A SURVEY ON MOBILE MULTIMEDIA APPLICATION DEVELOPMENT FRAMEWORKS

Ramzi N. Sansour
Al-Quds University
rsansour2004@gmail.com

Nidal Kafri
Palestine Technical University-Kadoorie
nkafri@science.alquds.edu

Muath N. Sabha
Arab American University
msabha@aauj.edu

ABSTRACT

Although Smart-Phones are becoming major in computer systems nowadays, and as multimedia is the most important applications running on the mobile platforms, there is no clear branch for smart phone software engineering. In this paper, we make a survey on the latest and most well-known frameworks on mobile multimedia software engineering.

Categorizing the mobile application frameworks is also a main issue that will be discussed. The concentration will be on the cross-platform environments (tools) as the most important applications resulted from the market statistics.

KEYWORDS

SmartPhone, Multimedia, Framework, cross-platform, software engineering, mobile development, platform independent, agile.

1. INTRODUCTION

As the vast development in technology these days, it became difficult to dispense mobile devices as they achieve ubiquity because of their portability and availability.

Over time, the mobile phones, namely smartphones, have many functionalities over other computer systems, like portability, mobility, availability, flexibility, low weight and small size.

Smartphones provide advanced features that lack in older cell phones like the processing power; some are comparable to computer speed that one may think of it like a miniature computer, in addition to making calls. Smartphones allow users to perform ranging from multimedia to Web browsing. It includes network connectivity functionality as Wi-Fi, 3G or 4G, GPS, and other functionalities like multi-touch screens, gyroscope sensors, high-resolution cameras, sound systems and other features that are not there in old mobile phones.

Smartphones have different platforms, and the most well-known platforms are Android, iPhone iOS, Windows Phone, Blackberry, and Symbian OS. Developers have to build applications for each platform, which takes more time and money.

There is a difference between traditional software engineering and mobile software engineering, where there are additional characteristics in mobile computing, like power consumption, security, testing complexity, user interface, native and web application, and context awareness. On the other hand, there are different hardware characteristics, which are screen size, battery life, sensors, and network connectivity.

Developers began to produce applications for these devices to let users deal with them effectively. It became important to develop applications in line with the capabilities of the Smartphones. Although these devices match PCs, they have some drawbacks like battery drains significantly, if it is used frequently, or by using graphical applications. Another drawback is small screen, which has some limitations in applications.

1.1 Mobile Applications Categories

There are many kinds of application development on Smartphones, mainly Native applications, Web applications, and Hybrid applications. Native applications run entirely on the mobile device; deal with the mobile hardware directly and effectively. Web applications run on web browsers and most of them execute on a remote server. They run on multiple platforms but with less
efficiency. The Hybrid application is a mixture of the Native and the Web application. Since developers find it difficult to write the code on each platform, the cross-platform frameworks facilitate this issue by achieving the "write-once-deploy-everywhere" solution.

1.2 Multimedia

There are different categories of applications in smartphone stores, which opened a wide range of areas in Entertainment like Games, Services, Social, Educational Applications, and Critical applications like Business, most of them are multimedia applications. These applications are implemented in a different way from other computer systems. As mobile phones became more important, some companies force their employees to use these devices in their job, especially outdoor.

Technology in mobile development precedes research in many steps, because of the huge competition between mobile companies. Many researches describe frameworks for application mobile developing, which can be divided into several categories: Applications, Tools, or Guidance.

2. MOBILE SOFTWARE ENGINEERING

There are some immature researches in finding some software engineering roadmap development. Some of the authors like Buthpitiya et. al. [2010] considered that mobile phones and portable PCs with connectivity and sensing capabilities would achieve ubiquitous context-aware computing. They have highlighted some challenges face the mobile developers, which are mostly the limitations of the mobile device [6]. Biegel & Cahill [2004] early considered the context awareness and mobility are the main concepts for the ubiquitous computing. They developed a framework that eases the development of mobile context-aware applications, which make users build their own applications fast and easily [4]. Wasserman [2011] stated the fact of the accelerated growth of mobile applications, in addition to some characteristics of these applications namely small size, one or two developers, native and web applications, and the needed security. He also worked on the differences between mobile and traditional software engineering [32]. Bareiss & Sedano [2011] tried to find some special needs for mobile applications [3]. Carbon & Hess [2011] put some methods for developing a business mobile application, in addition to some guiding principles like user-centric, lightweight, iterative and integrative [7].

