 200 milyonu yalnızca 2013 yılında satılmıştır. iOS içeren cihazların satışı hala ivme kazandığı için, beş yıl sonra bile bu cihazların büyük miktarının hala aktif kullanımda olduğu ve iOS 6 veya iOS 7'yi kullandığı öngörülebilir.

App Store şu anda, üçüncü parti geliştiricilerin geliştirdiği ve toplamda 60 milyar defa indirilen 1.000.000 uygulama³ içermektedir. Apple'a göre geliştiricilere on milyar dolardan⁴ fazla ödeme yapılmaktadır.

1 en.wikipedia.org/wiki/List_of_iOS_devices

2 theverge.com/2013/9/10/4715256/apple-700-million-ios-devices-sold-by-end-of-september

3 theverge.com/2013/10/22/4866302/apple-announces-1-million-apps-in-the-app-store

4 digitaltrends.com/mobile/app-store-specs-wwdc-2013/

Parçalanma

4.0" cihazlarının (iPhone 5 ve beşinci nesil iPod touch) piyasa çıkışı ile Apple ilk defa iPhone ve iPod touch modellerinin boyutunu değiştirmiştir. Bu da, her iki cihaz boyutu için geliştirme yapmak anlamına geldiği için biraz daha çaba gösterilmesini gerektirmektedir – özellikle de evrensel bir uygulama optimizasyonu (hem iPhone, hem de iPod touch ve iPad için) gerekli olduğunda.

Güncellenmiş CPU mimarileri uygulama projelerine çok fazla yük yaratmasa da Apple'ın tanıttığı 64-bit ARM çipi Apple A7⁵ 32-bit ve 64-bit mimarileri destekleme noktasında biraz daha fazla çaba gerektirmektedir. Ancak çoğu uygulama için bu sadece Apple'ın geliştirici araçlarının yeni sürümünü kullanarak derleme yapılması anlamına gelmektedir.

Yazılım tarafında ise iOS 7, tamamen yenilenmiş kullanıcı arayüzü ve estetikle piyasaya çıkmıştır. Daha önceki sürümlerde desteklenen süslü ve derin zengin görünüm yerine yeni işletim sistemi ikonografide daha az ayrıntı kullanan düz bir görünümle gelmiştir. Bu değişiklik ile hem yeni iOS 7'yi, hem de iOS 6 gibi eski sürümleri destekleyen uygulamalarda ilave çalışmaların yapılması gerekli olmuştur.

Apple Aralık 2013 itibarıyla hâlâ Retina Ekran özelliğini kullanmayan (iPad 2 veya orijinal iPad mini gibi) donanımların eski modellerini satmaktadır. Bu sebeple geliştiricilere optimize edilmiş iki ayrı sürüme de grafik kaynaklarını dahil etmeleri tavsiye edilmektedir.

Kullanıcılar için ücretsiz olan işletim sistemi güncellemeleri ve bazı durumlarda dört yıllık cihazların bile hala bu güncellemeleri alması sayesinde, kullanıcıların yeni sürümlere adapte olması diğer mobil platformlara kıyasla daha hızlı bir şekilde

⁵ en.wikipedia.org/wiki/Apple_A7

gerçekleşmektedir. Apple'ın App Store Destek Merkezi'ne⁶ göre, Ocak 2014 itibarıyla bütün cihazların %80'i, yalnızca 3 aydır piyasada olmasına rağmen iOS 7'yi kullanmaktaydı. Bu hızlı geçiş süreci, geliştiricilerin yeni sürümlere daha hızlı odaklanabilmeleri ve eski sürümler için desteği geride bırakabilmeleri anlamına gelmektedir.

Teknolojiye Genel Bakış

Geliştirme Çatısı ve Dil(ler)

iOS, Mac OS X temeli üzerine inşa edildiği için Coca Touch katmanı (kullanıcı arayüzünü yöneten ve çizen) dışında birçok teknolojinin ve geliştirme çatısının aynısını kullanmaktadır. Bunun yanı sıra, her sisteme özel olan küçük geliştirme çatıları da bulunmaktadır. Bu durum dokunmatik cihazlar için yeniden tasarlanması gereken kullanıcı arayüzü değişiklikleri içeren birçok uygulama için hayatı kolaylaştırmaktadır.

Apple, önceden yalnızca OS X ve benzerlerinde mevcut olan geliştirme çatılarını iOS üzerinde ya da tam tersi şekilde yalnızca iOS üzerinde mevcut olan geliştirme çatılarını OS X'te de kullanabilir kılarak platformları birbirlerine yaklaştırmaya devam etmektedir. 2009'da iPhone OS 3.0'te mevcut olan ama 2013 yılında Mac OS X 10.9'e (Mavericks) de dahil edilen MapKit bu duruma örnek olarak verilebilir.

iOS için Apple tarafından sunulan çoğu geliştirme çatısı, Smalltalk'dan ilham alınan ve C üzerinde kullanılan, C uyumluluğunu tamamen koruyan hafif bir dil olan Objective-C (veya farklı bir uç üzerinde Objective-C API'ları sunan) içerisinde yazılmıştır. Çok az sayıda geliştirme çatısı C API'ları sunmaktadır ve bunlar genellikle ses ve video programlamada

⁶ developer.apple.com/support/appstore/

kullanılır. Sistem ayrıca C++ ve Objective-C++ dillerini ve bunlar için standart geliştirme çatılarını da desteklemektedir.

iOS piyasaya çıkmadan önce Objective-C, yalnızca Apple'ın masa üstü platformu olan Mac OS X içerisinde kullanıldığı için belli belirsiz bir varlık göstermiş ve TIOBE endeks,⁷ puanı %0,03'e kadar düşmüştü. 2007'nin Aralık ayında en popüler programlama dilleri arasında 57. sırada olmasına rağmen 2011 ve 2012 yıllarında kazandığı "Yılın Programlama Dili" ödülleriyle, 2013'teki sıralamada 3. sıraya çıkarak Java ve C'nin arkasında yerini aldı.

Geçtiğimiz yıllarda Apple, otomatik hafıza yönetimi, bloklar (bir closure biçimi) ve otomatik olarak sentezlenmiş özellikler gibi yeni unsurları dile eklemek için hem Objective-C çalışmasında hem de LLVM derleyicisinde çeşitli iyileştirmeler yaparak geliştiricilerin daha az kod yazmasını hedefledi.

Geliştirici web sitesinde⁸ Apple, yazılım indirmeleri, eğitim videoları, başlangıç kılavuzları, dokümantasyonlar, şablon kodlar ve forumlar gibi iOS geliştiricilerinin kullanabileceği tonlarca kaynak sunmaktadır.

Bu kaynakların çoğu, her geliştiricinin okuması gereken İnsan Arayüzü Kılavuzları gibi çok değerli bilgiler içermektedir.

Alternatif Programlama Dilleri

Diğer platformlardan ve dillerden gelerek Objective-C ve iOS SDK API'larını öğrenmek dilin alışılmadık sözdizimi söz konusu olduğunda çok zor olabilir. Bu yüzden, Apple'ın araçlarını ve dillerini kullanmaktansa başka bir alternatifi düşünmek faydalı olabilir.

⁷ tiobe.com/index.php/content/paperinfo/tpci/index.html

⁸ developer.apple.com/devcenter/ios/

Yıllar içerisinde, iOS SDK API'sini farklı dillerde de desteklemek için köprü ve örtü görevi gören pek çok ürün piyasaya çıktı. Bu ürünlerden en popüler, önceden MonoTouch ismiyle bilinen Xamarin.iOS'tir⁹. Ürün, en yüksek olası işletim hızını koruyarak mevcut C# kod zeminini iOS'a (ve Xamarin.Android kullanarak Android'e) taşımayı amaçlamaktadır. Ticari ürün olan Xamarin.iOS, Apple'ın sunduğu en yeni değişiklikleri desteklemek için genellikle hızlı bir şekilde güncellenir. Bu da Xamarin.iOS'u, .NET iş mantığını iOS platformuna taşımak isteyen geliştiriciler için tutarlı bir seçenek haline getirmektedir.

iOS geliştirmesine ilgi duyan Ruby geliştiricileri, Ruby'nin hem OS X, hem de iOS üzerinde çalışan bir uyarlaması olan RubyMotion'a¹⁰ bakmak isteyebilirler. Ürün, mevcut Objective-C kütüphaneleri (CocoaPods¹¹ paket yöneticisi üzerinden veya manuel olarak) ve kaynak kodu derlemeleri (çalışma hızı ve gizleme amaçları için) ile uyumluluk sunmaktadır. Xamarin.iOS gibi RubyMotion da tempolu bir güncelleme hızına sahip ticari üründür.

Xcode ve Alternatifleri

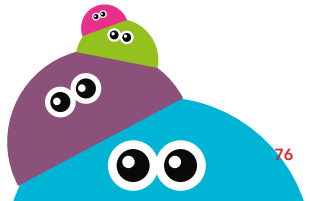
iOS (ve Mac OS X) geliştirmesi için Apple, tamamen ücretsiz olarak kendi geliştirici paketini sunmaktadır. Bunun içerisinde şunlar bulunur:

- **Xcode:** Bütünleşik geliştirme ortamı (IDE)
- **Instruments:** DTrace üzerinde çalışan performans analizi aracı

⁹ xamarin.com/ios

¹⁰ rubymotion.com

¹¹ cocoapods.org



- **Dashcode:** Dashboard widget'leri (Mac OS X) ve diğer HTML ile ilgili içerik için geliştirme ortamı
- **iOS Simulator:** Hızlı test için bir iOS ortamını simüle eder

Xcode için ticari bir alternatif IDE'si ise, JetBrains'in XCode'dan daha ayrıntılı özellikler sunan bir Java uygulaması olan AppCode¹² programıdır.

iOS ile Geliştirmeye Başlarken

iOS ortamında geliştirmeye başlarken gerekli olan unsurlar:

- Mac OS X 10.8/10.9 yüklü Intel tabanlı Mac
- Ücretsiz Apple Developer Connection üyeliği
- Xcode 4.x/5.x ve iOS SDK'sı (developer.apple.com'da ücretsiz olarak mevcuttur)

Bu unsurlar uygulamalar yazmanızı, Xcode ve iOS SDK içerisine dahil edilen iOS simülatörü içerisinde de bunları test etmenize fırsat verir. Bu kadarı, iOS yazılım geliştirmesine başlamak ve buna alışmak için yeterlidir ancak iOS cihazlarına ve AppStore'a uygulama yazabilmek için Apple Developer Connection'da ücretli bir üyelik gerekmektedir (yıllık 99\$). iOS Geliştirici Programı'nda ücretli üyelik ise, iOS ve Xcode'un gelecek sürümlerinin beta versiyonları gibi sürüm öncesi yazılımlara erişim sunar.

Genelde, her türlü cihazda olabildiğince fazla test yapılması tavsiye edilmektedir. Daha fazla bilgi için Test Etme ve Hata Ayıklama bölümüne bakabilirsiniz.

iOS uygulama geliştiriciliğine başlamak isteyenler için

¹² jetbrains.com/objc

Apple, kendi web sitesinde¹³ birçok başlama noktası ve kılavuz sunmaktadır. Yeni başlayanlar için okunabilecek en iyi belge, iOS uygulamalarını geliştirmek için ayrıntılı adımların ve gerekli becerilerin yanı sıra, temel kurulum eğitimlerinden kullanıcı etkileşimli arayüz tasarımına kadar bütün süreç hakkında ayrıntılı makalelere atıfta bulunan Start Developing iOS Apps Today¹⁴ belgesidir.

iOS Geliştirmesine Derin Bakış

iOS platformuna, Objective-C'ye ve Cocoa Touch geliştirme çatısına ilişkin temel konseptleri anladıktan sonra Apple'ın sunduğu diğer geliştirme çatıları ve bunların iOS uygulaması yapımında size nasıl yardımcı olacaklarını öğrenmenin vakti gelmiş demektir.

Bilgilerinizi genişletmek için en iyi başlama noktalarından biri, iOS geliştirme hakkında her türlü unsuru kapsayan bir kitaptır. En tecrübeli iOS geliştiricileri, Aaron Hillegrass'ın "iOS Programming: The Big Nerd Ranch Guide"¹⁵ adlı kitabını önereceklerdir. iOS geliştirme hakkındaki kitaplar hızlı bir şekilde zaman aşımına uğrasalar da, hızlı gelişim döngüsü sayesinde bu kitaplar, diğer kitaplara kıyasla belli bir süre boyunca geçerliliği olan bilgileri daha ayrıntılı bir şekilde anlatmaktadırlar.

¹³ developer.apple.com/library/ios/navigation/#section=Resource%20Types&topic=Getting%20Started

¹⁴ developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/chapters/Introduction.html

¹⁵ bignerdranch.com/book/ios_programming_the_big_nerd_ranch_guide_rd_edition_

Bir sınıf ortamında öğrenmek isteyen kişiler için Big Nerd Ranch, Avrupa ve Kuzey Amerika'da çeşitli eğitim sınıfları¹⁶ sunmaktadır.

Daha ayrıntılı kaynaklar için, Pragmatic Programmers, Apress, Sams ve O'Reilly gibi yayıncıların mevcut kataloglarına bakabilirsiniz. Bu yayıncılar, yeni başlayanlar, orta seviyedekiler ve ileri seviyedekiler için geliştirme çatıları hakkında düzenli olarak belli bir derinliğe sahip kitaplar yayınlamaktadırlar.

Test Etme ve Hata Ayıklama

iOS geliştirici araçları unit testinin yanı sıra, UIAutomation geliştirme çatısı ile otomatikleştirilmiş kullanıcı arayüzü testi imkanı da sunmaktadır. Bu araçlar, Xcode'un komut satırını kullanarak otomatik kabul testi gibi testler için sürekli entegrasyon sistemlerine entegre edilebilirler.

Bunların yanı sıra, çeşitli harici test otomasyon araçları ve geliştirme çatıları da mevcuttur. Bunlardan bazıları tescilli ticari ürünler olsalar da büyük çoğunluğu açık kaynaklıdır. İçlerinde otomatikleştirilmiş testlerinizin daha kolay ve güçlü olması için size hizmet satmak isteyen ticari şirketlerinkilerin ürünleri de vardır. Harici test otomasyon araçlarının çoğu, kullanım sırasında test için gereken bir kütüphaneyi, uygulamanın özel bir inşasına (build) entegre etmeyi gerektirir. App Store için çıkartılacak olan "release" inşalarınızı özel inşalardan ayrı tutmaya özen gösterin, aksi takdirde uygulamanızı App Store'a gönderdiğinizde beklemediğiniz bir ret ile karşı karşıya kalabilirsiniz.

Xcode hem gdp, hem de lldb'yi içerir ve uygulama için uygun derleyiciye göre otomatik seçim yapabilir. Hata ayıklayıcı

¹⁶ bignerdranch.com/training

üzerinde tam bir kontrol sağlarsanız da, Xcode, breakpoint ayarlama, düzenleme ve silme, değişkenleri ve bellek içeriğini görüntüleme gibi sıklıkla kullanılan eylemler için bir kullanıcı arayüzü sunmaktadır.

Araçlar ayrıca, geliştiricilerin hataları, performans darboğazlarını ve hafıza yönetim problemlerini bulabilmeleri için çeşitli özellikler içermektedir.

iOS simülatörleri normal iOS cihazlarına kıyasla farklı bir işletim niteliği taşımaktadır ve iOS ortamının tamamını taklit edemez. Bu yüzden, uygulamanın test edilmesi sırasında ortaya çıkmayan çeşitli sorunlar gerçek cihazlarda görülebilir. Neyse ki, desteklenen bütün testler cihazlarda da çalıştırılabilir ve Apple, test için geliştiricilerin 100 iOS cihazına kadar yükleme yapmalarına izin vermektedir.

Üçüncü parti yazılım testleri için kişiler bulmak ve bunları yönetmek (yerel olarak bir firma içerisinde yapılamıyorsa) TestFlight¹⁷ ve HockeyApp¹⁸ programları ile mümkündür. Her ikisi de beta testçileri için otomatik kod imzalaması, çökme raporlarının toplanması ve uygulama içi güncellemeleri gibi çeşitli yardımcı özellikler sunmaktadır.

Daha Fazla Bilgi

Çevrimiçi Kaynaklar

Geçtiğimiz birkaç sene içerisinde Dave Verwer'in "iOS Dev Weekly"¹⁹ isimli e-posta listesi, her ciddi iOS geliştiricisi için en önemli kaynaklardan biri haline geldi. Dave, her hafta en ilginç haberleri derlemekte ve her Cuma bunları dağıtmaktadır.

¹⁷ testflightapp.com/

¹⁸ hockeyapp.net

¹⁹ iosdevweekly.com

Çok yakın zamanda, Chris Eidhof, Daniel Eggert ve Florian Kugler'in başını çektiği bir grup iOS geliştiricisi, kendi web sitelerinde aylık ücretsiz olarak veya App Store içerisindeki Newsstand'de küçük bir üyelik ücreti karşılığında yayınlanan çevrimiçi bir dergi olan objc.io'yu²⁰ çıkarmaya başladılar. Dergi, her ay farklı bir konuyu incelemekte ve o konu hakkındaki teknolojileri kapsayan makaleleri içermektedir.

Geçtiğimiz sene Matt Thompson, "NSHipster"²¹ adresinde Objective-C ve Cocoa içerisinde kısmen bilinmeyen veya çok fazla dikkate alınmamış API'ler hakkında mükemmel bir makale arşivi oluşturdu. Haftalık olarak ilginç makaleleri yayınlamaya devam ediyor ve yakın zamanda, bu makalelerin tümünü bir kitapta topladı.

Ole Begemann, genellikle başka yerde bulunmayan ayrıntılı iOS geliştirme konuları hakkında görüşlerini sunduğu bir blog yazmaktadır. RSS beslemesine oleb.net/blog adresinden ulaşabilirsiniz.

Kaliteli iOS geliştirme kurslarının en büyük koleksiyonunu Ray Wenderlich²² ve ekibi sayesinde bulabilirsiniz. Başlangıç seviyesinden, her türlü ayrıntıya sahip 3 boyutlu oyun geliştirmesine kadar geniş çapta konuya ve bilgiye buradan ulaşabilirsiniz.

Diğer birçok geliştirici, mevcut gelişmeler hakkında çeşitli bilgileri kendi bloglarına yazmaktadırlar. Öne çıkan bir blog, Objective-C ve Cocoa geliştirmesi hakkında çok ilginç soru-cevap kısmına sahip olan Mike Ash'in²³ blogudur. Daha çeşitli bloglar www.planetcocoa.org üzerinden bulunabilir.

²⁰ www.objc.io

²¹ nshipster.com

²² raywenderlich.com

²³ mikeash.com/pyblog

Etkinlikler

iOS'un sürekli olarak artan popülerliği sayesinde, her sene dünyanın çeşitli yerlerinde buraya yazılamayacak kadar çok sayıda iOS merkezli konferanslar düzenlenmektedir. Yine de belirtilmesi gereken iki önemli konferans bulunmaktadır:

- Her sene Haziran ayında Apple Worldwide Developer Conference'I (WWDC)²⁴ düzenlemektedir. San Francisco'da bir hafta boyunca süren konferans, Apple mühendisleri ile beraber yapılan birçok Mac OS X ve iOS geliştirmenin yanı sıra, geliştiricilerin kendi uygulamaları hakkında Apple mühendislerinden tavsiye alabilecekleri laboratuvar gibi etkinlikleri de içermektedir.
- Mac OS X ve iOS geliştirmeye ilişkin olarak en büyük ve en başarılı Avrupa konferansı, her sene İngiltere'de Mart ayında düzenlenen NSConference'dır²⁵.

Bu iki toplantının biletleri genellikle birkaç gün – bazen saat – içerisinde bitmektedir. Bu yüzden bu konferanslardan herhangi bir tanesine katılmak istiyorsanız planınızı önceden yapın ve bilet satışlarını duyuran ilanlara üye olun.

Eğer benzer şekilde düşünen geliştiriciler ile iletişime geçmek istiyorsanız, büyük ihtimalle size yakın bir CocoaHeads²⁶ toplantısı bulabilirsiniz. Çoğu CocoaHeads topluluğu, sıklıkla toplanıp, kendi tecrübelerini paylaşan ve birbirlerine yardımcı olan gayri resmi Mac ve iOS geliştiricilerinden oluşmaktadır.

²⁴ developer.apple.com/wwdc

²⁵ ideveloper.tv/nsconference/

²⁶ cocoaheads.org



Java ME (J2ME)

Ekosistem

J2ME (resmi olarak "Java ME"), hala geniş çapta kullanılan en eski mobil uygulama platformudur. Oracle tarafından satın alınan Sun Microsystems tarafından geliştirilen J2ME, genellikle temel özelliklere sahip olan cep telefonları için tasarlanmıştır. Günümüzde piyasada bulunan bir çok temel özellikli cep telefonu J2ME'yi desteklemektedir.

Yaşı ve temel pazar segmenti sebebiyle J2ME, API'lar, donanım gücü ve gelir anlamında modern akıllı telefon platformları ile mücadele edememektedir. Sonuç olarak J2ME'nin popülerliği son yıllarda ciddi anlamda azalmıştır.

Peki J2ME için neden geliştirme yapmalısınız? Temel olarak bir sebep yüzünden: piyasaya ulaşmak için. Küresel akıllı telefon satışı, 2013'ün ikinci çeyreğinde ilk defa temel özelliklere sahip telefon satışını geçse de¹, kullanımdaki cep telefonlarının çoğu temel özelliklere sahip telefonlardır. Ve temel özelliklere sahip telefonlar genellikle Java ME'yi desteklemektedir. Bu yüzden eğer iş modeliniz, olabildiğince fazla müşteriye ulaşma odaklı ise, J2ME sizin için mükemmel bir seçenek olabilir – özellikle belirli Afrika ülkelerini veya Hindistan gibi pazarları hedefliyorsanız.

Ancak eğer iş modeliniz, doğrudan uygulama satışlarına bağlı ise veya uygulamanızın son teknolojiye sahip özelliklere ve donanıma ihtiyacı varsa, akıllı telefon platformları daha iyi bir seçenek olacaktır.

1 gartner.com/newsroom/id/2573415

Ön Koşullar

Bir Java ME uygulaması geliştirmek için şunlara ihtiyaç duyacaksınız:

- Java SDK'sı² ve Eclipse Pulsar for Mobile Developers³, Java ME eklentisine sahip NetBeans⁴ veya IntelliJ⁵ gibi bir IDE. Yeni başlayanlar genellikle NetBeans'i seçmektedirler.
- Wireless Toolkit⁶, Micro Emulator⁷ gibi bir emülatör veya satıcı özel belli bir SDK veya emülatör.
- Kurulumla bağlı olarak, ProGuard⁸ gibi bir gizleme cihazına ihtiyaç duyabilirsiniz. Profesyonel geliştirme için Maven⁹ veya Ant¹⁰ gibi bir yapı aracını kullanmayı düşünebilirsiniz.
- Çeşitli cihazlar için uygulamanızı oluşturmak amacıyla açık kaynaklı çerçeve olan J2ME Polish'i¹¹ denemek isteyebilirsiniz.

Bütün bir kurulum ve ayar talimatı, bu kılavuzu kapsamı dışında kalmaktadır; bunun için lütfen ilgili araçların kılavuzlarına bakın.

Ayrıca en önemli teknolojiler ve API'lar için JavaDocs'u indirin ve okuyun: en güncel Java-Docs'u www.jcp.org

2 oracle.com/technetwork/java/javame/downloads

3 eclipse.org

4 netbeans.org

5 jetbrains.com

6 oracle.com/technetwork/java/download-135801.html

7 microemu.org

8 proguard.sourceforge.net

9 maven.apache.org

10 ant.apache.org

11 j2mepolish.org

adresinden indirebilirsiniz. Üreticiye özel API'lar için çeşitli dokümantasyonlar genellikle satıcının web sitesinde (Örneğin, Nokia UI API¹²) mevcuttur.

Uygulama

Java ME platformu, Connected Limited Device Configuration (Bağlı Sınırlı Cihaz Konfigürasyonu) (CLDC)¹³ ve Mobile Internet Device Profile'ı (Mobil İnternet Cihaz Profili) (MIDP)¹⁴ oluşturmaktadır. CLDC ve MIDP'nin her ikisi de, on yıl önce tasarlanmıştır; sundukları özellikler, günümüzün standartlarına göre tam olarak yeterli değildir.

Üreticiler bu yetersiz özellikleri, çeşitli Java Belirleme Talepleri (JSR'ler) uygulayarak tamamlayabilmektedirler. Bunlara örnek olarak kullanıcı veri ve dosya sistemi erişimi (JSR 75) veya GPS desteği (JSR 179) gösterilebilir. Java ME geliştirmesi ile ilgili ayrıntılı JSR listesi için Java Community Process'in JCP Teknolojisi¹⁵ listesine bakınız.

Kullanmak istediğiniz JSR'lerin bütün cihazlar için mevcut olamayacağını bilmek çok önemlidir: bir cihazda mevcut olan özellikler başka bir cihazda bulunmayabilir.

İşletim Ortamı

J2ME uygulamalarına MIDlet'ler denir. Bir MIDlet'in yaşam döngüsü basittir: Yalnızca başlatılabilir, duraklatılabilir ve ortadan kaldırılabilir. Çoğu cihazlarda bir MIDlet, simge durumunda küçültüldüğünde otomatik olarak duraklatılır; arka planda çalıştırlamaz. MIDlet'ler ayrıca birbirlerinden

¹² www.developer.nokia.com/Community/Wiki/Nokia_UI_API

¹³ java.sun.com/products/clcdc/overview.html

¹⁴ java.sun.com/products/midp/overview.html

¹⁵ jcp.org/en/jsr/tech?listBy=1&listByType=platform

aynı bir şekilde çalıştırılırlar ve temel işletim sistemi ile olan etkileşimlerinde oldukça sınırlıdır – bu özellikler tamamen opsiyonel JSR’ler (örnek olarak JSR 75) ve satıcıya özel API’lar vasıtasıyla sunulmaktadırlar.

UI’ler Oluşturmak

Uygulamanızın UI’sini farklı yöntemlerle oluşturabilirsiniz:

1. Yüksek seviyedeki LCDUI bileşenleri: Biçim ve Liste gibi standart UI bileşenlerini kullanabilirsiniz
2. Düşük seviyedeki LCDUI: düşük seviyedeki grafik fonksiyonlarını kullanarak UI’nizin her bir pikselini manüel olarak kontrol edebilirsiniz
3. SVG: UI’yi ölçeklenebilir vektör grafiklerinde çizip of JSR 226¹⁶ veya JSR 287¹⁷ API’larını kullanabilirsiniz.

Ek olarak, bazı üreticiler ilave UI uzantılarını sunmaktadırlar. Örnek olarak, Nokia’nın temel özelliklere sahip en son serisi (Nokia Asha), cihaz modeline göre Full Touch¹⁸ veya Touch and Type¹⁹ kullanıcı arayüzü paradigmalarını içermektedir. Samsung, en yeni Java ME API’larında²⁰ en yeni pinch yakınlaştırma özelliğini sunmaktadır.

UI geliştirmesine yardımcı olabilecek araçlarda bulunmaktadır.

¹⁶ www.jcp.org/en/jsr/detail?id=226

¹⁷ jcp.org/en/jsr/detail?id=287

¹⁸ www.developer.nokia.com/Resources/Library/Full_Touch/

¹⁹ www.developer.nokia.com/Community/Wiki/Nokia_UI_API_1.1b

²⁰ developer.samsung.com/java/technical-docs/Multi-Touch-in-Samsung-Devices

1. **J2ME Polish²¹**: Bu araç CSS’de tasarımı ayırır ve kullanıcı arayüzü için HTML’yi kullanabilirsiniz. Yüksek seviyedeki LCDUI çerçevesi ile geçmişle bağdaşır özelliğe sahiptir.
2. **LWUIT²²**: Swing’den ilham alınan bir UI çerçevesi.
3. **Mewt²³**: UI’yi tanımlamak için XML kullanır.

Java ME için ekran çözünürlükleri 176x280/220’den 360x640’a kadar değişiklik göstermektedir ve bunların en popüler olanı 240x320’dir. Bu kadar farklı çözünürlüğü desteklemek zor olabilir ancak yukarıdaki araçları kullanarak otomatik olarak ölçeklendiren UI düzen tasarımlarını oluşturabilirsiniz. Her bir çözünürlük için özel UI’ler oluşturmak mümkündür ancak bu tavsiye edilmemektedir: oldukça fazla zaman alır, hataya açıktır ve pahalıdır.

Grafiksel kaynaklar daima optimize edilmelidir. Bunun için mükemmel ve ücretsiz bir araç PNGGauntlet²⁴.

Test Etme

Cihaz parçalanması sebebiyle uygulamaların test edilmesi çok önemlidir. Farklı cihazlar üzerinde mümkün olan en erken ve en sık şekilde test yapın. Bazı emülatörler gerçekten iyidir ancak cihazlar üzerinde test edilmesi gereken unsurlar da bulunmaktadır. Nokia²⁵ ve Samsung²⁶ gibi satıcılar, seçilmiş cihazlar için ödenmiş veya hatta ücretsiz uzaktan hizmet sunmaktadır.

²¹ j2mepolish.org

²² lwuit.java.net/

²³ mewt.sourceforge.net

²⁴ pnggauntlet.com

²⁵ forum.nokia.com/rda

²⁶ developer.samsung.com

Otomatik Test

Java Me için, J2MEUnit²⁷, MoMEUnit²⁸ ve CLDC Unit²⁹ gibi çeşitli cihaz test çerçeveleri bulunmaktadır. JInjector³⁰ gibi gelişmiş araçlar da kod kapsamı ve UI test desteği sunmaktadır.

Taşıma

En temelinde Java Me, satıcıların genellikle farklı bir şekilde yorumladıkları çeşitli standart ve özellik setlerinden oluşmaktadır. Bu, her türlü hataya ve standart olmayan davranışa neden olmaktadır. Aşağıdaki bölümlerde, uygulamalarınızı farklı Java ME telefonlarına ve platformlarına taşımanız için çeşitli stratejileri özetleyeceğiz.

En Küçük Ortak Payda

Uygulamanızın fonksiyonelliğini en küçük ortak payda ile sınırlayarak bir çok taşıma sorununu önleyebilirsiniz. Bu genellikle CLDC 1.0 ve MIDP 1.0'ı veya uygulamanızı daha gelişmiş ülkelerde/bölgelerde çıkarmak istiyorsanız, CLDC 1.1 ve MIDP 2.0 anlamına gelmektedir.

Bu yaklaşım basit uygulamalar için iyi olsa da, ayrıntılı ve özelliği bol uygulamalar bunun yüzünden kısıtlanmış olacaktırlar. Bu durumda, temel olarak Kablosuz Endüstri için (JTWI, JSR 185) Java Teknolojisini veya Mobil Hizmet Mimarisini (MSA, JSR 248) kullanmak isteyebilirsiniz ancak, bunların piyasada sınırlı desteğe sahip olduklarını göz önünde bulundurmanız gerekir.

²⁷ j2meunit.sourceforge.net

²⁸ momeunit.sourceforge.net

²⁹ snapshot.pyx4me.com/pyx4me-cldcunit

³⁰ code.google.com/p/jinjector

Çerçeveleri Taşımak

Çerçeveleri taşımak, uygulamanızı farklı cihazlara ve platformlara otomatik olarak adapte ederek parçalama işlemiyle uğraşmanızda size yardımcı olur. Bunu yapabilmek için süreci neredeyse kusursuz hale getirmek için birbiri ile çalışan özel işletim istemci kütüphaneleri ve yapı-zamanı araçları (çaprak derleyiciler gibi) sunarlar.

İyi taşıma çerçeveleri, projeleriniz içerisinde platforma ve cihaza özel kodu kullanmanıza fırsat verir. Diğer bir deyişle, iyi bir çerçeve cihaz parçalanmasını saklamaz ancak bunu daha iyi idare edilebilir bir şekle sokar.

Java ME için seçeneklerinizden bir tanesi Enough Software³¹ tarafından sunulan J2ME Polish'tir (hem GPL Open Source lisansı, hem de ticari lisans altında mevcut olan). Açık kaynaklı MoSync SDK³² ile C++'dan Java ME'ye taşıma yapmak da mümkündür.

Çapraz platform geliştirmesi ve mevcut araç setleri hakkında daha fazla bilgi için “Çapraz Platformu Seçmek” bölümüne bakın.

İmzalama

Mobil cihazlar için Java standardı, imzalanan ve imzalanmayan uygulamalar arasında farklılık göstermektedir. Bazı telefon fonksiyonları, yalnızca güvenilir uygulamalar için mevcuttur.

Bir cihaz üreticisi veya taşıyıcısı tarafından imzalanmış olan uygulamalar, en yüksek güvenlik seviyesine sahiptir ve telefon üzerinde bulunan her türlü java API'ına erişimde bulunabilir.

³¹ enough.de

³² mosync.com

JavaVerified³³, Verisign³⁴ veya Thawte³⁵ tarafından imzalanan uygulamalar daha düşük güvenlik seviyesindeyken, imzalanmamış uygulamalar en düşük güvenlik seviyesinde bulunmaktadır.

Uygulama imzalanmadığında hangi özelliklerin etkileneceği ve nelerin gerçekleşebileceği, büyük ölçüde uyarlamaya göre değişecektir. Ayrıca, bütün telefonlar gerekli olan kök sertifikalarını taşımamaktadırlar. Bu, bir karşılıklıya neden olacağı için uygulamanızı yalnızca gerektiğinde imzalamayı düşünebilirsiniz. Bazı durumlarda bir uygulama mağazası (Nokia Store'un yaptığı gibi) sizin yerinize imzalama işlemini gerçekleştirebilir.

Başka bir seçenek ise, bir test ve sertifika hizmet sunucusunu kullanmak ve bütün bu karmaşıklığı onlara bırakmaktır. Intertek³⁶ büyük ihtimalle bu konu ile alakalı en büyük tedarikçidir.

Dağıtım

Uygulama mağazaları, uygulamalarınızı dağıtmak için muhtemelen en etkin yol olacaktır.

- Handmark³⁷ ve Mobile Rated³⁸ taşıyıcıdan ve satıcıdan bağımsız uygulama mağazaları sunmaktadır.
- GetJar³⁹ yalnızca Java uygulamaları için değil, ücretsiz

³³ javaverified.com

³⁴ verisign.com

³⁵ thawte.com

³⁶ intertek.com/wireless-mobile

³⁷ store.handmark.com

³⁸ mobilerated.com

³⁹ getjar.com

mobil uygulamaları için de bilinen en eski dağıtıcılardan bir tanesidir.

- Nokia Store⁴⁰ dünya çapında Nokia kullanıcılarını hedefler ve kredi kartı ile yapılan alışverişlerde gelirlerin %70'ini ve operatör faturalandırması ile yapılan alışverişlerde yine gelirlerin %60'ını geliştiricilere sunar.
- Orange⁴¹ ve O2⁴² gibi taşıyıcılarda bu sisteme ortak olmaktadır.

Mevcut uygulama mağazalarının (yalnızca J2ME uygulamalarını satanlar hariç) genel bir özeti, WIP App Store Kataloğunda⁴³ bulunabilir. Ayrıca daha fazla bilgi almak için, Uygulama Mağazaları hakkındaki bölüme bakınız.

Daha Fazla Bilgi

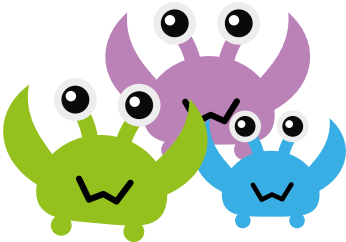
Eğer Java ME geliştirmesi hakkında daha fazla bilgi almak istiyorsanız, aşağıdaki kaynaklar size yardımcı olabilir.

⁴⁰ publish.oivi.com

⁴¹ www.orangepartner.com/distribute

⁴² mobileapps.o2online.de

⁴³ wipconnector.com/appstores/



Online

Java ME, kullanılan en eski mobil platformlarından bir tanesi olduğu için bununla ilgili kaynaklar bulmak çok kolaydır.

Örneğin:

- J2ME Salsa⁴⁴ gibi sitelerde bulunan dersler.
- billday.com/javame gibi sitelerde bulunan kaynak arşivleri.
- opensource.ngphone.com adresinde bulunan blog üzerinden ulaşılabilecek ilginç projeler veya Bluetooth projesi Marge⁴⁵ gibi java.net'in⁴⁶ mobil ve gömülü sayfaları.

Kitaplar

Yıllar içerisinde, Java ME hakkında çok iyi kitaplar yazılmıştır. Bunlar:

- **Beginning J2ME: From Novice to Professional** by Jonathan Knudsen ve Sing Li
- **Pro Java Me Apps: Building Commercial Quality Java ME Apps** - Ovidiu Iliescu
- **Pro J2ME Polish: Open Source Wireless Java Tools Suite** - Robert Virkus, J2ME Polish geliştirmesi ile ilgili olarak.
- **LWUIT 1.1 for Java ME Developers** - Biswajit Sarkar; LWUIT geliştirmesi ile ilgili olarak.

Ne yazık ki, Java ME'nin azalan popülerliği sebebiyle, geçmiş yıllarda yayınlanan Java ME kitaplarının sayısı çok azdır.

⁴⁴ j2mesalsa.com

⁴⁵ marge.java.net

⁴⁶ community.java.net/mobileandembedded



Ekosistem

Tizen, açık kaynaklı, Linux tabanlı bir işletim sistemi olup, akıllı telefonlarda, netbook'larda, araç içi bilgi-eğlence sistemlerinde ve diğer akıllı cihazlarda çalışması için tasarlanmıştır. Nokia/Intel'in Meego ve Samsung'un LiMo'suna (Linux tabanlı akıllı telefon işletim sistemleri) bir varis olarak görülebilir. Samsung da ayrıca artık geliştirmekten vazgeçtiği Bada işletim sistemini Tizen ile birleştirerek native uygulamalar için yeni bir uygulama geliştirme çerçevesi sundu. Linux Vakfı'nın bir markası olarak Tizen ilk önce Aralık 2011'de Tizen Vakfı tarafından duyuruldu ve sürüm 1.0 (kod ismi 'Larkspur') Nisan 2012'de çıktı. 0 tarihten beri sistem sürekli geliştirme içerisinde oldu ve Tizen 3.0'ın 2014 sonlarında çıkarılması planlanıyor. Tizen Projesi'nin temel yüklenicileri; mobil cihazlara yönelik tarafta Samsung ve araç içi bilgi-eğlence sistemlerine yönelik geliştirme süreçlerindeyse Intel'dir. Katkıda bulunan diğer firmalar Fujitsu, NTT Docomo, Huawei, Vodafone ve Orange'dır.

İlk Tizen akıllı telefonlarının 2014 içerisinde çıkması beklenmektedir; bu yazının yazıldığı tarihte Samsung'un Tizen içeren yalnızca birkaç dijital kamera modeli bulunmaktadır.

Geliştirme

Tizen'in odaklandığı temel nokta, HTML5 ile yazılan uygulamaları çalıştırmak için standartlara uyumlu bir işletim sistemine olanak vermektir. Tizen bunun yanı sıra, C++ içerisinde de native uygulamaları yazmanıza fırsat sunmaktadır. Böylece donanım becerilerini en yüksek seviyeye çıkartabilirsiniz. Her iki geliştirme yöntemi de, ilk Tizen uygulamanız için size iyi bir başlangıç sunmak amacıyla, JQuery gibi çeşitli popüler yazılım geliştirme çerçeveleri ve kütüphaneler tarafından desteklenmektedir. Eğer daha önce native Bada uygulamaları yazdıysanız, bunları kolaylıkla taşıyabilirsiniz. Bada uygulamalarını Tizen içerisinde çalıştırmak için yalnızca birkaç kod değişikliği yeterlidir ve bu süreci daha da kolaylaştırmak için bir taşıma aracı mevcuttur.

Web uygulamaları geliştiricileri, mükemmel uygulamalar yaratmak için kapsamlı HTML5 özelliklerini, JQuery ve JQuery Mobile gibi Tizen cihazı API ve kütüphanelerini kullanabilirler. Eğer Bada için uygulama geliştirdiyseniz, yazmış olduğunuz kodun büyük kısmını Tizen üzerinde de kullanabilirsiniz. Native uygulamalarda olduğu gibi, yalnızca çok az sayıda kod değişikliği gerekli olacaktır.

Resmi Tizen SDK'sı, hem web hem de native uygulama geliştirmesi için kullanılabilen Eclipse bazlı bir entegre geliştirme ortamı içermektedir. Daha önce Bada ile çalışmış geliştiriciler muhtemelen bu SDK'nin kökenini hatırlayacaklardır: Samsung'un Bada SDK'sı. Bir kod editörü, UI tasarımcısı, cihaz emülatörü... bunların hepsi kullanıma hazır bir şekilde mevcuttur. Web tabanlı uygulamalar için eğer isterseniz Intel XDK'yi kullanabilirsiniz.

Test Süreci

Elbette en iyi testler, bir cihaz üzerinde gerçekleştirilebilir olanlardır. Bu yazının yazıldığı tarihte cihaz üzerinde test için tek seçenek Samsung'un zor bulunan Tizen test cihazları olan RD-PQ ve RD-210 cihazlarıydı. Bu yüzden, ilk seri üretim Tizen telefonlarının piyasa çıkmasını beklerken muhtemelen Tizen SDK'sı içinde sunulan simülatörü ve emülatörü kullanmanız gerekecektir. Simülatör, web tabanlı uygulamaları test etmek için basit bir yaklaşım sunmaktadır ancak özellikleri sınırlıdır ve bir cihaz üzerinde yapılan gerçek durum testiyle kıyaslanamaz. Emülatör, cihaz deneyimine daha yakındır ve native uygulamaları test etmek için de kullanılabilir. QEMU bazı bir sanal makinedir ve mevcut bir Tizen mobil kurulumunun kalıbını (ROM) kullanmaktadır.

Gerçek bir cihaza ihtiyacınız olduğu durumlarda Samsung Remote Test Lab'i¹ kullanabilirsiniz. Bu laboratuvarlar dünyanın her yerine yayılmıştır ve Tizen SDK'sı içinden cihazlara uzaktan bağlanabilmenizi sağlar.

Dağıtım

Tizen uygulamaları, Tizen SDK'sı tarafından yaratılan Tizen paket dosyaları (*.tpk) ve widget dosyaları (*.wgt) ile dağıtılmaktadır. Android'in *.apk dosyalarına çok benzer olan bu dosyalar bir Tizen cihazına kopyalanarak ve cihazın dosya gezgini içerisinde tıklanarak kurulabilir. Uygulama dağıtımı için ana merkez TizenStore² olacaktır. Apple Appstore, Google Play ve Microsoft Windows Phone Market'ın aksine, Seller Office³

¹ developer.samsung.com/remotetestlab

² www.tizenstore.com

³ seller.tizenstore.com

içerisinde kayıtlı geliştirici olmak için ödeme yapmanıza gerek yoktur. Yayınlanması için gönderdiğiniz uygulamalar Tizen Store şartlarına göre incelenecektir. Uygulamanın kabulü genellikle, uygulamanızın karmaşıklığına göre 2-3 gün sürmektedir. Uygulamanız yayımlandıktan sonra elde edilen gelirin %80'ini alırsınız.

Daha Fazla Bilgi

Bir geliştirici olarak bakmanız gereken ilk yer developer.tizen.org olmalıdır. Bu site, Tizen geliştirmesi için her türlü belgelendirme, araç ve destek hizmetlerini içermektedir. Orada ayrıca, çok aktif olan ve yardımsever Tizen uygulama geliştiricileri topluluğunu barındıran bir forum da bulabilirsiniz.