Some authors, Charland & Leroux [2011], compared the native and the web applications. They stated the problems that face the mobile vendors, especially the platform requirements and cross platform compatibility. They tried to distinguish between applications and could not recommend but hybrid applications [8]. On the other hand, Dehlinger & Dixon [2011] counted on the rapid development of the mobile platforms. They also categorized four mobile application challenges for software engineering, which are Creating Universal User Interfaces, Enabling Software Reuse across Mobile Platforms, Designing Context-Aware Mobile Applications, and Balancing Agility and Uncertainty in Requirements [11].

Rosa et. al. [2012] considered that mobile new features improved communication for users’ efficiency to access information by new approaches like social networks, blogs, and Web pages. GPS is used dramatically in applications, because of the existence of location-aware and context-aware technologies in mobile devices. The authors proposed a multimedia application called CityEvents, which follows location, context, and context-awareness approaches. This application allows users to have information about culture events with multimedia contents, anywhere and anytime showing information to the user on a table or on a map [25].
Ha et al. [2012] describe the benefits of cloud computing, in which it helps to improve processing, storage, and energy limits, as it will reduce latency significantly in the critical path of user interaction. Cloud can reduce the mobile energy consumption by the processor, memory and storage. They perform a set of experiments on some familiar applications (Face Recognition, Speech Recognition, Object and Pose Identification, Mobile Augmented Reality, and Physical Simulation and Rendering) for desktops, laptops, and Smartphones to prove the benefits of cloud computing [13].

3. MOBILE DEVELOPMENT FRAMEWORKS

Many researchers are interested in mobile applications development, in which they concentrated on the problems facing them, producing frameworks to meet the requirements of application development. Some of the researches focus on context-aware development to achieve ubiquity, as the existence of sensors, Wi-Fi, and GPS in the mobile devices that help users to benefit from these functionalities to get data from mobile environment. Other researches address providing tools helping developers in their application development. We can categorize the mobile development frameworks as follows:

3.1 Context-Aware Frameworks

Many researches wrote about ways for effective mobile applications, concentrating on getting data from mobile environment.

Buthpitiya et al. [2010] develop a framework for developing powerful Context-Aware applications. It is a tool that provides a protocol and ontology in which a widget can discover other widgets providing the context information it requires. Each widget ensures high level of security for the application, which is focused on authentication of other widgets, ensuring widget integrity, and securing communication between widgets. It also keeps track on the raw context information used to generate its output [6].

Biegel & Cahill [2004] developed a framework that eases the development of mobile context-aware applications, allowing the developers to collect and manage data from sensors, context, and reason about context. This framework fulfills the two major goals necessary for successful development of ubiquitous, context-aware applications, which are: 1- Ease of application designing, prototyping and testing. 2- Designers and end-users can build their own applications. It provides visual programming tool without the need to write a complex code [4].

Wissen et al. [2010] describe “ContextDroid”, a centralized framework which helps programmers to develop applications that deals with multiple context-aware, reduces development time, and putting more attention on the efficiency of the battery life [34].

Simon & Fröhlich [2007] presented an application framework that takes advantage from the web geospatial content from interaction of the user interface and the high end mobile phone devices. Its main goal is to reduce the inaccuracy of the GPS. It encourages developers to exploit the device capabilities [27].

Sz`ant´o [2010] extended the Java Context Awareness Framework to adapt to specific situation by making use of information, to help developing context aware applications. It aims to run on the Android platform by supporting the development of context-aware applications to solve part of the communication problems [31].