Windows Phone ve Windows RT

Windows Phone ve Windows RT, neredeyse birleşme noktasındalar. Henüz birleşmemiş olsalar da, bu durum yakında gerçekleşecek - tasarım paradigması, API'ler ve araçlar gibi birçok önemli ayrıntıyı halihazırda paylaşıyorlar. Her iki platform için de geliştirme yapmak, Android veya iOS'taki farklı form faktörleri kadar açık bir şekilde gerçekleşmese de, kodu ve UI yapılarını paylaşmak çok daha kolaydır. Kitabın bu baskısıyla beraber Windows Phone ve Windows 8/RT'ye aynı bölümde değineceğiz.

Ekosistem

Kitabın bir önceki baskısıyla ilgili Windows Phone'a ilişkin bazı şikayetler aldık:

- Piyasa çekişmesindeki ciddi düşüş
- Microsoft'un servis sağlayıcının onayından önce önce geliştiricilere ve yetkili kullanıcılara işletim sisteminin yeni sürümlerini kurma sözünü veren kendi "meraklılar programı"nı devreye sokmaması.

Neyse ki her iki durumda da iyi sonuçlar elde ettik: Microsoft, bir erken erişim programı¹ başlattı. Windows

¹ neowin.net/news/developers-finally-get-early-access-to-windows-phone-updates

Phone da ayrıca Kantar World Panel'a² göre, en büyük beş Avrupa ülkesinde (Birleşik Krallık, Almanya, Fransa, İtalya ve İspanya) %10'u aşarak kendi pazar payını ciddi ölçüde arttırdı. İtalya ve Latin Amerika'da Windows Phone, ciddi anlamda iPhone'dan³ daha fazla satış yapmaktadır. Tek dikkat edilmesi gereken nokta, bu büyümenin Nokia Lumia 520 gibi daha ucuz telefonlarla sağlanmış olmasıdır.

Böylece eleştirilerimizin dinlendiği açıkça gördük ki görmek istediğimiz şey de buydu! Bu doğrultuda, bu sürüm ile bazı eksiklikleri öne çıkartalım: İlk olarak API'lerin ciddi miktarda artması ve özelliklerinin aynısını yavaşça yapmak yerine diğer platformları bu konuda geçmesi gerekiyor. ikinci olarak, Windows Phone arka planda çalışan uygulamalar için desteğini arttırmalı; üçüncü olarak Windows Phone, üreticiler için daha iyi özelleştirme seçenekleri sunmalıdır (Çinli şirketlerin kendi yerel pazarlarına daha iyi şekilde sunum yapabilmeleri gibi).

Aktif Windows Phone satıcıları Nokia, Samsung, HTC, ZTE ve Huawei'dir. HTC'nin Windows Phone desteğini durduracağı konuşulmaktadır. Nokia Lumia cihaz aralığı bu piyasanın %80'ine⁴ sahiptir. Microsoft, kendi 'hizmetler ve cihazlar' vizyonunu yalnızca Surface Pro ve Surface tabletlerini sunarak değil, ayrıca Nokia'yı⁵ satın alarak gerçekleştirmektedir.

Windows Phone Store şu anda 200.000'den fazla uygulama-

2 kantarworldpanel.com/global/News/news-articles/Apple-iPhone-5S-outsells-5C-three-to-one-in-Great-Britain

3 microsoft.com/en-us/news/press/2013/aug13/08-21wplatampr.aspx

4 blog.gsmarena.com/over-80-of-windows-phone-devices-made-by-nokia

5 microsoft.com/nokia

maya⁶ ev sahipliği yapmaktadır ve ortalama Windows Phone kullanıcısı şu anda 51 adet uygulama⁷ yüklemektedir.

Windows 8 ve Windows RT tarafından bakıldığında Microsoft, Surface tabletlerinin ikinci iterasyonunu 2013'ün dördüncü çeyreğinde çıkarmıştır. Orijinal Surface RT benzeri RT tabletleri bu kadar başarılı olmamıştır ancak Surface'in kısmen daha başarılı olduğu belirtilebilir. Çoğu satıcı, Nokia haricinde, diğer Windows RT donanımlarını satmaktan vazgeçmiştir. Nokia, Lumia 2520 tableti Microsoft tarafından satın alınmadan önce çıkarmıştı. Windows 8.1, kullanıcılardan daha olumlu bir geri bildirim aldı ve Windows 8 piyasası artık Net Applications'a⁸ göre bütün Mac OS X sürümlerinin toplamını geride bıraktı.

Windows 8 için getirilen eleştirilerden en önemlisi klasik masaüstü ve Metro 'Windows Store Apps' dünyası ile öne çıkan çift kişilikli farklılıkla ilgiliydi.

Diller ve Araç İşletimi

Windows Phone geliştirmesi, Microsoft Visual Studio IDE veya Expression Blend⁹ kullanılarak C/C++, C# veya VB.NET içerisinde yapılmaktadır.

Esas olarak olay yönelimli (event-driven) uygulamalar Silverlight kullanarak oluşturulur. "Oyun döngüsü" temelli oyunlar için de DirectX kullanılır.

Ancak her iki teknoloji de tek bir uygulama içerisinde

⁶ neowin.net/news/microsoft-windows-phone-store-now-has-over-200000-apps

⁷ phonearena.com/news/Are-the-apps-in-the-Windows-Phone-Marketplace-of-a-higher-quality-than-the-ones-in-Google-Play-Store_id32045

⁸ <http://netmarketshare.com/operating-system-market-share.aspx>
netmarketshare.com/operating-system-market-share.aspx

⁹ dev.windowsphone.com

kullanılabilir. İlave olarak PhoneGap¹⁰ kullanarak HTML5 bazı uygulamalar yaratabilirsiniz. Ancak web geliştirmesine bu bölümde değinilmemiştir. Son olarak, Windows Phone App Studio¹¹ kullanarak kod olmadan basit Windows Phone uygulamaları yaratabilirsiniz.

Windows RT geliştirmesi, benzer dillerin ve araçların yanı sıra native WinRT uygulamalarının geliştirilmesi ile JavaScript ve HTML5 kullanımı ile gerçekleştirilebilir. Microsoft ayrıca iş uygulamaları yaratmak için basit bir ortam sunan Project Siena¹² uygulamasını çıkarmıştır.

Uygulamanızı yaratırken hangi platforma odaklanacağınızı bilmeniz önemlidir.

	C/C++	C#/VB.NET	JavaScript
WinRT	Evet	Evet	Evet
Windows Phone	Evet	Evet	Hayır (sadece webview'de mevcut)
Silverlight/XAML	Evet	Evet	Hayır
HTML	Hayır (ancak webview mevcut)	Hayır (ancak webview mevcut)	Evet
DirectX	Evet	Evet (SharpDX ile birlikte)	Hayır

¹⁰ phonegap.com

¹¹ apps.windowsstore.com

¹² microsoft.com/en-us/projectsiena

	C/C++	C#/VB.NET	JavaScript
Codesharing	Eski native Windows uygulamaları, professional Xbox, diğer platformlar, ...	Legacy .NET Windows Apps, indie Xbox, Windows Phone apps, ...	Web sayfaları, HTML5 apps, ...

Eğer Direct X'i C# ile kullanmak isterseniz SharpDX.org'u, anxframework.codeplex.com veya monogame.codeplex.com gibi onu temel alan oyun kütüphanelerini kullanabilirsiniz.

En yaygın senaryo, uygulamalar için Silverlight'ı ve oyunlar için DirectX'i kullanmak olsa da, ihtiyaçlarınıza göre Silverlight oyunlarını ve DirectX uygulamalarını yaratabilirsiniz. Silverlight uygulamanız içerisinde de Direct3D'yi yönetmek mümkün olabilir. Bu olay yönelimli (event-driven) Silverlight uygulaması içerisinde 3 boyutlu bir model görüntülemek veya tamamen DirectX kullanılarak yapılmış bir oyunda Silverlight tabanlı menüler oluşturmak için yapılabilir.

Metro Tasarım Paradigması

Windows Phone'un en belirgin özelliği, tipografiye ve içeriğe odaklanan, kullanımı basit ve benzersiz arayüzüdür. Bu UI paradigmasının adı Metro veya Modern UI'dir¹³ ve Xbox ve Windows 8'de de kullanılmaktadır. Bu UI paradigması aşağıdaki prensipleri içerir:

- **Chrome olmayan içerik** Gereksiz süsleri ortadan kaldırır ve asıl odak noktasının içerik olmasını sağlar. Beyaz alan içeriğe denge ve vurgu sağladığı için, mevcut bütün pikselleri kullanmaktan kaçınılmalıdır.

¹³ [wikipedia.org/wiki/Metro_\(design_language\)](http://wikipedia.org/wiki/Metro_(design_language))

- **Hareketle canlılık** Zengin animasyonlar ile normalde düz olan tasarım derinlik katar.
- **Tipografi güzeldir** Yazı tiplerini Metro içerisindeki birinci sınıf unsur haline getirir. Helvetica'dan esinlenen Windows Phone yazı tipi Segoe, modern bir yaklaşımla eşleşmektedir.
- **Kendinden dijital** tasarım, gerçek dünyadaki nesneleri taklit etmeye çalışmaz; bunun yerine dijital çözümler için mevcut olan etkileşimlere odaklanır.

Microsoft, hukuki sorun ihtimallerine karşı tasarım paradigması için 'Metro' isminde vazgeçmiş olsa da 'Modern UI' gibi alternatif isimler bu kadar popüler olmadı.

Özellikle mevcut uygulamalar üzerinden taşıma yaparken Metro UI tasarım prensiplerini benimsemeniz gerekir. Tasarımcılar birçok ilham kaynağını ve bilgiyi, dev.windowsphone.com/design ve design.windows.com adreslerinde bulabilirler.

Genel deneyim için önemli olan şey, başlangıç ekranında bulunan küçük widget'lar olan 'canlı desenler'dir. Bunları programlanabilir şekilde veya push bildirimlerini kullanarak uzaktan güncelleyebilirsiniz.

Windows RT ve Windows Phone Arasındaki Kod Paylaşımı

Hem Windows Phone 8 hem de Windows 8 aynı kerneli kullandığı için, her iki işletim sisteminde de birçok API bulunmaktadır. Bazı API'lerin, Windows Phone içerisinde mevcut olduğunu ancak (tam olarak) uyarlanmadığını göz önünde bulundurun. [HttpClient](http://nuget.org/packages/Microsoft.Net.Http)¹⁴ gibi popüler API'ler için bazı Nuget uyumluluk paketleri bulunmaktadır. Bunun yanı sıra, BCL

¹⁴ nuget.org/packages/Microsoft.Net.Http

paketi¹⁵ eski Windows Phone 7 platformunda bulunan "async" ve "await" desenlerini kullanmanıza fırsat vermektedir.

Bir Portable Class Library Project (Taşınabilir Sınıf Kütüphanesi Projesi) kullanarak çeşitli platformlar arasında paylaşılmış olan kütüphaneler yaratabilirsiniz. Bu projeler, Visual Studio Professional veya daha üstü sürümlerde oluşturulabilir; ücretsiz Expression sürümleri bunu desteklememektedir.

Ayrıca projeler arasında kaynakları bağlayarak Visual Studio içerisinde doğrudan kodları paylaşabilirsiniz.

Platforma Entegrasyon

Windows Phone da Windows RT de, uygulama listesinin ötesinde platforma derin bir entegrasyonu destekler.

Her iki sistem de, push mesajlarıyla güncellenebilen, isteğe göre ekstradan bilgi gösterebilen 'canlı desenler' (live tiles) destekler. Her iki sistemde de kilit ekranındaki görüntüyü kontrol edebileceğiniz kilit ekranı uygulamalarını kullanabilirsiniz.

Örnek olarak, Windows Phone'da lenses¹⁶ adı verilen kamera uzantısı uygulamalarını yaratabilir, müzik merkezini¹⁷ genişletebilir veya arama deneyimine¹⁸ entegre edebilirsiniz.

Daha geniş bir bilgi için entegrasyon belgelendirmesini¹⁹ okuyabilirsiniz.

Windows RT üzerinde benzer görevleri gerçekleştiren sözleşmeler²⁰ ile belirli dosya uzantılarını idare edebilir,

¹⁵ blogs.msdn.com/b/bclteam/archive/2012/10/22/using-async-await-without-net-framework-4-5

¹⁶ msdn.microsoft.com/library/windowsphone/develop/jj206990

¹⁷ msdn.microsoft.com/library/windowsphone/develop/ff769558

¹⁸ msdn.microsoft.com/library/windowsphone/develop/hh202957

¹⁹ msdn.microsoft.com/library/windowsphone/develop/hh202969

²⁰ msdn.microsoft.com/library/windows/apps/hh464906

paylaşım içeriğine katılım sağlayabilir ve çok daha fazlasını yapabilirsiniz.

MVVM

Diğer platformlardan gelen uygulama geliştiricileri için XAML'nin veri ilişkilendirme (data binding) konseptleri yeni olacaktır. Her bir sayfa için, o sayfanın verilerini içeren bir View Model olmalıdır. View'ın kendisi yalnızca UI'yi açıklar; asıl görüntülenen veriler, View Model'den alınan verileri içermektedir. Model View View Model (MVVM) konsepti, karmaşık uygulamaların geliştirilmesini ciddi anlamda kolaylaştırmaktadır.

Oyun Motorları

Native uygulama becerileri sayesinde, Windows Phone 8 ve Windows RT için bazı oyun motorları mevcuttur. Bunlar:

- Cocos2d-x²¹
- Havok²²
- Marmalade²³
- OGRE²⁴
- Unity 3D²⁵

²¹ cocos2d-x.org/wiki/Windows_Phone_8_Environment_Setup

²² havok.com/products/havok-windows-ecosystem

²³ developer.madewithmarmalade.com/develop/supported-platforms

²⁴ ogre3d.org/2012/10/30/ogre-now-supports-windows-phone-8

²⁵ unity3d.com/pages/windows

Hizmetler

Uygulamanızın canlı desenlerini güncellemeniz için Push bildirimleri²⁶ mevcuttur. Ayrıca ücretsiz olan SkyDrive bulut alanını kullanmak ve uygulamanız içerisindeki diğer Windows Live hizmetlerine²⁷ entegre etmek isteyebilirsiniz. Bunun yanı sıra çok sayıda üçüncü parti seçeneği²⁸ mevcuttur.

Çoklu Görev ve Uygulama Yaşam Döngüsü

Windows Phone, uygulamaları arkaplanda askıya alan ve daha hızlı uygulama değişimine fırsat veren sınırlı çoklu görev biçimi içermektedir. Uygulama açıldıktan sonra arkaplanda çalışabilen tek işlemler ses çalma, yer izleme ve dosya transferidir. Uygulamalar ayrıca belirli aralıklarla arkaplanda çalışacak kodları zamanlayabilirler (Background Agents olarak da bilinir). Background Agents, sınırlı sayıda kaynak kullanabilir ve eğer OS, telefonun kaynakları koruması gerektiğini belirlerse, durdurulabilir veya atlanabilir.

Arkaplanda askıya alınan uygulamalar, eğer OS kaynakların başka bir yerde kullanılması gerektiğini belirlerse, otomatik olarak kapatılabilir.

Hiç kapanmamış olan bir uygulamanın görünümünü yaratmak için Windows Phone'un Tombstoning²⁹ adlı, iyi bir şekilde belgelenmiş olan uygulama yaşam döngüsü bulunmaktadır. Tombstoning'i mümkün kılmak için Windows Phone altyapısı, uygulama yaşam döngüsünün farklı aşamaları sırasında (ön belleğe alma ve veri ve UI durumlarını yenilemek gibi) eylemler uygulamak için gerekli olan parçaları sağlamaktadır. Windows

²⁶ msdn.microsoft.com/library/windowsphone/develop/ff402558

²⁷ msdn.microsoft.com/live

²⁸ dev.windowsphone.com/en-us/featured/partners

²⁹ msdn.microsoft.com/library/windowsphone/develop/ff817008

Phone 8 ile geliştiriciler için yeni bir "hızlı uygulama devam ettirme" özelliği de bulunmaktadır

Windows Store uygulamaları benzer yaşam döngüsüne³⁰ sahiptir.

Test ve İstatistikler

Birim (unit) testler Visual Studio içerisine entegre edilmiştir; yapmanız gereken tek şey projenize ilave bir Unit Test projesi oluşturmak ve test etmek istediğiniz projeleri referans göstermektir.

Davranış odaklı (behaviour-driven) geliştirme için Expensify'ın³¹ Windows Phone Test Framework'ü mevcuttur.

Localytics³² and Flurry³³, Windows Phone ve Windows RT ile uyumlu istatistik araçları ve hizmetleri sunar. Geliştiriciler ayrıca Google Analytics gibi üçüncü şahıs izleme hizmetlerine bağlanmak için Silverlight Analytics Framework'ü³⁴ kullanabilirler. Bunlar, Visual Studio içerisinde mevcut olan sağlam performans izleme araçlarıdır.

Dağıtım

Uygulamalarınızı Microsoft Windows Store aracılığıyla dağıtabilirsiniz. Uygulama içeriği, Apple App Store'daki gibi incelenir ve sınırlandırılabilir, Microsoft Dev Center³⁵ adresinde bulunan ayrıntılı uygulama sunumu kılavuzu

³⁰ msdn.microsoft.com/library/windows/apps/hh464925

³¹ github.com/Expensify/WindowsPhoneTestFramework

³² localytics.com/docs/sdks-integration-guides/#winphone7

³³ flurry.com/flurry-analytics.html

³⁴ msaf.codeplex.com

³⁵ dev.windowsphone.com

sunmaktadır. Geliştirici araçları ücretsiz olarak sunulmasına rağmen, yazılımın Windows Store vasıtasıyla sunulması için ücretli bir Store hesabının alınması gereklidir. Şu anda bir geliştirici hesabının ücreti bireyler için yıllık 19 dolar ve şirketler için yıllık 99 dolardır. Bu ücret, DreamSpark³⁶ programı içerisindeki öğrencilerden alınmamaktadır. Store ayrıca zaman-sınırlı beta dağıtımı ve girişimler için şirket merkezi³⁷ sunmaktadır. Uygulamalarınızı sunmadan önce onları yerel olarak test edebilmek için Visual Studio'ya entegre edilmiş olan Windows Phone Store Test Kitini veya Windows Sertifikasyon Kitini kullanabilirsiniz.

Uygulamanızın geliri 25.000 dolardan fazla olduğundan %70'lik standart gelir payı %80'e çıkmaktadır. Windows Store, küresel anlamda her yere ulaşabilmeniz için 200'den fazla ülke ve bölgeyi ve 100'den fazla dili desteklemektedir.

Uygulamalar cihaz tarafından değil, müşteri tarafından yönetilir. Böylece bir kullanıcı uygulamanızı, bir masaüstü PC veya bir tablet gibi çok farklı platformlar üzerinde kullanabilir.

Ücretli uygulamalar için Windows Phone altyapısı, uygulamanızın "deneme modunda" olup olmadığını belirleme becerisini ve kullanımı buna göre ayarlama özelliğini sunar. Microsoft özellikle süreli sınırlamayı (30 dakikalık deneme gibi) tavsiye etmemekte, bunun yerine özellikleri sınırlamayı³⁸ tavsiye etmektedir.

Reklam bazlı gelir kaynağı için birden fazla seçenek bulunmaktadır. Microsoft'un kendi Microsoft Reklam Kontrolü³⁹ (şu anda 18 ülkede mevcut) bulunur. Bunun yanı sıra

³⁶ www.dreamspark.com

³⁷ msdn.microsoft.com/library/windowsphone/develop/jj206943

³⁸ msdn.microsoft.com/library/windowsphone/develop/ff967558

³⁹ advertising.microsoft.com/mobile-apps

Smaato⁴⁰, inneractive⁴¹, AdDuplex⁴² ve Google⁴³ gibi örnekler de bulunmaktadır. Bunların hepsi alternatif reklam çözümleri sunmaktadır. Gelir kaynağı hakkında daha fazla bilgi için bu kılavuzda bulunan ilgili bölüme bakın.

Daha Fazla Bilgi

Haberler, geliştirici araçları ve forumlar için dev.windowsphone.com ve dev.windows.com adreslerini ziyaret edin.

Geliştirme ekibi, haberleri ve güncellemeleri, blogs.windows.com/windows_phone/b/wpdev ve blogs.windows.com/windows/b/appbuilder adreslerinde bulunan bloglara veya @wpdev Twitter hesaplarına koymaktadır. Geliştirici ve tasarımcı kaynaklarına ilişkin geniş bir kaynağı, windowsphonegeek.com ve reddit.com/r/wpdev adreslerinde bulabilirsiniz.

Şu anda, içerik menüsü, tarih belirleyici ve diğerleri gibi Windows Phone SDK'sına dahil edilmemiş olan birçok dahili OS kontrolü bulunmaktadır. Bu kontroller, phone.codeplex.com adresinde bulunabilen Windows Phone için Phone Araç Kitinin bir parçası olarak mevcuttur. Diğer popüler Windows Phone projeleri arasında coding4fun.codeplex.com, cimbolino.org ve mvvmlight.codeplex.com bulunmaktadır. XAML temelli kullanıcı arayüzlerinin visual tree, bağlama (binding) ve özelliklerini denetlemek için xamlspy.com mevcuttur.

channel9.msdn.com adresinde Windows ve Windows Phone ile ilgili birçok videolu eğitimi bulabilirsiniz.

Örnek kodu, code.msdn.microsoft.com/windowsapps, code.msdn.microsoft.com/wpapps adreslerinde, çeşitli codeplex.com

⁴⁰ smaato.com

⁴¹ inner-active.com

⁴² adduplex.com

⁴³ developers.google.com/mobile-ads-sdk/

projelerinde ve msdn.microsoft.com/library/windows/apps/br211375'de mevcut olan baştan sona örneklerde bulabilirsiniz. Uygulama geliřtiricileri Windows RT uygulamalarını planlamaya, tasarlamaya ve geliřtirmeye iliřkin bir yol haritası msdn.microsoft.com/library/windows/apps/xaml/br229583 adresinde sunulmaktadır.

Eęer iOS'tan veya Android'den geęiř yapıyorsanız, dev.windowsphone.com/en-us/featured/porting ve msdn.microsoft.com/library/windows/apps/dn436165 adreslerinde çeřitli bilgiler bulabilirsiniz.



Çapraz Platformu Seçmek

Ne kadar çok platform, o kadar az zaman: Bu söz, mobil alanda içinde bulunduğumuz durumu çok iyi bir şekilde özetlemektedir. Aralarından seçim yapabileceğiniz onlarca platform bulunmaktadır: Android, BlackBerry 10, Firefox OS, iOS, Tizen, Windows 8 ve Windows Phone en önemli akıllı telefon ve tablet platformları olmaya devam ederken Brew MP ve Java ME, temel özellikli (feature phone) telefonlarda üstünlük kurmaya devam edecektir (öneme göre değil alfabetik sıraya göre sıralanmıştır).

Birçok uygulama sponsoru, Queen'in ünlü şarkı sözünde belirttiği gibi geliştiriciye "hepsini istiyorum, hepsini istiyorum ve hepsini şimdi istiyorum!" şeklinde seslenecektir. Bu yüzden, birden fazla paralel geliştirme ekibi veya bir çapraz platform stratejisi arasında bir seçim yapıp parayı buraya yönlendirmek gereklidir.

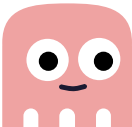
Mobil Platformları Arasındaki Temel Farklılıklar

Eğer uygulamanızı farklı platformlara sunmak istiyorsanız, bazı engelleri aşmanız gerekecektir. Bazı zorluklar diğerlerine göre daha kolay aşılabilmektedir:

Programlama Dili

Şimdiye kadar çoğu mobil platformun yayınladığı SDK'ların, platformun desteklediği programlama dilleriyle uygulama geliştirmenize olanak sağladığını gördünüz.

Ancak bu diller genellikle kök dillerinin bazı ailelerine bağlıdır ve aşağıdaki tablo bunların ve desteklendikleri platformun kısa bilgilerini içermektedir.



Dil	1. Sınıf Vatandaş	2. Sınıf Vatandaş
Action-Script	BlackBerry 10, BlackBerry PlayBook OS (QNX)	Hiçbiri
C, C++	BlackBerry 10, Brew MP, Sailfish OS, Windows RT, Windows Phone 8	Android (Kısmen, NDK kullanarak), iOS (kısmen)
C#	Windows 8, Windows Phone	Hiçbiri
Java	Android, BlackBerry, Java ME cihazları	Hiçbiri
JavaScript	BlackBerry PlayBook OS, Firefox OS, Tizen, Windows 8	BlackBerry (WebWorks), Nokia (WRT)
Objective-C	iOS	Hiçbiri

- 1 Yerel olarak destekleyen platformlar (Uygulama yazmak için ilk ya da tek dil)
- 2 Yerel olan dile alternatif olarak destekleyen platformlar (Genelde yerel dilindeki kadar platformun özelliklerine erişemezsiniz)

Çapraz platform geliştirme çatıları (framework), farklı yollar kullanarak programlama dil bariyerlerini aşmaktadırlar:

- **Web Teknolojileri:** Bu yaklaşım, native uygulamalarda gömülü olan 'webview'lar vasıtasıyla web teknolojileri için çoğu platformun doğrudan destek sunduğu gerçeğini kullanmaktadır. HTML ve CSS gibi, bu yaklaşım da JavaScript'i kullanmaktadır.
- **Yorumlama:** Burada geliştirme çatısı, ortak veya çatıya özel bir dili yorumlayan her bir platform için bir motor sunar. Örnek olarak, oyun geliştirmesi için popüler bir seçenek, Lua komutlarıdır.
- **Çapraz Derleme:** Çapraz platform geliştirme çatılarının neticede yaptıkları şey çapraz derlemedir ve bu aynı

zamanda en karmaşık teknik çözümdür. Uygulamayı bir dilde yazmanızı sağlar. Bununla beraber platformun native diline uygun bir çevrim yapılı ve native çalışma zamanı hızı sağlanır.

Çoğu geliştirme çatısı, bir cihazın konum belirleme becerileri gibi belli platform ve cihaz özelliklerine erişim sağlamada yardımcı olan çapraz platform API'ları sunar. SMS mesajlaşması gibi özellikler için cihazdan bağımsız olan ağ API'larını¹ kullanabilirsiniz.

OS Sürümleri

Platformlar evrim geçirir ve eninde sonunda avantaj sağlamak istediğiniz, sürüme özel özellikler olacaktır. Bu durum uygulamanıza başka bir karmaşıklık katmanı ekler ve ayrıca çapraz platform araçlarının bir dezavantajını ortaya koyar: bazen yeni bir OS sürümü piyasa çıktığında çapraz platform araçları geride kalırlar.

UI ve UX

Çapraz platform yaklaşımı için zor bir engel de bireysel platformlardaki hakim Kullanıcı Arayüzü (UI) ve Kullanıcı Tecrübesi (UX) kalıpları ile ortaya çıkar.

Farklı platformlarda aynı şekilde çalışan, görünüşü güzel bir UI yaratmak nispeten kolaydır. Ancak bu tür bir yaklaşım, kullanıcı deneyimini ciddi anlamda arttırabilen ve yalnızca o platforma özel, önemli UI ayrıntılarını kaçırmınıza neden olabilir. Bir yandan platformların tasarım felsefesi söz konusu olduğunda çeşitli farkları da yoksayacaktır: Çoğu platform, uygulamaların gerçek hayattaki emsallerine benzediği gerçekçi

¹ www.developergarden.com/apis/

tasarımın peşinden koşsa da, Windows Phone'un Metro arayüzü, içeriğe odaklanılan, çeşitli süslerle karmaşık hale getirilmeyen "orijinal dijitallik" fikrine ulaşmaya çalışmaktadır. Bir çapraz platform UI'nın ortaya çıkardığı başka bir zorluk ise kullanıcıların alışık oldukları native UI'dan farklı bir şekilde çalışması ve uygulamaların kullanıcılar için "çalışmamasıdır". Basit bir örnek olarak, belli bir platformda geri tuşu gibi bir donanımsal özelliğin doğru bir şekilde desteklenmemesidir. Diğer bir zorluk ise, aynı görünen, ancak aynı şekilde çalışmayan native UI unsurlarının taklit edilmesinden doğan "tekinsiz vadi"² sendromudur. Bu sendrom, gerçeğine benzemeyene karşı duyulan iticiliği ifade eder. Bu yüzden native kontrolleri taklit etmek yerine, native olmayanları veya yalnızca 'gerçek olanları' kullanmanız gerekir.

Doğrudan nihai kullanıcıları (B2C) hedefliyorsanız, iş kullanıcılarını (B2B) hedeflerken kullandığınız yaklaşıma kıyasla, platforma özel kullanıcı deneyimini daha fazla göz önünde bulundurmanız gerekir. Her durumda, UI ve UX'i her bir platforma göre özelleştirmek ve tasarlamak, uygulama geliştirme çabanızın büyük bir kısmını oluşturabilir ve muhtemelen, çapraz platform stratejisinin en zorlu unsuru olacaktır.

Ana Ekran ile Entegrasyon Desteği

Uygulamanızın cihazların ana ekranına entegre edilmesi platforma göre farklılık gösterecektir: iOS'ta yalnızca, uygulamanızın simgesine numara ile bir rozet ekleyebilirsiniz. Windows Phone'da, ana ekrana yapısal bilgiler içeren canlı desenler yaratabilirken Android'de, rastgele veriler görüntüleyen ve herhangi bir görseli kullanabilen her türlü ayrıntıya sahip bir widget ekleyebilirsiniz.

2 <http://www.acikbilim.com/2011/12/dosyalar/tekinsiz-vadi.html>

Masaüstü entegrasyonunu kullanmak, kullanıcılarınızla olan etkileşimi ciddi bir şekilde arttırabilir.

Çoklu Görev Desteği

Çoklu görev desteği, aynı anda arkaplan hizmetlerinin ve birden fazla uygulamanın çalışmasını sağlar. Çoklu görev, işletim sistemleri üzerinde farklı bir şekilde gerçekleştirilen başka bir özelliktir. Android, BlackBerry ve Sailfish OS'leri üzerinde aynı anda birden fazla uygulama çalıştırabileceğiniz arkaplan hizmetleri bulunur: Android'de kullanıcının uygulamalardan çıkması mümkün değildir; bu işlem, kaynaklar azaldığında otomatik olarak OS tarafından gerçekleştirilir. iOS ve Windows Phone'da, uygulamadan çıkış sonrasında çalışmaya devam eden sınırlı sayıda arkaplan görevi bulunur. Böylece, eğer arkaplan hizmetleri uygulamanızın sunduklarını geliştirebilecek düzeyde ise, telefonun özelliklerine tam olarak erişim sağlama fırsatı verdiğinden emin olmak amacıyla çapraz platform stratejilerini dikkatli bir şekilde değerlendirmelisiniz.

Batarya Tüketimi ve Performans

Çoklu görev ile yakın bir ilişki içerisinde olan başka bir unsur ise uygulamanızın batarya kullanımudur.

Her iki senede bir CPU gücü neredeyse iki katına çıkarken (Moore'un yasası, transistör sayısının her 18 ayda bir iki katına çıktığını belirtmektedir), bunun tam tersine şarj kapasitesi yalnızca her yedi senede bir iki katına çıkmaktadır. Akıllı telefonların şarjda uzun zaman geçirmesinin nedeni budur. Bir çapraz platform soyutlama katmanı içerisinde platforma ne kadar yakın olursanız, uygulamanızın şarj tüketimini ve performansını o kadar iyi bir şekilde kontrol edebilirsiniz. Temel bir kural olarak, uygulamanızın tek seferde ne kadar uzun çalışması gerekirse o kadar az soyutlamaya ihtiyaç duyarsınız.

Ayrıca bazı platformların mükemmel bir performans

çeşitliliği bulunmaktadır; bunlardan öne çıkanı Android'dir - Android cihaz aralığında çok yavaştan süper hızlıya cihazlar bulabilirsiniz.

Push Hizmetleri

Push hizmetleri, uygulamanıza çalışmadığı durumlarda bile çalışıyormuş hissi vermek için iyi bir yöntemdir. Örnek olarak bir sohbet uygulamasında, kullanıcıya gelen sohbet mesajlarını bir push mekanizması kullanarak iletebilirsiniz. Yeniden belirtmek gerekir ki push hizmetlerinin çalışma şekli ve kullandıkları protokoller her bir platformda farklı olacaktır. Örnek olarak mevcut olan veri boyutu iOS'ta 256 byte ve BlackBerry'de 8 kb'dır. Urban Airship³ gibi hizmet sağlayıcılar, çok çeşitli platformlar üzerinde hizmet sunmaktadırlar.

Uygulama İçi Satın Alma

Uygulama içerisindeki satın alma mekanizmaları, uygulamanız içerisinden çeşitli hizmetler veya mallar satmanıza fırsat verir. Bunun farklı platformlarda farklı bir şekilde çalışacağı yine belirtilmelidir. Daha fazla ayrıntı için gelir elde etme bölümüne bakın.

Uygulama İçi Reklam

Mobil uygulamalar içerisinde reklamlar görüntülemek için farklı seçenekler bulunmaktadır; bunlardan bazıları satıcıdan bağımsız olan üçüncü şahıs çözümleridir. Ancak platforma özel reklam hizmetleri, daha iyi gelirler ve daha iyi bir kullanıcı deneyimi sunmaktadır. Bu satıcı hizmetlerinin platformlar üzerinde farklı çalıştıklarının altı çizilmesi gerekir. Gelir elde etme bölümü, yine bu konu hakkında daha fazla bilgiyi sunacaktır.

3 urbanairship.com

Çapraz Platfom Stratejileri

Bu bölüm, uygulamalarınızı farklı platformlara uyarlamak için kullanabileceğiniz stratejilerin bazılarını açıklamaktadır.

Doğrudan Destek

Her bir hedef platform için özel bir ekip kurarak farklı platformlara destek verebilirsiniz. Bu yöntem çok fazla kaynak kullansa da, büyük ihtimalle her sistemde en iyi entegrasyon ve kullanıcı deneyimi yaklaşımını sunacaktır. Kolay bir giriş yöntemi tek bir platformla başlayıp, uygulamanız gerçek hayatta kendini kanıtladıktan sonra diğer platformlara geçiş yapmaktır. Bileşen kütüphaneleri, native geliştirmeyi hızlandırmanızda size yardımcı olabilir. Popüler örnekler aşağıda verilmiştir:

Bileşen Kütüphanesi	Hedef Platformlar
cocoacontrols.com	iOS
chupamobile.com	Android, iOS
verious.com	Android, iOS, HTML5, Windows Phone
windowsphonegeek.com/Marketplace	Windows Phone

Varlık Paylaşımı

Farklı platformlar için birden fazla ekibi yönettiğinizde, bazı uygulama yapılarını paylaştığınızda bile efordan tasarruf edebilirsiniz:

- **Konsept ve varlıklar:** Bunu çoğunlukla otomatik olarak yapacaksınız: uygulama fikrini ve konseptini; UI akışını; uygulamanın girdilerini, çıktılarını ve tasarım, tasarım

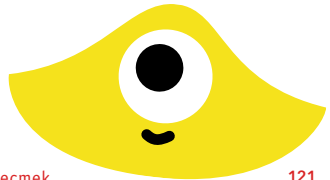
varlıklarını paylaşın (platforma özel UI yapılarını destekleme ihtiyacını gözardı etmeyin).

- **Veri yapıları ve algoritmalar:** Platformlar arasında veri yapılarını ve algoritmaları paylaşarak bir adım öne geçin.
- **İş modelinin kod paylaşımı:** Çapraz platform derleyicilerini kullanarak platformlar arasında iş modellerini de paylaşabilirsiniz. Alternatif olarak platformlar arasında bir tercüman veya sanal bir makine ve tek bir ortak dil kullanabilirsiniz.
- **Eksiksiz Soyutlama:** Bazı çapraz platform araçları, iş modelinizi tamamen soyutlamanıza, farklı platformlar için uygulamanızı görüntülemenize ve kontrol etmenize fırsat vermektedir.

Oyuncu ve Sanal Makineler

Oyuncu konseptleri farklı platformlar üzerinde tipik olarak ortak bir API seti sunmaktadır. İyi bilinen örnekler arasında Flash, Java ME ve Lua bulunur. Bu yaklaşım, geliştirme sürecini çok kolay bir hale getirir. Ancak yeni özellikler açısından platform sağlayıcısına bağlı kalırsınız ve burada ortaya çıkan bir zorluk, bu özellikler yalnızca tek bir platformda çıktığında kendini göstermektedir. Oyuncu konseptleri, sıklıkla farklı platformlardaki implementasyonlar arasında müşterekliği sağlamak adına sunulan özellikler için bir “en küçük ortak payda” yaklaşımı kullanmaktadır. Applause⁴ gibi üretici konseptleri,

⁴ applause.github.com



oyuncu konseptini bir adım öteye taşıyabilir; bunlar genellikle alan adına özeldir ve herhangi bir veri içerisinde uygulamalar oluşturunuzun sağlar. Programlanabilir çözümlerle kıyaslandığında genellikle esneklik eksiklikleri bulunur.

Çapraz Dil Derlemesi

Çapraz dil derlemesi, tek bir dilde kodlayıp ardından platforma özel dile çevrilmeyi mümkün kılar. Performans açısından bu genellikle en iyi çapraz platform çözümdür ancak native uygulamalar ile kıyaslandığında performans farklılıkları olabilir. Bu farklılıklar belirli programlama yapıları optimal şekilde kaynak dilden hedef dile çevrelemediği zaman ortaya çıkabilir.

Çapraz dil derlemesi için üç yaygın yaklaşım bulunmaktadır: kaynaktan kaynağa doğrudan çevirme, kaynak kodunu orta seviyedeki soyut dile dolaylı çevirme ve platformun ikili (binary) formatına doğrudan derleme. Dolaylı yaklaşım tipik olarak daha az sayıda okunabilir kod sunar. Hedef platform üzerinde geliştirmeye devam etmek istediğinizde ve başlangıç noktası olarak çevrilen kaynak kodu kullanmak istediğinizde ortaya çıkma ihtimali bulunan bir sorundur.

Hibrit Web Uygulamaları

Mevcut web uygulama geliştirme çatılarının bir kısmı aşağıdaki tabloda verilmiştir. Bu çatılar ile çevrimdışı yetenekler de dahil olmak üzere neredeyse gerçek uygulamalar gibi işlev gören web uygulamaları yaratabilirsiniz. Ancak platform entegrasyonu, performans ve diğer unsurlar söz konusu olduğunda teknolojilerin sınırları olduğunu unutmayın. Mobil web geliştirmesi hakkında daha fazla bilgi için web bölümünü okuyun.

Web Uygulaması Çözümü	Lisans	Hedef Platformlar
Chrome Apps developer. chrome.com/apps	BSD	Android, Mac, Windows
jQuery Mobile www. jquerymobile.com	MIT ve GPL	Android, BlackBerry, Firefox, iOS, Windows Phone
Sencha Touch www.sencha. com/products/touch	GPL	Android, BlackBerry, iOS, Windows Phone
The M Project the-m- project.org	MIT ve GPL	Android, BlackBerry, Firefox, iOS, Windows

Genellikle donanım özelliklerine ve native UI elementlerine erişiminiz yoktur. Bu yüzden bizce bu listedekiler “gerçek” çapraz platform çözümü olarak sayılmamalıdır: bu yüzden bu çözümler, bölümün sonunda bulunan tabloda listelenmemiştir.

Hibrit web geliştirme, bir native uygulama içerisinde bir Webview’ı gömme anlamına gelmektedir. Bu yaklaşım, uygulamalarınızın web parçaları içerisinde native fonksiyonelliğe erişim sağlamanıza fırsat verir. Native kodu ayrıca uygulamanızın performans veya kullanıcı deneyimi açısından önemli olan unsurları için kullanabilirsiniz. Hibrit uygulamalar, seçtiğiniz platform üzerinde web geliştirme parçalarını yeniden kullanmanıza izin verir – iyi bilinen bir hibrit web geliştirme çatısı örneği PhoneGap’tir.

ANSI C

HTML ve web programlaması, çok yüksek bir soyutlama seviyesinden başladığı için ANSI C’yi kullanarak bunun tam tersi bir yolu izleyebilirsiniz. ANCI C kodunu, Android, Black-Berry 10, iOS ve Windows 8/Windows Phone gibi bütün önemli platformlarda çalıştırabilirsiniz. Bu yaklaşımdaki temel sorun,

ANSI C içerisinde platforma özel API'lara veya UI kontrollerine ulaşamıyor olmanızdır. C kullanımı genellikle, ses kodlayıcılar gibi karmaşık algoritmalar için geçerlidir. Buna karşılık gelen kütüphaneler daha sonra, belli bir platform için her bir uygulama projesinde kullanılabilir.

Çapraz Platform Uygulama Geliştirme Çatıları

Birçok çapraz platform çözümü bulunmaktadır, bu yüzden bunların hepsinden bahsetmek çok zordur. Buna siz parçalanma diyebilirsiniz; biz rekabet ediyoruz. Ancak uyarmamız gerekir: bütün çözümler hakkında bilgimiz bulunmuyor, bu yüzden eğer sizin bir çözümünüz olursa lütfen adresinden bizimle iletişime geçerek bunu bildirin. Bir geliştirme çatısının, listelenecek olan en az iki mobil platformu desteklemesi gerekmektedir.

Çapraz platform araçlarını değerlendirirken sormanız gereken bazı sorular bulunmaktadır. Bunların hepsi konu ile ilgili olmayabilir, bu yüzden elinizdeki seçenekleri uygun bir şekilde değerlendirmeniz gerekir. İlk önce, uygulama fikrinizi, içeriğinizi, hedef kitlenizi ve hedef platformlarınızı dikkatli bir şekilde inceleyin. Ayrıca, farklı platformlarda aktif olan rakiplerinizi, pazarlama bütçenizi ve geliştirme ekibinizin teknik bilgisini göz önünde bulundurmanız gerekir.

Research 2 Guidance ayrıca çeşitli çapraz platform araçları hakkında bir rapor⁵ yayınlamıştır.

- Çapraz platforma aracı zinciriniz nasıl çalışıyor? Hangi programlama dilini ve API'ı kullanabilirim?
- Platforma özel fonksiyonelliğe erişim sağlayabilir miyim? Nasıl?

5 research2guidance.com/cross-platform-tool-benchmarking-2013

- Native UI bileşenlerini kullanabilir miyim? Nasıl?
- Devam eden geliştirme projemin temeli olarak platforma özel yapıları kullanabilir miyim? Çevrilen/oluşturulan kaynak kodu neye benzer?
- Masaüstü entegrasyonu mevcut mu?
- Çoklu görevleri kontrol edebilir miyim? Arkaplan hizmetleri bulunuyor mu?
- Bu çözüm push hizmetleri ile nasıl çalışıyor?
- Uygulama için satın alma ve uygulama içi reklam özel-liklerini nasıl kullanabilirim?
- Geliştirme çatısı, yeni OS sürümlerine ayak uydurabiliyor mu?

Çözüm	Lisans	Girdi	Çıktı
Akula verivo.com	Ticari	(Visual)	Android, BlackBerry, iOS, Windows Phone
Application Craft applicationcraft.com	Ticari	HTML, CSS, JavaScript	Android, BlackBerry 10, iOS, Windows Phone, mobil siteler
Codename One codenameone.com	Ticari	Java	Android, BlackBerry, iOS, J2ME, Windows Phone
Corona coronalabs.com (Corona Labs)	Ticari	JavaScript	Android, iOS, Kindle, Nook, Windows Phone, Windows RT

Çözüm	Lisans	Girdi	Çıktı
J2ME Polish j2mepolish.org (Enough Software)	Açık Kaynaklı + Ticari	Java ME, HTML, CSS	Android, BlackBerry, J2ME, PC
Flash Builder adobe.com/devnet/devices.html (Adobe)	Ticari	Flash	Android, BlackBerry Tablet OS, iOS, PC
Feedhenry feedhenry.com	Ticari	HTML, CSS, JavaScript	Android, iOS, HTML5, Windows Phone
Kony One kony.com/products/develop/studio	Ticari	HTML, CSS, JavaScript, RSS	Android, BlackBerry, iOS, J2ME, Windows Phone, PC, Web
LiveCode runrev.com (RunRev)	Ticari	English-like	Android, iOS, PC ve Web
M2Active service2media.com (Service2Media)	Ticari	Drag and Drop + Lua	Android, BlackBerry, iOS, Windows Phone
MobiForms mobiforms.com (MobiForms)	Ticari	Drag and Drop + MobiScript	Android, iOS, PC, Windows Mobile
MoSync mosync.com	Açık Kaynaklı + Ticari	C/C++, HTML5/JS	Android, BlackBerry, iOS, J2ME, Windows Phone
NeoMAD neomades.com	Ticari	Java	Android, BlackBerry, iOS, J2ME, Windows Phone

Çözüm	Lisans	Girdi	Çıktı
Orubase orubase.com	Ticari	ASP .NET MCV	Android, iOS, Windows Phone, Windows RT
PhoneGap/Cordova phonegap.com (Adobe/Apache)	Açık Kaynaklı	HTML, CSS , JavaScript	Android, BlackBerry 10, iOS, Windows Phone
Qt qt.digia.com (Digia)	Açık Kaynaklı + Ticari	C++	Android, BlackBerry 10, iOS, Sailfish OS, Windows RT
Rhodes motorolasolutions.com/US-EN/ RhoMobile+Suite/ Rhodes (Motorola)	Açık Kaynaklı + Ticari	Ruby, HTML, CSS, JavaScript	Android, BlackBerry, iOS, Windows Phone
Titanium appcelerator.com (Appcelerator)	Açık Kaynaklı	JavaScript	Android, iOS, Tizen, Mobile Web
trigger.io trigger.io (Triggger Corp)	Ticari	HTML5, JavaScript	Android, iOS, Windows Phone
webinos webinos.org	Açık Kaynaklı	JavaScript	Android, BlackBerry, iOS, PC, TV
webMethods Mobile Designer(formerly Metismo Bedrock) metismo.com (Software AG)	Ticari	Java ME	Android, BlackBerry, brew, Consoles, iOS, PC, Windows Phone, Windows Mobile

Çözüm	Lisans	Girdi	Çıktı
Xamarin xamarin.com	Ticari	C#	iOS, Android, Windows Phone, PC
XDK xdk.intel.com (Intel)	Ücretsiz	HTML, CSS, JavaScript	Android, iOS, Windows Phone, Windows RT
XML VM xmlvm.org	Açık Kaynaklı + Ticari	Java, .NET, Ruby	C++, Java, JavaScript, .NET, Objective-C, Python

Çapraz Platform Oyun Motorları

Oyunlar çok daha fazla içerik odaklıdır ve platformla çok derin bir şekilde entegre olmaları gerekmez. Bu yüzden, çapraz platform geliştirmesi genellikle oyunlar için uygulamalara kıyasla daha çekicidir.

Çözüm	Lisans	Girdi	Çıktı
Cocos 2D cocos2d-x.org	Açık Kaynaklı	C++, HTML5, JavaScript	Android, BlackBerry, iOS, Windows 8, Windows Phone
Corona coronalabs.com (Corona Labs)	Ticari	Lua	Android, iOS, Kindle, nook, Windows Phone, Windows RT

Çözüm	Lisans	Girdi	Çıktı
EDGELIB edgelib.com (elements interactive)	Ticari	C++	Android, iOS, PC
Esenthel esenthel.com (elements interactive)	Ticari	C++	Android, iOS, PC
GameSalad gamesalad.com	Ticari	Drag and drop	Android, iOS, Tizen, Windows Phone, PC, web
Gideros Mobile giderosmobile.com	Ticari	Lua	Android, iOS
id Tech 5 idsoftware.com (id)	Ticari	C++	Consoles, iOS, PC
Irrlicht irrlicht.sourceforge.net	Açık Kaynaklı	C++	Android & iOS with OpenGL-ES version, PC
IwGame drmpop.com/index.php/iwgame-engine	Açık Kaynaklı	C++	Android, BlackBerry Playbook OS, iOS, PC
Marmalade madewithmarmalade.com (Ideaworks3D)	Ticari	C++, HTML5, JavaScript	Android, BlackBerry 10, iOS, LG Smart TV, Windows Phone, Windows RT
Moai [Ticari]	Lua	Android, iOS, PC, Web	MonoGame monogame.codeplex.com
Açık Kaynaklı	C#, XNA	Android, iOS, PC, Windows Phone, Windows RT	Ogre 3D ogre3d.org

Çözüm	Lisans	Girdi	Çıktı
Açık Kaynaklı	C++	Windows Phone, Window RT, PC	orx orx-project.org
Açık Kaynaklı	C, C++, Objective-C	Android, iOS, PC	ShiVa 3D stonetrip.com
Ticari	C++	Android, BlackBerry 10, iOS, PC, Consoles	SI02 sio2interactive.com (sio2interactive)
Ticari	C, Lua	Android, iOS, PC	Unigine unigine.com (Unigine corp.)
Ticari	C++, Unigine-Script	Android, iOS, PC, PS3	Unity3D unity3d.com (Unity Technologies)
Ticari	C#, JavaScript, Boo	Android, BlackBerry 10, iOS, Windows Phone, Windows RT, PC, consoles, web	



Mobil Siteler ve Web Teknolojileri

Web teknolojisinin sürekli gelişmesi ile beraber internet destekli cihazların artışı, giderek artan mobil web kitlesi için mükemmel bir gelecek vadetmektedir. Küresel mobil internet trafiği hızlı bir şekilde artmaktadır ve halihazırda %20¹ seviyesini aşmıştır. İnterneti cihazlarla (mobil, tablet, masaüstü ve Smart TV) dolaşmakla geçirilen süre ülkeden ülkeye değişmektedir. ABD'deki kullanıcılar internette dolaştıkları sürenin %16'sını mobil cihazlarda kullanırken bu oran, Çin'de %34 ve Hindistan'da %79'dur. Çoğu bölgelerde mobil internet trafiği masaüstü internet trafiğini geçmiştir. İnternet trafiği sürekli olarak gelişmekte ve çeşitli bölgelerde popüler olmaktadır. Küresel anlamda akıllı telefon kullanımı artarken, gelişen piyasalar için temel özellikli telefon (feature phone) sistemlerine yatırım yapmak için hâlâ büyük bir fırsat mevcuttur. Örnek olarak Hindistan'ı ele alalım. Hindistanlıların yalnızca %15'inin akıllı telefonu bulunmaktadır. Yine de bu 180 milyon kullanıcı anlamına gelmektedir. Aynı zamanda nüfusun %68'i, %9,5'i internet özelliğini bulunduran temel özellikli telefonlara² sahiptir. Benzer şekilde, basit bir mobil site 116 milyon kullanıcıya ulaşmanızı sağlayabilir.

Web teknolojilerinin en bariz kullanım şekli mobil siteler oluşturmaktır ve bu bölümde odaklanacağımız ana konusu oluşturmaktadır. Yine de web teknolojilerinin yoğun bir şekilde web ve hibrit mobil uygulamalarda, çapraz platform çözümlerinde ve en yeni native uygulama geliştirmeleri (Firefox

1 gs.statcounter.com

2 discovermobilelife.com

OS) içerisinde kullanıldığını belirtmek gerekir. Çapraz platform geliřtirmesi ve yeni Firefox OS'i hakkında daha fazla bilgi için bu kılavuz içerisindeki ilgili bölümlere bakınız.

Web teknolojilerinin en büyük avantajlarından biri, mobil geliřtirme için en kolay yolu sunmalarıdır. HTML, CSS ve JavaScript gibi web teknolojileri uzun senelerden beri geliřtirilmektedir. Bu teknolojiler mobil site geliřtirmesinin temel aktörleri olmuřlardır ve olmaya devam edeceklerdir. Native uygulama geliřtirmesi için diğerk karmařık dillere kıyasla, web teknolojilerinin öğrenilmesi daha kolaydır. Mobil web siteleri ve web uygulamaları, çeřitli platformlar için native geliřtirmeye kıyasla herhangi bir platform üzerinde içeriğeri erişimi daha kolay hale getirmektedirler. Bunun anlamı, mobil web sitelerinin otomatik olarak daha geniş bir alana ulaşabilmeleridir. Dolayısıyla mobil web geliřtirmesi yalnızca geliřtirme süresi ve masraflarından tasarruf etmez, ayrıca bakım söz konusu olduğunda vakitten ve maliyetten de tasarruf eden bir alternatif sunar. Bunun yanı sıra uygulama mağazalarından bağımsız olmanız, uygulama mağazasının onay politikasına uymak zorunda kalmadan herhangi bir içeriğeri hızlı bir şekilde sunabilmenize fırsat vermektedir.

Bütün avantajlarına rağmen web teknolojilerinin çeřitli eksiklikleri bulunmaktadır. Web teknolojileri, native uygulama geliřtirmesinin sunabileceğeri derin platform entegrasyonu ve donanım özelliklerine doğrudan erişim becerilerine ulaşmakta zorlanmaktadır. Web teknolojilerinin performansı, yüksek ölçüde bağlantı durumuna bağılıdır; Facebook ve LinkedIn gibi büyük siteler hafıza sorunları yaşamaktadır ve mevcut web geliřtirme araçları, native uygulama geliřtirme için kullanılan araçlara göre geridedir.

Mobil sitelerin gelir yaratma süreçleri de zorlu olabilir; kullanıcılar, mobil sitelere ücretsiz giriş yapmak isterler. Mobil siteler için en yaygın gelir yaratma aracı reklam entegrasyonu-

dur. Mobil siteler için ödeme çözümleri hâlâ ilk aşamalarında ve bunların uygulanması bazen zor olabilir. Bunun tersine mevcut uygulama mağazalarının gelir yaratma araçlarının kurulumu kolaydır ve nihai kullanıcı için yüksek güvenlik seviyesi sunmaktadırlar.

Gelir yaratma hususu en önemli gereksinimlerden bir tanesiye, hibrit veya web uygulaması stratejisi iyi bir uzlaşma seçeneği ortaya koyabilir. Bu durumda asıl sorun, gerçekten kullanıcı dostu bir ürün yaratmak için native ve web teknolojilerinin benzersiz becerilerini birleştirmek olacaktır. Bu kitaptaki çapraz platform bölümünde, hibrit uygulamaları yaratmak için mevcut geliştirme çatılarını bulabilirsiniz.

Temel bir prensip olarak kullanıcılar yüklenmesi uzun süren, yüksek veri masrafı çıkartan veya hiç çalışmayan bir siteye yönlendirilerek sinirlendirilmemeli ve hayal kırıklığına uğratılmamalıdır. En kötü durumda kullanıcı, basit ancak gereken bütün içeriği bulunduran bir siteye yönlendirilmelidir. Herhangi bir geliştirmeden önce göz önünde bulundurulması gereken en önemli unsur, hedef kitlenizin cihaz özelliklerini, gezinme alışkanlıklarını ve bant genişliği/veri planlarını bilmektir.

UX bakış açısına göre, Google SMB'ler için aygıt dönüşümü için en iyi 10 uygulamayı³ sunmaktadır:

- **Parmak dostu olun** - sitelerinizi, büyük ellere sahip kullanıcıların bile kolaylıkla etkileşime geçebilmeleri için tasarlayın
- **Görünürlük için tasarlayın** - içeriğinizin, bir kol uzunluğundaki mesafeden bile okunabildiğinden emin olun
- **Gezinimi basitleştirin** - net gezinim, hiyerarşi ve dikey kaydırma, bilgiye ulaşmaya yardımcı olur

3 www.dudamobile.com/webinar/Google_DudaMobile_Webinar.pdf

- **Erişilebilir hale getirin** - ideal olarak, mobil siteniz tüm mobil cihazlarda ve tüm telefon yönlendirmelerinde çalışmalıdır
- **Dönüşümü kolaylaştırın** - dönüşüme yardımcı olacak bilgiye odaklanın
- **Yerel hale getirin** - insanların size ulaşmasına yardımcı olacak dil gibi işlevleri ekleyin
- **Mobil site yönlendirmelerini kullanın** - kullanıcılara, masaüstü siteye gitme seçeneği verin ancak mobil siteye dönüşü de mümkün kılın
- **Hızlı hale getirin** - mobil kullanıcılara yardımcı olun: sitenizin hızlı bir şekilde yüklenmesini ve taranmasını sağlayın
- **Kusursuz hale getirin** - masaüstü sitenizdeki işlevleri mümkün mertebe mobil siteye de uyarlayın
- **Öğrenin, dinleyin ve tekrarlayın** - iyi mobil siteler kullanıcı odaklıdır; kitleden alınan geri bildirim ile inşa edilirler.

Google halihazırda mobil arama sonuçları üzerinde gerekli değişiklikleri yapmıştır ve kendi tavsiyelerine uymayan siteleri cezalandıracağını duyurmuştur. Mobil sitenizi optimize etme hakkında daha güncel bilgilere ulaşmak için Google'ın geliştirici sitesine⁴ göz atın.

HTML5

HTML standardının beşinci sürümü, önceden yalnızca tescillenmiş teknolojilerle mümkün olan özellikleri yeniden sunmayı taahhüt ediyor. HTML5, geliştiricilerin ve karar verici kişilerin, native uygulama yerine mobil site ve web uygulaması

⁴ developers.google.com/webmasters/smartphone-sites

geliştirmeye karar vermelerini sağlayan temel unsurlardan bir tanesidir. Belli başlı popüler cihaz için tek kod zemininde bir araya getirilmiş, görünüm ve hissiyatı konusunda yakın hissettiren uygulamalar, kamera ve mikrofon gibi cihaz donanımına erişim imkanı, çevrimdışı kullanım için yerel veri depolama alanı ve ekran boyutuna göre iyileştirme sağlaması gibi özellikler, HTML5'i native uygulama geliştirmeye göre çekici kılmaktadır.

Tüm çekiciliğine rağmen HTML5, evrensel tarayıcı desteği konusunda halen gerilerdedir. Eski Facebook CTO'su Brent Taylor, durumu şu şekilde açıklamaktadır: 'Mobil taracılar üzerinde aşırı boyutlara ulaşmış bir teknoloji parçalanması var ve bu yüzden geliştiriciler, hangi HTML5 parçasını kullanabileceklerini bilmiyorlar. HTML5, tek bir standart olarak tanıtıldı ancak her mobil cihaz için farklı sürümlerle geliyor. Donanım hızlandırması ve dijital hakların yönetimi gibi sorunlar çok tutarsız bir şekilde uyarlanıyor. Bu da birçok farklı telefon platformu üzerinde çalışan bir yazılım yaratma ve geniş bir kitleye ulaşma konusunda geliştiriciler için çok zorlu bir ortam yaratıyor.'

Çok yakın zamanda LinkedIn, kendi web tabanlı mobil uygulamalarının yerine native uygulamaları kullanmaya başladı. LinkedIn'in mobil mühendislik bölümünün kıdemli müdürü Kiran Prasad, hem HTML5 sitesi hem de zengin native uygulamalar yaratmaya karar verdi. Bunun nedeni HTML5'in iş için hazır ve önemli olmasına karşın ekosistem tarafından yeterli kadar desteklenmemesiydi. 'Bunun için araçlar var ancak bunlar daha başlangıç seviyesinde. İnsanlar bunları daha yeni yeni öğreniyorlar.'⁵

Tarayıcı uyumluluğu hakkında daha fazla bilgi için HT-

5 venturebeat.com/2013/04/17/linkedin-mobile-web-breakup

ML5Test⁶ sitesine göz atın. Bu site hangi HTML5 özelliklerinin hangi tarayıcı tarafından desteklendiğini ilişkin hem özet hem de ayrıntı bilgi içeriyor. Facebook ayrıca, HTML5 özellikleri için web tarayıcılarını 3 seviyeye göre test eden ringmark⁷ programını sundu. Bu araç geliştiricilerin çeşitli mobil (ve masaüstü) web tarayıcılarının destek seviyesini hızlı bir şekilde kontrol edebilmelerini sağlıyor.

Özetlemek gerekirse, mobil sektör içerisinde neredeyse herkes HTML5'in uzun vadede başarılı olacağını kabul ediyor. ABI araştırması, HTML ile uyumlu tarayıcılara sahip mobil cihazların 2013 yılı sonunda dünya çapında toplamda 4 milyar adede⁸ ulaşacağını tahmin etmektedir. İşletim sistemleri HTML5 özellikleri için desteği yavaş bir şekilde arttıracaklar ve tarayıcılar da genel olarak uyumluluğu ve hızı arttıracaklardır. Firefox OS, Sailfish, Tizen ve Ubuntu gibi açık kaynaklı platformlar, bu uyumlulaşma sürecini hızlandırmaya yardımcı olacaklardır. Ayrıca Dünya Çapında Ağ Konsorsiyumu (W3C) nihayet HTML5 özelliğinin tamamlandığını duyurdu ve HTML5'in 2014 yılında resmi bir web standardı olacağını 2014⁹ öngörmektedir. The Developer Economics 2014 araştırması¹⁰ halihazırda HTML5'i, %52'lik tüketici bilinci ile en popüler üçüncü mobil uygulama platformu (tabii ki, Android ve iOS'tan sonra) olarak sıralamıştır.

⁶ html5test.com/results/mobile.html

⁷ rng.io

⁸ www.abiresearch.com/press/14-billion-html5-capable-mobile-devices-in-2013-bu

⁹ www.w3.org

¹⁰ DeveloperEconomics.com

Parçalanma Uyarlamayı Gerektirir

Mobil site geliřtirmedeki en büyük zorluk parçalanmadır. Teoride internet özelliđi olan bütün cihazlar web tarayıcılarıyla herhangi bir mobil siteye bağlanabilir. Gerçekteyse, geliřtiricilerin mobil site içeriđini, sayısı sürekli artan, farklı yazılım ve donanım becerilerine sahip cihaz ve tarayıcıya uyarlaması ve eniyileřtirmesi gerekiyor.

Mobil cihazlar için içeriđin eniyileřtirilmesiyle ilgili kabul gören iki yaklařım vardır: İstemci Tarafı ve Sunucu Tarafı Uyarlama.

- İstemci Tarafı Uyarlama, cihazda mobil dostu bir deneyim sağlamak için CSS ve JavaScript'in bir arada kullanımından yararlanır.
- Sunucu Tarafı Uyarlama, içerik istemciye aktarılmadan önce mantık yürütmek için sunucuyu kullanır.

Ařađıdaki bölüm, mevcut ve gelecekte internet özelliđine sahip olacak olan cihazların çođu için mobil sitelerin erişilebilir olmasını sađlayan istemci tarafı ve sunucu tarafı teknikleri içermektedir.

İstemci Tarafı Uyarlama

Duyarlı (Responsive) Web Tasarımı

Duyarlı Web Tasarımı, pazarlamacılar ve web geliřtiricileri arasında moda bir sözcük halini aldı. En basit řekliyle duyarlı tasarım, farklı ekran çözünürlüđu veya tipleri olan cihazlara hitap etmek için hareket kabiliyeti yüksek bölmelerden (grid)ve resimlerden ve CSS medya sorgulayıcılardan (CSS media queries) meydana gelir.

Bu yöntemin bařlı bařına ürettiđi sonuç olarak, dar kapsamlı bir cihaz aralıđı için cihaza duyarlı bir deneyim sunmasını ve

karmaşılaşmış içeriği uyarlamadaki noksanlığını söyleyebiliriz. Aynı içerik bütün cihazlara sunulur. Karmaşık masaüstü ve mobil web siteleri sunmak için teknik olarak tavsiye edilmemektedir.

Olumlu Yönler

- Saf istemci tarafı uyarlamanın mevcut altyapıya herhangi bir zararı olmaz.
- İçerik ve yerleşimin otomatik ayarlaması mümkündür

Olumsuz Yönler

- Normal web sitesinde mevcut olan içeriğin aynısı, mobil sürümde de (görünür olsun ya da olmasın) olacaktır.
- Sitedeki sayfaların yüklülüğü, mobil cihazlar üzerinde performans açısından önemli bir etkiye sahip olabilir.
- Mobil dostu cihaz eniyileştirmesi yerine esasen daha genel bir yaklaşımdır.

Kademeli Geliştirme

Kademeli Geliştirme, her çeşit mobil cihaza hizmet edebilme imkanı sağlar. Tek bir HTML sayfası bütün cihazlara gönderilir. Daha sonra JavaScript kodu, belli cihazlar için kademeli olarak işlevselliği en iyi seviyeye çıkartmak için kullanılır. Yalnızca mobil odağına aldığından tek dezavantajı performanstır. Kademeli inşa etme süreci zamana ihtiyaç duyar ve cihaza ve ağa göre farklılık gösterir. Masaüstü ve mobil çözüm olarak tek dezavantajı tek bir HTML'nin bütün cihazlara gönderilmesidir. Kademeli geliştirmeyi kullanan çok iyi bilinen bir geliştirme çatısı jQuery Mobile'dır¹¹.

¹¹ jquerymobile.com

Olumlu Yönler

- Saf istemci tarafı uyarlamanın mevcut altyapıya herhangi bir zararı olmaz.
- İçerik, işlevler ve yerleşimin kademeli bir şekilde ayarlanması mümkündür

Olumsuz Yönler

- Algılamayı tarayıcı gerçekleştirdiği için kontrol kaybı.
- Tarayıcı algılaması hâlâ istenilen seviyede değil
- İstemci tarafında gerçekleştirilen algılama, sitenin genel performans üzerinde bir etkiye sahip
- Benzer HTML sayfası bütün cihazlara sunulur

Sunucu Tarafı Uyarlama

Cihaz Veri Tabanları

Cihaz veritabanları web sitesine erişim sağlayan her bir cihazı algılar ve sunucuya cihazın kapasitesini içeren bir liste gönderir. Bu bilgi daha sonra cihaz becerilerinin ihtiyacını karşılayan mobil siteler sunmak için kullanılır. Sunucu tarafı uyarlama, en eski ve en güvenilir çözümlerden bir tanesidir. Popüler cihaz veri tabanları arasında WURFL¹² ve DeviceAtlas¹³ bulunur. Cihaz veri tabanlarının ana dezavantajı, çoğunluğunun bir ticari lisansın parçası olarak mevcut olmalarıdır.

Olumlu Yönler

¹² wurfl.sourceforge.net

¹³ deviceatlas.com

- En yaygın kullanılan çözüm (Google, Facebook, Amazon ve benzerleri)
- Maksimum kontrol
- Cihaz optimizasyonu mümkün (iPhone, Samsung Galaxy ve benzerleri gibi)

Olumsuz Yönler

- Cihaz Tanımlama Depoları donanıma odaklanmıştır
- Verinin yanı sıra bir algılama mekanizması da gerekmektedir (basit bir 'User-Agent' eşleşmesi düzgün çalışmaz)

Hibrit Uyarlama

Gerçekten de her iki yöntemi bir araya getiren, istemci ve sunucu tarafı uyarlamanın birleşimi hibrit uyarlamadır. Sunucu tarafı uyarlama sayesinde yüksek performans garanti eder ve cihaz yeteneği kaynakları (bir sonraki ziyaretlerde) mobil deneyimi arttırmak için kullanılabilir.

Hibrit uyarlama çözümleri ticari olarak Sevenval¹⁴ ve Netbiscuits¹⁵ gibi şirketlerde ve FITML¹⁶ gibi topluluk temelli bulut çözümlerinde mevcuttur.

¹⁴ sevenval.com

¹⁵ netbiscuits.com

¹⁶ fitml.com



Daha İyi Veri Girişİ

Küçük, genelde ekran üzerinde bulunan klavyeler ile metin girmek kulfetli ve zaman alıcı olabilir; özellikle kullanıcının sayılar, eposta adresleri veya benzeri metinleri girmesi gerekiyorsa. Neyse ki geliştiriciler, kolaylıkla beklenen girdi türünü belirleyebilirler; böylece akıllı telefonlar en uygun klavyeyi ekranda görüntüleyecektir. mobileinputtypes.com, çeşitli örnekleri sunmaktadır.

Daha İyi Performans

Mobili kullanıcılar sitelerin 2-5 saniye arasında yüklenmesini beklemektedir. Bu değere ulaşmak özellikle karmaşık mobil siteler için şu anda zorlu bir görevdir. Aşağıdaki bölümler yükleme süresini azaltmak ve performansı arttırmak amacıyla transfer boyutu, içerik ve HTTP taleplerini azaltmak için çeşitli ipuçları sunmaktadır.

Transfer Boyutunun Azaltılması

Görüntü boyutunu değiştirmeyi kullanın ve görüntü kalitesini ağ kalitesine göre ayarlayın.

İçeriğin Azaltılması

Hem site hem de sitede kullanılan varlıkların yüklemesi gün geçtikçe daha önemli bir hal almaktadır. JavaScript ve CSS dosyaları gibi varlıkların küçültülmesi, genel varlık yükleme sürelerini azaltmaya yardımcı olabilir. Tek tipteki birden fazla dosya bir dosyaya sıkıştırılır ve alfabe dışı karakterler ortadan kaldırılır. Bütün bunlar sonucunda daha az sayıda HTTP talebi ve tabii ki, daha hızlı yükleme süresi elde edilir.

Aynı zamanda kullanıcının neler olduğunu bilmesi de önemlidir. Benzer şekilde, eğer içerik yükleniyorsa kullanıcının bundan haberdar olması önemlidir ve kullanıcı, boş bir kutu veya sayfa ile karşılaşmamalıdır. Herhangi bir mobil deneyim

için pürüzsüz bir deneyim çok önemlidir ve bu siteden, site içerisindeki içerik yüklemesine ve bunu çevreleyen herhangi bir animasyona kadar her türlü unsuru içermektedir.

HTTP Taleplerinin Azaltılması

Görselleri, kodları ve stilleri satırıcı düzende ekleyin ve JavaScript veri yolunu ve Uygulama Önbelliğini ekleyin. Bunların temel faydası, kodların sayfa başına tek bir talep üzerine sunulması, HTTP gidiş ve gelişlerinin en düşük seviyeye inmesi ve çekirdek kodlarının uygulama önbelliğinde depolanmasıdır. Uygulama yeniden yüklemeyi etkilemeyecektir ve kodlar açık bir şekilde ön belleğe alınabilir özelliğe (CDN) sahip olacaktır.

Mobil web performansı hakkında daha fazla bilgi için slideshare üzerinde Roland Guelle'nin sunumuna¹⁷ göz atın.

Web Teknolojilerinin Test Edilmesi

Web teknolojilerinin çeşitli mobil telefonlarda nasıl çalıştığı farklı yöntemlerle test edilebilir. En basit yöntem, web sitesini veya web uygulamasını, mobil cihazlar üzerinde çalışan çeşitli web tarayıcıları içerisinde test etmektir. Bunlar içerisinde en popüler mobil web tarayıcıları bulunur. Örnek olarak çevrimiçi kullanılabilir kamu verileri¹⁸ gösterilebilir. Test edilecek cihaz kümesi, mevcut web logları ve buna benzer kaynaklardan elde edilen verilerin analiz edilmesi için küçültülebilir. Ayrıca çeşitli form faktörleri üzerinde test yapılması, yerleşim ve biçim sorunlarını ortaya çıkarmamıza yardımcı olabilir.

Otomatik test açısından, WebDriver¹⁹ en etkili geliştirme çatısıdır. İki adet tamamlayıcı yaklaşım bulunmaktadır:

¹⁷ www.slideshare.net/sevenval/mobile-web-performance-dwx13

¹⁸ gs.statcounter.com/#mobile_browser-ww-monthly-201207-201306

¹⁹ seleniumhq.org/projects/webdriver

1. Android ve iOS içerisinde gömülü olan WebView kontrollerini kullanarak otomatik test
2. Çeşitli mobil web tarayıcılarını taklit etmek için yapılandırılmış, Google Chrome veya Mozilla Firefox kullanılarak yapılan User-Agent yanıltması

Her iki yaklaşımın da olumlu ve olumsuz yönleri bulunmaktadır:

- Gömülü WebView'lar, hedef platform işletim sistemi üzerinde çalışmaktadır. Bunların birçok davranışsal hatayı bulması muhtemeldir. Ancak yapılandırma kısmı sürece daha fazla etki eder ve bazı platform işletim sistemlerince desteklenmez.
- Yanıltma, web sunucuları karşısında tarayıcının, Nokia Asha 201 telefonu gibi gömülü WebView'a benzer mevcut olmayan mobil tarayıcılar dahil olmak üzere sanki birçok platformdan gelmiş gibi davranmasını sağlayabilir. Ancak tarayıcı davranışı ve oluşturma (render) gerçekçi değildir; bu yüzden cihazlar üzerinde gerçekleşmeyen diğer 'yanlış pozitif' hatalar bulunurken, birçok hata algılanmayacaktır.

Daha Fazla Bilgi

Çevrimiçi Kaynaklar

- W3Schools ve CSS Tricks (temel düzeyde HTML, CSS ve JavaScript öğrenmek için iyi kaynaklar): w3schools.com, css-tricks.com
- HTML5 Rocks (eğitimler, slayt gösterileri, makaleler ve

- daha fazlası dahil olmak üzere HTML5 hakkında mükemmel bir kaynak): html5rocks.com/en
- Breaking the Mobile Web (mobil web programlaması hakkında birçok kitabın yazarı olan Max Firtman kendi mobil blogunda güncel haberleri sunuyor: mobilexweb.com
 - Mobi Thinking (Mobil pazarlama uzmanlarından yorumlar, analizler ve fikirler ile DotMobi'nin pazarlamacılar için kaynağı): mobithinking.com
 - (Mobil) Web Uygulamalarının Testi: docs.webplatform.org/wiki/tutorials/Testing_web_apps
 - Hangi özelliklerin web üzerinde çalıştığını öğrenin : caniuse.com ve beta.theexpressiveweb.com
 - WHATWG (HTML topluluğunun web sayfası): whatwg.org
 - Word Wide Web Consortium (Web standartlarını belirleyen organizasyon): w3.org

Kitaplar

- Mobile First - Luke Wroblewski
- Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement by Aaron Gustafson ve Jeffrey Zeldman
- Responsive Web Design - Ethan Marcotte
- Programming the Mobile Web - Max Firtman
- jQuery Mobile: Up and Running - Max Firtman



Erişilebilirlik

Dünya nüfusunun yaklaşık %20'si bir nedenle fiziksel engellidir. Erişilebilirlik, engelli olsun ya da olmasın bütün kullanıcılarınız için mükemmel bir kullanıcı deneyimi yaratmak ve uygulamanızın olabildiğince çok kişi tarafından kullanılmasını sağlamayı hedefler.

Uygulamalarınızı herkes tarafından erişilebilir kılmak için çeşitli nedenler şu şekilde sıralanabilir:

- Erişilebilirliğin genel kurallarına uymak, uygulamanızın genel kullanımını artırır. Örnek olarak, uygulamanızda konuşma özelliğini bulundurmak görme engelli kişilere yardımcı olabilir ve sürücüler için araba içi kullanıma olanak sağlayabilir.
- Uygulamanız, ABD'de Rehabilitasyon Kanunu'nun 508. fıkrası uyarınca mevzuatın erişilebilir bir çözümü zorunlu koştığı durumlar gibi, hükümet destekli piyasa sektörlerine girmenize yardımcı olabilir. Benzer çalışmalar ülkemizde de TÜBİTAK, KOSGEB, AB fonları ve ilgili kamu ve özel kuruluşların açtığı çeşitli fon ve olanaklardan faydalanmanızı sağlayabilir.
- Apple, Google ve Microsoft'un mobil platformları, kullanıcı arayüzü otomasyonu testi için erişilebilirlik API'lerini kullanabilirler; böylece uygulamanızı erişilebilir özelliklere sahip bir şekilde programladığınızda otomatik test daha kolay bir hale gelir.

Birçok potansiyel kullanıcıınızın engel sorunu olabilir; bu da mobil teknolojileri kullanmalarını zorlaştırabilir. Bu engeller arasında, çeşitli seviyede görme veya duyma bozuklukları, kavramsal engeller, el becerisi sorunları, teknofobi ve ben-

zerleri bulunabilir. Bu kullanıcıların çoğu, kendi cihazlarını kullanırken onlara yardımcı olmak için üçüncü parti araçlara bağlı kalmaktadır. Bu yazılım kimi zaman 'Yardımcı Donanım' olarak adlandırılır ve ekran okuma ve büyütme uygulamaları gibi yardımcı programları içerir. iOS içerisinde, mobil telefonlar üzerinde erişilebilir bir arayüz sağlama adına önemli bir program olan VoiceOver¹ bulunmaktadır. Android'in 'Yardımcı Donanım' için bir eklenti yaklaşımı bulunur; en yaygın eklenti TalkBack'tir. Çeşitli mobil platformlar, kullanıcı arayüzünü daha erişilebilir hale getirmek için ekran büyütme ve diğer ayarları içermektedir.

Bu kullanıcıların deneyimleri, bir uygulamanın, yardımcı teknoloji ile ne kadar iyi çalıştığından etkilenir.

Erişilebilirlik API'leri için standart UI öğelerinin özel davranışları ile ilgili metne bakılır. VoiceOver ve TalkBack gibi ekran okuyucular, metni kullanıcının dinleyebileceği sesli konuşmaya dönüştürür. Ekran okuyucu yazılımı, özellikle metin yokken, kullanıcıya daha kavramsal bilgi sunmaya yardımcı olmak amacıyla kontrol türünü ve ilgili davranışları da belirleyebilir.

İşaret dilleri, ekran okuyucuları devreye sokulduğunda etkilenebilir. VoiceOver ve Android'in Explore-By-Touch programı dahil olmak üzere çeşitli ekran okuyucuları, kullanıcıya bir unsura dokunmadan önce ekranı keşfederek hangi unsurun ne işe yaradığını anlama fırsatı vermektedir. Değişiklikler, kullanıcıların uygulamanızla etkileşimini olumsuz bir şekilde etkileyebilir. Uygulamanızı bu ekran okuyucuları ile test ederek bu sorunları erkenden, nihai kullanıcıların etkilenmesine izin vermeden yakalayabilirsiniz. Ayrıca, uygulamayı yeniden tasarlayabilir ve ekran okuyucuları aktif ve devre dışı iken iyi bir şekilde çalışan uygun işaret dillerini bulabilirsiniz.

1 apple.com/accessibility/iphone/vision.html

Yazılımınızı engelli kullanıcılar için erişilebilir hale getirmek için bazı genel anahatları takip etmeniz gerekir. Eğer bunlara uyarırsanız, kullanıcının sizin yazılımınız ile beraber kullandığı yardımcı teknolojiyi dahil edebilmek için en iyi fırsatı yakalamış olursunuz:

- Platformunuzun hangi erişilebilirlik özelliklerine ve API'lerine sahip olduğunu bulun ve bu API'ler mevcutsa, bunların en iyi şekilde kullanıldıklarından emin olun.
- Mümkün olan durumlarda, özel kullanıcı arayüzü unsurları yerine standart unsurları kullanın. Bu, platformunuzun bir erişilebilirlik altyapısı bulunuyorsa veya gelecekte buna ihtiyaç duyacaksa, uygulamanızın kullanıcılarınıza erişilebilirlik hizmeti sunmasının mümkün olmasını sağlayacaktır.
- Platformunuzdaki standart tasarım ilkelerini izleyin. Bu durum, uygunluğu artırır ve varsayılandan daha erişilebilir bir tasarım anlamına gelebilir.
- Bütün resimleri, resmin ne olduğunu anlatan kısa bir tanım ile etiketleyin (Çalma düğmesi için “Çalma”)
- Bir eylemi farklılaştırmak için renk unsurunu tek yöntem olarak kullanmaktan kaçının. Örnek olarak, renk körü bir kullanıcıdan kırmızı ile işaretlenmiş olan alanları düzeltmesi istenirse, kullanıcı bu hatayı algılayamayacaktır.
- Uygulamanızın bütününde iyi bir renk kontrastına sahip olduğunuzdan emin olun.
- Eğer mevcutsa, platformunuz için Erişilebilirlik API'nı kullanın. Bu, özel kullanıcı arayüzü unsurlarını daha erişilebilir hale getirmenize fırsat verecektir ve uygulamanızın bütününde daha az yorulmanızı sağlayacaktır.
- Uygulama arayüzünüzün düğmelere ya da bir klavyeye programlı gezinimini destekleyin. Bu yalnızca, uygulamanızın harici klavyeler ile kullanılmasına izin

- vermekle kalmayacak, ayrıca gezinimin bir trackball veya sanal d-pad ile yapılabildiği Android gibi platformlar üzerinde uygulamanızın erişilebilirliğini artıracaktır.
- iPhone'daki VoiceOver gibi yardımcı teknolojiler ile hedef cihaz üzerinde uygulamanızı test edin.

Online olarak çok daha ayrıntılı kılavuzlar² bulabilirsiniz. BBC de, erişilebilir mobil uygulamaları hakkında çeşitli kılavuzlar³ yayınlamıştır.

Apple ve yakın zamanda Google ve Microsoft, kendi kullanıcı arayüz test otomasyon çerçevelerini desteklemek adına erişilebilirlik arayüzünü kullanarak ilgili erişilebilirlik desteklerine daha fazla önem vermeye başlamışlardır. Bu, kendi uygulamalarının daha erişilebilir olması için geliştiricilerin göz önünde bulundurabilecekleri başka bir teşviği de beraberinde getirmektedir.

Farklı mobil platformlarına yakından bakıldığında, erişilebilirlik özellikleri ve API'ları açısından büyük farklara sahip oldukları görülmektedir.

Özel Kontroller ve Unsurlar

Eğer uygulamanız içerisinde özel kullanıcı arayüzü unsurlarını kullanıyorsanız, Erişilebilirlik API'na olan bu platformlar özel kontrollerinizi erişilebilir hale getirmenize yardımcı olabilir. Bunu, cihaz üzerinde çalışan yardımcı teknolojinin kontrol özelliklerini sorgulayabilmesi ve ona erişilebilirlik sağlayabilmesi için, kontrol ile karşı karşıya getirerek sağlayabilirsiniz.

² slideshare.net/berryaccess/designing-accessible-usable-application-user-interfaces-for-mobile-phones

³ bbc.co.uk/guidelines/futuremedia/accessibility/mobile_access.shtml

Android üzerindeki bu süreç hakkında daha fazla bilgiyi Google IO 2012 sunumlarından⁴ alabilirsiniz.

Eğer Apple geliştirici programının üyesiyseniz, iOS Geliştirici Merkezi⁵ üzerinde mevcut olan, WWDC 2012 ve 2013 sunumlarındaki erişilebilirlik videosuna bir göz atın.

Erişilebilir Android Uygulamaları

Android'in en yeni ana sürümü olan sürüm 4, erişilebilirlik tarafında birçok yeniliği getiriyor. Bunlar arasında erişilebilirlik odağı, Braille alfabesi desteği ve daha fazlası bulunmaktadır. Geliştirici belgeleri de iyileştirilmiştir. 4.4 dahil olmak üzere bir sonraki küçük sürümler de erişilebilirlik desteğini artırmıştır.

Bunlar arasında, erişilebilir canlı bölgeler ve altyazı yerleştirme desteği bulunmaktadır Android Services⁶

Önceki Android sürümlerini kullanan kişiler de dahil olmak üzere, uygulamanızın hedef kitlesini arttırmak amacıyla:

- Uygun olan yerlerde standart UI kontrollerini kullanın
- Ayrıca özel görünümünün de erişilebilirliğini artıran Destek Kütüphanesi'ni⁷ kullanmayı düşünün.
- Kullanıcılarınızın, uygulamanız içerisinde bir trackball veya D-Pad ile gezinebileceklerinden emin olun; bu onlara,

⁴ youtube.com/watch?v=q3HliaMjL38 ve youtube.com/watch?v=ld7kZRpMGb8

⁵ developer.apple.com/wwdc/videos

⁶ developer.android.com/about/versions/android-4.3.html#A11y

⁷ developer.android.com/tools/support-library/index.html

Talkback ve diğer yardımcı teknoloji uygulamalarını kullanabilmek için en iyi fırsatı sunacaktır.

Android erişilebilirliğindeki en başarılı örnekleri ayrıntıları ile birlikte Android erişilebilirlik API'sinin nasıl kullanılacağı ile ilgili bilgi için, lütfen Google'ın Uygulamaları Erişilebilir Hale Getirmek⁸ başlıklı belgesine bakın.

Ayrıca, Erişilebilirliği Uyarlamak⁹ ismindeki bölüm içerisindeki geliştirici eğitim alanında daha fazla örnek bulabilirsiniz. Erişilebilirliğin test edilmesine de online¹⁰ olarak değinilmiştir.

Metinden konuşmaya çevirme API'nı kullanmak da dahil olmak üzere Android erişilebilirliği hakkında daha fazla bilgi için Eyes-Free projesine¹¹ bakın.

Erişilebilir BlackBerry Uygulamaları

Eğer BlackBerry OS 7.1'i hedefliyorsanız, erişilebilirlik API'nın kullanımı hakkında çeşitli bilgileri ve erişilebilir UI tasarımına ilişkin birçok ipucunu geliştiriciler için hazırlanmış olan web sayfasında¹² bulabilirsiniz.

Mayıs 2012 tarihinde BlackBerry, çeşitli BlackBerry Curve telefonları için BlackBerry Ekran Okuyucusu'nu¹³ piyasa sürdü. Bu program, uygulamalarınızın erişilebilirliğini test etmek için ücretsiz bir indirme olarak mevcuttur.

⁸ developer.android.com/guide/topics/ui/accessibility/apps.html

⁹ developer.android.com/guide/topics/ui/accessibility/apps.html

¹⁰ developer.android.com/tools/testing/testing_accessibility.html

¹¹ code.google.com/p/eyes-free

¹² developer.blackberry.com/java/documentation/intro_accessibility_1984611_11.html

¹³ blackberry.com/screenreader



Blackberry 10, kullanıcıların kendi cihazlarını özelleştirebilmeleri için çeşitli erişilebilirlik ayarı sunmaktadır. Bunlar arasında işaret dili ile yakınlaştırmayı kontrol eden, ekran çevresinde dolaşan ve yazıtipi boyutlarını değiştirebilen Yakınlaştırma Modu bulunur. Uygulamanızın nasıl görüldüğünü test etmek ve bunların nasıl kullanılabileceğini görmek için uygulamanızı yakınlaştırma ile kullanın ve yazıtipi boyutlarını değiştirin. Blackberry 10.2 ve sonraki sürümler, bir ekran okuyucusu ile piyasaya sürülmektedir. Erişilebilir Blackberry 10 uygulamaları hakkında belge Erişilebilirlik özellikleri ve en iyi uygulamalar - BlackBerry Native¹⁴ adlı kaynakta bulunabilir.