Cugola et al. [2011] describe the “SelfMotion” approach which allows to be modeled in terms of the functionalities they provide and the overall goal. Whenever a change happens in external environment, a middleware tries to find an alternative path to continue executing the application. The “SelfMotion” approach has activities at design-time which requires
intervention of domain experts and software engineers, and at run-time which are supported by a middleware [10].

3.2 Guidance Frameworks

Other researches describe frameworks as support and guidance for developing mobile applications.

Intel [2012] created a mobile application development framework as a set of specific capabilities, tools and resources to enable building successfully planned applications. This framework has been created for Intel’s employees’ expectations to use mobile devices for accessing web-based and native line-of-business applications in the corporate environment. It enables developers to provide feedback about what guidance worked best. This framework provides Guidance Documentation, Enabling Capabilities, and Supporting Resources [16].

Cheng & Yuan [2007] proposed designing and implementing a Generic Mobile Application (GMA) development framework. This framework is capable to fit different devices or situations according to mobile user interface formats, computing power and functionality. Weak devices will be supported by a server to function jobs they were not able to do before. GMA has three-tier architecture in which the GMApp is designed to run depending on the computing power and functionalities. The three tiers are front-tier (Standalone mode: Computing power is good enough; the device supports all application functionalities), middle-tier (Browser mode: Computing power is not good; the device cannot run other than built-in applications), or both tiers (Master-slave mode: Computer power is good enough, but the device cannot support all application functionalities). The third tier is the backend-tier [9].

Sierra Systems [2011] provides guidance on how to select mobile development framework to help reuse of existing enterprise assets in web applications. The key requirements of how developers choose architecture are Audience (Enterprise mobile applications for employees). Devices (The capability of the employees’ devices). Legacy Enterprise Apps (Enterprise application use centralized data, using database model). Usability and Security (If both usability and security are important, it is recommended to implement the Mobile Devices Management MDM product). Cost (Developers, Tools, and Runtimes are the cost of developing an enterprise application). Fine Tuning Apps (Platform differences, and achieve look and feel, responsiveness, and battery life impact). Finally Performance (Any achievement of the previous requirements may introduce an overhead, which affect performance) [26].

Smith [2012] presents the AVIARC Framework, which is a declarative application framework. It enables developers to implement browser-based applications quickly by providing a toolset to allow extracting application requirements quickly taking the advantage of the user experience. AVIARC interfaces with any data service (data-centric systems). In AVIARC, there is not any need to address the implementation details of the Web. Developers in AVIARC do not need to write code to address the nature of the Web [29].

3.3 Cross-Platform Frameworks

Other researches make a comparison and description for Cross-platform frameworks as tools for developing successful mobile applications.

Oracle [2011] created the Oracle Application Development Framework Mobile, which makes developers quickly develop applications for multiple mobile platforms. It provides tools, services, and infrastructure as enterprise/internet applications to mobile clients. The programming model is web-based, which offers consistency and migration to new platforms. The application can access the device services, offering more experience for users than the browser can offer. Developers with Oracle ADF Mobile only need to
write the code once and then deploy it across multiple platforms [22].

Singh & Palmieri [2011] made a comparison between four major available cross-platform tools in the market, which are: Rhomobile which aims to manage enterprise applications. DragonRad which focuses on database driven mobile enterprise applications. PhoneGap is an open source framework that produces hybrid applications. MoSync is an open source for building all sizes of applications [28].

Microsoft [2011] presents two mobile applications frameworks Silverlight and XNA. The Silverlight Framework is used for building Native and web GUI applications on the Windows Phone operating system. It uses very rich user interface markup. The XNA Framework is an advanced graphics-focused framework. It enables creating 2D and 3D portable games. Silverlight and XNA applications require only a few small changes to run on multiple platforms [20].

Raj & Tolety [2012] divided the approaches for building the appropriate cross-platform mobile applications into four approaches according to construction and execution phases. Web Approach: The application is designed to be executed in the web browser and the data is server driven. Hybrid Approach: Is between web and native, using web technology and run inside native container. Interpreted Approach: The code is deployed to the mobile device and interpreted later at the runtime. And Cross Compiled Approach: Where the source code is converted to native binaries. The Cross Compiler generates the executable code for a particular platform [23].