Erişilebilir iOS Uygulamaları

iOS, erişilebilirlik için iyi bir desteğe sahiptir. Örnek olarak, iOS cihazları şu unsurları içermektedir:

- **VoiceOver**, bir ekran okuyucudur. Ekrandaki nesneleri ve metni okuyarak, ekranı iyi bir şekilde göremeyen kişilerin uygulamanızı kullanabilmesini sağlar.
- **Zoom**: Ekrandaki bütün içeriği yakınlaştırır.
- **White on Black**: Bu, ekrandaki renkleri tersine çevirerek siyah ve beyaz kontrasta ihtiyaç duyan ancak beyaz arkaplan ışığından rahatsız olan kişilere yardımcı olur.
- **Altyazı** İşitme kaybına sahip kişiler için altyazı.
- **Sesli, görsel ve titreşimli uyarılar** kendileri için en iyisini seçmek isteyen kişiler için.
- **Sesli Kontrol ve Siri**: Bu, kullanıcıların sesli kontroller ile telefon konuşmaları yapabilmelerine ve telefonun diğer özelliklerini kullanabilmelerine fırsat verir.

¹⁴ developer.blackberry.com/native/documentation/cascades/best_practices/accessibility/accessibility_features_best_practices.html

iOS 7, kullanıcı tarafından yapılandırılabilen yeni erişilebilirlik ayarlarının yanı sıra, Dynamic Type ve Guided Access API'larını da sunmaktadır.

Eğer iOS üzerinde çalışıyorsanız, Apple'ın erişilebilirlik prensiplerini¹⁵ kullandığınızdan emin olun. Bu prensipler API'ı ayrıntılı bir şekilde açıklar ve uygulamalarınızın kullanıcı deneyimini en yüksek seviyeye çıkarmak için mükemmel ipucu kaynağını oluşturur.

Erişilebilir Windows Phone ve Windows 8 Uygulamaları

Windows Phone/Windows 8 platformu üzerinde erişilebilir uygulamalar yazmak için iki geliştirme yöntemi bulunmaktadır. Bunlar XAML ve HTML 5/JavaScript'tir.

Eğer uygulamanız C#, C++ veya Visual Basic içerisinde yazılmışsa, uygulamanızı erişilebilir kılmak için Accessibility in Metro style apps using C++, C#, or Visual Basic¹⁶ belgesinde ayrıntılı bilgileri bulabilirsiniz.

Eğer HTML5 ve JavaScript'i kullanmayı seçtiyseniz, Accessibility in Metro style apps using JavaScript'e¹⁷ ihtiyaç duyacaksınız.

Uygulamanızın erişilebilirliğini¹⁸, test ettikten sonra, Microsoft benzersiz bir yöntemle uygulamanızı Windows Store içerisinde erişilebilir¹⁹ olarak tanıtmaya fırsatı sunmaktadır; böy-

¹⁵ developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility

¹⁶ msdn.microsoft.com/en-us/library/windows/apps/xaml/hh452680.aspx

¹⁷ msdn.microsoft.com/en-us/library/windows/apps/hh452702.aspx

¹⁸ msdn.microsoft.com/en-us/library/windows/apps/xaml/hh994937.aspx

¹⁹ msdn.microsoft.com/en-us/library/windows/apps/xaml/jj161016.aspx

lece, aramalarında erişilebilirlik filtresi kullanan kullanıcıların uygulamaları bulabilmeleri daha kolay olacaktır.

Windows Phone 8, yüksek kontrast, ekran büyütme ve metin boyutu gibi erişim kolaylığı ayarlarını sunmaktadır. Bunlar geliştirdiğimiz çeşitli uygulamalar için geçerli olabilmektedir, bu yüzden bu farklı ayarlarla uyumlu olduğundan emin olmak için uygulamamızı test etmemiz yararlı olacaktır. Windows Phone 8.1 ayrıca, ayrı bir ekran okuyucusu sunmaktadır. Sınırlı sayıda dahili özellikler içerir ve diğer uygulamaları desteklemez.

Erişilebilir Mobil Web Uygulamaları

Web erişilebilirliği hakkında birçok yazı yazılmıştır ancak bu yazının yazıldığı tarihte, erişilebilir mobil cep geliştirmesi için en iyi uygulamayı sunan bir standart bulunmamaktadır.

Eğer uygulamanız, native bir uygulama görünümü ve tavrını taklit edecekse, bu bölümde sunulan ve yukarıda bulunan prensipleri takip etmeniz gereklidir.

Eğer bir web içeriği geliştiriciyseniz, Web İçeriği Erişilebilirlik Prensipleri (WCAG) Özeti'ne²⁰ bakmanız gerekir.

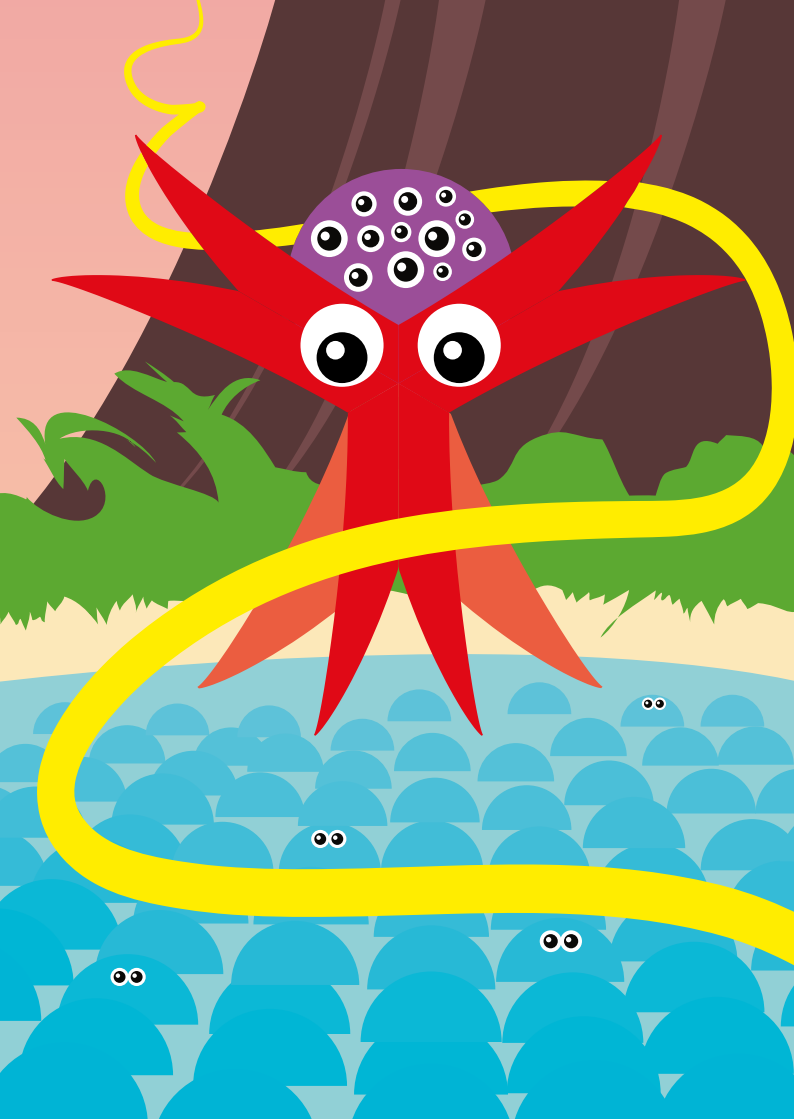
HTML5 desteğinin, çeşitli mobil platformlar üzerinde gittikçe benimsenmesi sebebiyle, gelecekte ortaya çıkabilecek herhangi bir mobil web uygulaması erişilebilirlik standardının temelini oluşturması muhtemel olduğundan Mobile Web Application Best Practices'ı²¹ okumanızı öneriyoruz.

Ayrıca, Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines'ı (WCAG)²² yararlı bir kaynak olarak kullanabilirsiniz.

²⁰ w3.org/WAI/intro/wcag

²¹ w3.org/TR/mwabp

²² w3.org/TR/mwbp-wcag/



Kurumsal Uygulamalar: Strateji ve Geliştirme

Kurumsal karar vericiler artık mobil uygulamaları, bir muhasebecinin hesap çizelgesindeki bir öğeden çok bir gereklilik, mobil bir stratejik faktör olarak görmektedir. Kurum içi uygulamalar, bir şirket içindeki bilgi birikiminin aktarımını hızlandırır. Herhangi bir zamanda ve yerde mevcut olan rekabetçi veriyi hazır tutarak çalışanın becerisini artırır. Uygulamalar ayrıca şirketlerin tedarikçilerine, müşterilerine ve nihai tüketicilerine vb. ulaşabilmelerini sağlar. Kurumsal uygulamalara örnek olarak; saha ve satış ekiplerinin işlerini kolaylaştıran yazılımlar, envanter yönetimi, tedarik zinciri yönetiminin yanı sıra B2C pazarlaması araçlarını gösterebiliriz.

Bunu söylemek gereksiz olabilir ancak günümüzde en önemli risk, bir kurumun mobil tarafta bir stratejiye **sahip olmamasıdır**. İşletmeler şu anda, önceden olduğu gibi BT yatırımlarını üst düzey yönetimle sınırlamak yerine **Herkes için Mobil** seçeneğini göz önünde bulundurmaktadır. Cihaz satın almaya ilişkin geleneksel BT yaklaşımını devreye sokmak ve bunları sadece yöneticilere kullanırmak artık tek seçenek değil.

Çalışanların normal hayatta kullandıkları telefonları işyerinde de kullanabilmelerini sağlayan ve onları tek tip bir cihaz kullanmaktan kurtaran **Kendi Cihazını Kendin Getir** (BYOD) yöntemi giderek popülerleşmekte ve BT altyapısına bağlanmak, emniyetli içerikleri indirmek ve ticari kuruluş uygulamalarını kullanabilmek için personelin alıştığı arayüzü kullanabilmesine fırsat verme-



ktedir. BYOD'un ortaya ıkması ile řirket, geleneksel olarak kurumsal BT stratejisinin hiçbir zaman parası olmayan risklere maruz kalabilir. İyi dřünlmř ve uyarlanmıř olan bir kurumsal mobil stratejisinin erkenden benimsenmesi, verilerin srekli olarak gvenli kalmasında ok nemli bir unsurdur.

Ticari Kuruluřları řekillendiren Mobil Uygulamalar iin nemli Noktalar

- Mevcut sistemlere kıyasla maliyetin azaltılması
- İř srelerinin modernleřtirilmesi
- Derhal hazır olabilen gncel verilerle beraber gelen rekabeti avantaj
- alıřan memnuniyetinin ve verimliliğinin artması
- Mevcut srelere kıyasla ok daha hızlı cevap

Kurumsal Strateji

Gnmzde pek ok řirket, bir Uzman Mobil Yetkilisi'ne (CMoO) sahiptir ya da CIO pozisyonlarını geniřletmiřlerdir. řirketin mobil taraftaki mobil teknoloji ve yatırımlarına yn vermenin yanı sıra, srdrlen iř ve yeni teknolojiler arasındaki mesafeyi kısaltacak kprleri kurmak da onun grevidir. řirketin byklğ ve faaliyet alanına gre değışmekle birlikte, kurum iinde bir mobil yazılım geliřtirme ekibi oluřturmak veya dıřardaki bir yazılım geliřtirme ofisiyle olan sreleri ynetmek de onun iři olabilmektedir.

Mobil yazılımın, alıřanların/kullanıcıların istediklerini sunduğundan, bunun teknik olarak mmkn olduğundan ve her řeyin genel řirket stratejisine uyduğundan emin olmak iin lider, bir Mobil Yenilik Konseyi (MIC) veya Mkemmelik Merkezi (CEO) dzenlemeyi dřnebilir. Bu grup, mobil geliřtirme

ekibinden gelen yetenekli temsilciler, şirket içerisinde mobil için mevcut olan paydaşlar ve en önemlisi, ilgili iş süreçlerinde tecrübesi olan ve çeşitli departmanlardan gelen nihai kullanıcılar gibi önemli üyeleri içermelidir.

CMoO/CIO'nun, MIC/COE ile beraber odaklanması gereken konular arasında şunlar bulunur:

- Strateji – genel mobil stratejisi ve uygulamalar için vizyon ve yön.
- Denetim politikaları - Kendi Cihazını Kendin Getir (BYOD) veya Kendi Cihazını Kendin Seç): bunlar temel olarak, bir Mobil Uyarılma Yönetimi (MAM) Politikası (BYOD) ve bir Mobil Cihaz Yönetimi & Emniyeti (MDM) politikası (CYOD) arasındaki farktır.
- Uygulama özellikleri
- Uygulama yol haritası
- Bütçe planlaması
- Kabul – uygulamanın üretim için onaylanması
- Uygulamanın hayata geçirilmesi – demolar ve prototipler hakkında ilk geri bildirimler, test etme, toplu bir şekilde uyarılma
- Teşvikler – mobil teknolojilerin benimsenmesinin teşvik edilmesi

Kurumsal mobil uygulama, dünyanın pek çok bölgesinde hâlâ yeni bir kavram. Üçüncü taraf kurumsal uygulamalar yazan bir şirketin veya dâhili bir ticari kuruluş stratejisini benimsemek isteyen bir geliştirme müdürünün karşılaştacağı temel engellerden bir tanesi, işletme ihtiyacı gereksinimi olacaktır. En yaygın soru büyük olasılıkla “Bütün bunlar çok güzel, ancak bunlara neden ihtiyacımız var?” olacaktır; bu yüzden bir mobil stratejisinin benimsemesi için şirkete ikna edici sebepler sunmaya hazır olmalısınız.

Mobil Kurumsal Uygulamalar için İş Planı Oluşturma

- İşinizi mobile taşıma süreci için bir **Vizyon Planı** oluşturun ve bu değişimin kurumunuza nasıl yardımcı olacağını ve şekillendireceğini önceden planlayın.
- Amaç ve hedef kitle belirtilerek, her bir uygulama için bir **Uygulama Tanımı** yazın.
- Satın alacağınız cihazlar için bir **Bütçe** oluşturun.
- Uygulama & Cihaz Yönetim Stratejisi & Emniyeti Altyapısı için bir plan oluşturun.
- Gelecekte de varlığını sürdürebilecek olan bir Geliştirme Platformu (MEAP/MADP) kullanarak, uygulama geliştirme ekibi için bir plan oluşturun.

Mevcut Sistemleri Mobile Uyarlamak

Eğer halihazırda müşterilerinize henüz mobile taşınmamış bir sistem sunuyorsanız, vermeniz gereken çeşitli kararlar olacak. Çözümün uygulamasına başlamadan önce sisteminize bir mobil ürün sunmanın yaratacağı etkiyi tamamen anlayabilmek çok önemlidir. Ürününüzü mobil dünyaya taşımak için mevcut olan yaygın sebepler arasında kamera ve GPS veya İnternet'e bağlanmadan yalnızca hareket halinde bilgiye ulaşabilme becerisi gibi telefon özellikleri bulunmaktadır. Mobil bir ürünün sürekli desteği, bakımı ve geliştirmesi, orijinal sisteminiz için ayrı bir yol haritası haline geleceğinden ve ilave bir maliyete sahip olacağından, mobil duruma geçmeniz için doğru sebeplere sahip olduğunuzdan emin olmanız gerekir.

Süreçleri Mobilleştirmeye Dair Önemli Noktalar

- Mobile geçmenin sebeplerini açık bir şekilde belirtin ve bu sebeplerin, mobile geçmek için yeterli olduğundan emin olun.
- Mobil ve masaüstü arasında farkı anlayın. Mevcut sisteminizi kopyalamayın; örnek vermek gerekirse, bilgiyi yakalamak için biçimi kullanmak yerine, sesi yakalayıp bunu sisteminize yükleyebilirsiniz. Böylece bir kullanıcı küçük bir cihaza yazı yazmasına gerek kalmadan hızlı bir şekilde not alabilir.
- Mevcut sisteminizin bütün özelliklerini uyarlamaya çalışmayın; mobile uygun olan, önemli noktaları dâhil edin.
- Müşterilerinizin hangi cihazları kullandıklarını ve hareketlenmek için sisteminizdeki hangi özelliklerin gerekli olduğunu anladığınızdan emin olun.
- Çapraz platform testi, birden fazla cihaz tipi ve işletim sistemini kapsayan, açıkça tanımlanmış bir mobil test stratejisine sahip olun

Kurum içerisinde Cihaz ve Uygulama Yönetimi

Bir kurumsal uygulama geliştirirken, hassas şirket bilgilerini içeren donanımın kaybolabileceğini veya çalınabileceğini daima göz önünde bulundurmalısınız. Cihazları, içeriği ve uygulamaları emniyet altına almak için artık iki yöntem bulunmaktadır. Mobil Cihaz Yönetimi (MDM) ve Mobil Uygulama Yönetimi (MAM).

MDM, ticari kuruluş, bir cihaz üzerinde temel bir kontrol sunar; böylece bir cihaz kaybolduğunda, çalındığında veya

alıřan iřten ayrıldıđında, ticari kuruluř cihazı silebilir ve alıřmasını engelleyebilir. Bu yntem genellikle, bir ticari kuruluřun cihazın mlkiyetini elinde bulundurduđu, bylece cihaz zerindeki verilerin ve uygulamaların řirkete ait olduđu ve cihaz zerinde saklanan herhangi bir kiřisel verinin, riski alıřana ait olacak řekilde depolanmasının da dhil olduđu durumlarda kullanılmaktadır.

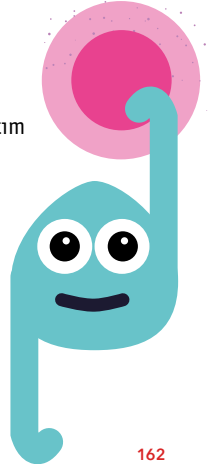
MAM, bir ticari kuruluřun, kontrol cihazın sahibinden tamamen almadan uygulamaları ve indirilen ieriđi emniyetli hale getirmesine olanak sađlar. Bir alıřan, iřten ayrıldıđında cihazı beraberinde gtrdđnde, iřletme kurumsal uygulamalarını devreden ıkartabilir ve fotođraflar ve tketiciler tarafından satın alınan uygulamalar gibi herhangi bir kiřisel veriyi etkilemeden indirilmiř olan herhangi bir veriyi silebilir. ođu MDM ve MAM czmleri apraz platform zelliđine sahiptir ve iOS, Android, Windows ve BlackBerry cihazlarını desteklemektedir. Bu husus, bir MDM veya MAM sunucusunu belirlerken daima gz nnde bulundurulmalıdır.

Bu ynetim czmleri vasıtasıyla eřitli gvenlik zellikleri mevcuttur. Bunlar arasında:

- Cihaz izleme
- Lisans kontrol
- Dhili bir Over the Air (OTA) czm ile dađıtım
- Yazılım envanteri
- Varlık kontrol
- Uzaktan kontrol
- Bađlantı ynetimi
- Uygulama desteđi & dađıtım

bulunur.

Gvenlik nlemleri arasında:



- Parola koruması
- Cihaz üzeri veri şifrelemesi
- OTA veri şifrelemesi
- Uzaktan kilitlemeli cihazlar
- Uzaktan silinebilir cihazlar
- Cihazların yeniden tedarik edilmesi

bulunmaktadır.

MDM ve MAM sunucularının örnekleri arasında şunlar bulunur:

- Airwatch¹
- App47²
- Apperian³
- BlackBerry Enterprise Server⁴
- Good⁵
- Microsoft⁶
- MobileIron⁷
- Mocana⁸

1 air-watch.com

2 app47.com

3 apperian.com

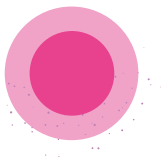
4 <http://us.blackberry.com/business/products-services/bes/overview.html>

5 good.com

6 microsoft.com/en-us/windows/windowsintune/explore.aspx

7 mobileiron.com

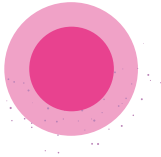
8 mocana.com



Kurumsal Uygulama Platformları (MEAP/MADP)

Kurumsal uygulamalarının önemli unsurlarından bir tanesi, veri senkronizasyonudur. Mobil cihazların, şirketin sunucularından alınan güncel veya ilgili veriler ile yenilenmesi, güncellenmiş veya toplanmış verilerin geri gönderilmesi gerekmektedir. Veri erişim kapsamı, kullanıcının iş sorumluluklarının yanı sıra gizlilik politikası tarafından belirlenmektedir. Herhangi bir durumda senkronizasyonun güvenli olması gereklidir. Kurumsal veri, en önemli varlıklarınızdan biridir. Bunun yanı sıra, şirketin bütünü tarafından kabul gören bir uygulama, birden fazla platforma hizmet etmek durumundadır. Bu anlamda, native SDK'ların yanı sıra, yaygın çoklu platform çözümlerinin eksikliklerini giderme anlamında, Mobil Kurumsal Uyarlama Platformu veya Mobil Uyarlama Geliştirme Platformu (MEAP/MADP) çözümlerini değerlendirmek isteyebilirsiniz.

MEAP/MADP'ler, mevcut arka plan programların veri kaynaklarına sahip çoklu mobil platformlar üzerinde çalışan kurumsal uygulamaları geliştirmek, test etmek, kullanıma açmak ve yönetmek için özel yazılım ve araçlar sunan geliştirme ortamlarıdır. Bunların amacı, birden fazla platform, araç ve kimlik doğrulama ve veri senkronizasyonu gibi karmaşık işlemler için becerilerin sürdürülebilmesi gerektiği durumlarda geliştirmeyi basitleştirmek ve geliştirme maliyetlerini azaltmaktır.



Mevcut çözümler arasında şunlar bulunur:

- Amp Chroma by Antenna⁹
- IBM MobileFirst Platform¹⁰
- Kony KonyOne¹¹
- SAP Mobile Platform¹²
- Spring Mobile Solutions¹³

Kurumsal Uygulamalarda Güvenlik

Herhangi bir IT departmanının temel görevlerinden bir tanesi de, şirket altyapısının tüm unsurlarını herhangi bir veri sızıntısı olmaması veya hiçbir verinin çalınmaması için olası saldırılara karşı güvenli getirmektir. Mobil cihazlar, şirketin IT altyapısının bir uzantısı olduğu için bütün Ticari Kuruluş uygulamaları, bir şirketin dâhili ağına hukuk dışı bir şekilde erişim sağlanamayacak şekilde tasarlanmış olmalıdır. Bir kurumsal

⁹ <http://www.antennasoftware.com>

¹⁰ <http://www.ibm.com/mobilefirst/us/en/why-ibm-for-mobile/platform.html>

¹¹ <http://www.kony.com/products>

¹² <http://www.sap.com/mobileplatform>

¹³ <http://www.springmobilesolutions.com>



uygulama geliştiricisi olarak sizden genellikle, şirketin kendi güvenlik politikasında belirtilmiş olan standartlara uymanız istenecektir; bu yüzden veri şifrelemesi, ağ iletişimi hakkındaki güvenlik konularına ve jail break veya kırma işlemlerine maruz kalmış cihazların işleme koyulmasına ilişkin çeşitli soruları cevaplandırmaya hazır olun.

Birçok MDM ve MAM sağlayıcısı, uygulama yazıcılarının kullanabileceği bir SDK sunma veya uygulama kaydırma teknikleri gibi unsurları kullanarak aslında uygulama güvenliğini artırmaktadırlar. Bu özelliklerin yanı sıra, bu platformların düzenli olarak güncellenmesi, ticari kuruluşun uygulamalarını uzaktan kilitleyebilmelerini ve ayrıca güvenlik açısından fazla vakit ve para harcamadan sürekli olarak değişen güvenlik ortamındaki güncellemeleri yakından takip edebilmelerini mümkün kılar.

Kurumsal Uygulamaların Güvenliğini Sağlamada Önemli Noktalar

- Eğer bir MDM veya MAM sunucusu kullanıyorsanız, sizin ticari kuruluşunuzun standartlarını sağlamaları gereken güvenlik seviyelerine sahip olduklarından emin olun.
- Cihaz üzerinde herhangi bir veri depolarken onun şifreli olduğundan emin olun
- Web hizmetleri ile iletişimde bulunurken daima https protokolünü kullanın
- https kullanımının yanı sıra, web hizmetleri ile iletişime geçerken, bağlandığınız sunucunun/cihazın geçerli olduğundan emin olmak için hem uygulama hem de web hizmeti içerisinde uç nokta kontrolünü gerçekleştirdiğinizden emin olun.
- Değerlerin cihaza gönderildikten sonra değiştirilmeyeceğinden emin olmak için uygulamanızın

paketlendiđi herhangi bir ayarın sađlama toplamına sahip olduđunu daima kontrol edin.

- Uygulamanın jail break yapılmıř veya kısılmıř cihazlar üzerinde alıřmasına izin vermeyin.
- Uygulamanın, kendi gvenliđini sađlayamadıđı durumlarda, kendisini devreden ıkartabilmesi iin bir ynteme sahip olduđundan emin olun.
- Kullanılan btn řifreleme yntemlerinin, uygulamanın kullanıldıđı ilgili blge(ler)in mevzuatı ve ihracat kanunları ile uyumlu olduđundan emin olun.



Mobil Mantıksal Analiz

Uygulamalarımız, belki de hiçbir zaman tanışmayacağımız insanlar tarafından uzaktan kullanılmaktadır. Mobil Mantıksal Analiz, uygulamanın bir sonraki sürümünü geliştirmek için uygulamanın nasıl kullanıldığını görmemize yardımcı olur. En iyi mobil uygulamalarının yarısından fazlası halihazırda¹ mobil mantıksal analiz aracını kullanmaktadır.

Seçeneklerin sayısı bizim şımarmıza neden olmuştur; kampanya izlemeden yazılım kalitesini iyileştirmeye kadar çok çeşitli mobil mantıksal analiz hizmeti sunan en az 20 şirket bulunmaktadır. BlackBerry bile hem Java-tabanlı eski platformu² hem de yeni platformun 10.2 sürümü için³ mobil mantıksal analiz hizmetini teşvik etmektedir. Bunların çoğu, hata raporlaması, müşteri ve gelir izlemesi gibi ilave özellikleri içermektedir. Neredeyse yarısı, belki de kendi kütüphanelerinin nasıl davranacağı konusundaki korkuları yatıştırmak amacıyla,⁴ açık kaynaklı uyarlamalar sunmaktadır.

Mobil mantıksal analizin, uygulamanızın nasıl kullanıldığını keşfetmeniz açısından size nasıl yardımcı olabileceği konusunda aşağıdaki ipuçları ve kılavuzları okumaya devam edin. Uygun bir çözümü alıp bunu uygulamanızda kullanmayı öğreneceksiniz.

- 1 blog.velt.com/mobclix-index-the-when-where-what-of-apps, static.usenix.org/event/sec11/tech/slides/enck.pdf
- 2 github.com/blackberry/WebWorks-Community-APIs/tree/master/Smartphone/Analytics
- 3 devblog.blackberry.com/2013/08/discover-whats-coming-in-the-blackberry-10-2-sdk/
- 4 readwrite.com/2013/12/05/why-mobile-developers-need-open-source-analytics-embedded-in-their-applications

Başlarken

Birçok mobil mantıksal analiz çözümü sağlayıcısı, kendi ürünlerini tanımanız için size bir 'Başlarken' bölümü sunar. Örnekler arasında Flurry⁵ ve KISSmetrics⁶ bulunur. Uygulamanız için özel bir 'anahtar' ile yapılandırma gerekli olduğundan genellikle ürünleri kullanmadan önce kaydolmanız gerekir.

Herhangi bir ürünü tam olarak kullanmadan önce potansiyel çözümlerin bazılarını deneyin. Belgelendirmeleri okuyun ve bunları uygulamanızda nasıl kullanabileceğinizi görmek için örnek kodu inceleyin. Tüm bunların yanı sıra gizlilik politikası dâhil olmak üzere hukuki sözleşmeleri de kontrol edin. Daha sonra mobil mantık analizini uygulamanızda deneyebilmek için bunlardan en az bir tanesini seçin. Onların kodlarını entegre ederek, uygulamanızda mobil mantıksal analizi kullandığınızda neleri izleyebileceğinizi ve mobil mantıksal analizin uygulamada nasıl çalıştığını daha kolay bir şekilde görebilir ve öğrenebilirsiniz.

Çoklu platform uygulamaları için, her bir platform üzerinde tutarlılık isteyebilirsiniz; aksi takdirde, özellikle farklı platformlar için farklı mobil mantıksal analiz çözümleri kullanılıyorsa farklı ve hatta kıyaslanamayan veri setlerini kıyaslamaya çalışmakla uğraşabilirsiniz. Uygulamanızı piyasa sürmeyi düşündüğünüz bütün platformları destekleyen ortak bir çözüm bulmaya çalışın.

Odaklanmak isteyebileceğiniz iki sunucu bulunmaktadır. Segment.io⁷ çok çeşitli diğer mobil mantıksal analiz ürünlerini sunmaktadır ve kodunu github.com/segmentio adresinde açık kaynaklı bir şekilde sunmuştur. Farklı mantıksal analiz

⁵ support.flurry.com

⁶ support.kissmetrics.com/getting-started/overview

⁷ segment.io

sağlayıcılarına alışmak için gerekli olan çabayı azaltmak için farklı izleme yöntemlerini uyarlamaktadır. Count.ly⁸, sunucularının yanı sıra istemci kütüphanelerinin açık kaynaklı uygulamalarını sunmakta ve kendi ürününü değerlendirmek isteyenler için bütünlüklü bir test ortamı yaratmaya teşvik etmektedirler.

Dikkatli olun; bazı mobil mantıksal analiz çözümü sağlayıcıları uygulamanız tarafından rapor edilen verileri kullanabilir ve bunları diğerlerine sunup satabilir. Bu verinin yaşam süresini kontrol edebilir; bu yüzden veriye ulaşmanız mümkün olmayabilir. Diğer taraftan bu bilgileri koruyabilir ve siz uygulamanızı piyasadan çektikten sonra bunları kullanabilir. Veri içerisinde kişisel anlamda tanımlanabilir bilgiler bulunuyorsa bu noktada bazı hukuki ve mahremiyetle ilgili ek sorunlar çıkabilir. Bu yüzden üçüncü kişilerin kendi yazılımları ve API'lar üzerinde rapor edilen bilgileri nasıl kullanacaklarını ve paylaşacaklarını bilmek önemlidir.

Neyin Ölçüleceğini Belirlemek

Uygulamanın nasıl kullanılacağını anlamak için neyi ölçmek istiyorsunuz? Aşağıda bazı tavsiyeleri bulabilirsiniz:

- **Önemli kullanım olaylar:** Kullanım; örnek olarak yeni bir arama seçeneği veya uygulamanız üzerinde sosyal ağ programlarını açtıklarında.
- **Gelir yaratıcı olaylar:** Kullanıcının, sizin için gelir yarattığı herhangi bir etkileşimi: Kullanıcılar ne sıklıkla uygulamanızın bir üst sürümünü veya yazılımınızı içerisindeki öğeleri satın alıyor? Alışverişi bitirmeden

önce siparişleri ne zaman iptal ediyor veya alışverişten vazgeçiyorlar?

- **Kullanılabilirlik ölçüleri:** Kullanıcılar, kullanım akışı içerisinde nerede takılıyorlar? Yazılımınızı kullanırken hızlı bir şekilde hedeflerine ulaşabiliyorlar mı?

Odaklanmak istediğiniz temel alanları belirledikten sonra, örnek olarak hangi veri unsurlarının raporlanacağı gibi, mantıksal analiz ölçütlerini tasarlamanız gerekecektir.

Nasıl Ölçüm Yapılacağının Belirlenmesi

Neleri ölçtüklerini kolay ve doğru bir şekilde hatırlayabilmeniz için etkileşim olaylarınız için mantıklı isimler oluşturun. Her bir olay için, hangi unsurların dâhil olacağını kayıt altına almanız ve buna karar vermeniz gerekir. Verilerin toplandıktan sonra nasıl kullanılacağını düşünün: her bir rapor ve grafiği oluşturmak için çeşitli veri unsurlarının nasıl işleneceğini belirleyin ve tipik raporları ve grafikleri tasarlayın.

Her bir unsurun zaman bilgisi gibi küreselleştirme sorunlarına değinmeyi unutmayın. Uygulama, cihazın yerine, cihazın ayarlarına göre bir olayın zamanını algılayabiliyor mu veya UTC zamanı⁹ gibi küresel bir zaman mı kullanıyor?

Bazı Mobil Mantıksal Analiz çözümleri, veri unsurlarını sunucuya otomatik olarak kaydedip raporlayacaktır. Bu unsurların ne olduğunu, nasıl ve ne zaman raporlandığını ve bunların nasıl biçimlendirildiğini bilmek önemlidir. Daha sonra bunları kullanıp kullanmayacağınızı ve otomatik olarak oluşturulan bu raporlara güvenip güvenmeyeceğinize karar verebilirsiniz.

Özel olay etiketleri, önceden tanımlanmış olayları çoğaltabilir ve birçok mobil mantıksal analiz çözüm sağlayıcısı,

⁹ en.wikipedia.org/wiki/Coordinated_Universal_Time

uygulamanızın bunları oluşturması için çeşitli yöntemler sunmaktadır. Özel olay mesajlarını biçimlendirmeniz gerekebilir. Eğer buna gerek duyarsanız, unsur ve ayıncıların dil kodlamalarına dikkat etmeniz gerekir. Örnek olarak, bunlar REST mesajları¹⁰ olarak gönderilmişse URL kodlu¹¹ olarak kodlandırılmaları gerekebilir.

Sunulan mantıksal analiz verisinin güvenilirliğini ve doğruluğunu etkileyebileceği için, mantıksal analiz sisteminin mevcut kapasitesinin aşılma riskini azaltmak amacıyla uygulamanın olayları ne sıklıkla rapor edeceğini ayarlamak isteyebilirsiniz. Mantıksal analiz çözümleri tarafından işlenen veri hacmini azaltmak için kullanılan bir yöntem örneklemedir. Adam Cassar, periscopix.co.uk/blog/should-you-be-worried-about-sampling/ adresinde bu konu hakkında ilginç bir blog makalesi yayınlamıştır.

Kodunuzu Ayarlamak

Uygulamanız ile kullanıldığında mobil mantıksal analizin düzgün bir şekilde çalışabilmesi için gerekli olan ilave becerileri belirlemeniz gerekebilir.

Android için bunlar izinler olarak bilinir. Bu mantıksal analizler, olayların online olarak rapor edilebilmesi için büyük ihtimalle İnternet izinleri ve çözümün, telefonun yerini kaydettiği durumlarda konum tabanlı izinler isteyecektir. Eğer uygulamanız halihazırda izinler kullanıyorsa, bunların kullanımını yeniden belirlemenize gerek yoktur.

iOS için `UIRequiredDeviceCapabilities`, iTunes ve App Store'a, uygulamanın cihaza ilişkin hangi özelliklere gerek duyduğunu bildirecektir. Unsurların, anahtar kullanarak

¹⁰ msdn.microsoft.com/en-us/library/live/hh243648

¹¹ en.wikipedia.org/wiki/Percent-encoding

belirlendiği bir sözlük olarak uyarlanır. Anahtarlar arasında WiFi, yer hizmetleri ve GPS bulunmaktadır.

Windows Phone için, uygulamanın kullanacağı unsurları belirlemek için kabiliyetler kullanılır. Localytics, `ID_CAP_IDENTITY_DEVICE` kabiliyetini ayarlayan bir örnek içeren online hızlı başlangıç kılavuzuna¹² sahiptir.

Sonuçlarla Uğraşmak

Bir uygulamanın bir mantıksal analiz olayı göndermesinden, bilginin işlenip size ulaşmasına kadar geçen süre içerisinde bir gecikme gerçekleşmektedir. Gecikme veya geri kalma, 'gerçek zamandan' birkaç saate kadar uzayabilir. Siz ve işletme sponsorlarınızla gerçek zamanlı olayları ne kadar geciktirebileceğiniz konusunda ortak bir karar vermelisiniz.

Bazı mantıksal analiz çözümü sunucuları, veriye erişim sağlayabilmeniz için bir API sunmaktadır. Bu, özel raporlar yaratmak için size daha geniş bir kapsam sunabilir. Bunlardan bazıları, verileri daha geniş ölçüde kontrol edebilmenize ve nasıl kullanılabileceğini belirlemenize yardımcı olan sunucuları dâhil etmenize yardımcı olur.

Sonuçların kalitesini değerlendirmek amacıyla bazı organizasyonlar uygulamaları içerisine birden fazla mantık analizi çözümlerini dâhil etmeye çalışırlar ve sonuçları karşılaştırırlar. Ancak, birbiriyle çakışan iki sonuç uyumlu olmayı zorlaştırabilir; bu yüzden üçgenleme¹³ yöntemi ile farklılıkları tanımlamak amacıyla üç farklı sonuç setini kullanmak gerekebilir.

KISSmetrics ile çalışmaya karar verdiyseniz, support.kissmetrics.com/getting-started/testing-km adresinde ölçümlerinizi

¹² localytics.com/docs/windows-phone-7-integration/

¹³ [en.wikipedia.org/wiki/Triangulation_\(social_science\)](http://en.wikipedia.org/wiki/Triangulation_(social_science))

test etmek için farklı yöntemleri belirten makaleye bir göz atabilirsiniz.

Gizlilik

Son kullanıcıya, uygulamanın nasıl kullanıldığı hakkında çeşitli bilgilerin kayıt altına alınacağını ve paylaşılacağını, tercihen kendi şartlarınız ve koşullarınız doğrultusunda, açıklamalar yapmayı unutmayın. Kullanıcıların, uygulama kullanımlarının izlenip izlenmeyeceğini karar vermelerini isteyebilir veya buna ihtiyaç duyabilirsiniz. Eğer bunu isterlerse, kullanıcı için ayarların kontrolünü kolaylaştırın ve kayıtlı verilere erişimde bulunmak, silmek veya mantıksal analiz çözüm sağlayıcısı ile iletişime geçebilmek için kullanıcıya bir yöntem sunun.

Üçüncü taraf kütüphane sağlayıcıları, çok farklı gizlilik tavırlarına sahiptir. Bazıları kullanıcıların gizliliğinin hayati önem taşıdığını düşünür ve izleyicileri izlememenin önemini vurgularlar. Google Analytics, kendi hizmet kullanım şartlarında¹⁴ açık bir şekilde, kişisel olarak tanımlanabilen bilgilerin izlenmesini yasaklamıştır. Diğerleri, kişisel olarak tanımlanabilen verilerin açık bir şekilde nasıl kaydedilebileceğini gösteren kaynak kod parçaları gibi örnekleri sunmaktadırlar. Örnek vermek gerekirse, KISSmetrics şu kod parçasını¹⁵ sunmaktadır: `[[KISSMetricsAPI sharedAPI] identify:@"isim@eposta.com"];` Bunun yanı sıra Mixpanel bir kullanıcının People Analytics kaydının nasıl güncelleneceğinin bir örneğini¹⁶ adresinde sunar.

¹⁴ google.com/analytics/terms/us.html

¹⁵ support.kissmetrics.com/apis/objective-c

¹⁶ mixpanel.com/docs/people-analytics/android

Gizlilik ve kullanıcılara ilişkin verilerle çalışmanın ahlaki yönleri hakkında daha fazla bilgi almak için çeşitli kaynaklardan faydalanabilirsiniz:

- Jeff Northrop'un Mobile Analytics'teki¹⁷ blog makalesi
- Kord Davis'in kitabı "Ethics of Big Data"
- Perakende endüstrisi, yararlı genel bilgilere sahip mobil uygulamalar hakkında bir davranış kuralı yayınlamıştır: futureofprivacy.org/issues/smart-stores

Daha Fazla Bilgi

Umuyoruz ki bu bölüm, Mobil Mantıksal Analiz hakkında daha fazla bilgi alabilmeniz için faydalı olacaktır. Aşağıda, devam eden araştırmalarınız için ilave bilgiler bulabilirsiniz:

- Bu kitabın kardeşi olan Mobile Developer's Guide to the Parallel Universe: Mobil Mantıksal Analizi, pazarlama bakış açısından ele almaktadır. www.wipconnector.com adresinden PDF formatında indirilebilir.
- thenextweb.com/dd/2013/08/11/9-tools-to-help-you-measure-mobile-analytics, konuya geliştiricinin bakış açısından bakan kullanışlı başka bir kaynaktır.
- kontagent.com/resources, hangi uyarlamayı kullanırsanız kullanın, mobil mantıksal analiz hakkında çok geniş çapta tanıtım belgelerini, popüler webinar'ları ve diğer malzemeleri içeren bir sitedir.

¹⁷ jnorthrop.me/2012/07/2/privacy-considerations-mixpanel-people-analytics/



Zengin Medya Desteği Vermek

“Telefon sayısı kadar standart sayısı”, cep telefonlarında desteklenen medya biçimleri söz konusu olduğunda bilinen bir gerçek olarak ortaya çıkmaktadır. Çoğu ses ve video biçimlerinin desteklendiği veya bunun için yalnızca bir kod çözücünün kolaylıkla yüklenilebildiği PC’lerin aksine cep telefonlarında bu sistem çok farklıdır. Ekran boyutunun ve bant genişliğinin optimizasyonuna olanak sağlamak için geçtiğimiz yıllarda çeşitli mobil formatları ve protokolleri geliştirilmiştir. Çözünürlük, bit hızı, taşıyıcı, protokol veya kod çözücülerdeki küçük farklılıklar, oynatmanın başarısız olması ile sonuçlanabilir, bu yüzden daima gerçek cihazlar üzerinde test yapmalısınız.

Bununla beraber, günümüzde akıllı telefonların çoğu MP4 h.264 640x480 AAC-LC’yi desteklemektedir. Ancak telefonlar arasında (aynı satıcı veya aygıt yazılımı olsa bile) birden fazla farklılık ortaya çıkabilir. Her sene, Android 4+ üzerinde çalışan açık kaynaklı video standardı WebM/vp8¹ gibi biçimler eklenmektedir. Ancak çoğu geliştirici tarafından başka bir h.264 türü gibi görüldüğü için gereken desteği bir türlü alamamaktadır. Şu anki eğitim, ekran çözünürlüklerinin gittikçe büyümesi ve Full

1 en.wikipedia.org/wiki/VP8



HD TV çözünürlüklerinin pil ve hafıza pahasına mobil ekranlara taşınmasıdır.

Aşağıda, en yüksek uyumluluk için tavsiye edilen tam ekran biçimleri bulunmaktadır:

Taşıyıcı	mp4, 3gp, avi (yalnızca Black-Berry), wmv (yalnızca Windows Phone + BB10)
Protokol	HTTP (progresif veya indirme) or RTSP (akıtma)
Video	H.264, H.263
Ses	AAC-LC, MP3, AAC+
Klasik Çözünürlükler	176x144 (Eski telefonlar), 320x240, 480x320 (J2ME)
Yaygın çözünürlükler	480x800, 640x480 (Blackberry), 960x640 (iPhone), 1024x768 (iPad 1+2), 2048x1536 (iPad 3+4)
HD Çözünürlükler	1280x720 (BB10, Samsung, Windows Phone 8), 1136x640 (iPhone 5)
Full HD / aka 1080p	1080x1920 (HTC One, Samsung Galaxy S4)
Quad HD / 2K aka 1440p	1440x2560 (Samsung Galaxy S5, Vivo xPlay)

HD çözünürlüklerini destekleyen cihazların ayrıntılı listesini² adresinde bulabilirsiniz.