3.4 Measurement Frameworks

Kim [2012] presented a model-based performance prediction at mobile software development time for project optimization. He presented the Goal-Question-Metric (GQM) as a top-down approach to establish a goal-driven measurement system for mobile software development. It focuses on its paradigm to support software development and maintenance. He also presented the Project Capability Model (PCM) framework, which measures an organization’s capability through completion and analysis of questionnaire. He used it to measure capability in terms of external effectiveness and efficiency of an organization [19].

From the above mentioned frameworks, we can see that each of them tried to solve a problem in mobile application development. Table 1 illustrates the good features in those frameworks.

4. CROSS-PLATFORM MOBILE DEVELOPING TOOLS

To support multiple OSs, Cross-Platform mobile development tools are used to deploy native applications for many OSs.

In this survey, we will explain many major available cross-platforms tools, which are Rhodes, PhoneGap, DragonRad, MoSync, Appcelerator Titanium, Sencha Touch 2, jQuery Mobile, Xamarin, Unity3D and Corona SDK. These mobile application development tools make it easy to develop an application and save time and requires less coding.

4.1 Rhodes

RhoMobile Rhodes is an open source framework for cross-platform Smartphone applications, developed by RhoMobile. It aims to manage enterprise application and data, and to provide a high level productivity and web programming portability. It is composed of many products, like Rhodes (Develop), RhoConnect (Integrate), Rhohub (Deploy) and RhoGallery (Manage). It can be used across Linux, Mac and Windows. RhoMobile Current application data remains on users’ device by a standalone server (RhoSync) which is provided by RhoMobile in addition to a hosted development environment (RhoHub) [28] [24] [21].
It provides an (IDE) called RhoStudio. The application can be written with any editor supports HTML and Ruby like Eclipse, Visual Studio, Netbeans, IntelliJ, Textmate and other [28] [24].

The supported platforms are Android, BlackBerry, iOS, Windows Phone and Symbian [28] [24].

4.2 PhoneGap

PhoneGap is an open-source framework for developing mobile applications. Nitobi Software under MIT License develops it. The applications developed by PhoneGap are hybrid and created by HTML5, CSS3 and JavaScript. It targets mainly web developers. It does not provide a unique IDE; so developers should execute the source code on each IDE (i.e. Eclipse for Android and XCode for iPhone). It provides a PhoneGap Build for compiling applications in the cloud [28] [24].

PhoneGap makes it easy to work with a short span of time, without the need of maintaining native SDK [46].

The supported platforms are Android, iOS, webOS, Windows Phone, Symbian, Blackberry and Bada, [28] [24].

4.3 DragonRad

Seregon develops DragonRad [24]. It focuses on database driven mobile enterprise applications with a WYSIWYG tool, which provides the drag and drop visual environment GUI and helps developers to create logics. Its application is compiled into Lua byte code. It supports several databases such as MySQL, Oracle or SQL Server. The application is written in D and D programming language. It has its own IDE, which is a DragonRad Designer [28] [24].

The supported platforms are Android, Windows Mobile, and Blackberry [28] [24].

4.4 MoSync

MoSync is an open source tool that helps developer to build all types of applications, simple, advanced and complex, sharing the same code base. The application is created by C++ and Java. It provides the full fledge Eclipse-based IDE with the use of standard C/C++ [28].

MoSync application is a native application, where a service layer supports many functions like file I/O, threading, networking, memory management, and other functions. It provides an IDE based on Eclipse [28]. Developers can add a map to their application by the MoSync Widget C API’s Map, available only for Windows Phone 7 and iOS [30].

The supported platforms are iOS, Windows Phone, Android, JavaME, BlackBerry and Symbian [28] [36].

4.5 Appcelerator Titanium

Appcelerator is an opensource platform for developing mobile applications using web technologies. It links JavaScript to native libraries. The applications are written in HTML, JavaScript and CSS. It has development tools that support PHP, Ruby and Python. The output application is a native code. It provides an IDE based on Eclipse called Titanium Studio. Appcelerator uses native UI and platform APIs [24], but to provide UI it does not use a browser engine on the device [15]. Data can be stored either in the cloud or on the device [21].