² http://en.wikipedia.org/wiki/List_of_mobile_phones_with_HD_display

Akış ve Yerel Depo Kıyaslaması

Medya içeriğini mobil cihazlara aktarmanın iki yolu bulunmaktadır. Yerel dosya sisteminden çalmak veya bir sunucu üzerinden gerçek zamanlı olarak "akıtmak".

Nispeten dengesiz mobil ağlar üzerinden bir içeriği akıtmak için, gecikme ve arabelleğe alma sorunlarını çözen, RTSP isimli bir protokol geliştirilmiştir. Tipik çerçeve hızları MP4 için 15 fps, 3gp için 25 fps iken, veri oranlarına baktığımızda GPRS için 48 kbps (yalnızca ses), Edge için 200 kbps, 3G/UMTS/WCMA için 300 kbps ve HSDPA, Wi-Fi ve 4G için 500 kbps gibi rakamlara ulaşılır. HD-video 2MBps'den başlamakta ve henüz internetten akışlar için tavsiye edilmemektedir.

Hedeflenen platform Windows Mobile/Phone olduğunda HTTP akışını desteklemek için Windows Media Services³ tercih edilir. Android 3.0 ve yukarısı da HTTP akışını desteklemektedir. Kademeli (progresif) indirme için "Atomich hinting" denilen özelliğin bu amaçlar için gerekli olduğunu ve mp4 dosyalarının dil kodlamasında çok katı (H.264 15 fps AAC-LC 48khz stereo kullanın) olduğunu unutmayın. Akış biçimlerinde yalnızca HTC Android cihazları ve Android 4.0 cihazları daha az katıdır ve diğer markalara kıyasla daha fazla kodlanmış içeriği oynatacaklardır.

Telefon üzerinde akışın mümkün olmadığı, taşıyıcının (GSM gibi) bloke ettiği veya kullanıcının her seferinde bağlantı olmadan medyayı oynatabilmesine fırsat vermek istediğiniz durumlarda, çok basit bir şekilde dosyanın bağlantısını sunup indirme fırsatını verebilirsiniz. Bu işlem web üzerinde normal bir indirme bağlantısı vermek kadar kolaydır. Ancak unutmamak gerekir ki cep telefonları, doğru MIME tiplerinin kontrolünde sıkı davranabilirler. 3gp dosyaları için "audio/3gp" veya

3 <http://technet.microsoft.com/en-us/windowsserver/dd448620.aspx>

"video/3gp" ve mp4 dosyaları için video/mp4 MIME tiplerini kullanmaya özen göstermelisiniz.

Bazı telefonlar veri tipini algılamak için yalnızca dosya uzantılarını kullanmaktadırlar. Bu yüzden bir komut dizisi (download. php gibi) kullandığınızda ki iyi bilenen bir yöntemdir, medyanın doğru bir şekilde işlendiğinden emin olmak için `download.php?dummy=.3gp` gibi bir parametreyi bağlantısının sonuna eklemelisiniz. Bazı telefonlar video olmadan 3gp ses dosyasını oynatamamaktadırlar ancak bunun üzerinden gelmek için dosya içerisinde boş bir video izi veya bir album kapağının görüntü resmi koyulabilir.

Uzantı ve protokole göre, farklı oynatıcılar bu talebi işleyebilirler. Android gibi bazı telefonlarda, birden fazla medya oynatıcısı, kullanıcının aralarından seçim yapabilmesine fırsat vermek için açılan bir ekranda görünebilir. Kullanıcı deneyimini geliştirmek için, sııırlı fonksiyona sahip olan gömülü oynatıcıları kullanmaya özen göstermelisiniz.

Son olarak, medya dosyalarınızı uygulamanız içerisine bir kaynak olarak dahil edebilirsiniz. Android cihazlarda, SD kartta bulunan medyaları oynatabilmek için (Android 3.1 ve yukarısı) `android.permission.READ_EXTERNAL_STORAGE` iznini aldığınızdan emin olmalısınız.

Kademeli (Progresif) İndirme

Bir akış sunucusunu yapılandırmakla uğraşmak yerine medya dosyalarının herhangi bir web sunucusundan alınabildiği kademeli indirmeyi kullanarak iyi bir alternatif sunabilirsiniz. Bunu yapmak için dosyalarınızla ilgili `_ipucu_` vermelisiniz. İpucu vermek, mobil oynatıcının dosyanın küçük bir kısmını (ilk 15 saniyesi) indirdikten sonra dosyayı oynatabilmesi için medya içerisindeki çeşitli yerleri işaretleme işlemidir. Mevcut telefon kapasiteleri ve veri ağları ile beraber bu yöntem,

medya dosyalarını idare etmek için tercih edilen bir yöntem haline gelmiştir. Not: Bir mp3 dosyası için ipucu vermeye gerek yoktur.

Büyük olasılıkla mevcut olan en güvenilir açık kaynaklı ipucu oluşturma yazılımı Mp4box'tır⁴. Ancak gün geçtikçe daha fazla medya aracı içerisinde ipucu desteğiyle beraber gelmektedir.

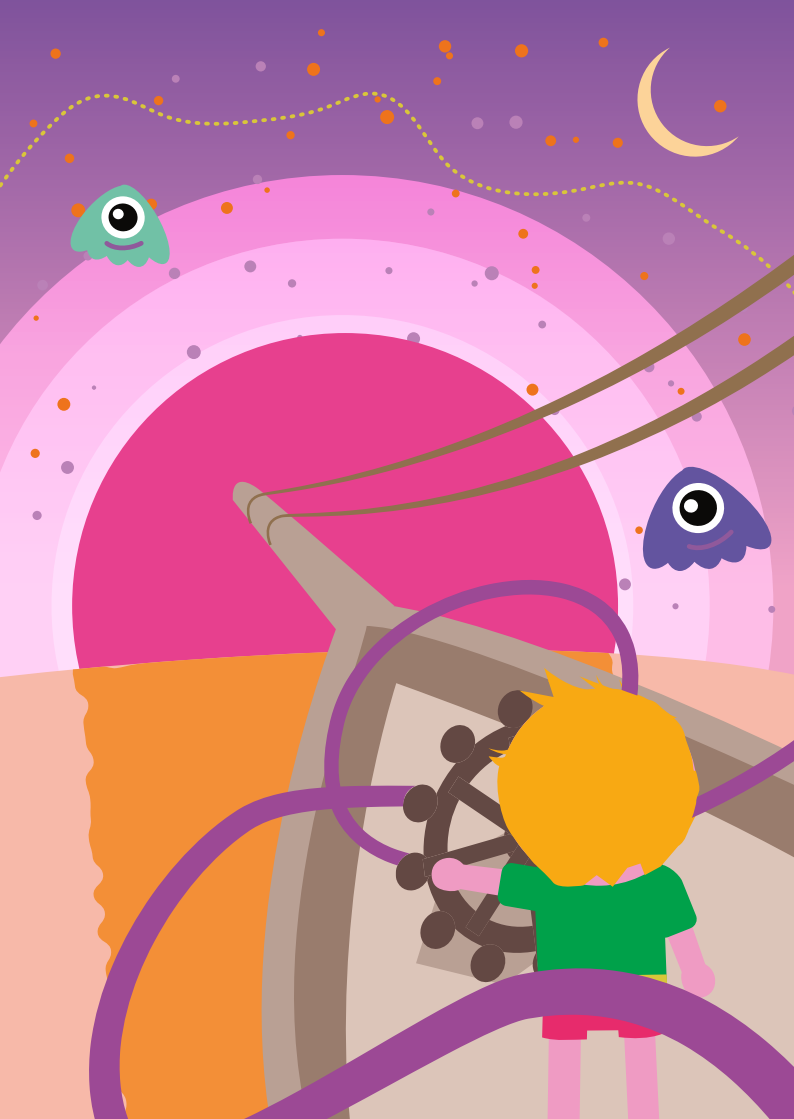
Medya Dönüştürücüleri

Mevcut çok çeşitli medyayı cep telefonunun desteklediği biçimlere dönüştürmek için FFMPEG, sahip olunması gereken bir medya biçim dönüştürücüsüdür (açık kaynaklı). Çerçeve hızını, bit hızını ve kanalları aynı anda ayarlayabilir. FFMPEG'i düzgün kullanabilmek için H263, H264, AAC ve AMR gibi dil kodlayıcılarını kurmuş olmalısınız. eRightSoft⁵ "Super" gibi FFMPEG tabanlı birçok iyi dönüştürücü bulunmaktadır. MAC kullanıcıları için, QuickTime pro (ücretli sürüm), 3gp ve mp4 dosyalarını kodlamak ve ipucu oluşturma işlemini gerçekleştirmek için iyi bir alternatiftir. Eğer Java/açık kaynak geçmişine sahip bütün bir ağ çözümü istiyorsanız Alembik'e⁶ bir göz atın.

⁴ gpac.wp.institut-telecom.fr/mp4box/

⁵ www.erightsoft.com/super

⁶ www.alembik.sourceforge.net



Konum Tabanlı Hizmetlerin Uygulanması

Konum tabanlı hizmetler, mobil uygulamalar için sıcak alanlardan biri olmaya devam ediyor.

Bir kullanıcının konumunun bilinmesi, onlara daha doğru bilgi sunulabilmesi, yakındaki bir vejetaryen restoranını bulmalarına, yerel hava tahminlerini gözden geçirmenize, bir toplantıda arkadaş bulmanıza veya diğer bisikletçilerden faydalanarak manzaralı yerel bisiklet yollarını bulmanıza yardımcı olunabileceği anlamına gelir. Konum verilerinin alınması hikâyenin sadece yarısıdır, kullanıcıya anlamlı bir tanıtım yapılması önemli bir etkidir. Bu genellikle, yollar, ilgi noktaları ve diğer alakalı bilgiler ile dolu olan bir grafik sunumunu ifade eder. Ancak basitliği göz ardı etmeyin; yakınlık ile sıralanmış olan ayrıntılı bir liste, kaydırmalı, yavaş bir harita görünümünden daha faydalı olabilir (özellikle kullanıcı, mb başına beş Avro'luk bir tarifeyi kullanıyorsa).

Konumlandırma Verileri Nasıl Alınır?

Konum tabanlı uygulamalar çok sayıda kaynaktan yerel bilgileri toplayabilir. Bunlar şöyle sıralanabilir; telefonun uygun şebeke bağlantılarından biri, GPS uyduları, görünür etiketlere dayanan kısa mesafeli sistemler veya yerel kısa mesafeli radyo ya da ekran veya klavye aracılığıyla veri girişi yapılan modası geçmiş yöntemler.

1. **Ağ konumlandırma:** Her bir GSM veya UMTS baz istasyonunun kendine özgü benzersiz bir kimliği (ID) vardır, bu ID ülke kodu, ağ kimliği, beş haneli Konum Alanı ve iki haneli Routing Alanından oluşur. Daha sonra, baz istasyonunun koordinatları operatörün veri tabanındaki beyanına bakılarak elde edilebilir. Sonuçta ortaya çıkan konum çok hassas değildir, hücre boyutuna bağlıdır (baz istasyonu kapsama alanı): şehir bölgelerinde yoğun bir şekilde konumlandırılmış olan hücreler, kırsal bölgelerde bulunan seyrek hücrelerden daha fazla seviyede doğruluk sunmalıdır. Sinyallerin yakında yer alan çok sayıda baz istasyonundan ulaşma süresindeki farkın ölçülmesi gibi teknikler (multilateration olarak bilinir) hassasiyeti artırmaya yardımcı olabilir, ancak mobil operatörler bu birinci sınıf ağ hizmetleri için ücret talep edebilir. WiFi özellikli telefonlar için kablosuz LAN erişim noktaları ile ilgili ayrıntılar mobil ağ konumlandırmaya alternatif olarak veya onu tamamlayacak şekilde kullanılabilir. Bu teknik, Google dâhil pek çok şirket tarafından kullanılıyor.
- **GPS konumlandırma:** Bütünleşik bir GPS modülü (veya harici) genellikle, donanımın kalitesine ve GPS modülünün sabitleme yaparken kaç adet uydu görebildiğine bağlı olarak 5-50 metre arasında size %50 hassasiyet sağlar. Hassasiyet aynı zamanda araziden, gölgelikten ve duvar malzemelerinden de etkilenir; bunların herhangi biri uydu sinyallerini engelleyebilir: kentlerde, yüksek bina yığınlarının oluşturduğu kentsel kanyonlarda sinyaller bozulabilir, bu durum hatalı veya kusur okumalara neden olur. GPS'in ağ konumlandırması ile birleştirilmesi giderek yaygınlaşıyor: Yardımlı GPS veya A-GPS ilk GPS sabitlemesinde gecikmeyi en aza indirmek amacıyla Yardımcı Sunucu adı verilen bir aracı kullanır. Sunucu, yörünge verilerini, hassas ağ zamanlamasını ve ağ tarafında GPS

bilgisi analizini kullanır. Ancak A-GPS daha hassas bir konum anlamına gelmez. GPS'in ilk olarak etkinleştirildiğinde veya GPS uydu kapsamının yetersiz olduğunda daha hızlı bir sonucu sunar. Bu durum, bir konum kilitlemesi için gerekli süreyi kısaltır.

Not: Çoğu A-GPS çözümü için aktif bir cep telefonu bağlantısı gerekir.

- **Kısa mesafeli konumlandırma:** Sensörlere dayalı sistemler; yakın alan iletişimi (NFC), Bluetooth (Bluetooth Low Energy dâhil) ve diğer radyo tabanlı etiket sistemleri – bir müzedeki sergi veya alışveriş merkezindeki mağazalar gibi ilgi noktalarının yakınında aktif veya pasif sensörler kullanılır. Düşük teknoloji çözümlere, fotoğrafı çekilebilen ve bir sunucuda veya yerel olarak telefonda analiz edilebilen barkod ve diğer görsel etiketler (QR kodları gibi) dahildir; bu tür etiketler bir id içerebilir ve buna göre bir konum bulunabilir. Kullanıcı, haritada bir konum seçerek, bir alan kodu veya fiziksel bir adres girişi yaparak konumunu belirtebilir. Bu seçenek genellikle, temel özellikli telefonlardaki uygulamalar için kullanılır. Bu cihazlarda, başka konum belirleme yolları mevcut değildir.

Apple'ın iBeacon'ı da dâhil olmak üzere yakın zamanda gerçekleştirilen Bluetooth 4.0 uygulamaları, bir üçgenleme yöntemi olarak radyo id'lerini alan bir sinyal gücü göstergesi (RSSI) ile birleştirmektedir. Ancak, bu tekniklerin doğruluğu henüz kesinleşmemiştir. Öte yandan ham konumlandırma ve gerçek zamanlı envanter için ekipman, değerli varlıklar, evcil hayvanlar ve benzerleri üzerine Bluetooth künyeler yerleştirmek, 2015 yılında gittikçe popüler bir hale gelecektir ve bunları yeni cihazlar ve hizmetler takip edecektir.

Haritalama Hizmetleri

Genellikle, bir harita hizmeti girdi parametreleri olarak bir pozisyonu ve metadata koleksiyonunu alır ve bağlamsal bir metadata katmanına sahip bir bit işlemi veya bir vektör haritasını sunar. Vektör verilerinin bit eşlemler (bitmaps) karşısında çok sayıda avantajı vardır. Bunlar: vektör sunumları daha az bant genişliği harcar ve bir harita görüntülemesinde gelişigüzel büyütmelemlere olanak sağlar. Bununla birlikte, müşteri için daha fazla işlem gerekir. Bit işlemleri genellikle, her biri kendi koordinatları ve yakınlaştırma seviyeleri tarafından adlandırılan kesik yakınlaştırma seviyeleri ile beraber sunulmaktadır.

Ücretsiz haritalar hem bit eşlemler (bitmaps) hem de vektör olarak sunulur ve şunları içerir: Open Street Map¹ veya Cloud-Made². Ticari haritalar ise şunları içerir: Garmin³ ve Microsoft's Bing resources⁴.

Google Maps⁵ gibi bazı çözümler, uygulamanız ücretsiz sunulduğu zaman ücretsizdir, ancak bir harita anahtarı elde etmeniz gerekir. Google'ın statik haritaları gibi diğer harita hizmetleri, bir harita anahtarı numarası veya IP adresine hizmet vermek ile sınırlıdır. Kaynakların çoğu benzer harita formatlarını paylaşır ve bu nedenle değiştirilebilirler.

¹ wiki.openstreetmap.org/wiki/Software

² www.cloudmade.com

³ garmin.com

⁴ www.microsoft.com/maps/developers

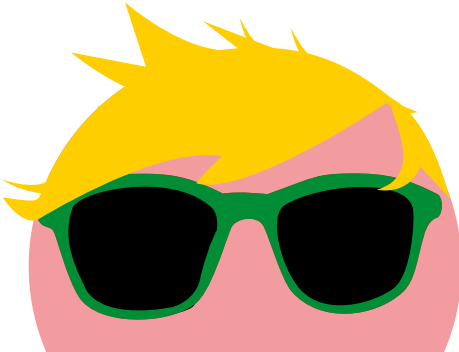
⁵ code.google.com/apis/maps

Konum Desteęinin Uygulanması

Java ME için konum API'si enlem ve boylam konumu, hassasiyet, cevaplama süresi ve bütünleşik GPS'den alınan rakımın yanı sıra gerçekleştirilen okumalara dayanarak hız gibi ayrıntıları sunar.

WiiOS'ta konum için entegre destek bulunur ama destekleyen işlevler tarafından konum verilerinin nasıl oluşturulabileceęi ve hangi harita kaynaklarının kullanılabileceęi ile ilgili kısıtlamalar mevcuttur. Şu anda, konum verilerinin iOS cihazlarında nasıl kaydedildięi ve depolandıęı ve Apple'ın bu verileri kendi için nasıl kullanmayı planladıęı konusunda devam eden bir tartışma vardır. Android geliştiricilerinin yüksek seviyedeki kütüphanelere erişimi bulunmaktadır ve bu cihazlar, Google'ın harita API'larını varsayılan olarak ele alsalar da harita kaynakları seçimi açısından daha serbesttirler.

Windows Phone 7 ve Windows 8 tabanlı cihazlarda, Windows Phone 8'deki tüm yeni Maps API'nin Windows Phone 7'de mevcut olan Bing Maps ile aynı olmadığını unutmayın. Windows Phone 8'de Bing Maps kontrolü hâlâ desteklenir ama artık kullanılmamaktadır.



iOS 3.x ve Android 2.0'dan beri, Web uygulaması geliştiriciler navigator.geolocation arayüzü aracılığıyla coğrafi bilgilere erişim sağlayabiliyor. Örneğin, `navigator.geolocation.getCurrentPosition(my_handle)`'ı çağırmak size, kullanıcıdan izin aldıktan sonra ve uyduların mevcut olması halinde `my_handle.coords.latitude` ve `my_geo_handle.coords.latitude` 'ü getirme fırsatı verir. Örnek vermek gerekirse JavaScript sözdizimi kullanıldığında gerçek bir parça örneği şöyle görünür:

```
function init_geolocation() {  
  // e.g. called from a ready() or onLoad function  
  navigator.geolocation.getCurrentPosition(  
    get_geolocation,process_errors);  
}  
  
function process_errors(error) {  
  switch(error.code) {  
    case error.PERMISSION_DENIED:  
      alert("user not sharing geolocation");  
      break;  
    case error.POSITION_UNAVAILABLE:  
      alert("could not calculate current position");  
      break;  
    case error.TIMEOUT:  
      alert("retrieving position timeout");  
      break;  
    default:  
      alert("misc. geolocation error");  
      break;  
  }  
}  
  
function get_geolocation(position){  
  alert('Latitude: ' + position.coords.latitude + '  
  Longitude: ' + position.coords.longitude);  
}
```

Bir hata, düşme yerine posta kodları ve benzerleri girilerek ağ kilitlerine geri dönme işlemiyle birleştirilebilir. Coğrafi

veriler, sadece koordinatların dışında, genellikle çeşitli formatlarda mevcut olan diğer bilgileri de sunar. En yaygın olarak kabul edilen standartlara geoRSS adı verilir ve tekli bir ilgi noktasında şu görünüme sahip olabilir:

```
<entry>
<title>Byviken's fortress</title>
<description>Swedish 1900-century army
installation, w. deep mote
</description>
<georss:point>18.425 59.401</georss:point>
</entry>
```

Konum verileri için başka formatlar vardır, ancak temel fikir benzerdir; veri akışlarını ve web hizmetlerini uyumlaştırarak farklı kullanıcı bağlamlarında sorun olmadan çalışabilmesi için sağlam sentezler yaratılabilir. Coğrafi bilgi için diğer önemli formatlar arasında, coğrafi bilgilerin taşınması ve depolanması için özel geliştirilmiş bir XML şifrelemesi olan Geography Markup Language (GML) ve Google Earth ile ilgili web hizmetlerinde kullanılan, ayrıntılı bir şekilde hazırlanmış coğrafi format olan KML yer alır.

LBS Uygulamaları için Araçlar

Çok sayıda şirket katma değerli hizmet olarak geliştirici dostu araçlar ve API'lar sunuyor. Bunların kullanılması, konum duyarlı hizmetlerin geliştirilmesi ve dağıtımını önemli miktarda hızlandırır. Araçların her biri normalde, bir veya birden fazla mobil platforma odaklanır. Admob gibi reklam şirketleri, kendi ürünlerini daha iyi hedefleyebilmek adına tek başına çalışan yer odaklı reklam programları sunmaktadır. Herhangi bir harita arayüzü sunulmaz, yalnızca koordinatlar gönderilir ve reklamların geriye dönmesi hedeflenir.

Aşağıda harita ve konum tabanlı hizmet kaynakları için bazı bağlantıları bulabilirsiniz:

- **Android çevrim dışı harita projesi:** code.google.com/p/big-planet-tracks/
- **BlackBerry:** us.blackberry.com/developers/ ("map api" ile arama yapın)
- **Garmin Mobile XT SDK:** developer.garmin.com
- **Google Maps kaynakları:** developers.google.com/maps
- **iBeacon:** support.apple.com/kb/HT6048
- **Nutiteq:** www.nutiteq.com
- **Nokia Maps:** developer.here.net
- **TeleAtlas:** developerlink.teleatlas.com
- **Windows Phone 8 Haritaları ve Navigasyon:** msdn.microsoft.com/en-us/library/windowsphone/develop/jj207045



Yakın Alan İletişimi (NFC)

2007 yılında ilk Yakın Alan İletişimi (NFC) destekli telefon (Nokia 6131 NFC) piyasaya çıktığında, teknoloji sadece son bir kaç yıldır popülerlik kazanmaya başlamıştı. İlk önemli hareket 2010 yılında Nokia ve Samsung'un akıllı telefonlarına NFC'yi eklemesi ile yapıldı. Böylece günümüzde artık Apple, çoğu Android, BlackBerry ve Windows Phone cihazı NFC çipi taşıyor.

Dört gözle beklenen teknoloji, NFC'nin güvenli bir şekilde gerçek bir cüzdanın yerini alma becerisini sunmalıydı. Cüzdan, insanların halihazırda para vermeye hazır oldukları bir kolaylıktı. Allied Market Research tarafından yapılan bir araştırma¹, 2013'ten 2020'ye kadar yıllık %127,5'luk bir büyüme oranı ile küresel mobil cüzdan piyasasının 2020 yılında 5.250 milyar dolara ulaşacağını öngörmektedir.

Bununla birlikte NFC, bugüne kadarki en büyük etkisini dokun-paylaş ve dokun-bağlan özellikleri sunarak göstermiştir.

NFC'nin kısa özeti

NFC, 90'lı yılların başından² beri kullanılan Radyo Frekanslı Tanımlama (RFID) teknolojisinin gelişmiş şeklidir. NFC'de RFID becerileri genişletilmiş ve aynı zamanda daha önceki teknolojilerle uyum korunmuştur. RFID uzun yıllardır başta nesnelerin izlenmesi ve dokun-öde uygulamaları için kullanılıyor.

NFC, mobil cihazlarda hedeflenen RFID ile bütünleşik bir arayüz ve protokoldür. Teknoloji, 3 cm veya daha az bir işlem aralığı sınırı ile 13.56 MHz üzerinde, lisansa sahip olmayan Endüstriye Özel Medikal (ISM) radyo bandı üzerinde çalışmaktadır. Bu teknoloji aynı zamanda mobil cihazlara,

¹ alliedmarketresearch.com/mobile-wallet-market

² nearfieldcommunication.org/history-nfc.html

herhangi bir ağ yapılandırmasına gerek olmadan güvenli iletişim kurma yolu sunar. Bu da iletişim kurmak için eşleştirme yapmayı gerektiren Bluetooth teknolojisine göre bir avantajdır.

NFC, elektromanyetik endüksiyon ile çalışır: bir bobindeki elektromanyetik dalgalar değiştiği zaman, bir gerilim indükle edilir. Bir donanım bakış açısından bakıldığında, yüksek ve düşük gerilim serisi bitleri temsil etmektedir. Veriler cihazlar arasında böyle gönderilir. 2006 yılında, NFC ekosistemini yöneten NFC Forumu, NFC Özelliklerini belirlemiştir. Forum, verileri içeren önemsiz bir ikili (binary) mesaj formatı olan NFC Veri Alışveriş Formatı (NDEF) için bir standart oluşturmuştur. Örneğin, bir URL'yi kodlamak için URL'yi takip eden 5 baytlık bir başlık bulunur; kısa bir URL, 12 bayta kadar bir alanı kullanabilir. Daha fazla bilgi için bakınız: online³.

NFC'nin Bluetooth karşısındaki avantajları⁴:

NFC, Bluetooth ile karşılaştırıldığında şu avantajları sunar:

- 1. Güvenlik:** Burada NFC'nin kısa mesafeli çalışması bir avantajdır, iletişimlerin yakalanmasını zorlaştırır. Buna ilave olarak, bazı akıllı telefon uygulamaları için ekranın aktif olması ve NFC donanımına erişime izin vermeden önce bir PIN girilmesi gerekir.
- 2. Düşük güç sarfiyatı:** İşlemin kısa mesafeli olması nedeniyle, NFC etiketlerinin okunması veya yazılması için sadece zayıf bir elektromanyetik alanın oluşturulması gerekir.
- 3. Hızlı eşleştirme:** NFC cihazları saniyenin onda biri kadar bir süre içinde bağlantı kurabilir.

³ [www.developer.nokia.com/Community/Wiki/Understanding_NFC_Data_Exchange_Format_\(NDEF\)_messages](http://www.developer.nokia.com/Community/Wiki/Understanding_NFC_Data_Exchange_Format_(NDEF)_messages) and www.nfc-forum.org/specs/

⁴ nearfieldcommunication.org/bluetooth.html

NFC çalışma modları⁵

NFC'de üç çalışma modu mevcuttur. Bunlar: Okuyucu / Yazıcı modu, Uç birimden uç birime modu ve Kart Emülasyon modu. Bu bölüm, bu modları ve kullanımlarını açıklar.

Okuyucu/Yazıcı modu

Bu modda, akıllı telefon NFC etiketinden verileri okuyabilir veya bu etikete veri yazabilir, telefonun NFC çipi NFC etiketleri ile etkileşim için bir elektromanyetik alan oluşturur. NFC etiketleri salt okunur olabilir. Bu nedenle, sadece yazma işlemi devreye sokulmuş bir NFC etiketi üzerine bir NDEF mesajı yazmak mümkündür.

Basit okuma uygulamaları genel veya uygulamaya özgü veriler taşıyan etiketler içerebilir. Bunun bir örneği, bir tedarikçinin ürününe veya hizmetin internet sitesine bağlantı sunan bir reklam posterinde yer alan NFC etiketinin bulunduğu akıllı posterlerdir. Bu durumda, etiket verilerinin özel bir formatı vardır ve telefon örneğin web tarayıcıda bir URL'i nasıl açacağını bilir. Uygulamaya özgü kullanımlar sadece hayal gücünüzle (ve NFC etiketinin muhafaza edebileceği veri miktarıyla) sınırlıdır. Okuma / Yazma uygulamaları uygulama tarafından okunmak için mesajların gönderilmesi veya etiketlerin yazılması için kullanılabilir.

Uç birimden uç birime modu

RFID teknolojisine önemli bir uzantıdır. P2P modunda iki akıllı telefon, vCards, URL gibi küçük miktarda bilgilerin alışı verişini yapabilir veya büyük veri aktarımları için bir Bluetooth bağlantısını başlatabilir. Android Beam teknolojisi genellikle

⁵ nfc.cc/technology/nfc

P2P modunda çalışır ve bu modda NFC ile kullanıcı kimliği belli edilmeden eşleşme gerçekleşir ve veri aktarımları daha hızlı olan Bluetooth bağlantısı üzerinden yapılır.

Uygulamalar bu modu kullanarak NFC destekli iki akıllı telefon arasında veri alışverişi yapabilir. Pek çok akıllı telefonda, Bluetooth eşleştirme ve veri alışverişi, kartvizitler gibi görevler için uygulamalar yer alır. Uygulamalarınızda bu modu örneğin bir oyundaki hareketler ile ilgili bilgi alışverişi için kullanabilirsiniz.

Kart Emülasyon Modu

Bu modda akıllı telefon, perakendecilerin EFTPOS terminali tarafından okunabilen pasif bir NFC etiketi gibi davranır. Bu modda önemli olan şey hassas bilgilerin depolandığı "secure element"dir (güvenli eleman – bunun hakkında daha fazla bilgi metnin ilerisinde verilecektir). Google Wallet ve Microsoft Wallet, NFC ile donatılmış bir akıllı telefonun dokun-öde için kullanılmasına ve kart emülasyonuna güvenmesine olanak sağlayan uygulamalara örnektir.

Genellikle kişiselleştirilen ve ödeme yöntemi veya hizmet sağlayıcıyla bir ilişki gerektiren bu mod, bilet hazırlama, ödeme, operasyonlar arasında geçiş (kapı açmak gibi) için kullanılabilir ve fiziksel kartların (örneğin; sağlık sigortası kartları, kredi kartları, ehliyet vb.) yerini alır.



Secure Element (Güvenli Eleman)

Kart emülyasyonlu cihazların temelinde secure element (SE)⁶ yatar. Bu da güvenli veriler içerir (kredi kartı bilgisi ve {diğerleri | smartcardalliance.org}). SE'nin muhafaza edilebileceği üç yer vardır:

1. SIM/UICC (SIM kartı ile NFC çipi arasında bağlantı kurmaya olanak sağlayan bir özellik olan Tek Kablolü Protokol aracılığı ile).
2. Telefonun NFC Çipi.
3. SD kartı.

Ek not olarak Google Wallet⁷, bir Google ön ödemeli kredi kartının kimlik bilgilerini sadece telefonun NFC çipindeki secure element'da muhafaza eder. Mevcut kredi kartı numaraları güvenli Google sunucularında tutulurken, sadece Google kredi kartı numaraları satıcılara aktarılır. Diğer yandan, Microsoft Wallet, hassas öğeleri SIM kartın secure element'inde muhafaza eder. Microsoft'a göre bu tür bir yöntem insanların cüzdanlarını bir telefondan bir diğer telefona aktarmalarına olanak sağlar.

NFC teknolojisini kullanan ödeme yöntemlerini yaygınlaştırmak için işbirliği yapan kredi kuruluşlarının ve Mobil Ağ Operatörlerinin (MNO'lar) sayısı dünya genelinde giderek artıyor. MasterCard PayPass ve VISA PayWave bu çözümlere örnek olarak verilebilir. NFC SIM kartları şu anda A1 (Avusturya), Orange France ve China Mobile gibi MNO'lar tarafından yayınlanıyor.

⁶ mp-nfc.org/nfc_near_field_communication_architecture.html

⁷ google.com/wallet/faq.html

Mevcut zorluklar

NFC, beraberinde daha ekonomik işlemler getirecek ilgi çeken bir teknoloji. Ancak NFC'yi kullanan ödeme yöntemlerinin geniş bir şekilde kullanılmasını görmemizden önce bütün bankalar, donanım imalatçıları, MNO'lar ve işletim sistemi geliştiricileri arasında tam bir anlayışın ve işbirliğinin oluşması gerekli olacaktır.

Secure element'in sınırlı depolama alanı vardır. Bu alanın oyuncular arasında nasıl paylaşıldığı pek net değildir. Son olarak, kabul edilmiş standartların eksikliği nedeniyle, bazı bankalar kendi çözümlerini uygulamıştır ve yeni ödeme modlarını kabul etmesi için satıcıları ikna etmek isterler.

NFC API'leri

Android: API Seviye 9'dan (Gingerbread 2.3) itibaren Android, NFC'nin kullanımını sorunsuz ve kolay hale getiren bir dizi yüksek düzeyli API sunuyor. Daha fazla bilgi için bakınız: Android geliştirici sayfası⁸.

BlackBerry: En son geliştirilen SDK'sı ile NFC etkileşimi için yüksek düzeyde API sunuyor. Bir NFC etiket okuyucu ve yazıcı uygulamak için örnek kod⁹ ve BlackBerry'de kart emülasyon modun nasıl kullanılır¹⁰ hakkında daha fazla bilgi BlackBerry web sitesinde bulunabilir.

Windows Phone 8: Windows Phone 8.0 SDK, Windows Phone 8 uygulamaları arasında P2P veri paylaşımını devreye sokmak için gerekli API'ları sunan bir sınıf setini içeren

⁸ developer.android.com/reference/android/nfc/package-summary.html

⁹ docs.blackberry.com/en/developers/deliverables/34480/Near_Field_Communication_1631111_11.jsp

¹⁰ supportforums.blackberry.com/t5/Java-Development/NFC-Card-Emulation-Primer/ta-p/1596893

Proximity paketlerini içermektedir. Bunun yanı sıra bir Android cihazındaki veri paketlerini bir WP8 cihazına aktarmak ve ters yönde aktarım yapmak da mümkündür. Ancak uygulama hala olgunlaşma sürecindedir ve büyük miktarda veri aktarımı mümkün değildir.

Uygulama hakkında daha fazla ayrıntı, online¹¹'de bulunabilir.



¹¹ msdn.microsoft.com/en-us/library/windowsphone/develop/jj207060



Dokunsal Titreřim (Haptic Vibration) Özelliđi

Dokunsal Titreřim Tasarımında Dikkate Alınacak Hususlar

Uygulamanızda neden dokunsal titreřim (Haptic Vibration) etkileřimini kullanmalısınız? Uygulamanız zaten dokunsal geribildirim olmadan da gayet düzgün alıřıyor, deđil mi? Evet, ancak sanal ortamınızı daha gereki ve ekici bir hale getiren duyuşsal bir unsuru kaybediyor olacaksınız. Margaret Atwood, "Dokunma görmekten, konuřmadan önce gelir" der ve sözlerine řöyle devam eder: "Dokunma, ilk ve son dildir ve her zaman dođruyu söyler." Görmek ve duymaktan daha derin olan bu dokunsal geribildirim hissi, gerek dünyadaki etkileřimlerimizden neler beklediđimizi bize öđretmektedir.

Kullanıcıların uygulamalardaki sanal dünyadan beklentilerini belirleyen řey, bizim gerek dünyadaki deneyimlerimizdir.

Örnek olarak bir düđmeye basmayı ele alalım. Gerek bir düđme basma dokunmayla ilgili bir deneyimdir. Bir bařlangıcı ve bitiři vardır. Yaptıđınız eylemin tamamlanmasına iliřkin olarak bir memnuniyet hissedersiniz. Buna kıyasla bir sanal düđme, aynı eylem onayını taklit etmek için temassal bir etki olmadan yetersiz görülebilir. Bunun yanı sıra, dokunma duyusuna dayalı bir onay olmadan kullanıcıyı, süreç aısından daha zorlu olan görsel/sesssel iřaretlere zorunlu hale getirirsiniz; bu da bir dokunma hissine kıyasla daha zor olur.

Dokunsal geri bildirim mobil video oyunlarında daha önemlidir. Bunu, konsol oyunlarından edindiđimiz deneyimden biliyoruz. "DualShock" Rumble pad motoru olmadan piyasaya

sürülen Sony PS3'ü hatırlayın. Oyun oynayanlar, durumdan memnun olmadıklarını belirtmişler ve Sony hemen ardından DualShock Rumble'ı piyasaya çıkarmıştı. Aynı dokunsal geri bildirim memnuniyeti mobil oyunlara uygulanır. Oyunlarınızda dokunsal etkiler kullanmanız mobil kullanıcılarınıza konsol platformlarından beklentilerini sunmanıza yardımcı olacaktır. İyi tasarım yaparsanız, oyunlarınız kullanıcılara daha gerçekçi ve inandırıcı görünür.

Dokunsal bir deneyim tasarlarken, kullanıcının en iyi deneyimini aklınızda tutun. Uygulamaya başlamadan önce planlama için biraz zaman harcayın. Proje belirlendikten ve zihninizde şekil aldıktan sonra, aşağıdaki ilkeleri göz önünde bulundurun:

- Basit duyular genellikle en etkili olanlardır. Çat Pat gibi çok basit bir hissin bile menü etkileşimini arttırması ve uygulama içerisinde kullanıcının kendine güven duymasını görmek bile şaşırtıcı olabilir.
- Sadece basit bir düğme tıklaması gibi sesli ve görsel olaylarla eşzamanlı duygular, bütünü, parçaların toplamından daha iyi hale getirir. Bir nesneyi veya faaliyeti görmek, duymak ve hissetmek, duysal uyumu sadece görmek veya duymaktan daha fazla etkiler.
- Kullanıcıyı uğraştırmak iyi değildir. Zayıf seçilmiş veya tasarlanmış dokunma duyuları can sıkıcı olabilir veya üretkenliği olumsuz etkileyebilir. Yüksek tondaki bir vızıltı, bir uyarı parçası olarak etkili olabilir ancak sürekli olarak tekrarlanan bir vızılama, uygulamayı kullanan kişi için sinir bozucu olabilir.
- Kullanıcının kafasını karıştırmak ve canını sıkmak iyi bir şey değildir. Tıpkı aynı anda çalan çok sayıda güzel sesin bir karma oluşturması gibi, aynı anda veya birbiriyle çok yakın zamanda devreye giren birden fazla dokunsal his kafa karıştırıcı ve bunaltıcı olabilir.

- Aşınalık kullanıcı deneyimini kolaylaştırır. Dokunsal efektler, görüntü veya ses yoluyla sunulması mümkün veya uygun olmayan önemli bilgileri kullanıcıya iletebilir. Standartlaşma ve uyum önemlidir. Dokunsal efekt dilinin yönetilebilir, yeniden kullanılabilen bir takım duyum ile sınırlamak, kullanıcının öğrenme sürecini kolaylaştıracaktır çünkü algılanması gereken daha az sayıda dokunsal etki mevcut olacaktır.

Neredeyse tüm mobil platformlar bir takım dokunsal titreme geribildirim kontrolü kullanımına olanak sağlamaktadır. Bu bölüm, bu platformlar arasındaki sınıfları ve yöntemleri anlamanız için kaynağınız olacaktır.

Android



Android, titreşim kontrolü için benzersizdir. Native destek verir ve iOS'dan daha fazla titreşim kontrolüne sahiptir. Ayrıca, Android'in bu titreşim kontrolünü geliştiricilere genişletmenin pek çok farklı yolu vardır. Böylece, konsola daha çok benzeyen X-Box veya PlayStation geri bildirim deneyimini oluşturabilirler.

Aşağıdaki temel veya geniş yöntemlerden hangisini kullanırsanız kullanın, daha iyi bir erişim için bir kullanıcının dokunsal efektleri etkin hale getirmiş olabileceğini lütfen unutmayın. Örneğin, KickBack Erişilebilirlik Hizmeti dokunsal geri bildirim sunar ve eyes-free¹ açık kaynak projesinin bir parçası olarak mevcuttur. Bu nedenle, uygulamanız tarafından yaratılmış olan dokunsal efektlerin bu tür hizmetlerle nasıl etkileşime girebileceğini veya bunları engelleyebileceğini göz önünde bulundurun.

¹ code.google.com/p/eyes-free

Android'de temel titreşim kontrolü için, uygulamanızın titreşim yapmasına olanak sağlamak amacıyla ilk önce `android.permission.VIBRATE`'e izin vermelisiniz. Daha sonra, titreşim hizmetini çağırmak için `getSystemService` fonksiyonlu ve `Context.Vibrator_Service`'lı `Vibrator Class`²'yı kullanırsınız.

Yukardaki yöntem içerisinde, gönderilen titreşimin süresini milisaniyeler bazında ayarlayabilir ve istediğiniz kadar başlangıç ve uyku olayı düzenleyerek titreşim modelleri oluşturabilirsiniz. Temel Android titreşim kontrol yöntemi sadece titreşim etkinliklerinin süresini kontrol etmenize olanak sağlar.

Genişletilmiş Android Titreşim Kontrolü

Android platformu açık kaynaklı olduğu için, Android'in titreşim kontrolünü genişletmek için ücretsiz yöntemler sunan en azından bir düzine şirket mevcut. Immersion Corporation'ın Haptic SDK'sı³, 124 adet önceden belirlenmiş Dokunsal titreşim geribildirim efekt kütüphanesiyle süre, seviye ve darbe frekansının tam olarak kontrol edilmesine olanak sağlar. Bu tipte bir kontrol ile uygulama geliştiricileri, pil gücünü korurken konsol oyun titreşim deneyimine rakip olan titreşim efektlerini yaratma kabiliyetine sahip olurlar.

Unity3D, Marmalade veya YoYo Games'in GameMaker Studio'sunu kullanan Android geliştiricileri için Immersion, eklenti desteği vasıtasıyla benzer genişletilmiş yöntemi sunmaktadır ve bu, kendi ana SDK web sayfalarında mevcuttur. Bu genişletilmiş titreşim kontrolüne ilgi duyan geliştiriciler, Eclipse ortamına kurulumunu ve önceden belirlenmiş bir kütüphaneden dokunsal efektleri çağırmak için `Launcher`

² developer.android.com/index.html?q=Vibrator

³ www.immersion.com/haptic/sdk

yöntemini kullanmayı açıklayan Hızlı Kullanım Kılavuzu'nu⁴ indirebilirler. Google Play üzerinde ayrıca, birkaç adet ücretsiz, önceden tasarlanmış Dokunsal efekti, kod bazında kullanmadan önce indirip deneyebilirsiniz. Uygulamalardan bir tanesinin adı "Haptic Muse", diğeri ise "Haptic Effect Preview"⁵dur.

Önceden tasarlanan efekt kütüphanesinin bir diğeri avantajı ise, mobil cihazlar arasında farklılıkları dengeleyen bir donanım ayırma katmanıdır; böylece, yarattığınız his sürekli olur.

BlackBerry 10

BlackBerry, size Android'in sunduğu ile aynı temel titreşim açma/kapama kontrolünü, genişletilmiş yöntem olmadan sunar. BlackBerry için, `startVibrate(int duration)` ve `stopVibrate(int duration)` bulunan `VibrationController Class`⁵ kullanılır.

Buna ilave olarak, BlackBerry'nin geliştiriciler için bir `intensity (yoğunluk)` (1-100) parametresi vardır.

Firefox OS

Mozilla Firefox Mobile, milisaniyelik süre parametrelerine sahip bir `window.navigator.vibrate` function call⁶ biçiminde tekli veya modelli titreşim dizilişlerine olanak tanıyan bir web view API'ı kullanmaktadır. Modelli titreşim dizilişleri, değişken titreme kapama ve açma özelliklerini kullanır. Örneğin, modelli diziliş `window.navigator.vibrate()`, titreşimi 1 saniye boyunca çalacak 200 milisaniye bekleyecek ve titreşimi 2 saniye daha devreye sokacaktır. Sıfır değerli süre veya boş

⁴ www.immersion.com/haptic/guide

⁵ developer.blackberry.com/search/?search=VibrationController

⁶ developer.mozilla.org/en-US/docs/WebAPI/Vibration



diziliş titreşim olayları, halihazırda çalışan herhangi bir titreşimi iptal edebilir.

iOS

iOS platformu geliştiricilere mobil cihazlarına ilişkin az miktarda titreşim kontrolü sunar. Aşağıdaki iOS titreşim kontrol yöntemi sadece iPhone'lara uygulanır. iPad'ler ve iPod'lar şu anda cihaz üzerinde titreşim oynatmasını desteklemez.

iPhone cihazınızda titreşimi tetiklemek için `AudioServicesPlaySystemSound` fonksiyonuna ve `kSystemSoundID_Vibrate` sabitine sahip `SysSoundViewController Class`⁷ kullanın. Bu sabiti aramak, 400 milisaneye boyunca motorunuzu çalıştıracaktır. Buna ilave olarak, yine 400 milisaniyelik bir titreşimi tetikleyecek, bir `kSystemSoundID_Vibrate` sabitini aramak için `AudioServicesPlaySystemSound` method⁸ kullanan bir iOS Audio Toolbox çerçevesi bulunmaktadır.

İlave iOS titreşim becerisi, Apple artık iOS platformu için bir "iPhone/iPad/iPod için üretilmiştir" (MFi) adlı çevresel mimariyi kullandığı için Logitech (PowerShell) ve BDA (Moga Ace Power) gibi çevresel mobil kontrolör üreticileri tarafından sunulabilir.

PhoneGap

PhoneGap, Adobe Systems'in sahip olduğu web görünümlü bir API'dir (daha fazla bilgi edinmek için çapraz platform başlığına bakın). PhoneGap titreşim yöntemi, mili saniye parametrelerine

⁷ developer.apple.com/search/index.php?q=SysSoundViewController

⁸ developer.apple.com/search/index.php?q=AudioServicesPlaySystemSound

sahip `navigator.notification.vibrate` function call⁹'ı kullanır. PhoneGap çok sayıda işletim sisteminde çalıştığı için, titreşim izinlerini her platform göre ayarladığınızdan emin olmanız gerekecek.

Buna ilave olarak, iOS'da PhoneGap süre parametresi göz ardı edilir ve iOS sabitini kullanarak titreşir.

Tizen

Samsung ve Intel'in desteklediği yeni işletim sistemi Tizen, iki temel geliştirme ortamına sahiptir¹⁰; bunlardan bir tanesi native uygulama geliştirmesi için, diğeri de WC3 Vibration API'ını kullanan web uygulama geliştirmesi içindir.

- Native uygulama geliştiricileri titreşim kontrolü için `Vibrator Class`¹¹ başlatma ve durdurma yöntemlerini kullanır. Geliştirici, model dizilişinin `start()` bağımsız değişkeni olarak geçirildiği değerleri ayarlayarak yoğunluk ve darbeyi belirleyebilir. Model dizilişindeki her bir eleman iki alana sahiptir. Süre alanı, titreşimin milisaniye olarak ne kadar süreyle oynatılacağını belirtir; yoğunluk alanı ise titreşimin ne kadar güçlü olacağını ayarlar. Yoğunluk değerleri 1-100'dür (minimum - maksimum), "Sistem Varsayılanı" için -1 ve "Sessiz" için 0. Eğer birden fazla dokunsal efekt oluşturmak istiyorsanız, ilave elemanları, model dizilişin ilerisine eklersiniz. `stop()` metodu sadece bir etkiyi erkenden sonlandırmak istemeniz durumunda gereklidir (Örn, bir telefon çağrısı geldiği için).

⁹ docs.phonegap.com/en/2.1.0/cordova_notification_notification.md.html

¹⁰ developer.tizen.org/documentation/dev-guide

¹¹ developer.tizen.org/help/topic/org.tizen.native.appprogramming/html/guide/system/vibrator_mgmt.htm

- Web uygulaması geliştiricileri `navigator.vibrate()` function call.¹²'yı kullanır. Yöntemin iki değişkeni vardır: milisaniye bazında tek bir süreli bağımsız değişkeni kullanan `singleVibration(int duration)`, ve bir süre dizilimini ele alan. İkinci işlev, darbe etkilerinin kontrolüne olanak sağlar `patternVibration(int duration on, int duration off)`, ancak doğal ortamda bulunan yoğunluk kontrolü için destek yoktur.

Web/HTML5

HTML5 titreşim kontrolü, Tizen gibi, web uygulamasının geliştirilmesi için WC3 Titreşim API'sının kullanılmasına bağlıdır. Aynı işlev çağrıları uygulanır.

Windows 8

Başlat & Durdur yöntemleri ile birlikte `VibrateController` Class¹³'ü kullanın. Daha hassas süre kontrolü için, milisaniye değerlerini kullanmak amacıyla `TimeSpan` metodunu ayarlamanız gerekecektir.

Yukarıda listelenen Windows 8 Class önceki Windows 7 Class ile aynıdır.

¹² developer.tizen.org/help/topic/org.tizen.web.appprogramming/html/tutorials/w3c_tutorial/device_tutorial/managing_vibration.htm

¹³ social.msdn.microsoft.com/search/en-us?query=VibrateController



Arttırılmış Gerçekliğin Uygulanması

Arttırılmış Gerçeklik (AR) son birkaç yıldır hakkında çok fazla bilgi okuduğunuz ve duyduğunuz bir şey. Her yerde karşımıza çıkıyor: AR oyunları, AR katalogları, AR posterleri, AR dövmeleri, araçlarda AR, AR uygulamaları, AR reklamları ve daha fazlası...

Ünlü pazar araştırma şirketi Research and Markets'a göre, AR piyasası katlanarak büyüyor: 2020 yılına kadar AR uygulamalarını kullanan tüketicilerin sayısı bir milyara ulaşacak. Geçen seneki TED Konferansı'nda Tomi Ahonen, AR'nin 8. kitle iletişim aracı olduğunu belirtti ve akıllı camların yaygınlaşması ile birlikte AR uygulamalarına olan talebin önemli miktarda artacağını ifade etti. Bu görüş, bu teknoloji ile ilgili gelirlerin 2016 yılına kadar 600 milyar dolara ulaşacağını tahmin edildiği 'Augmented Reality: Envision a More Intelligent World'¹ Semico raporu gibi çeşitli şirketlerin çalışmaları ile desteklenir. Buna ilave olarak 'Research and Markets', AR uygulamalarının 2016 itibarıyla 5 milyar 155 milyon dolar değere ulaşacağını ve bunun %95.35'lik bir yıllık büyüme oranı olacağını söylüyor.²

AR tam olarak nedir? Sanal elemanlar ekleyerek gerçek dünyayı geliştiren bir teknolojidir. Görsel arttırılmış gerçeklik en yaygın şeklidir, genellikle kullanıcının gerçek hayattaki görünümüne eklenen 3 boyutlu nesneleri veya 2 boyutlu kaplamalara sahip metin veya görüntüleri içerir.

1 www.semico.com/content/augmented-reality-envision-more-intelligent-world

2 www.researchandmarkets.com/reports/1963197

Mobil İçerikte AR Senaryoları

Mobil Arttırılmış Gerçeklik, ilave bilginin hareket halindeyken verimliliği, etkinliği ve kullanım eğlencesini arttırabileceği durumlarda kullanılır. Mobil AR özellikle insanların çok sayıda veri ve kısa sürede bunları işleme durumu ile karşı karşıya olduğu süreçler için uygundur. Bilgiyi, mobil cihazın ekranı vasıtasıyla görünür olan canlı akışa entegre ederek, kullanıcının dikkatinin artık mobil cihaz ve ortam arasında bölünmesi gerekmemektedir. Mobil Arttırılmış Gerçeklik çözümlerinin çok sayıda uygulaması vardır (örneğin; işletmelerde, pazarlamada, eğitimde veya eğlence). İşte bazı örnekler:

- Layar³ ve Wikitude⁴ gibi Arttırılmış Gerçeklik Tarayıcıları, gerçek dünyanın canlı görünümü üzerinde yer tabanlı verileri üst üste koyar. Kullanıcının konumu GPS tarafından belirlenir ve yakındaki ilgi alanları (POI'ler) hakkındaki bilgiler akıllı telefonun ekranında görüntülenir. Wikitude ayrıca, daha fazla bilgi edinmek için Wikipedia'ya bir bağlantı sunar. Blippar⁵, reklamları yaşama dahil etmek için AR kullanan bir diğer örnektir.
- IKEA Katalog Uygulaması

Kullanıcılar, IKEA AR Uygulamasını kullanarak, cep telefonları ile IKEA kataloğundan özel olarak işaretlenmiş içerikleri tarar, kişiye özel seçenekler ve ilave ürün fotoğrafları gibi ilave ürün bilgilerini görüntüler. Uygulama, daha yaygın olan QR kodlarının kullanılmasından ziyade Metaio⁶ görüntü

³ layar.com/products/app/

⁴ wikitude.com/app/

⁵ blippar.com

⁶ metaio.com

tanıma yazılımına bağlıdır. Uygulama aynı zamanda kullanıcının mobilyayı sanal olarak evine veya bürosuna yerleştirmesine olanak sağlar.

— AR Jump n' Run

AR Jump n' Run, DroidAR⁷ ile birlikte Android akıllı telefonları için geliştirilmiş konum tabanlı bir uygulamadır. Oyun, GPS, basamak tanıma veya her ikisini de kullanarak hem iç mekanlarda hem dış mekanlarda oynanabilir. Cihaz bir görüntü kapısı gibi hareket ederken oyuncu, sanal olarak zenginleştirilmiş bir dünya içerisinde yürür ve oyun deneyimi kazanır. Yardımcı 3D nesneleri toplamaya ve zararlı olanlardan kaçınmaya çalışırlar. Oyun ayrıca, oyuncuların yeni 3D nesneler ekleyerek haritalar oluşturmalarına veya haritaları düzenlemesine olanak sağlayan oyun içi bir harita editörü içeriyor.

— Ingress

"Etrafınızdaki dünya görüldüğü gibi değildir. Geleceğimiz değişme açıktır ve bir tarafı seçmelisiniz". Bu söz, Google'ın dâhili bir kurulum şirketi olan Niantic Labs tarafından geliştirilen Ingress⁸ oyununa ait. Oyunun tabanında, tüm dünyaya yerleştirilmiş portallar bulunur. Kullanıcılar bir takıma katılır ve görevleri de karşı takımın portallarını yok edip kendi portallarını korumaktır.

— Audi eKurzinfo

Metaio tarafından kurulan Audi, A3 modeli aracı için iOS için bir arttırılmış gerçeklik uygulaması⁹ şeklinde bir kılavuz sunuyor. Kılavuzda, araçta yer alan ve tümü telefon kamerası

⁷ code.google.com/p/droidar/

⁸ www.ingress.com

⁹ youtube.com/watch?v=TDTW0lbWBXI

ile algılaması kolay, 300'den fazla farklı öge yer alıyor. Örneğin, aracın içinde bir uyarı simgesinin belirmesi durumunda kullanıcı, sorunla nasıl baş edeceğini anlamak için uygulama ile bu simgeyi tarayabilir.

AR Developing 101

Bu bölüm AR uygulamalarını oluşturmak için gerekli olan önemli kavramlara genel bir giriş sunar. Bu kavramları anladığınızda, projeniz için doğru yapıyı seçebilmelisiniz.

Gerçek ve Sanal Dünya

Artırılmış Gerçeklik teknolojileri, sanal bir katman kullanarak gerçek dünyaya yapay nesneler yerleştirmenizi sağlar. Bu sanal katman veya sanal dünya ve onun koordinat sistemi, referans noktalarıyla gerçek dünyaya bağlıdır. Bu referans bir GPS konumu, bir görsel işaretçi veya bir görüntü olabilir. Örneğin, belirli bir GPS konumuna sanal bir nesne yerleştirebilirsiniz. Sonra, nesne bu koordinatlara bağlanır. Görsel işaretler veya resimler ile arttırmayı gerçek dünyadaki bir nesne ile bağlarsınız. Örneğin, görsel referans noktası olarak mısır gevreği kutusu olduğu sürece, bir mısır gevreği kutusuna basılmış bir oyun dünyanın her yerinde oynanabilir.

Referans noktaları takip edilerek belirlenir. Yaygın olarak kullanılan takip teknolojilerinde GPS, optik sensörler, pusula, ivmeölçer, yönölçer (jiroskop) ve adımölçer (pedometre) bulunur. Diğer konseptler işaret bazlı, işaretli ve karma izlemedir. İşaretçiler, nesneleri belirlemenin basit, ucuz ve doğru bir yoldur. Görüntü işleme sistemi, işaretçinin görüntüsünü ve bir görüntünün boyutunu işleyerek cihaz ile işaretçi arasındaki mesafeyi hesaplar. Diğer taraftan işaretçiler olmadan yapılan izlemede, işaretçiler yerine doğal özellikler kullanılır. Bu özellikler iki boyutlu nesneler (Örn: reklam

posterleri) veya etraftaki üç boyutlu cisimler (Örn; binalar) olabilir. Bir eşleşme algılamak için kaydedilen görüntüler veri tabanı ile karşılaştırılır. Bunun için karmaşık bir algoritma ve yüksek işleme gücü gerekir. Karma izleme teknolojisinde GPS, 3D özellik algılama, işaretçi algılama ve basamak algılama gibi farklı konum kaynakları birleştirilir. Bu durum, daha yüksek düzeyde konumlandırmaya ve hassas hareket algılamaya olanak sağlar.

İki Dünya Arasında Eşleştirme

Konum tabanlı AR uygulamaları için, gerçek dünyanın sürekli değişen pozisyonu ve sanal dünyanın pozisyonu arasındaki karşılıklı eşleşmeye ihtiyaç duyulur. OpenGL gibi oluşturma motorları, bu süreci karmaşıklığını azaltır ve gerçek zamanlı hassasiyeti ve hızı artırır. Bu motor aynı zamanda kameranın sanal ve gerçek dünya konumunun eşleştirilmesine dikkat eder ve bu konumlar değiştiğinde akıcı kamera geçişlerini garantiler. Geliştirici olarak yaşamınızı kolaylaştırmak için [gameplay¹⁰](http://www.gameplay3d.org/) veya [Unity¹¹](http://unity3d.com) gibi genişletilmiş motorlar kullanabilirsiniz.

Temel öge olan kamera verileri de diğer bileşenlere aktarılabilir. Bu durum örneğin bir çıkışma bileşeninin sanal nesneler ile kamera tarafından yakalanan görüntü arasındaki mesafenin kolayca hesaplanabileceği anlamına geliyor.

Sanal Elemanların Oluşturulması

Sanal öğeler, 3D veya 2D nesneler olarak temsil edilir ve çeşitli davranışlar sergileyebilir: Bazı nesneler toplanabilir olabilir, diğerleri kullanıcıyı takip edebilir veya sabit durup herhangi bir etkileşime izin vermeyebilir. Metaio Creator, Layar Creator veya Wikitude Studio gibi yazılımlarla kendi AR içeriğinizi

¹⁰ www.gameplay3d.org/

¹¹ unity3d.com

kolayca oluřturmak artık mmkn. Bir web uygulaması ile hedef grnt zerinde bir arttırmayı yerleřtirebilirsiniz. Arttırmalarınızı seip yerleřtirdikten sonra projenizi bir AR uygulaması olarak ıkartabilirsiniz. Bu uygulama ile arttırmalar, hedef grnt uygulama tarafından algılandığında grntlenecektir.

Metaio, Wikitude ve Layar'ın sunduėu aralar yalnızca grntlerin arttırılmasını saėlamaktadır. Eėer bařka Arttırılmış Gereklik senaryoları oluřturmak isterseniz, Eřzamanlı Konumlandırma ve Haritalama (Simultaneous Localization And Mapping - SLAM) adı verilen bir tekniėe ihtiyaınız var. Bu teknik, bir grnt ierisinde benzersiz bir řekilde tanımlanabilen alanlara dayalı olan zelliklere ve bu zelliklere baėlı olan ok gl bir Computer Vision tekniėidir. 3 boyutlu zellikler gerek dnyaya, cihazın hareket etme yntemi ile baėlanır. Bunlar bir btn olarak 3 boyutlu bir bulut oluřtururlar. Bu 3 boyutlu bulut, gerek dnyadaki nesne algılaması veya boyut algılamasının yanı sıra, nesnenin arttırılması iin de kullanılabilir. Computer Vision'ın bir bařka parası da grnt tanımlanmasıdır. Bu tanımlanan grnt pozisyonu, sanal dnya iin bir referans pozisyonu olarak kullanılabilir (mısıır gevreėi kutusu rneėi ile karřılařtırın). SLAM ve Computer Vision tekniėi, gerek zamanlı alıřma iin muazzam hesaplama gcne ihtiya duyar. Donanımın sınırlandırmaları ve becerileri hesaba katılmalıdır.

Uygulama Tabakalarının Birleřtirilmesi

Grsel AR'ın nemli bir elemanı, kameradan alınan grntnn paraları zerine bir řeyler yerleřtirebilmesidir. Uygulamanızın ihtiyalarına baėlı olarak 2D veya 3D grafikler yerleřtirmek ve ilgili API'leri kullanmak isteyeceksiniz. 2 boyutlu bir katman basit POI tarayıcıları iin genellikle yeterlidir.

Tekerleėi yeniden icat etmek yerine OpenGL gibi bir grnt oluřturma atısı kullanmanızı řiddetle tavsiye ediyoruz. Bu tr

yapılar kullanıcının konumunu, cihazın yönünü ve diğer sensor bilgilerini veya görüntü analiz verilerini kullanır ve içeriğinizi uygun şekilde görüntülemek amacıyla bunu çevirir. Eğer görüntü oluşturma bileşeni, uygulama kodunun geri kalanından ayrışır, gelecekte bu değiştirilebilir; örnek olarak, daha gelişmiş görüntü oluşturma çözümlerine geçiş yapılabilir.

Sanal nesnelerin konumunu görüntülemek için küçük bir radar UI'si veya sanal dünya ile etkileşime girmek için bazit bazı butonlar gibi şeyler eklemek isteyebilirsiniz. Platforma özel numuneler ve tasarımlar yapıştırdığınızdan emin olun. Ayrıca esnekliği sürdürmek için bu öğeleri ayrı bir tabakaya uygulayın.

Çok Tabakalı Sanal Bir Dünya Oluşturmak

Sanal dünya oluşturma en iyi yolu, bir ağaç yapısı kullanmak ve sanal nesneleri farklı tabakalara yerleştirmektir: Güncellenmesine ihtiyaç duyulmayan ve kullanıcı ile etkileşime geçmeyen sabit “arkaplan” nesneleri için bir katman ve hareket edebilir nesneler veya UI elemanları için diğer katmanlar.

Sadece kullanıcıya yakın olan nesneleri güncelleyip işlemeli ve dörtlü ağaç yapısından faydalanmalısınız. Bir dörtlü ağaç, uygulamanızın bir karakter kutusu içerisinde bütün nesneleri verimli bir şekilde saklamanıza yardımcı olan bir veri yapısıdır. En iyi uygulama performansının alınmasını sağlamak amacıyla



cihaz donanımına bağılı olarak farklı bir görüntü yarı çapı kullanılabilir.

Güncellemeleri tetikleyen ve örneğin her 20 ms’de bir güncelleme metodu çağıran bir güncelleme mekanizması kullanılmasını tavsiye ediyoruz. Nesne ağacındaki devreler, bu güncelleme aramalarının hangi alt kısımlara iletileceğini belirler. Örnek olarak bir dörtlü ağaç, güncelleme prosedürünü verimli hale getirmek açısından yalnızca kullanıcıya yakın olan nesneleri güncelleyecektir. Basit bir liste yapısı bütün alt kısımları güncelleyecektir ve sanal bir pozisyona sahip olmayan (mantıksal oyun istatistikleri gibi) veya beraber güncellemeleri gerekmeyen nesneler için daha uygundur. Tam bir nesne bileşimi konsepti, uygulama senaryosuna bağılıdır ve evrensel bir şekilde tanımlanamaz.

Arttırılmış Gerçeklik SDK’leri

ARLab

ARLab, farklı takip yöntemleri için ayrı ayrı ADK’ler sunar. Bunlardan bir tanesi 2 boyutlu konum ve Layar ve Wikitude’un ilk sürümlerine benzer POI’ler gibi sensör bazlı içeriğim yanı sıra görüntü eşleştirme, görüntü izleme, nesne izleme, sanal düğmeler yaratma ve bir 3 boyutlu motor SDK’sı yaratma amacıyla atanır. SDK lisanslarının iOS ve Android için ayrı ayrı satın alınması gerektiği ve her SDK’nın 99€ ile 299€ arasında değişen fiyatları olduğu unutulmamalıdır.

www.arlab.com

ARPA SDK

ARPA SDK, görüntülerin, işaretçilerin ve GPS’in izlenmesini sağlar ve yüz takibi için de bir modül sunar. İnteraktif AR senaryolarının mümkün olması için bir Unity Extension’ı sunar.

Ücretler hakkında mevcut somut bilgi yoktur ve SDK kullanan tüm uygulamalar ayrı ayrı lisanslanmalıdır. Belirtilmesi gereken bir diğer önemli engelse, sadece Android 4.0 veya daha yüksek sürümlerin bulunduğu Android cihazlarında çalışıyor olmasıdır. Şimdiye kadar masaüstü ve iPad için şirketin kendisi tarafından oluşturulmuş AR Uygulamalarıyla ilgili iyi performanslar gördük.
www.arpa-solutions.net/ARPA_SDK

D'Fusion

Total Immersion'ın D'Fusion SDK'sı, görüntüleri arttırmak için 2 boyutlu özütleme özelliğini kullanır ve metaio, Qualcomm ve Layar gibi diğer SDK'lara epey benzer. Bazı SDK paketlerinde yüz izleme ve hareket algılama kütüphaneleri mevcuttur. Diğer pek çok SDK karşısındaki bir avantajı da ücretsiz kullanıma izin veren bir lisansının da olmasıdır.
www.t-immersion.com

DroidAR

DroidAR SDK, konum tabanlı ve işaretçi tabanlı interaktif AR Android uygulamalarının geliştirilmesi için oluşturulmuştur. SLAM ve 3 boyutlu nesne algılaması ve yeniden inşası için 3 boyutlu pozisyon izlemesini mümkün kılar. İzleme için geniş iç mekân senaryolarının üstesinden gelebilmek amacıyla konum bazlı izleme, işaret izlemesi ve adım algılama algoritması kombinasyonunu kullanır. DroidAR v1, tamamen uzatılabilen ve açık kaynaklı AR SDK'larından bir tanesidir. DroidAR, yaklaşık 6.000 kişilik geliştirici topluluğu tarafından desteklenir ve SHIO 2013 uygulaması gibi çeşitli uygulamalarda kullanılmaktadır.
github.com/bitstars/droidar

Layar

Başlangıçta Layar, kullanıcı için çok sınırlı etkileşim fırsatları sunan, görünür olan katmanlara sahip, tamamen konum bazlı bir AR platformuydu. Artık Layar, görüntüleri arttırmak amacıyla 2 boyutlu özellik özütlemesine artık odaklanmamaktadır.

Yeni Vision SDK ve Layar oluşturucusu, baskı medya içeriğini uzatmak için tasarlanmıştır. iOS için Layar Player SDK, Layar tarayıcısına ihtiyaç duymayan Layar uygulamalarını oluşturmayı mümkün kılar. Vision SDK lisansı yıllık 2.500 Euro, Geo SDK lisansı ise yıllık 7.500 Euro'dur. SDK'lar filigranlı olarak gelir, bu filigran yıllık 7.500 Euro ekstra ücretle kaldırılabilir.

www.layar.com

metaio

metaio GmbH, Arttırılmış Gerçeklik piyasasında ilk taşıyıcı olmuştur. Metaio SDK'larının ana odak noktası, örneğin, dergiler veya kataloglarda 2D görüntülerin arttırılmasıdır. SDK'nin pro sürümü aynı zamanda bir ürün ambalajı, bir heykel veya bina cephesi gibi 3D nesnelerin tanınmasını destekler. Metaio SDK ile birlikte güçlü bir 3D oluşturma motoru standart olarak sunulur. Kullanıcıya sürekli olarak bir metaio filigranı görüntüleyerek SDK'yi ücretsiz kullanmak mümkündür.

Filigransız temel sürüm 2.950€'dur ve filigransız pro sürüm 4,950€'dur. SDK, 50.000 defadan fazla indirilmiş ve 1.000'in üzerinde uygulamada kullanılmıştır.

www.metaio.com

PointCloud SDK

PointCloud SDK'sı, masaüstü oyunlar ve diğer küçük alanlı senaryolar için küçük çapta SLAM izlemesi işlemine odaklanmıştır. Unity eklentisi beta olarak mevcuttur. SDK halen erken aşamadır ancak gelecek vadetmektedir. SDK ücretsiz kullanılabilir, ardından PointCloud logosu uygulamada daimi olarak görüntülenmelidir. Eğer filigranı çıkarmak isterseniz,

daha fazla ayrıntı için 13. Lab ile irtibat kurabilirsiniz. iOS sürümü PointCloud internet sitesinden indirilebilir ve talep üzerine Android alfabaya da erişilebilir.
developer.pointcloud.io/sdk

Vuforia

Qualcomm Vuforia SDK ücretsiz kullanılabilir ve Unity uzantısı bulunmaktadır; bu yüzden görsel algılama ve gerçeklik arttırma senaryoları için iyi bir seçenektir. Sadece 2D görüntü algılamasını destekler ve yerel olarak depolanan özel görüntüleri tanır. Birden fazla 2 boyutlu işaretçi tek bir işaretçi içerisinde oluşturulabilir ancak 3 boyutlu nesneler henüz algılanamamaktadır. Ayrıca hareket verilerini veya coğrafi yer tanımlamalarını kullanan konum bazlı uygulamaları oluşturmak için kullanılamamaktadır. 80.000 tescilli geliştirici mevcuttur ve Vuforia SDK, yaklaşık 100.000 defa indirilmiştir. Vuforia SDK ile çok sayıda büyük ticari proje oluşturulmuştur.
www.vuforia.com

Wikitude SDK

Wikitude, konum tabanlı bir İlgi Alanı (POI) tarayıcısıdır. Wikitude'un yeterli bir çözüm sunduğu klasik bir kullanım örneği, POI aramasıdır (Örneğin; "En yakın postane nerede?" sorusuna cevap vermek). Wikitude, statik içerik için tasarlanmıştır ve interaktif senaryoları devreye sokmaz. Eğer kendi Wikitude tarayıcı uygulamanızı oluşturmak için daha yeni sürümler kullanırsanız, HTML5, Titanium veya PhoneGap'ı kullanabilirsiniz. BlackBerry 10 desteklenir. Wikitude 32 dilde mevcuttur ve dünya genelinde 13 milyondan fazla kullanıcısı vardır. Wikitude AR tarayıcısının yanında, görüntüleri arttırmaya yardımcı bir araç olan Wikitude Studio da sunulmaktadır. Bu, Metaio Creator ve Layar Creator'e benzer şekilde çalışır.
www.wikitude.com/developer



Güvenlik

Bu rehberi okuyanlar akıllı mobil cihazların ne kadar yaygın olduğunu ve mobil uygulamaların ne kadar faydalı olabileceğini öğreniyor. Mobil cihazlar, kişisel bilgisayarlardan daha kişiseldir. İnsanlar telefonları ile uyanıyor, tüm gün boyunca telefonlarını kendilerine yakın tutuyor ve tüm gece telefonları yanlarında uyuyor. Zamanla telefonların bizim güvenilir 'ortaklarımız' haline geliyor.

Uygulamaların çoğu yakınlık ve güvenilirlik avantajlarından faydalanyor. Örneğin, telefonunuz banka hesabınıza erişimin doğrulanması için kullanılabilir veya tabletiniz, satın aldığınız çevirim içi filmlere doğrudan erişim sunabiliyor. Cihaz, Near Field Communications (NFC) veya Bitcoin gibi sanal para ile yapılan ödemeler için gerçek para cüzdanı görevini bile görebiliyor. Mobil uygulamalar korsanların ve 99 sente satılan uygulamaların ötesine ilgi duyan hırsızların dikkatini çekiyor. Kaspersky Lab, Haziran 2013'de Android uygulamalarında, 629 malware ailesinden oluşan 100,000 kötü amaçlı kod numunesi sayıldığını bildirmiştir¹. Eskimiş ağ ve uç nokta (endpoint) bazlı savunma araçları (anti-virüs araçları gibi) artık yeterli değil. Mobil uygulamaya güvenliğin yerleştirilmesinin önemi artıyor.

Mobil uygulamaların mimarisi gelişmeye devam ediyor. Bazı uygulamalar sadece native ve her farklı mobil işletim sistemi için tamamen farklı kod tabanlarıyla yazılması gerekiyor. Bazıları bir uygulama simgesi içerisine yerleştirilmiş bir web sitesi URL'sinden biraz fazlası olan WebView tabanlı çalışıyor. Kalanlar da native app işlevlerinin WebView ile birleştiği "hibrit" uygulamalar. Birçok mobil uygulama, bilgi almak veya

1 securelist.com/en/analysis/204792299/IT_Threat_Evolution_Q2_2013#16

güncellemek için arkayüz (backend) hizmetlerine bağlanmak zorundadır. Mobil uygulamalar ile birlikte web uygulamaları gibi klasik uygulama güvenlik ihtiyaçlarının da kullanılması gerekir. Girişlerin, izin verilen boyut, tür ve değere göre değerlendirilmesi gerekir. Hata kullanımının, hassas bilgileri sızdırmadan yararlı hata mesajları sunularak gerçekleştirilmesi gereklidir. Tanımlama, kimlik doğrulama ve kimlik doğrulama kontrollerinin atlanmadığından emin olmak amacıyla uygulamaların sızma testlerinin gerçekleştirilmesi gerekir. Hassas verilerin ve şifreleme anahtarlarının düz metin olarak depolanmasını engellemek amacıyla cihazlardaki depolama alanlarının incelenmesi ve test edilmesi gerekir. Günlük (log) dosyaları, parolaları veya diğer hassas bilgileri içermemelidir. SSL yapılandırmalarının da test edilmesi gerekir.

Kullanıcılar uygulamalarınızı güvenli şekilde kullanmak istiyor; hoşla gitmeyen sürprizler istemiyor. Mobil telefonlar, kullanıcıları, olası GPS üzerinden konum izlenmesi gibi çeşitli saldırılara maruz bırakabilir. Kamera ve mikrofon, gizli tutmayı tercih ettikleri bilgileri vs. yakalamak için kullanılabilir. Uygulamalar, iletişim bilgileri gibi hassas bilgilere erişmek amacıyla da yazılabilir. Kötü amaçlı uygulamalar, pahalı numaralara gizli telefon çağrıları yapabilir ve SMS mesajları gönderebilir.

Uygulama geliştirici saygınlığını, gelir kaybı ve fikri mülkiyet hakkı kaybı gibi konularda tehlikeye düşürebilir. Kuruluşlar, kullanıcıların muhtemelen uygulamanızı kullanarak, mobil cihazlarından erişebileceği işle ilgili verileri korumak ister. Kullanıcı ne yüklerse yüklesin, verileri gizli tutulabilir mi?

Uygulamanızı Tehdit Eden Unsurlar

Bazı platformlarda (özellikle iOS ve Android), dâhili imza kontrollerini devreden çıkartmak, yaygın olarak yapılan bir işlemdir. Birisinin kodunuzu değiştirip, bunu "jailbreak" yapılmış bir telefonda çalıştırmasının sizin için önemli olup

olmadığını düşünmeniz gereklidir. Açık bir sorun, lisans kontrolünün kaldırılması ve bu sebepten ötürü uygulamanızın çalınması ve ücretsiz olarak kullanılması olacaktır. Bu kadar belirgin olmayan ancak daha ciddi bir sorun ise, kullanıcılarınızın verilerini çalabilen veya uygulamanız içerisine yasadışı bir içerik yerleştirip markanızın itibarını zedeleyebilen kötü amaçlı bir kodun (malware) yerleştirilmesi olacaktır.

Uygulamanız üzerinde ters mühendislik işlemi yapılması, bir bilgisayar korsanına, DRM korumalı filmler için kriptolu anahtarlar, çevrimiçi oyun sunucusu ile konuşmanız için protokoller gibi hassas içeriğe erişim fırsatı veya mobil ödeme sisteminde saklanan kredilere erişimde bulunma fırsatı verebilir. Bu tehditlerden herhangi bir tanesini açığa çıkarmak için yalnızca bir bilgisayar korsanı ve jailbreak işlemi uygulanmış tek bir iPhone telefon yeterlidir.

Uygulamanızda gerçek para veya değerli içerik bulunuyorsa, Man-At-The-End (MATE) saldırılarına karşı korumak için gerekli her türlü adımı atmanız gerekir. Sayısal Hak Yönetimi (DRM) teknolojilerini uyguluyorsanız, kendi kendine korumayı zorunlu hale getiren sağlamlık kurallarına uymanız gerekecek.

Uygulamanızın Korunması

Kod Haritasının Gizlenmesi

Bazı mobil platformlar, doğrudan CPU üzerinde çalıştırılmak yerine sanal bir makine tarafından çalıştırılan bayt kodu kullanılarak (Java veya .NET) programlanır. Bu platformlar için ikili (binary) biçimler, sınıf hiyerarşisini içeren ve her bir sınıfın ismini ve tipini, değişkenini, metodunu ve parametresini sunan metadata'ları içermektedir. Metadata, sanal makinenin dil özelliklerinden bazılarını (Örneğin, Reflection özelliği) uygulamasına yardımcı olur. Ancak metadata, kod üzerinde

ters mühendislik işlemi yapmak isteyen bir bilgisayar korsanı için çok yararlıdır. Ücretsiz olarak mevcut olan ters derleme programları, bayt kodundan kaynak kodunu oluşturmakta ve ters mühendislik işlemini kolaylaştırmaktadır.

Android platformu, C'de yazılan ve native kod olarak derlenen fonksiyonlara erişim sağlamak amacıyla Java Native Interface'i (JNI) kullanma seçeneğine sahiptir. Native kod üzerinde ters mühendislik işlemi yapılması, Java'ya kıyasla daha zordur ve güvenliğin en fazla öneme sahip olduğu durumlarda bu yöntem tavsiye edilmektedir.

"gcc" normalde Android için native kod oluşturmak amacıyla kullanılan bir derleyicidir, ikiz kardeşi "clang" ise iOS için kullanılır. Bu derleyiciler için varsayılan ayar, her bir işlevin paylaşılan bir nesneden çıkartılması ve ikili değer içerisindeki dinamik sembol tablosuna eklenmesidir. Dinamik sembol tablosu, hata ayıklama için kullanılan sembol tablosundan farklıdır ve bunun çıkartılması derlemeden sonra çok zor olmaktadır. Dinamik sembollerin boşaltılması, bir bilgisayar korsanına native kod üzerinde çok yardımcı bir işlev endeksi sunar. `{-f visibility}` derleyici parametresinin² doğru şekilde kullanılması, kodun korsan tarafından anlaşılmasını zorlaştırmanın kolay yoludur.

Derlenen Objective-C kodu, makine kodu ve saldıran kişiye isimler ve uygulamanın arama yapısı hakkında birçok bilgiyi içeren tonlarca metadatayı içermektedir. Şu anda bu metadatayı okuyabilecek ve bilgisayar korsanlarına kılavuzluk yapabilecek birçok araç ve komut dizisi bulunmaktadır. Ancak bunları saklayabilecek herhangi bir araç mevcut değildir. iOS için bir GUI oluşturmanın en yaygın yöntemi Objective-C kullanmaktan geçmektedir. Yine de en güvenli yol, bunun kullanımını en

² <http://gcc.gnu.org/wiki/Visibility>

düşük seviyeye getirmek ve GUI ötesindeki bütün unsurlar için düz C veya C++ kullanmaktır.

Kontrol Akışının Gizlenmesi

Tüm isimler gizlenmiş olmasına rağmen, iyi bir korsan, yazılımın nasıl çalıştığını yine de çözebilir. Ticari olarak yönetilen kod koruması araçları, operasyonların yeniden kodlanması ve talimat bloklarını ayırma yöntemleri ile bu yolu daha da karmaşık hale (obfuscate) getirir; bu da kaynak koda dönüştürme işlemini çok daha zor hale getirir.

İyi bir koruma aracı kullanıldığında, korunan bir ikili (binary) kodu geri derleme (de-compile) girişimi, geri derleyicinin çökmesi veya geçersiz kaynak kodu ile sonuçlanacaktır.

Native kodun kaynak koda dönüştürülmesi daha zordur ama hâlâ yapılabilir. Bir araç olmadan bile derleyici tarafından oluşturulan "assembler" kodu içerisindeki kontrol akışını takip etmek gayet kolaydır. Güçlü bir güvenlik gereksinimine sahip uygulamalar, native kodun yanı sıra yönetilen kod için bir karıştırma (obfuscate) aracına ihtiyaç duyacaklardır.

Ağ İletişimlerinin Korunması

Ağ iletişimleri hassas olabilir; özellikle, ağ analizcilerinin ücretsiz bir şekilde mevcut olduğu ve ağ trafiğini izleyip kesiştirdikleri durumlarda uygulamalar emülatörlere veya simülatörlere yüklenebildikleri zaman. Örneğin, uygulamanız ile sunucular arasında HTTP için SSL kullanarak, hassas ağ iletişimlerinin korunmasına dikkat edin. SSL hareket halindeki veriyi korur ancak bu işlem yalnızca uygulamadan, SSL oturumunun sonlandığı web sunucusu üzerinde gerçekleşir. Özellikle WiFi ağları üzerinden yapılan MATE saldırıları bile hassas verileri ifşa edebilir. Taşıma güvenliğinin güçlendirmenin bir yolu da, uçtan uca güvenlik sağlamak amacıyla sunucu ile mobil uygulama arasında asimetrik bir şifreleme kullanmaktır

(kamuya açık/özel anahtar çiftleri kullanarak). Hassas kurumsal veriler ve uygulamalar için mobil cihazlara Virtual Private Network (VPN) sunucuları ve VPN istemcilerinin kurulumunu yapmalısınız. VPN'ler genellikle güçlü kimlik doğrulama ve SSL üzerinde ve ötesinde güvenli bir taşıma sağlarlar.

Kurcalamaya Karşı Koruma

Kod tabanınızı uygulamayı kurcalamaya çalışan unsurları aktif bir şekilde algılayarak ve bu saldırılara cevap vererek daha iyi koruyabilirsiniz. Kriptografi kodu daima standart ve kısmen güvenli şifre algoritmalarını kullanmalıdır (örnek AES, RSA, ECC), Ancak bir saldırgan, ikili (binary) sisteminiz içerisinde veya çalıştırma kodu içerisinde şifre anahtarlarını bulduğunda ne olur? Bu durum, saldırganın değerli bir unsuru içeren bir kapıyı açması ile sonuçlanabilir. Ortak anahtar kriptografisini kullansanız ve anahtar çiftinin yalnızca yarısı bile açık olsa da saldırganın bir anahtarı değiştirdiği ve diğer yarısına sahip olduğu senaryolar karşısında nelerin olabileceğini düşünmeniz gerekir. Kodunuz kurcalandığında bunu algılamak için bir tekniğe ihtiyacınız var. Kodu şifrelemek/şifresini çözmek, kurcalamayı algılamak amacıyla kod üzerinde denetim toplamaları gerçekleştirmek ve kod değiştirildiğinden cevap vermek için mevcut araçlar bulunmaktadır.

Mobil uygulama ve arkayüz (backend) hizmetleri arasındaki iletişimler gözlemlenebilir ve haklanabilir. SSL kullanırken bile SSL trafiğini denetleyebilecek olan bir web vekili (proxy - Paros gibi), bir WiFi bağlantısı üzerinde kurulabilir. Saldırganlar daha sonra taşınan veriyi para kazanmak veya yalnızca eğlenmek için kurcalayabilirler. Bu nedenle eğer HTTPS aracılığı ile gerçekten hassas veriler gönderiliyorsa, mobil uygulama ve sunucu arasında bir şifreleme/şifreyi çözme uygulaması koymayı düşünün ki böylece ağ dinleyicilerin (sniffer) yalnızca şifreli veriyi görmelerini sağlayın.

Kriptografik Algoritmaların Korunması

Kurcalama engelleyici aktif bir araç, kripto anahtarları üzerinde yapılan bazı saldırıları algılamaya veya engellemeye yardımcı olabilir ancak anahtarları kalıcı bir şekilde gizleyemeyecektir. Beyaz kutu kriptografisi, anahtarların gizli kalmasını sağlayan bir standart şifre algoritmasını uygulamayı amaçlamaktadır. Beyaz kutu kriptografisinin bazı sürümleri, ters mühendislik işleminin zor olacağı ve benzer nümerik değerlerin elde edildiği bazı karmaşık matematiksel yaklaşımları kullanmaktadır. Diğer kriptografiler anahtarları, ters mühendislik işleminin zor olduğu arama tablolarına ve sonlu otomatlara gömmektedirler. Eğer DRM kodu yazacaksanız veya yüksek güvenliğe sahip veri depolamasına ihtiyacınız varsa, beyaz kutu kriptografisine kesinlikle ihtiyaç duyacaksınız.

En İyi Uygulama Alışkanlıkları

Sırları veya Gizli Bilgileri Depolamayın

Cihazda depolanan hassas bilgi miktarını en aza indirin. Güvenli depolama, karmaşık bir şifre tarafından koruma altında bulunmuyorsa, referansları veya şifre anahtarlarını saklamayın. Bunun yerine sınırlı yaşam süresi ve işleve sahip olan kimlik doğrulama jetonlarını saklayın.

Günlük (log) dosyaları, sistem hatalarının teşhis edilmesi ve uygulamaların kullanımının takibi için faydalıdır. Ancak kullanıcıların konum bilgilerini depolayarak veya kişisel bilgilerini kaydederek onların gizliliğini ihlal etmemeye dikkat edin. Bazı ülkelerde, toplanabilen bilgilerin izlenmesini engelleyen kanunlar vardır – bu nedenle uygulamanızın kullanılacağı ülkedeki kanunları kontrol ettiğinizden emin olun.



Bilgisayar korsanlarının daha da derin seviyelere inmek için kullanabilecekleri yığın (stack) izleme veya sistem tanılamalarını yazdırmayın.

Cihaza Güvenmeyin

Bir uygulama tasarımı yaparken, cihazın uygulamayı kötüye kullanmayı deneyen bir saldırgana ait olabileceğini varsayın. Mobil uygulamaları yaratırken, arkayüz (backend) hizmetleri için kullanacağınız benzer güvenli yazılım geliştirme yaşam döngüsünü uygulayın. Mobil uygulamalarınız için oluşturduğunuz veri tabanlarına bile güvenmeyin – bir korsan buranın düzenini de değiştirebilir. İşletim sisteminizin koruma sağlamasına güvenmeyin – birçok OS koruması, cihaz üzerinde jailbreak işlemi yapılarak (Android'de bu işleme "root olmak" denir) önemsiz bir şekilde devreden çıkartılabilir. Native anahtar saklayıcılarının verileri gizli tutacaklarına inanmayın – kullanıcı cihazı uzun ve karmaşık bir şifre ile korumadığı sürece bazı anahtar saklayıcılar kaba kuvvet tahmini ile kırılabilir.

İzinlerin En Aza İndirilmesi

Uygulamanın GPS gibi sensörlere ve hassas içeriğe erişimine izin veren çeşitli izinler vardır; Android'de bu izin konseptidir, iOS ise yetki vermeyi kullanır. Android'de bu izinlerin Android-Manifest.xml dosyasında uygulama oluşturmanın bir parçası olarak belirtilmesi gerekir. Kullanıcı, uygulamayı cihazına kurmayı seçtiğinde bunlar kullanıcıya sunulur.

Her bir izin, uygulamanızın çirkin şeyler yapma olasılığını artırır ve hatta bazı kullanıcıların uygulamanızı kullanmaktan vazgeçmelerine bile neden olabilir. Bu nedenle uygulamanızın ihtiyacı olan izinlerin ve özelliklerin sayısını en aza indirmeyi amaçlayın.

Araçlar

Koruma

Temel Java kodunuzdaki isimleri değiştirme işlemi Proguard³ ile ve açık kaynaklı bir araç olan Arxan's GuardIT⁴ aracılığıyla yapılabilir.

Yönetilen kod (Java ve .NET) koruma araçları olarak Arxan Technologies⁵ ve PreEmptive Solutions⁶ isimli iki satıcının aracı mevcuttur.

Native kod koruma araçları ve beyaz kutu kriptografi kütüphaneleri için temel satıcılar Arxan ve Irdeto'dur⁷.

Android kodunu onaysız değişikliğe karşı koruma tedbirleri androidcracking.blogspot.com/ adresinde belgelendirilmiştir. Arxan'ın EnsureIT'si, hata ayıklayıcıları algılayacak olan ekstra kodu, inşa (build) aşamasında yerleştirmenize, hafızadaki kod üzerindeki değişiklikleri görebilmek için denetim toplamalarını kullanmanıza ve kodun, kullanım aşamasında şifresinin çözülmesine veya tamir edilmesine fırsat tanır.

Dinleme (Sniffing)

Paros⁸ standart, ücretsiz bir web proxy aracıdır. Ortak platformda mevcut olan standart bir ağ dinleme aracı da Wireshark⁹'tır.

³ www.proguard.sourceforge.net

⁴ arxan.com

⁵ arxan.com

⁶ preemptive.com

⁷ www.irdeto.com

⁸ sourceforge.net/projects/paros

⁹ sourceforge.net/projects/wireshark

Geri Derleme (De-Compiling)
Bakınız: Hex Rays de-compiler¹⁰.

Daha Fazlasını Öğrenin

İşte size yardımcı olabilecek bazı faydalı kaynaklar ve referanslar:

- Apple yazılım güvenliği¹¹ konusunda genel bir rehber sunuyor. Bu rehberde aynı zamanda platformlar için daha ayrıntılı konulara çok sayıda bağlantı yer alır.
- iOS ve Android için ticari eğitim kursları mevcuttur. Lancelot Institute¹² iOS ve Android'i kapsayan güvenli kodlama kursları sunmaktadır.
- O'Reilly'nin yayınladığı Jeff Six'in Android güvenliği hakkındaki Application Security For The Android Platform¹³ adlı kitabı ve iOS için Jonathan Zdziarski tarafından hazırlanan Hacking and Securing iOS Applications¹⁴ kitabı.
- Charlie Miller et al. tarafından yayınlanan iOS Hackers Handbook¹⁵ kitabı, iOS cihazlarından kod ve veri çalmanın ne kadar kolay olduğunu gösteriyor.
- Akademik araştırmacılar USENIX 2011¹⁶ adresinden, halka açık Android uygulamalarından ne kadar bilginin derlenebileceğini gösterir.

¹⁰ www.hex-rays.com

¹¹ developer.apple.com/library/mac/navigation/#section=Topics&topic=Security

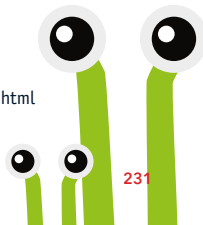
¹² www.lancelotinstitute.com

¹³ shop.oreilly.com/product/0636920022596.do

¹⁴ shop.oreilly.com/product/0636920023234.do

¹⁵ www.wiley.com/WileyCDA/WileyTitle/productCd-1118204123.html

¹⁶ static.usenix.org/event/sec11/tech/slides/enck.pdf



- Qualsys Labs¹⁷ tarafından ücretsiz bir SSL tester sunulur.
- OWASP¹⁸ tarafından kapsamlı ücretsiz uygulama güvenlik rehberliği ve test araçları sunulur. Buna OWASP Mobile Security Project¹⁹ dahildir.
- AT&T'nin Uygulama Kaynaklarını İyileştirme aracı²⁰, Android için açık kaynaklı bir mobil uygulama performans izleme aracı sağlar.

Sonuç

Mobil uygulamalar zamanla daha güvenilir hale geliyor ancak bu güveni, kendileri için kullanmak isteyen birçok kişi karşısında çoğu zaman korunmasız bir şekilde bürünüyorlar. Uygun düzeyde uygulama güvenliği, her uygulama için dikkat edilmesi gereken bir şeydir. Sonuçta uygulamanız yabani ortam içerisinde tek başına kalacaktır ve nereye giderse gitsin, kendisini bilgisayar korsanlarına ve diğer tehditlere karşı koruması gerekecektir.

Hedef almak istediğiniz güvenlik özellikleri ve becerileri hakkında bilgi edinmek için zaman ayırın. Uygulamanıza ilişkin olası tehditleri belirlemek adına tehdit modellemesi gibi teknikleri kullanın. Kod incelemelerini gerçekleştirin ve gerekli olmayan kayıt ve hata giderme yöntemlerini çıkartın. Bir bilgisayar korsanının kodunuzu nasıl analiz edeceğini düşünün ve daha sonra, güvenli ve emniyetli bir ortam içerisinde açıklıkları ortaya çıkarmak için benzer teknikleri kullanın ve uygulamanızı piyasaya sürmeden önce bu açıkları ortadan kaldırın.

¹⁷ www.ssllabs.com/ssltest

¹⁸ www.owasp.org

¹⁹ www.owasp.org/index.php/OWASP_Mobile_Security_Project

²⁰ developer.att.com



Test Etme

Uygulamanızı yaratmak için o kadar çalıştıktan sonra dünyaya sunmadan önce test etmeye ne dersiniz? Test, bir engel gibi görülebilir ancak uygulamanızdaki hataların herkesin gözü önünde ortaya çıkması sizin için acı verici olabilir. Uygulamanızın App Store'daki puanı, çok düşük seviyedeyken itibarınızı toparlamak çok zordur. Mobil uygulamaların etkili şekilde test edilmesi karmaşık ve zorlu olabilir. Bunun için otomatik ve interaktif testleri cihazlarda birleştirmeniz gerekir. Neyse ki Android ve iOS da dahil olmak üzere, birçok büyük mobil geliştirme platformu çekirdek araçları içerisinde test otomasyon araçlarını içermektedir. Popüler platformlar için çapraz platform test otomasyon araçları mevcuttur; bunların bazıları ücretsiz ve açık kaynaklıdır bazılarıysa ticaridir.

Sürekli entegrasyon (Continuous delivery) için sürekli test etmeniz gerekir. Tutarlı uygulamaların üretim ortamındayken süregelen bir şekilde güncel kalması gerekir. Güncellemeler yeni platform sürümleri veya cihaz modelleri için düzeltmeleri, yeni işlevsellikleri ve diğer iyileştirmeleri içerebilir. Bu yüzden test etmek bir kereye mahsus bir işlem değildir; yüksek kaliteye sahip uygulamalar sürekli olarak optimize bir şekilde test edilmektedir. Üretim (production) testi, test sürecini, uygulamanız hızla büyümeden önce ortaya çıkabilecek olası sorunların erkenden tespit edilmesini ve testin doğrulanması aşamalarını içerir.

Bölümün bu kısmında test üzerine genel konulara değindik. Bölümün bundan sonraki kısmında platforma özel testi işleyeceğiz.

Test Edilebilirlik: En Büyük Kazanım

Eğer uygulamanızı etkili ve verimli bir şekilde test etmek için yollar bulmak istiyorsanız, test etme yolları tasarlamaya ve bunları uygulamaya başlayın; bu durum özellikle otomatik testte uygulanır. Örneğin, kodunuz içerisinde Dependency Injection gibi teknikleri kullanmak, gerçek sunucuları (yavaş ve tuhaf) sahte (mock) sunucularla (kontrol edilebilir ve hızlı) değiştirmenize olanak verir. Önemli UI öğeleri için benzeri olmayan, net tanımlayıcılar kullanın. Eğer tanımlayıcılarını değiştirmeden muhafaza ederseniz, testleriniz için daha az bakım gerekir.

Kodunuzu test edilebilir modüllere ayırın. Yıllar önce mobil cihazlar ve yazılım araçları çok sınırlıyken, geliştiriciler kendi mobil kodlarını tek parçalı kod parçalarına 'optimize' etmeyi seçtiler ancak mevcut cihazlar ve mobil platformlar, bu 'optimizasyon' biçiminin gereksiz olduğunu ve hatta büyük olasılıkla zarar verici olduğunu göstermiştir.

Mümkünse, özel bir hata ayıklama arayüzü üzerinden, uygulamanın durumunu sorgulamak için yöntemler sunun. Siz veya sizin testçileriniz aksi takdirde, uygulama istediğiniz gibi çalışmadığında bunun nedenini bulmak için oldukça fazla zaman harcayabilirsiniz.

Test GÜdümlü Geliştirme

Yazılım tasarlamının ve uygulamanın çok çeşitli yolları vardır. Test GÜdümlü Geliştirme (Test-Driven Development - TDD) geliştiricilerin uygulama için ana kod yazmayla paralel olarak otomatik testler yazdığı bir yaklaşımdır. Otomatik testlere birim testleri dahil olacaktır, bunlara bir sonraki konuya değinilmiştir.

TDD hem bir zihniyet hem de bir uygulamadır. Süreç zorlaşsa bile, testleri yazabilmek için belli bir disiplin seviyesi gerektirmektedir. Ve TDD'yi özenle uygulayan geliştiricilerin daha iyi kalitede, daha basit, açık kodlar yazmaları muhtemeldir; bu da gelecekte daha iyi bir kontrolü mümkün kılar (uygulamanın kaynak kodunu korurken ve revize ederken çalıştırılabilen otomatik testler tarafından korunur ve desteklenirler).

TDD, sade bir yaklaşım olarak yeni uygulama kodunun yazılmasından önce testlerin yazıldığı ve çalıştırıldığı bir yöntemdir. Yeni testlerin başarısız olması beklenir; yani uygulamanın davranışı hakkındaki arızalı kısımlar rapor etmelidirler. Hatalar uygulamanın ne yapması gerektiği ile şu anda yaptığı şey arasındaki uyumsuzluğu ifade eder. Artık uygulamanın kaynak kodu üzerindeki değişikliklerini test etmek için geliştiricinin otomatikleştirilmiş bir yöntemi bulunmaktadır. Bütün testleri geçmek için yeterli sayıda yazılım yazıldıktan sonra, bu testlerin gereksinimlerini karşılayan bir uygulamaya güveniniz tam olacaktır.

İş gereksinimlerini karşılamış olsanız da, işinizin henüz 'bitmediğini' düşünebilirsiniz. Örnek olarak, uyarlama içerisinde birden fazla, gereksiz karmaşıklık ve bilinen diğer kusurlar olabilir. Bu sebepten ötürü 'refactoring' denilen bir süreç vasıtasıyla kaynak kodu revize edip iyileştirme fırsatına sahipsiniz. Refactoring, otomatik testlerin, geliştirilmiş kod karşısında çalıştırıldığında testleri tamamladıkları durumlarda, geliştiricilerin uyarlamayı (implementasyonu) iyileştirdikleri bir durumdur.

Mevcut Mobil Test Otomasyon araçlarını kullanırken TDD bir zorluk olarak görünse de, Graham Lee'nin kitabı Test-Driven iOS Development¹da görüldüğü gibi çeşitli kişiler TDD'nin başarılı bir şekilde kullanılmasına ilişkin olarak farklı örnekler

1 informit.com/store/product.aspx?isbn=0321774183

sunmuşlardır. Ayrıca uygulamanın jenerik yönleri açısından TDD'yi kullanmayı düşünebilirsiniz.

Birim (Unit) Testi

Birim testi, genellikle kaynak kodun yalnızca birkaç satırı olmak üzere, küçük miktarda kodu test eden otomatik testlerin yazılmasını içerir. Genel olarak bunlar, bu bireysel küçük parçalı kodların nasıl tepki göstereceğini bilecekleri için uygulamanın kaynak kodunu yazan geliştiriciler tarafından yazılmalıdır.

Birim testleri, yazılım geliştirilmesinde uzun bir kökene sahiptir. JUnit²'in, neredeyse mobil uygulama geliştirmek için kullanılan bütün programlama dilleri için benzer geliştirme çatıları vardır.

Birim testleri, otomatik testlerin yalnızca bir unsurudur; uygulamanın nasıl çalıştığını göstermek adına tek başlarına yeterli değildir. Geliştiricilerin, yazılımın hangi bireysel parçalarının nasıl çalışacaklarını görmelerine yardımcı olur. Diğer otomatik test türleri de dahil olmak üzere ilave testlerin yapılması, uygulamaya duyduğunuz güvenin artmasına yardımcı olacaktır.

Etkili Test Uygulamaları

Kullanıcılarınız tarafından bulunmadan önce sorunları bulmanız için etkili test yapmanız gerekir. Daha büyük ekiplerde uzman 'testçiler' bulunabilir. Etkili test içerisinde cihazları, platformları ve benzer uygulamaların davranışlarını bilmek vardır. Cihazları yapılandırmak için, örneğin çok dilli uygulamaları test etmek için etkili yollar bulmanız gerekir. Kullanıcılarınızın uygulamanızı kullanması olası olduğu olduğu

² en.wikipedia.org/wiki/JUnit

gerçekçi koşullarda test yapmanız gerekir. Testdroid'in, doğru test deneyimini ekibinize sunmanız için iyi bir kontrol listesi³ bulunmaktadır.

Aşağıdaki çeşitli konular, başlangıçta size yardımcı olabilir.

Etkileşimli Test Etme

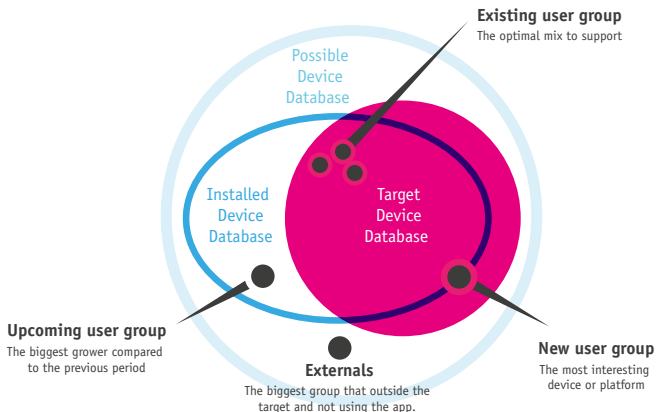
İşyeriniz gibi sabit bir konumda, çeşitli cihazları test ederken gözden kaçacak hataları bulmak adına çeşitlilik ve hareket size yardımcı olabilir. Kullanıcılarınızdan uygulamanızı veya benzer uygulamaları nasıl kullandıklarını öğrenin. Ardından uygulamaları ve cihazları kullanma şekillerini taklit eden testler düzenleyin.

Test senaryolarını (test case) düzenlerken appquality-alliance.org/resources adresindeki ilkelere dikkat edilmelidir. Örneğin, telefona bir arama geldiğinde ve kullanıcılar telefonu 'uçuş moduna' aldığı anda neler olduğunu görmek için uygulamanın test edilmesini içerir.

Bundan sonraki birkaç bölüm, etkileşimli testlere üç farklı yaklaşım sunacaktır.

³ testdroid.com/testdroid/6336/get-the-superb-expertise-in-your-testingqa-team





- **Fiziksel cihazlar:** gerçek telefonlarla test yapmak önemli olduğu için.
- **Uzaktan kontrol:** fiziksel olarak elinizde olmayan telefonları kullanarak test yapabilmenin bir yoludur, binlerce kilometre uzakta ve hatta başka bir kıtada bile olabilir.
- **Kitle kaynaklı test etme:** diğer test ediciler sizin adınıza test gerçekleştirir.

Fiziksel ve Sanal Cihazlar

Fiziksel cihazlar gerçektir, onları elinizde tutabilirsiniz. Sanal cihazlar, bir başka bilgisayarın içinde, yazılım olarak çalışır. Her ikisi de mobil uygulamaların test edilmesi için faydalı bir yer sağlar.

Sanal cihazlar genellikle ücretsizdir ve hemen kurulum ve kullanım için uygundur. Android dahil bazı platformlar, kişiye

özel cihazlar (örneğin, yeni bir ekran çözünürlüğüne sahip) oluşturmaya olanak sağlar. Bunları uygun bir donanım bulmadan önce, uygulamalarınızı test etmek için kullanabilirsiniz. Uygulamalarınız için sağlam ve işe yarar testler sunabilirler. Önemli farklar: performans, güvenlik ve fiziksel cihazlar ile karşılaştırdığımızda nasıl etkileşime girdiğimiz. Bu farklar bazı test sonuçlarının geçerliliğini etkileyebilir.

Uygulama ve ekosistem geliştikçe, kullanılacak test cihazlarının ayarının sürekli olarak gözden geçirilmesi gerekiyor. Değerlendirmeleriniz esnasında, aynı zamanda şu anda uygulamanızın desteklemediği yeni cihazları belirleyebilirsiniz. Aşağıdaki şekil bu kavramları gösterir.

— ŞEKLİ BURAYA YERLEŞTİRİN

Yazılımınızın hedef kullanıcılar tarafından yapıldığı gibi gerçek, fiziksel telefonlarda çalıştırılması gerekir. Çeşitli telefon modellerinin performans özellikleri birbirlerinden ve bilgisayarınızdaki sanal cihazlardan oldukça farklıdır. Bu nedenle test amacıyla fiziksel cihaz bulmak için elinizden geleni yapın. Dokunmatik ekran, fiziksel klavye, ekran çözünürlüğü, şebeke oluşturma çip seti vb. özellikleri olan popüler, yeni modellerden karışık olarak seçim yapmak iyi bir başlangıç olur. Düşük seviyedeki veya eski cihazlar üzerinde de yazılımınızı deneyin; bu cihazları kullanan kişileri de mutlu etmeniz gerekir.

İşte, fiziksel cihazlarda test edilecek alanlara bazı örnekler:

- **UI içerisinde gezinim:** Örneğin, kullanıcılar uygulamanızı tek elle kullanabilir mi? Farklı aydınlatma koşullarının etkileri: kullanıcı arayüzü deneyimi dışarıdayken, gerçek güneş ışığında değişiklik gösterebilir. Mobil bir cihazda kullanıcıların çoğu hareket halinde olacak. Ekranı

döndürün ve uygulamanın eşit derecede çekici ve işlevsel olduğundan emin olun.

- **Konum:** Eğer uygulamanızdaki konum bilgisini kullanırsanız, hem hızlı hem de yavaş hareket edin. Uygulamanızın nasıl davrandığını görmek için, düzgün çekmeyen ağı ve GPS kapsamı olan yerlere gidin.
- **Multimedia:** ses, video oynatma ve kaydetme imkanlarını test etmek fiziksel cihazlar ile emülatörler arasında önemli derecede farklılık gösterebilir.
- **İnternet bağlanabilirliği:** bir internet bağlantısının oluşturulması oldukça fazla zaman alabilir. Bağlantı gecikmesi ve bant genişliği şebekeye, mevcut şebeke gücüne ve eş zamanlı bağlantı sayısına bağlıdır. Kesintili bağlantının etkilerini test edin ve bunun karşısında uygulamanın nasıl yanıt vereceğini görün.

Uzak Cihazlar

Eğer elinizde fiziksel cihazlar yoksa veya uygulamanızı diğer ağlarda test etmeniz gerekiyorsa (özellikle yurt dışında ve diğer bölgelerde), ‘uzak cihaz hizmetleri’nden biri size yardımcı olabilir. Bunlar testinizin genişliğini ve derinliğini az bir ücret karşılığında veya ücretsiz olarak arttırmanıza yardımcı olabilir.

Çeşitli üreticiler bu hizmeti, kayıtlı yazılım geliştiricileri için kendi telefon modellerinin alt setlerinde ücretsiz sunmaktadırlar. Hem Nokia⁴ (kendi platformları için) hem de Samsung⁵ (Android ve Tizen için), sınırlı ancak ücretsiz günlük erişim sunmaktadırlar.

Çeşitli cihazlarda ve platformlarda benzer testler için

⁴ developer.nokia.com/Devices/Remote_device_access/

⁵ developer.samsung.com/remotetestlab/rtlDeviceList.action

SauceLabs, testdroid, PerfectoMobile veya DeviceAnywhere gibi şirketlerin ticari hizmetlerini de kullanabilirsiniz. Bazı üreticiler bu hizmetleri markalar ve tanıtımını yapar ancak, genellikle kısa bir deneme süresinin ardından bunlar için ödeme yapmanız gerekir. Bazı ticari hizmetler, otomatik testler oluşturmanıza olanak sağlamak amacıyla API'ler sağlar.

Uzaktaki cihazları, uzak ofislerde ve konumlarda barındırma hizmeti vererek özel bir depolama yeri bile oluşturabilirsiniz.

Ortak cihazları kullanırken kişiselliğe ve gizliliğe dikkat etmelisiniz.

Kitle Kaynaklı (Crowd-Sourced) Test Etme

Dünya genelinde cep telefonu olan milyarlarca kullanıcı var. Bunların bazıları profesyonel yazılım test ediciler ve bunların da bazıları uest.com, testhub.com ve mob4hire.com gibi profesyonel, dış kaynaklı test hizmeti şirketleri için çalışmaktadır. Daha büyük özel yazılım test ekiplerinin bulundurulmasına oranla uygulamanızı hızlı ve nispeten daha ucuza test edebilirler.

Bu hizmetler yaptığınız diğer testleri arttırabilir, ancak bunları tek resmi test olarak kullanmanızı tavsiye etmiyoruz. İyi sonuçlar elde edebilmek için yapmak istediğiniz testleri belirlemek adına zamanınızın ve çalışmalarınızın bir kısmını buna ayırmanızı ve sonuçları incelemek için şirketle beraber çalışmanızı tavsiye ediyoruz.

Belirli Hususlara Dikkat Etmek

Platformlar, ağlar, cihazlar ve hatta firmware bile kendine özel, özgüldür. Bunların herhangi biri, uygulamalarınız için sorunlara neden olabilir. Eğer hızlı ve erken geribildirim alma fırsatınız varsa, ilk önce bunları elle test edin. == Test Otomasyonu == Otomatik testler, olası sorunlarla ilgili erken geribildirim sunarak, özellikleri sunma hızınızı arttırmaya ve bunu korumaya yardımcı olur. Bunu yapmak için iyi tasarlanmış

ve uygulanmış olmaları gerekir. Aksi takdirde, sorunlu ve güvenilir olmayan otomatik testleri sürdürmeye çalışmanın yanı sıra, bozuk ve güvensiz bir uygulama ile uğraşarak iş yükünüzü arttırma riskine girersiniz. İyi otomatikleştirilmiş testler, iyi yazılım geliştirme uygulamalarını taklit eder; örnek olarak {Tasarım Desenleri en.wikipedia.org/wiki/Design_Patterns}nin modülerliğini kullanmak, kod incelemelerini gerçekleştirmek, vb.

Kullanmayı planladığınız test otomasyon araçlarının uzun ömürlülüğünü ve geçerliliğini değerlendirmek önemlidir. Aksi takdirde, desteklenmeyen test otomasyonu kodları ile uğraşmanız gerekebilir. Geliştirme SDK'sının bir parçası olarak sunulan test otomasyon araçları dikkate değerdir. Bunlar genellikle ücretsizdir, belli bir platform için doğal olarak mevcuttur ve büyük şirketler tarafından desteklenirler.

Test Otomasyonu

BDD Test Otomasyonu

BDD'nin açılımı Davranış Güdümlü Geliştirme'dir⁶. Davranış, otomatikleştirilmiş testler olarak çalıştırılabilen, belli bir biçim sahibi metin dosyaları ile tanımlanmıştır. Testlerin formatı yazılım projesine dahil olan herhangi biri tarafından okunabilir ve anlaşılabilir olmalıdır. Sanal olarak her dilde yazılabilir. Örneğin, Japonca⁷. Test betiklerini yapılandırmak amacıyla **Given, When, Then** gibi durumlar için basit ve tutarlı bir yol olarak kullanılırlar.

Mobil uygulamaları test etmek için çeşitli BDD yapıları vardır. Bunlar:

⁶ en.wikipedia.org/wiki/Behavior-driven_development

⁷ <http://github.com/cucumber/cucumber/tree/master/examples/i18n/ja>

- Android ve iOS için Calabash: <http://github.com/calabash>
- iOS için Frank: : www.testingwithfrank.com
- Android için RoboGerk: : <http://github.com/leandog/RoboGherk>
- iOS için Zucchini: : www.zucchiniframework.org

ve iOS ve Android üzerindeki web uygulamaları dahil olmak üzere, web uygulamalarını test etmek için Selenium-WebDriver ile entegre olan çeşitli uyarlamalardır.

Sıklıkla, özel 'adım tanımları' (test edilen uygulama ile etkileşime giren küçük komut dizileri), kod becerisine sahip birisi tarafından yazılmalıdır.

GUI Test Otomasyonu

Otomatik testler Grafiksel Kullanıcı Arayüzü (GUI) aracılığı ile uygulama ile etkileşime girer. Bu yöntem, test endüstrisinin hayat iksirlerinden bir tanesidir; birçok kişi denemiş ancak çok azı mobil uygulamalar için yararlı ve tutarlı GUI test otomasyonu yaratmada başarılı olmuştur. GUI test otomasyonunun çok zorlu olmasının temel nedenlerinden bir tanesi, Kullanıcı Arayüzünün, otomatik testlerin uygulama ile etkileşim yöntemine etki eden sonuçları ortaya çıkarma ihtimalidir.

Testlerin uzun vadede ve uygulama değiştikçe etkili olabilmesi için, geliştiricilerin etiketleri ve otomatik GUI testleri tarafından kullanılan diğer unsurları tasarlamaları, uyarlamaları ve desteklemeleri gerekmektedir. UI Otomasyonu⁸ ile Apple ve daha yakın zamanda Android⁹ UI otomasyonu için fiili arayüz

⁸ developer.apple.com/library/ios/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/UsingtheAutomationInstrument/UsingtheAutomationInstrument.html

⁹ developer.android.com/tools/testing/testing_ui.html

olarak UI unsurlarına atanmış olan Erişilebilirlik etiketini kullanmaktadırlar.

Bazı ticari şirketler kendi araçlarını açık kaynaklı hale getirmişlerdir; Örneğin GorillaLogic'in MonkeyTalk¹⁰'u ve Xamarin'in Calabash¹¹'i. Bu araçlar özellikle Android ve iOS için çapraz platform desteği sağlamayı amaçlar. Başarılı açık kaynaklı diğer yapılar: Robotium¹² ve Frank¹³.

Arayüzsüz (Headless) İstemci

Modern bir mobil uygulamanın kullanıcı arayüzü (UI) tüm kod tabanının %50'sinden fazlasını oluşturabilir. Eğer test süreciniz, kullanıcılar için tasarlanmış olan GUI ile sınırlıysa, test ve hata ayıklama sürecinizi gereksiz yere karmaşık hale getirmeniz gerekebilir. Bir diğer yöntem çekirdek kodun etrafında (genellikle bu ağ ve işletme katmanlarını da içerir) çok basit bir UI tabakası oluşturmaktır. Bu 'headless' istemci, cihaz, taşıyıcı ve diğer çevresel sorunlara ilişkin hataları hızlı bir şekilde izole edip belirlemenize yardımcı olabilir.

Bir headless client yaratmanın başka bir faydası, bazı test işlemlerini daha da basitleştirmesidir; bütün önemli iş fonksiyonlarını gerçekleştirebilir ve/veya test sonuçlarını otomatik hale getirip raporlayabilir.

Dosya IO operasyonlarına izin vermek için kullanıcıya sorguların sunulduğu (birçok sefer) Dosya İşlenmesini test eden bir J2ME uygulaması gibi çeşitli telefon modelleri üzerinde önemli özellikleri ve becerileri 'arayan' iskelet programlarını da yaratmayı düşünebilirsiniz. Olgun platformların parçalanmışlığı ve tuhaf davranışları göz önünde

¹⁰ gorillalogic.com/testing-tools/monkeytalk

¹¹ github.com/calabash

¹² code.google.com/p/robotium

¹³ testingwithfrank.com

bulundurulduğunda, yaptığınız yatırımın karşılığını hızlı bir şekilde size geri verecektir.

Bir Uygulamanın Yaşam Döngüsünü Beş Aşamada Test Etmek

Mobil bir uygulamanın tüm yaşam döngüsü 5 aşamadan oluşur: uyarlama (implementasyon), doğrulama, piyasaya sürme, gerçek hayat ve onaylama. Testler her aşamaya uygulanır. Daha önceki aşamalar için alınan kararların bazıları, ileriki aşamalarda testlerinizi etkileyebilir. Örneğin, eğer ilk aşamada otomatik bir sistem testi istediğinize karar verirsiniz, ileriki aşamalarda uygulaması daha kolay olacaktır.

Aşama 1: Uyarlama

Uyarlamaya tasarım, kod, birim testleri ve görevlerin oluşturulması dahildir. Genellikle testçiler bu görevlerde yer almazlar. Ancak iyi bir test, uygulamanın iyi yapıldığından emin olmamıza yardımcı olarak uygulamanın kalitesini ve başarısını bariz şekilde arttırabilir.

Test açısından aşağıdaki sorulara karar vermelisiniz:

- Test güdümlü geliştirme (TDD) kullanıyor musunuz?
- TDD kullanmasanız bile birim testleri yazıyor musunuz?
- Otomatik sistem testleriniz olacak mı? Eğer olacaksa bu otomatik sistem testlerini nasıl kolaylaştıracaksınız? Örneğin, UI'deki önemli nesnelere uygun etiketler ekleyerek.
- Uygulamalarınızı nasıl değerlendireceksiniz? Örneğin; Mobil Mantıksal Analizler kullanarak mı? Arıza bildirimi? Kullanıcılardan aldığınız geri bildirimlerle mi?

Tasarımı sorgulayın. Doğal olarak hedeflenen amaca uygun olduğundan emin olmak istiyorsunuz ve ayrıca ciddi hatalar yapmaktan kaçınmak istiyorsunuz. Phillip Armour'un cahilliğin beş emri¹⁴ isimli yazısı yaklaşımınızı yapılandırmanıza yardımcı olacak harika bir kaynak.

Bu aşamada aynı zamanda uygulamanızın test edilebilirliğini nasıl arttıracığınızı düşünün. Böylece uygulamanızın daha etkili ve verimli şekilde test edilmesini kolaylaştırabilirsiniz. Birim testleri ve Test Güdümlü Geliştirme (TDD) uyarlama aşamasında gerçekleştirilir. Derleme süreçlerinizi ve komut dizilerinizi, etkin, güvenilir ve verimli olduklarından emin olmak amacıyla test etmeyi unutmayın; aksi takdirde uygulamanın yaşam döngüsü içerisinde yetersiz derleme etkilerinden muzdarip olacaksınız.

Aşama 2: Doğrulama

Doğrulama adımı birim testlerinin değerlendirilmesi, dahili kurulum ve sistem testleri dahildir.

Birim testlerinizi gözden geçirin ve potansiyellerini değerlendirin: bunlar gerçekten faydalı ve güvenilir mi?

Not: Bunlar, uyarlama aşamasının bir parçası olarak incelenmelidirler ancak mevcut kod temeli için geliştirmenin 'tamamlanmış' olarak ele alınmasından önce maddi eksiklikleri görmek için iyi bir zamandır.

Yükleyeceğiniz uygulamaları "piyasaya sürme öncesi" aşamasında belirli cihazlar için kullanmayı göz önünde bulundurmalısınız. Bazı platformlarda (Android, iOS ve Windows Phone dahil) telefonların geliştirme amaçlı uygulamaların da yüklenebilmesi için ayarlanması gerekir. Bir yandan uygulamanın hangi telefonlarda test edileceğine de karar vermeniz gerekir. Örneğin; uygulamayı mobil platformun her

¹⁴ www.plan.cs.colorado.edu/diwan/3308-07/p17-armour.pdf

uygun sürümünde test etmek akıllıcadır. iOS için bu sadece en son sürümleri içerebilir. Android için ayrıca, sürüm 2.x ile satılan ve hiçbir zaman 4.x'e güncellenmeyecek olan düşük seviyedeki cihazları göz önünde bulundurmanız gerekir.

Ayrıca cihazların farklı biçim faktörlerini test etmek (ekran boyutlarının oranının farklılık gösterdiği durumlar gibi) isteyebilirsiniz. iPhone 5'in yeni ekran boyutları birçok UI hatasını ortaya çıkarmıştır. Android geliştiricileri, farklı ekran boyutlarının tetikleyebildiği çeşitli sorunların farkındadır.

Sistem testleri, testçiler tarafından genellikle etkileşimli olarak gerçekleştirilir. Bazı sistem testleriniz için test otomasyon araçlarını ve geliştirme çatılarını değerlendirmeyi unutmayın. Daha ileride daha fazla ayrıntıya gireceğiz.

Ayrıca uygulamanızın aşağıdaki koşulları nasıl sağladığını bilmek isteyebilirsiniz:

- Kullanılabilirlik, kullanıcı deneyimi ve estetik gereksinimler
- Özellikle nihai kullanıcılar tarafından algılanan performans
- Uluslararasılaştırma ve yerel testler

Aşama 3: Piyasaya Sürme

Bu aşama yayınlama öncesini ve yayınlamayı içerir.

Henüz büyük uygulama mağazaları ile çalışmamış olan kişiler, uygulamanızın onaylanma süresi gibi çoğu unsurun sizin kontrolünüz dışında olduğu zorlu bir deneyimle karşılaşmaya hazır olun. Ayrıca bazı uygulama mağazalarında yeni sürümü geri alamazsınız. Bu yüzden eğer mevcut sürümünüzün temel hataları bulunuyorsa, bu hataları ortadan kaldıran yeni bir sürümü sunmanız gerekir. Daha sonra uygulamanızın çalışan bir sürümünün kullanıcılar tarafından indirilmesi öncesinde, yeni sürümünüzün uygulama mağazası tarafından onaylanmasını beklemeniz gerekecektir.

Bu kısıtlamalar ele alındığında, uygulamanın hedeflenen cihazlara olan uyumluluğu gibi çeşitli ön-yayın kontrollerini yapacak şekilde test işleminizi genişletmeniz faydalı olacaktır. Ana platformların sağlayıcıları artık kendi sunum kriterlerine uyum açısından uygulamanızı test etmeniz için kılavuzlar yayınlamaktadırlar. Diğer uygulama mağazalarını hedefleseniz de bu ilkeler size yardımcı olabilir.

Apple	https://developer.apple.com/appstore/resources/approval/guidelines.html (Erişim için Apple hesabı gereklidir)
Android	http://developer.android.com/distribute/googleplay/publish/preparing.html#core-app-quality
Windows Phone	http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394032(v=vs.105).aspx
BlackBerry	http://developer.blackberry.com/devzone/appworld/tips_for_app_approval.html

Aşama 4: Gerçek Hayat

Bu aşamaya arama, güven, indirme ve kurulum dahildir. Uygulamanız yayımlandıktan sonra, kullanıcıların onu bulması, güvenmesi, indirmesi ve kurması gerekir. Bu aşamayı her yönüyle üretim ortamında test edebilirsiniz. Uygulamanızı ilgili uygulama mağazasında ve popüler arama motorlarında aramayı deneyin. Hedef kullanıcılar tarafından ne kadar farklı yöntemle bulunabiliyor? Peki hedef grubu dışındaki kullanıcılar – onların da uygulamanızı bulmalarını istiyor musunuz? Kullanıcılar,

uygulamalarınızı indirip denemek için size nasıl güvenebilirler? Uygulamanız için gerçekten çok fazla izin gerekiyor mu? İndirme boyutu ne kadar ve mobil ağ üzerinden indirmek ne kadar pratik? Kullanıcının telefonunda uygulama için yeterli yer bulunuyor mu? Uygulama doğru bir şekilde yükleniyor mu? Uygulamanın bazı telefonlar tarafından reddedilebileceği imza sorunları olabilir.

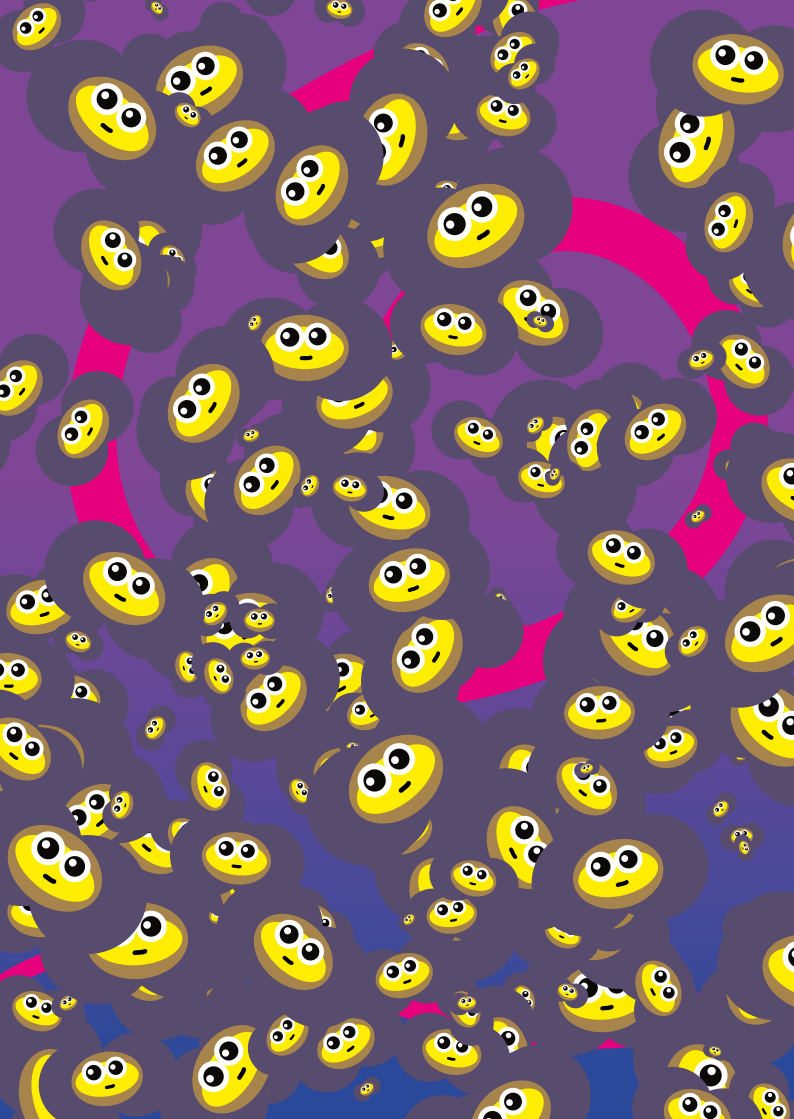
Sürekli yeni sürümlerin ortaya çıkması ile gerçek hayat unsurlarını izlemek için proaktif bir yaklaşım gerekebilir.

Aşama 5: Onaylama

Bu aşamaya ödeme, kullanım ve geri bildirim dahildir. Halihazırda bildiğiniz gibi, kötü bir geribildirime sahip olan bir mobil uygulamanın başarılı olması pek muhtemel değildir. Bunun yanı sıra, birçok uygulamanın kullanıcının telefonu içerisindeki yaşam döngüsü çok kısa olmaktadır. Eğer uygulama, kullanıcıları birkaç dakika içerisinde cezbedip kendisine çekmezse, uygulamanın silinmesi muhtemeldir. Ödemeyi bekleyenleriniz için özellikle uygulama içi ödemeler açısından, çeşitli ödeme türlerini test etmeniz önemli olacaktır.

Aşağıdaki unsuru olabildiğince kısa bir süre içerisinde test etmek için farklı yöntemler arayın:

- Problem algılama ve raporlama. Bunlar kendi kodunuz, üçüncü kişi hizmetleri ve çevrimiçi hizmetler olabilir.
- Mobil Mantıksal Analizler. Toplanan verilerin anlamı var mı? Bildirilen verilerde ne anormallik var? Sonuçları ve diğer unsurları alırken bir gecikme mevcut mu?



Gelir Edinmek

Sonunda uygulamanızı veya mobil web sitesini bitirdiniz ve beta testi geribildirimlerine göre son rötuşları uyguladınız. Bunu bir hobi, marka teşhiri, vb. için yapmadığınızı farz ederek, artık para kazanma zamanınızın geldiğini söyleyebiliriz. Ancak bunu nasıl yapabilirsiniz, seçenekleriniz neler?

Genelde, aşağıdaki gelir kaynağı oluşturma seçenekleri bulunmaktadır:

1. **İndirme başına ödeme:** Uygulamanızdan indirme başı para kazanın
2. **Uygulama içerisinde ödeme:** Uygulamanıza ödeme seçenekleri ekleyin
3. **Mobil reklamcılık:** Reklamlardan para kazanın
4. **Sponsorluk:** Sponsorunuzda oturum açan kullanıcıların her biri için para alın
5. **Gelir paylaşımı:** Uygulamanızdan kaynaklanan operatör hizmetlerinizden gelir kazanın
6. **Dolaylı satışlar:** Bağlı kuruluşlar, veri raporlama ve fiziksel ürünler.
7. **Bileşen piyasası:** Uygulamanızın bileşenlerini veya beyaz etiket sürümünü diğer geliştiricilere satın

Kendi geliştirmenizi planlarken, gelir edinme iş modeli, uygulamanın fonksiyonel ve teknik davranışını etkileyebileceği için erken tasarımınızın önemli unsurlarından bir tanesi olabilir.

İndirme Başına Ödeme

İndirme başına ödeme (Pay-per-download - PPD) ile uygulamanız her kullanıcıya bir defa satılır, kullanıcılar uygulamayı indirip telefonlarına kurarlar. Ödeme, bir uygulama mağazası veya mobil operatör tarafından düzenlenebilir veya kendiniz de bir düzen oluşturabilirsiniz.

Uygulamanız bir uygulama mağazasında dağıtıldığında mağaza sizin için ödeme işlemlerini takip edecektir. Bunun karşılığında mağaza bütün satışlar üzerinden bir gelir payı alır (tipik olarak %30). Çoğu durumda mağazalar ülkeye ve para birimine göre özel bir fiyatlandırma için sabit bir fiyat noktası matrisi sunarlar (\$0.99, EUR 0.79, \$3 gibi)

İndirme başına ödeme uygulamaları genellikle iki şekilde işler: operatör faturalaması veya kredi kartı ödemeleri.

Operatör faturalamasında satış ücretinin müşterilerinizin cep telefonu faturalarına ekleneceğini belirten bir yazıyı onaylaması veya SMS göndererek onay vermesi ile uygulama için ödeme yapmanıza olanak sağlar. Bazı durumlarda operatör faturalandırması bir uygulama mağazası tarafından yapılır (dünyadaki çeşitli taşıyıcılar için operatör faturalandırmasını destekleyen Google Play gibi). Diğer durumlarda doğrudan geliştirici tarafından uygulanabilir.

Her operatör, satış fiyatından belli bir gelir payı elde edecektir (genel olarak %30-%65 arası) ve eğer bir tanesini kullanıyorsanız, bir kaynak da kendi payını alacaktır. Güvenlik (uygulamanızın kopyalanmasını nasıl önlediğiniz) ve yönetilebilirlik, PPD modelinde yaygın görülen sorunlardandır ancak bazı senaryolarda tek gelir kaynağı yaratma seçeneği olabilir. Operatör faturalandırmasını kendi kendinize yapmanız zor olabilir; özellikle birden fazla ülkede, her ülkenin operatörü ile sözleşmeler imzalamanız gerektiğinden dolayı baş ağrıtıcı olabilir. Bilinmeyen sebeplerle, Vodafone gibi bazı operatörler

Birleşik Krallık ve Almanya gibi bazı önemli piyasalarda operatör faturalamasını Google Play için bir seçenek olmaktan çıkıyorlar. Bunun nedeni yerel mobil banka ödemeleri gibi daha iyi alternatiflerin uygulanması olabilir.

Çoğu satıcı uygulama mağazalarının, operatör faturalandırma sözleşmelerini takip ettiklerini belirtmek önemli olabilir; Nokia Store'un hem eski Nokia Store, hem de Microsoft'un Windows Phone Marketplace için 60 ülkede operatör faturalandırma kapsamı bulunur. Google ve Blackberry'nin benzer seçenekleri bulunmaktadır. Bunun yapılmasındaki temel neden, kredi kartı ödemesi ve operatör faturalaması seçeneği sunulduğunda, kullanıcıların genellikle operatör faturalamasını tercih etmeleridir (Nokia araştırmalarına göre kredi kartı ile yapılan ödemelerden elde edilen gelir artışı 10 kata kadar arttı). Nokia, faturalandırma gelirinin %70'ini geliştiricilere sunarak geliştiricileri en azından operatör payından uzak tutmaktadır.

Kredi kartı faturalaması Apple, Google (bazı durumlarda), Amazon ve diğer mağazalarda kullanılıyor. Apple'da, iPhone kullanıcılarının kayıta kredi kartı bilgilerinin sunması gerekiyor. Google'da ve Android'de de artık bu gerekiyor. Analistlere göre satın alma işleminde önce bu bilginin girilmesi, uygulama geliri başına aylık elde edilen tutar üzerinde önemli bir fark yaratmaktadır.

En son ödeme seçeneği, kendi web sitenizi yaratıp PayPal mobile, Dutch initiative [eM! Payment](#)¹, dial-in to premium landlines² veya diğerleri gibi bir ödeme mekanizmasını dahil etmektir.

PPD kullanımı genellikle, uygulamanız için herhangi bir özel tasarım ve kod gereksinimi olmadan uyarlanabilir ve

¹ empayment.com

² daopay.com

başlangıç olarak, daha düşük kurulum maliyeti ve az sayıda idari masraflar içerdiği için uygulama mağazası seçeneklerini kullanmanızı tavsiye ediyoruz.

Her bir ödeme türü için talebin fiyat esnekliğini göz önünde bulundurmak önemlidir PED³. Fiyatın yükseltilmesi, daha yüksek gelir elde edilmesi (veya tam tersine, fiyatın düşürülmesi daha az gelir elde edilmesi) anlamına gelmiyor, fiyatınızın kullanıcı tabanınızın beklentilerine uyması gerekiyor.

Uygulama İçi Ödeme

Uygulama içi ödeme, belirli eylemlerinizi veya varlıklarınızı için uygulama içerisinde satış yapmanıza fırsat veren bir yöntemdir. Temel bir kullanım yöntemi, bir deneme süresinden sonra uygulamanızın tek seferlik satışını gerçekleştirmektir – eğer uygulamanızın özelliklerinin daha yüksek bir fiyatı hak ettiğini düşünüyorsanız bu yöntem, PPD'ye kıyasla çok daha fazla satış size kazandırabilir. Alternatif olarak uygulamanızın temel özelliklerini ücretsiz olarak sunup, özel içeriği (videolar, sanal krediler, özel bilgiler, ilave özellikler gibi) ücretli bir şekilde sunabilirsiniz. Çoğu uygulama mağazası uygulamada satın alma seçeneği sunar veya kendi ödeme mekanizmanınızı uygulayabilirsiniz. Eğer yalnızca tek seferlik “tam lisans” ödemesinden başka seçeneklere bakmak isterseniz, kullanıcılarınızın ne zaman, nasıl ve ne ödeyeceğini düşünüp uygulamanızı ona göre tasarlamanız gerekir.

Uygulama içi satın almalar, kullanıcıların ilk önce “ücretsiz” oyunları kullanıp, onların alışmasından sonra ilave özellikleri ücretli bir şekilde onlara sunma yöntemi ile birçok piyasada önde gelen gelir yaratma modeli haline gelmiştir. Distimo'nun⁴

³ en.wikipedia.org/wiki/Price_elasticity_of_demand

⁴ <http://www.distimo.com/publications>, “2013 Yılı İncelemesi'ne”

göre, 2013 yılında küresel iOS uygulama gelirlerinin %92'si ve Android uygulaması gelirinin %98'i uygulama içi satın alma işleminden gelmektedir.

Bu tür ödeme özellikle oyunlarda popülerdir (ekstra güç, ekstra seviye, sanal kredi vb. satın alma gibi özellikler için) ve temel uygulamayı ücretsiz olarak sunarken daha geniş bir kurulum temeli oluşturmaya yardımcı olabilir. Ancak göz önünde bulundurmanız gereken unsur, bazı uygulama mağazalarının (Apple'ınki gibi) uygulamanız içerisinde üçüncü kişi ödeme seçeneklerine izin vermemesidir. Bu durum uygulama mağazasını ücretsiz bir dağıtım yöntemi olarak kullanırken, mağazanın gelir payını ödemekten kaçınmanızı engellemek için yapılmaktadır.

Ayrıca uygulama içi ödeme yöntemini devreye sokmak amacıyla uygulamanızı buna göre geliştirip tasarlamanız gerektiğini bilmelisiniz. Eğer uygulamanız çeşitli platformlarda uygulanıyorsa, her platform için farklı bir mekanizma uygulamanız gerekebilir (her uygulama mağazasına ilave olarak).

PPD'de olduğu gibi, bir uygulama mağazası tarafından sunulan bir uygulama içi satın alma mekanizması ile başlamanızı tavsiye ediyoruz; bunlardan bazıları operatör faturalandırma hizmetlerini kullanabilirler veya önceden verilen kredi kartı bilgilerini devreye sokabilirler (Apple veya Amazon gibi) veya doğrudan operatörler tarafından sunulan uygulama içi ödeme kullanılabilir. Kullanıcı bakış açısına göre bu, ödeme yapmanın en kolay ve uygun yolu (bir veya iki tıklama, kredi kartı bilgisi, kullanıcı adı veya diğer bilgileri girmeye gerek yok), bu nedenle geliştiriciler en yüksek kullanıcı Kabul ve dönüşüm oranlarını bekleyebilir.



Mobil Reklamcılık

Web sitelerinde yaygın şekilde görüldüğü gibi, reklamları görüntüleyerek para kazanmaya karar verebilirsiniz. Mobil reklamları göstermek için size bazı araçları sunabilecek olan çeşitli oyuncular bulunur ve bu, mobil tarayıcı uygulamalarında para kazanmanın en kolay biçimidir. Admob.com, Buzzcity.com ve inmobi.com mobil reklamcılık sunan taraflardan bazılarıdır. Ancak, çok çeşitli cihaz, ülke ve özellik olduğu için şu anda 50'den fazla büyük mobil reklam ağı bulunmaktadır. Her bir ağ belli belirsiz farklı yöntemler sunar ve uygulamanızın hedef kitlesi için en iyi şekilde gelir kaynağı oluşturan bir tanesini doğrudan bulamayabilirsiniz. Altın bir kural yoktur; sizin için en iyisini bulabilmek amacıyla deneme yanılma yöntemini kullanmanız gerekebilir. Ancak hızlı bir başlangıç için 40'dan fazla mobil reklam ağındaki içeriği birleştirip optimize eden Madgic⁵, smaato⁶ veya inneractive⁷ gibi bir mobil reklam kaynağını kullanabilirsiniz. Çoğu reklam ağı, reklam gelirlerinin %30-%50'ini almakta ve kaynaklar ise ilave bir %15-%20 payı almaktadırlar.

Eğer uygulamanızın satışları iyiye ve belli bir ülkede geniş bir hacme sahipse, reklamları doğrudan bir reklam ajansına veya markasına (Premium reklamcılık) satmayı veya bunu sizin adınıza yapması için bir medya ajansını görevlendirmeyi düşünebilirsiniz.

Tekrar etmek gerekirse, cihaz satıcılarının bir çoğu, uygulama mağazası sunumlarının bir parçası olarak mobil reklam hizmetlerini sunmaktadırlar ve bu mekanizmalara bir göz atmanız faydalı olabilir. Bazı durumlarda, uygulamanızı onların

⁵ madgic.com

⁶ smaato.net

⁷ inner-active.com

mağazalarına sokabilmeniz için satıcının teklif ettiği hizmetleri kullanmanız gerekebilir.

Uygulama içi reklamcılıkta uygulamanızı dikkatli bir şekilde tasarlamanız ve kodlamanız gerekecek.

Uygulamanız içerisindeki reklamların görüntülenme yerlerinin yanı sıra, çeşitler ve vazgeçme mekanizmasının da dikkatli bir şekilde düşünülmesi gerekir. Eğer reklamlar çok yoğun olursa kullanıcılarınız uygulamayı silebilirler; az olursa, bu da daha düşük veya sıfır gelir anlamına gelir. Geleneksel büyük başlık reklamcılığına kıyasla yeni olan unsur arayer reklamcılığıdır: Bu terim genellikle bütün ekranı kaplayan ve alt kısımda "ekranı atla" seçeneği bulunan reklamlar için kullanılır. Reklamları yerleştirmek için çeşitli denemeler yapmanız ve doğru seviyeyi ve pozisyonu bulmanız gerekebilir.

Sponsorluk

German startup Apponsor⁸ reklam görüntülemeye veya indirme ücreti almaya gerek kalmadan yeni bir para kazanma yolu sunuyor: Kullanıcı uygulamanızı ücretsiz ediniyor ve sponsorunuzun haber bültenine üyelik için sorgu çıkartılıyor. Sponsor bunun karşılığında geliştiriciye, her bir haber bülteni kaydı için belli bir tutar ödüyor.

Dolaylı Satışlar

Satışlara yön vermek için bir başka seçenek de uygulamanızı kullanmak.

Burada, genellikle uygulamanız veya web siteniz ücretsiz sunulur ve ardından aşağıdaki mekanizmalar kullanılır:

⁸ apponsor.com



1. Bağlı programlar: Ücretsiz bir uygulama içerisinde üçüncü kişi veya sizin ücretli uygulamalarınızın reklamını yapın Bakınız: MobPartner⁹. Bu, mobil reklamcılıkta bir değişim olarak kabul edilebilir.
2. Veri raporlama: Davranışı izler ve verileri ilgili taraflara satar. Gizlilik nedeniyle, her türlü kişisel bilgiyi gizli tutmanız gerektiğini unutmayın, tüm verilerin gizli kalmasını ve raporların toplanmasını sağlayın.
3. Sanal ve gerçek dünyanın karşılaştırılması: malları gerçek dünyada satmak için, uygulamanızı bir pazarlama aracı olarak kullanın. Bunun tipik örnekleri; araç uygulamaları, dergi uygulamaları ve McDonald's ve Starbucks gibi büyük markalar ile Groupon'un genellikle kullandığı iş modeli gibi kupon uygulamaları.

Eğer isterseniz, bu seçeneği diğer gelir yaratma seçenekleri ile birleştirmenizi engelleyen hiçbir şey bulunmamaktadır, ancak yoğun ve zorlama promosyon hissini vermediğinizden emin olmanız gerekir.

Bileşen Piyasası

Bir Bileşen Piyasası (Component Marketplace - CMP), yazılım bileşenleri veya beyaz etiketli uygulamalar satarak geliştiricilerin kendi ürünleri üzerinden gelir yaratabilmeleri için başka bir fırsat sunmaktadır. Bir yazılım bileşeni, yazılımın yapı taşıdır ve daha yüksek seviyedeki bir yazılım tarafından kullanılacak olan tanımlı bir işlevsellik sunar.

Bu noktada ortaya çıkan soru, CMP'lerin açık kaynak ile farkının ne olduğudur. Bir kullanıcı açısından açık kaynak genellikle ücretsizdir. Kaynak kodu sunulmalıdır ve

⁹ mobpartner.com

kullanıcıların kaynak kodunu modifiye etme ve türetilmiş işi dağıtma hakkı bulunmaktadır.

Bazı bileşen tedarikçilerine lisans ücreti gerekir. Geliştiricinin daha düşük seviyedeki bir kodu hatadan ayıklamasına yardımcı olan tam kaynak kodunu sağlarlar. Bazı CMP'ler tüm modelleri destekler: Kaynak kodlu veya kaynak kodsuz ücretli bileşenlerin yanı sıra kaynak kodlu veya kaynak kodsuz ücretsiz bileşenler.

Eğer bileşen arayan bir geliştiriciyseniz, CMP'ler iki temel avantaj sunar: Bunlardan ilkinde, yalnızca yazılım bileşeni kullandığınız için kodunuzu açık kaynaklı hale getirmenize gerek olmamasıdır. Tüm açık kaynaklar lisansla sunulur. Apache gibi bazı lisanslar ticari anlamda uygundur; AGPL ve OSL gibi diğerleri, kendi kodları ile birleştirilebilecek şekilde kodunuzu açık kaynaklı hale getirmeyi şart koşarlar. Bunu istemeyebilirsiniz.

İkinci olarak, CMP'ler bileşenleri bulmak ve indirmek için kolay yollar sunar. Kullanılacak doğru unsuru bulmak için açık kaynak veri havuzlarını günlerce dolaşabilirsiniz. Bileşen piyasaları onlarca senedir var olmaktadırlar. Bileşenler için en meşhur piyasa, Windows topluluğundaki Visual Basic ve .NET'tir. componentOne gibi piyasalar ve Infragistics gibi tedarikçiler, kendi alanlarında ünlüdürler. Mobil arena içerisindeki bileşen piyasası fikri henüz yenidir. Deutsche Telekom'un Developer Garden¹⁰, ChupaMobile¹¹ ve Verious¹² bu alandaki ilgili oyunculardır.

¹⁰ www.developergarden.com/component-marketplace/

¹¹ www.chupamobile.com

¹² www.verious.com

Gelir Yaratma Modelini Seçmek

Tüm bu seçeneklerden sonra stratejiniz ne olmalı? Bunlar elbette ki hedeflerinize bağlı; şimdi birkaç tanesine bakalım:

- Geniş bir kullanıcı tabanı mı istiyorsunuz? Uygulamanızı uygulama içi satın alımlar veya mobil reklamcılık ile ücretsiz dağıtmaya çalışın (ücretli, reklamsız bir sürüm bile sunabilirsiniz)
- Kullanıcılarınızın uygulamanızı hemen satın alacaklarından emin misiniz? O zaman PPD olarak 0,99\$'a satın, ancak unutmayın: gün başına birkaç bin dolar kazansanız da, öngörünüz yanlış çıkarsa veya zorlu rakiplerle uğraşmak zorunda kalırsanız, bunun haftada birkaç yüz dolara düşebileceğini unutmamalısınız.
- Birinci sınıf özellikleri birinci sınıf fiyata mı sunuyorsunuz? Zaman ve özellik sınırlı bir deneme uygulaması düşünün ve tam sürümü, sürekli olarak veya belli bir süre boyunca uygulama içi satın alma ile etkinleştirmeyi göz önünde bulundurun.
- Bir oyun mu geliştiriyorsunuz? Uygulama içi reklam içeren ücretsiz bir uygulama sunmayı düşünün; veya temel bir sürüm sunun ve uygulama içi satın alma ile yeni özellikleri, daha fazla seviyeyi, farklı araçları veya unsurları açma fırsatı verin.
- Mobil uygulamanız mevcut bir PC web mağazasının bir uzantısı mı? Uygulamayı ücretsiz sunun ve gerçek dünyadaki ürünlerden ve hizmetlerden gelir elde edin.

Market Stratejileri

Gelir edinmenin başka bir yüzü pazarlama ve promosyondur. Bu ihtiyaç, uygulamanızı kendi sitenizden sunduğunuzda kendisini gösterecektir ve bir satıcının uygulama mağazası üzerinden satış yaptığınızda da buna ihtiyaç duyacaksınız. Uygulama mağazaları, mobil geliştiricilerin hem laneti hem de nimetidir. İyi yönden bakıldığında, normalde elde edilmesi zor olan bir teşhir ve satış fırsatını geliştiricilere sunarlar. Karanlık taraf ise, popüler olanların yüz binlerce uygulaması olması ve bunların aralarından çıkıp kendilerini göstermenin çok zor olmasıdır. Birçok kişi uygulama mağazası içerisinde başarı yakalamamanın piyango talihlisi olmaktan çok daha zor olduğunu belirtir.

Aşağıda, size yardımcı olabilecek birkaç ipucunu bulabilirsiniz.

Yükselmek için Temel Stratejiler

Uygulama mağazaları hakkında anlaşılması gereken en önemli şey, onların pazarlama makineleri değil, bir dağıtım kanalı olduğudur. Bunun anlamı, uygulama mağazaları uygulamalarınızı kullanıcıların cihazlarına iletmek için mükemmel bir yöntem iken, uygulamanızı sizi için pazarlamayacak olmalarıdır (ana başlık veya liste konumlandırması üzerinde ücretli bir yerleştirme seçeneğini seçmemişseniz). En iyi on uygulama listesine girmediğiniz sürece, kendi indirmelerinizi teşvik etmek için uygulama mağazalarına güvenemezsiniz. Ancak uygulamanız ile piyango oynamayın; uygulamanızı pazarlamak için bir stratejiye ve plana sahip olun.

Birçok geliştiriciye, uygulama mağazalarında kendi uygulamalarına dikkat çeken ve yüksek konumlar kazandıran taktikleri sorduk.

Cevaplar geldiğinde ortak bir fikir ortaya çıkmıştı: sihirli

bir değnek yok – bütün cephelerde en sert şekilde savaşmak zorundasınız! Ancak bazı unsurları dikkate almanız size yardımcı olabilir:

- Mükemmel bir uygulamaya ihtiyacınız var: eğlenceli olmalı, kolay bir şekilde kullanılmalı ve hatalı olmamalı. Mağazaya koymadan önce beta testçilerinin test yapmalarını sağlayın.
- Uygulama mağazasındaki simgelerinizi ve resimlerinizi cilalayın, uygulama tanıtımı üzerinde kafa yorun ve anahtar kelimelerinizi ve kategorilerinizi iyi düşünün. Eğer sonuçlardan memnun kalmazsanız deneme yanılma yöntemini kullanın.
- Dikkat çekmek için en iyi yöntemlerden bir tanesi blog ve magazin yazarlarının uygulamanız üzerinde incelemeler yapmalarıdır. Bunun karşılığında bazıları para, bazıları özel fırsatlar, bazıları da erken erişim isteyeceklerdir.
- Mümkün olduğu kadar çabuk (pozitif) görüş alın. Değerlendirme için düzenli olarak arkadaşlarınızı arayın veya kullanıcılara sorun.
- Eğer reklam yapacaksanız, birkaç gün sürecek olan yoğun bir reklam programı kullanın. Bu, yavaş ve tempolu bir ivme yerine ani bir popülerlik sunacağı için aynı miktarda parayı iki hafta boyunca kullanmaya kıyasla daha etkili olacaktır.
- Uygulama mağazasını dolaşan kişilerin yarattığı trafiğe aldanmayın; web siteniz, SEO ve sosyal medya aracılığı ile trafik yarattığınızdan emin olun.

Birden Fazla Mağazaya Karşı Tek Mağaza

Geliştiriciler için mevcut olan 120'den fazla uygulama mağazası ile birçok uygulama dağıtım seçeneği bulunmaktadır. Ancak bir uygulamayı, uygulama mağazasına göndermek için harcanan ortalama 20 dakika, birkaç indirme alabilmek için uygulamanızı bilinmeyen bir sürü yere göndererek vakit harcayacağınız anlamına gelir. Bu yüzden geliştiricilerin çoğu yalnızca 1 veya 2 mağazaya uygulamalarını gönderirler: olası büyük fırsatları kaçırmazlar ancak kodlama gibi daha önemli şeylere vakitleri kalır! Bu yüzden hangisini seçeceksiniz?

Birden fazla mağaza	Tek mağaza
Temel platformlu uygulama mağazalarının ödeme mekanizmaları, bazı ülkelerde sorun ve içerik prensipleri gibi ciddi sınırlamaları olabilir.	Akıllı telefon kullanıcılarının %90'dan fazlası sadece tek bir uygulama mağazası kullanıyor; bu da genellikle telefonla beraber gelen platform uygulama mağazasıdır.
Daha küçük mağazalar size daha fazla görünürlük seçeneği sunabilir (öne çıkan uygulama)	Kendinize ait web siteniz, size uygulama mağazalarından çok daha fazla trafik getirebilir (özellikle iyi bilinen bir markanız varsa)
Küçük mağazalar, büyüklerine oranla daha fazla sosyal medya dostudur.	Birçok küçük uygulama mağazası verileri büyük mağazalardan alır, bu yüzden uygulamanız halihazırda orada olabilir.
Operatör mağazalarının çok sıkı içerik prensipleri bulunmaktadır ve bazı türdeki uygulamalar için giriş yapmak zor olabilir.	Büyük içerikler için operatör veya platform mağazaları, birden fazla mağaza stratejisini seçmemeniz için yeterli seviyede teşhir sunabilir.

Birden fazla mağaza	Tek mağaza
Küçük mağazalar, çeşitli ödeme veya iş modeli seçenekleri sunabilir veya pek çok ülkede mevcut olabilir.	Bazı operatör mağazalarında daha kolay faturalama süreçleri vardır (bir kullanıcının mobil hesabına doğrudan faturalama gibi) bu da, daha yüksek dönüşüm oranlarına neden olur.
Bazı geliştiriciler, Android gelirlerinin %50'sinin Android Market'den geldiğini bildiriyor.	iOS geliştiricilerinin sadece 1 uygulama mağazasına ihtiyacı vardır

Platform uygulama mağazaları, sizin adınıza kullanıcılar için genel bir kapsam sunabilirler ancak zaman içerisinde hedeflenen kullanıcı temelinizi benimsemek amacıyla stratejinizi geliştirmeniz sizin elinizde olacaktır. Bunun anlamı, belirli operatör mağazalarını, belli ülkelerde popüler olan mağazaları kullanmak veya yalnızca platform mağazalarını kullanmaya devam etmektir. Büyük kitlelere sahip üçüncü kişi uygulama mağazaları bulunmaktadır. Bunlardan bir tanesi, çeşitli ülkelerde PPD ve uygulama için ödemeler gibi gelir yaratma yöntemlerini sunan Android için Amazon uygulama mağazasıdır. Ayrıca, bazı ülkelerde, Almanya'daki AndroidPit¹³ veya Çin'e özgü Android mağazalarından biri gibi yerel olarak popüler uygulama mağazaları bulunur.

Ne Kazanabilirsiniz?

En yaygın olarak sorulan geliştirici sorularından bir tanesi de, bir mobil uygulama ile ne kadar para kazanılacağıdır. Bazı uygulamalar geliştiricilerini milyoner yapabilirken, diğerleri yakın zamanda şimdiki günlük işlerini bırakamayacaklardır.

¹³ androidpit.de

Forbes.com tarafından 2012 yılında yapılan bir araştırmaya¹⁴ göre, birçok uygulama geliştiricisi, geliştirme maliyetlerini amorti edecek düzeyde bile gelir elde edememektedir ve tek platform geliştiricileri, bunun kendi başına yetecek bir şirket haline gelmesi için yeterli olmayacağını savunmaktadırlar. VisionMobile'in 6000 mobil geliştirici ile 2013 yılında yaptığı Developer Economics araştırmasına göre bunların %67'si aylık 500\$'lık "uygulama yoksulluk sınırı" altındadır¹⁵.

Sonuçta, elde edebileceğiniz şey bir ihtiyacı gidermek ve etkili pazarlamayı öğrenmektir. Tecrübeler göstermiştir ki en popüler uygulamalar kullanıcıya para ve vakit kazandıranlardır (otel indirimleri, ücretsiz müzik ve benzerleri); bunu oyunlar takip eder (Angry Birds'ün başarısı bir örnek olabilir). İşletme araçları (ofis dokümanı görüntüleyicileri, eşleştirme araçları, yedekleme araçları, vb.) bunları takip etmektedir. Ancak çoğunlukla tek bir uygulamanın başarısı öngörülemez. Başarı genellikle bir deneme yanılma tecrübesi ve büyük miktarda sabır gerektirir.

Daha Fazlasını Öğrenin

Eğer uygulama pazarlamasına daha da derinlemesine girmek istiyorsanız WIP¹⁶ tarafından yayınlanan "Mobile Developer's Guide To The Parallel Universe" kitabına bir göz atın.

¹⁴ www.forbes.com/sites/tristanlouis/2013/08/10/how-much-do-average-apps-make

¹⁵ www.visionmobile.com/products/research

¹⁶ wip.org



Sonuç

Mobil Geliştirici Kılavuzumuzun 14. Baskısını okuduğunuz için teşekkür ederiz. Okurken eğlendiğinizi ve fikirlerinizi netleştirmenize yardımcı olduğunu umuyoruz. Belki de artık, bir mobil uygulama geliştirmeye hazırsınız veya mobil uygulama işinde yeni seçenekleri keşfettiniz. Biz de öyle umuyoruz. Lütfen topluluğa katılın ve deneyimlerinizi ve fikirlerinizi bizimle ve diğerleri ile paylaşın.

Eğer bu rehberde katkıda bulunmak veya gelecek baskılara sponsorluk yapmak isterseniz, lütfen geri bildirimlerinizi adresine gönderin. Eğer Twitter kullanıyorsanız, twitter.com/enoughsoftware adresinden bizi takip edebilir ve #mdgg hashtag'ini kullanarak proje hakkındaki bilgileri tüm dünyaya yayabilirsiniz.

Bu rehberi e-kitap olarak da edinebilirsiniz bunun için; amazon.com ve kobobooks.com adreslerini kontrol edin. Alternatif olarak, pdf dosyasını www.enough.de/mdgg adresinden indirebilirsiniz. Bu yazının yazıldığı tarihte bu kitabın içeriğini www.mobiledevelopersguide.com, adresinde bulabilmeniz için çalışmaktayız. Burada, katılım göstermek ve geribildiriminizi gönderebilmek için yeni yollar bulacaksınız.





Yazarlar Hakkında

Mostafa Akbari / bitstars

Mo, son birkaç yıldır yazılım mühendisliğinde ve insan etkileşimi araştırmalarında çalıştı, yeşil hareketlilik projelerine dahil oldu. Şimdi Mo, arttırılmış gerçeklik araştırması ve geliştirmesi için RWTH Aachen Üniversitesi'nden Simon Heinen ile yan bir proje üzerinde çalışmaktadır. Kişisel konum bazlı veri ve bilgisayar vizyonu üzerine AR etkileşimleri konusuna odaklanmaktadır. Karmaşık gerçeklik oyunlarına duyduğu tutku, masa oyunlarına ve define avı oyunlarına dayanmaktadır.

 Twitter: @mosworld www.bitstars.com

Anna Alfut

Anna profesyonel yaşamına Kreatif Tasarımcı olarak başladı. Arayüz tasarımına olan tutkusunu keşfettikten sonra, iOS ve Android platformları için bir uygulamanın yazılmasına yardımcı oldu ve hem ajans, hem de müşteri tarafında çeşitli projelerde danışmanlık görevini üstlendi. Şu anda, tüketicilerin mobil ve masa üstü cihazlarında kullandıkları ürünler için UI ve UX tasarımcısı olarak evden çalışıyor. Düşünme ve çizimin yanı sıra, resimle ilgilenmekte ve Londra'da yaşamın tadını çıkarmaktadır.

www.alfutka.net

Andrej Balaz / Enough Software

Arts Bremen Üniversitesinden mezun olan Andrej, mobil uygulamalar ve diğer interaktif teknolojiler için UI, UX ve görsel tasarıma odaklanıyor. Kendisi ayrıca, bu kılavuzun düzeni ve tasarımından sorumlu. Mobil ile ilgili işlerle uğraşmadığında dijital sanatlar ve çizimle ilgilenmeyi seviyor.

 Twitter: @Designamye www.enough.de www.balaz.de

Richard Bloor / Sherpa Consulting Ltd

Richard, 2000 yılından beri mobil uygulama gelişmesi konusunda yazıyor. AllAboutSymbian.com gibi popüler internet sitelerinde katkıda bulundu, ama şu anda şirketlerin geliştiriciler için kaynak yaratmasına yardımcı olmaya odaklanıyor. Land Information NZ entegreli arazi mülkiyeti ve keşif sistemi gibi birkaç önemli IT projelerinde geliştirme ve test görevini yöneten Richard, işine güçlü bir teknik deneyimi sunmaktadır. Mobil geliştirme hakkında yazmadığı zamanlarda Richard, Wellington, Yeni Zelanda'daki arazisinde doğal çalılar yetiştirmektedir.

Davoc Bradley / Rivo Software

Devoc 1999 yılından beri yazılım mühendisi olarak çalışıyor. Yüksek kullanım oranına sahip web ve mobil sistemlerin yapısı ve tasarımı konusunda uzmanlaşıyor. Yakın zamanda ödül kazanan Mobil Uygulama Yönetim sisteminin mimarisi, tasarımı ve geliştirmesi ile uğraşmaktaydı. Şu anda Rivo Software'in Teknik Mimarı. Davoc ayrıca istekli bir müzisyen, sıkı bir kriket hayranı ve seyahat etmeyi çok seviyor

 Twitter: @davocbradley www.rivosoftware.co.uk

Marco Büttner / SciDev

25 yaşındaki Marco, 2011 yılından beri mobil geliştiriciliği yapıyor. Berlin Humboldt Üniversitesinde bilgisayar bilimleri okudu ve bada, Tizen ve gelişmekte olan diğer mobil platformlar için web uygulamaları geliştirmeye odaklanan SciDev mobil geliştirme projesini kurdu. bada ve Tizen topluluğu ile bağlantısını koruyor ve her zaman teknik bilgisini paylaşmaktan mutluluk duyuyor.

 Twitter: @scionbln www.scidev.eu

Dean Churchill / AT&T

Dean, AT&T'de uygulamaların güvenli şekilde tasarımı, geliştirilmesi ve test edilmesi konusunda çalışıyor. Son bir kaç yıldır, dahii AT&T mobil uygulamalarının yanı sıra tüketici uygulamaları için güvenlik gerekliliklerini yönlendirmeye odaklanıyor. Bu günlerde AT&T'nin Mobil Sağlık ve Dijital Yaşam ürün hatlarını desteklemekle meşgul. Seattle bölgesinde yaşıyor ve kayak ile uçurma balıkçılığını seviyor.

Julian Harty / Commercetest

Julian 2006 yılında, Google'ın mobil uygulamalarını test etmek için, ABD dışındaki ilk Test Mühendisi olarak Google'da çalışmaya başladı. Google'ın içerisinde ve dışında, çeşitli görevlerin nasıl gerçekleştirileceğine ilişkin yardımcı oldu ve konu hakkındaki ilk kitabı yazdı. Ardından, eBay için çalıştı. Burada, görevi küresel olarak test işlemini modernleştirmekti. Şu anda bağımsız olarak çalışıyor, mobil uygulamalar ve uygun test otomasyon araçları yazıyor ve diğerlerinin mobil uygulamalarını iyileştirmesine yardımcı oluyor. Ayrıca, mobil uygulamalar için test etme işlemi ve test otomasyonu konusunda yeni bir kitap yazıyor.

 Twitter: @julianharty

Bob Heubel / Immersion Corp.

Bob Heubel, Immersion Corporation'da çalışan bir dokunsal teknoloji uzmanı ve force-feedback, tactile-feedback or rumble-feedback efektleri olarak bilinen unsurlar hakkında geliştiricilere yardımcı olmaya odaklanmıştır. Dokunma Bilimi alanında birkaç adet patenti bulunmaktadır ve oyun ve etkileşim tecrübelerine dokunsal hissi tasarlayıp programlamak için on dört yıldan beri geliştiriciler, taşıyıcılar ve donanım OEM'leri ile beraber çalışmaktadır. Bob'ın bazı işlerini Android için Rockstar Games'in Grand Theft Auto: Vice City & Max Payne içerisinde görebilirsiniz.. Bob, 1989 yılında UC Berkeley İngiliz Edebiyatı Bölümünden mezun oldu.



Twitter: @bobheubel www.immersion.com

Ovidiu Iliescu / Enough Software

Ovidiu, uzun yıllar boyunca masa üstü ve web tabanlı uygulamalar geliştirdikten sonra, mobil yazılımın kendisine daha uygun olduğunu gördü. 2009 yılından beri Enough Software için Java ME ve Blackberry geliştirmelerine dahil oldu. Etkili kodlama, algoritmalar ve bilgisayar grafikleri ile ilgili her türlü işten heyecan duyuyor.



Twitter: @ovvyblabla www.enough.de www.oviduiiliescu.com

Alex Jonsson / EvoThings

Alex; uygulamalar ve web teknolojilerinin yanı sıra fiziksel şeyleri dijital şeylere bağlamak da dahil olmak üzere mobil olan her şeyi seviyor. Stockholm'deki Royal Institute of Technology'de medya Teknolojisi alanında doktora yaptı ve fikirleri ile görüşlerini hem sektör hem de akademisyenler ile serbestçe paylaşıyor. Dr Jonsson'ın ayrıca uygulamaların ve hizmetlerin yeni işletmeleri nasıl teşvik ettiklerini araştırmak gibi bir ilgi alanı bulunmakta. Bunların birbirleri ile nasıl çalıştığını araştırıyor ve böylece bu evrenin nasıl akıllıca ortaya çıktığını görmeyi seviyor. Alex, EvoThings'in kurucusu ve CTO'su çünkü bazı şeyler birbirine bağlı olduğunda daha güzeldir.



Twitter: @dr_alexj www.evthings.com

Matos Kapetanakis / VisionMobile

VisionMobile Pazarlama Müdürü olarak faaliyetleri: VisionMobile web sitesinin ve blogunun yönetilmesinin yanı sıra, şirket tarafından yayınlanan çizimler ve bilgi grafikleri için kavramlar ve pazarlama konularından sorumludur. Matos, Developer Economics araştırmalarının ve diğer geliştirme araştırması projelerinin de Proje Müdürü olarak çalışıyor. www.visionmobile.com

Michael Koch / Enough Software

1988 yılından beri yazılım geliştiren Michael, 2005 yılında Enough Software geliştirme ekibine katıldı. Burada CTO görevinde bulunuyor. Çok sayıda mobil uygulama geliştirme projesinde (başta Java ME, Android, Windows Mobile ve BlackBerry için) lider konumda bulundu ve bunun yanı sıra sunucu teknolojisi alanında uzmanlık yaptı. Michael, GNU classpath gibi pek çok serbest projeye açık kaynak tutkunu olarak dahil oldu.



Twitter: @linux_penguin www.enough.de

Daniel Kranz / Joule

Daniel, danışmanlık, temsilcilik ve teknik geçmişı olan çok kanallı bir stratejist. DAha önce önde gelen bir reklam ajansında teknik proje müdürlüğü ve çok kanallı web uzmanlığı görevini üstlenen Kranz şimdi, mobili genel stratejilerinin bir parçası olarak nasıl entegre edecekleri konusunda küresel stratejik planlama alanında markalara danışmanlık yapıyor.

www.jouleww.com

Carlo Longino / WIP

Carlo'nun mobil endüstride on yılı aşkın deneyimi bulunuyor. Bu deneyimi edinmeye, Nokia'nın Finlandiya'daki merkezinde çalıştığı zaman başladı. 2010 yılında, geliştirici pazarlama hizmetleri müdürü olarak Wireless Industry Partnership'e (WIP) katılmadan önce, MBA tamamladı ve serbest danışman ve yazar olarak çalıştı.

Bundan önce ise, Silikon Vadisi merkezli bir analist firması olan Floor64 için kıdemli analist olarak çalıştı. Burada mobil ve sabit telekom endüstrilerinde görev aldı. Ayrıca, Nokia'nın sahip olduğu bir düşünce liderlik sitesi olan TheFeature.com'u kuru ve beş sene boyunca işletti. Carlo ayrıca The Wall Street Journal, Business 2.0 ve Dow Jones Newswires'da yer aldı. . Bunun yanı sıra, Mobile World Congress, SXSW, MobileBeat ve CTIA gibi sektördeki çok sayıda etkinlikte konuşma yaptı.

 Twitter: @caaarlo www.wip.org

Tim Messerschmidt / PayPal

Tim, 2008 yılından beri Android uygulamaları geliştiriyor. İşletme enformatiği eğitimi aldıktan sonra, 2011 yılında Mobil Yazılım Geliştirici olarak Berlin merkezli Neofonie Mobile'a katıldı ve 2010 yılından beri Samsung Almanya için Andorid ve bada Geliştirici Destekçisi olarak danışmanlık yapıyor. 2012 yılında, Geliştirici Uzman olarak PayPal'e geçti. Genel olarak, mobil ödemeler, UI, UX ve Android geliştirme konusunda oldukça tutkulu. Ayrıca, konferanslarda konuşma yapmayı, makale yazmayı ve her tür sosyal medyayı seviyor.

 Twitter: @seraandroid & @PayPalDev www.timmesserschmidt.de

Patrick Mortara

Patrick, Frankfurt'ta bilgisayar bilimi eğitimi aldı. Doksanlı yılların ortasından beri, hem serbest olarak, hem de günlük işinde masa üstü tabanlı yazılım geliştiriyor. Mobil gelişme alanında çalışmaya 2010 yılında, Samsung bada'lı ilk akıllı telefonu Samsung Wave I'i piyasaya sürdüğünde başladı.

 Twitter: @pmortara www.mortara.org

Gary Readfern-Gray / RNIB

Gary Royal National Institute of the Blind için çalışan bir Erişilebilirlik uzmanı olarak çalışıyor. Yenilik Biriminde yer alan Gary, mobil boşluk ve özellikle geliştirici toplulukları ile çalışarak çok farklı platformlarda erişilebilir uygulama geliştirmesi konusuna odaklanmıştır.

www.rnib.org.uk

Alexander Repty

Alexander, 2004 yılından beri Mac OS X için yazılım geliştiriyor. 2008 yılında iPhone SDK piyasaya çıktığında, program için kayıt yaptıran ilk geliştiriciler arasındaydı. Enough Software'in bir çalışanı olarak, çok sayıda uygulamada çalıştı. Bunlardan biri, Apple TV reklamlarında ön plana çıkarıldı. iPhone gelişmeleri hakkında bir dizi makale yazdı. Nisan 2011 itibarıyla, bağımsız bir yazılım geliştiricisi ve yüklenici olarak kendi işine başladı.



Twitter: @arepty www.alexrepty.com

Marcus Ross

Marcus serbest çalışan bir geliştirici ve eğitimci. 10 yıl boyunca pek çok şirkette çalıştıktan sonra, şu anda SQL-BI projeleri ve mobil çapraz platformundaki her şeyi yapıyor. Aman dergisi "mobileWebDeveloper"da düzenli olarak yazarlık yapıyor. Boş zamanında, genellikle konferanslarda görülüyor, mobil konularda ve JavaScript konusunda konuşma yapıyor. Bunun yanı sıra mobil gelişme konusunda makaleler, kitaplar ve tweet'ler yazıyor.



Twitter: @zahlenhelfer www.zahlenhelfer-consulting.de

Michel Shuqair / AppValley

90'lı yıllarda, siyah ve beyaz WAP uygulamaları, iMode e SMS oyunları ile başlayan Michel, daha sonra m.wauwee.com mobil sosyal ağa liderlik yaptı. Yaklaşık 1.000.000 üyeye hizmet veren Michel, Amsterdam'daki merkezde bulunan Symbian, iPhone, BlackBerry ve Android uzmanlarından oluşan bir ekip tarafından destekleniyor. m.wauwee.com MobiLuck tarafından satın alındı.

www.appvalley.nl

Marco Tabor / Enough Software

Marco, Enough Software'de PR'dan, satıştan ve çok daha fazlasından sorumlu. Bu projeyi koordine ediyor, bunun yanı sıra sponsor bulma ve mobil topluluk tarafından sunulan girdileri birleştirme sorumluluğunu da üstleniyor.

 Twitter: @enoughmarco www.enough.de

Ian Thain / SAP

Ian SAP'de Mobil Evangelist, bu işe, 13 yıl önce Sybase Inc. ile başladı. İşletme için mobil bilgi ve deneyim sunarak tüm dünyadaki hedef kitleye düzenli olarak hitap ediyor. İşletmelerde Mobil olmak konusunda makaleler, bloglar ve tweet'ler yazıyor ve Kurumsal dünyada / İşletme dünyasında, Geliştirici ve Mobil Deneyim konusunda tutkulu şekilde çalışıyor.

 Twitter: @ithain scn.sap.com/blogs/ithain/ www.sap.com

Marc van 't Veer / Polteq

Marc, Polteq'de test danışmanı ve 7 yılı aşkın süredir koordinatör ve sistem testçisi olarak çalışıyor. Telekom, SOA, test otomasyonu, saptama modülleri ve sürücülerin geliştirilmesi ve API'lerin test edilmesi gibi teknik odaklı testler yapmada çok fazla deneyime sahip. Şu andaki işinde, büyük bir Flemenk süpermarket için tüm mobil uygulamaların testlerini koordine ediyor.

www.polteq.com

Robert Virkus / Enough Software

Robert 1998 yılından beri mobil alanında çalışıyor. Bütünleşik Java VM bulunan ilk kitle piyasa telefonu olan Siemens SL42i'deki mobil istemciyi geliştirip taşıırken, Java parçalanmasını ilk elden deneyimledi. Bu deneyimden sonra, 2004 yılında Open Source J2ME Polish projesini başlattı. J2ME Polish geliştiricilerin cihaz parçalamasının üstesinden gelmesine yardımcı oluyor. Robert, J2ME Polish'in, pek ok mobil uygulamanın ve bu kitabın arakasındaki firma olan Enough Software'in kurucusu ve CEO'sudur.

 Twitter: @robert_virkus www.enough.de www.j2mepolish.org



Proje hakkında bahsetmekten çekinmeyin.
Twitter'daki #mdgg etiketini kullanabilirsiniz.

Teşekkürler

Girişim



Sponsorlar:



**BU PROJE, GELİŞTİRİCİLER VE KARAR VERİCİLER
İÇİN MOBİL TEKNOLOJİLER ÜZERİNE TİCARİ
OLMAYAN, TOPLULUK GÜDÜMLÜ BİR ÇABADIR.**

Daniel Hudson, www.webtechman.com

Çok güzel bir iş ortaya çıktı! Rehberde açıklanan basit adımlar sayesinde mobil marketteki hakimiyetinizi ne kadar hızlı sağladığınızı gördüğünüzde şaşıracağınız.

Monika Lischke, Community Manager, Intel AppUp developer program
*Geliştirici olmayanlar için bile fevkalade yardımcı bir içerik olmuş.
Ve tasarımı, söylemeye bile gerek yok şahane olmuş!*