The supported platforms are iOS, Android and BlackBerry [24].

4.6 Sencha Touch 2

Sencha Touch 2 is a high performance open source framework for mobile applications. It enables developers to build fast applications. The output application is hybrid, written by HTML5, CSS3 and JavaScript. It takes advantage of hardware acceleration [45]. A web server is needed to run locally for developing applications [12].

The supported platforms are iOS, Android, BlackBerry, Kindle and Bada, with a free
commercial license for application development, and a paid commercial license for OEM uses [45].

4.7 jQuery Mobile

jQuery is a unified system for all popular mobile device platforms that is an HTML5-based user interface. The code is lightweight and built with progressive enhancement, and has a flexible design. It is built on the rock-solid jQuery and jQuery UI foundation. It adopts the “write less, do more” concept. It does not create native applications. It has a broad support for the vast majority of all Smartphones, feature phones and older browsers. The application is written in HTML5, CSS [43].

The supported platforms are iOS, Android, BlackBerry, Windows Phone, Bada, palm WebOS, Symbian and MeeGo, with free under the MIT and GPL license [12] [43].

4.8 Xamarin

Xamarin allows developing cross-platform applications written in C#. It produces a native application or integrated .NET application. It provides IDE (MonoDevelop IDE and Xamarin plug-in for Visual Studio). It is not necessary to have XCode IDE installed to develop for iOS when using Xamarin, as it integrates with XCode Interface Builder [48] [35]. Java SDK must not be installed to develop for Android when using Xamarin, it provides some of the functionality familiar to Visual Studio developers [2].

On each platform the application user interface uses native controls, taking advantage of native UI toolkits [35]. With Xamarin context-sensitive template, developers can edit and restructure their code [48].

The supported platforms are iOS, Android, and Windows Phone. [48].

4.9 Unity 3D

Unity 3D is a cross platform 3D game engine, focusing on asset centric as a 3D modeling application. Developers can make motion applications (i.e. games) using Unity 3D engine. The application can be written in C#, JavaScript and Boo [33] [1].

Unity 3D uses 3D modeling tools models and game objects like Maya, 3DS Max, etc. With Unity 3D, developers can access different motion sensors of mobile devices from run-time classes. Unity 3D engine works on Mac OS X and Windows [33] [17]. Unity 3D tool provides a visual aid that reduces the amount of time for coding, as developers can simply drag codes from one object to another [1].

The supported platforms are iOS, Android, and Windows Phone [33]. It offers a 30 days trial for iPhone users [17] [1].

4.10 Corona SDK

Corona SDK is the leading mobile development cross-platform framework for building rich interactive applications, based on the Lua scripting language. Building application in Corona SDK is easy and quick in a text editor [17] [1]. Also developers can add features like Facebook and physics by writing a very short code, with Corona APIs [1] [37]. Corona uses HTML5 compliant web views, combined with OpenGL-based graphics [37].

The supported platforms are iOS, Android, Kindle Fire and NOOK. Corona tool is a free unlimited trial, but to publish the application to the App Store and Android marketplace, developers have to pay $199/year for the license [17] [1].

Table 2 illustrates a comparison between Cross-Platform Mobile Development Tools.

5. STATISTICS

On February 28, 2013, the total number of applications in the Apple App Store was 803137 applications [42]. On February 27, 2013, the total number of applications in Android market was 652659 applications [44]. Microsoft published
126,530 applications for Windows Phone in the same period [5]. The BlackBerry10 applications reached around 70000 applications [14]. Table 3 and Figure 1 shows a list of categorized applications for largest three platforms market, the Apple App Store, the Android Store, and the Windows Phone Store [42] [44] [5]. Figure 2 illustrates the worldwide Smartphones sales to end users by Operating Systems in 2011, 2012, and 1Q/2013 [39] [40] [41] [38]. Figure 3 illustrates the worldwide Mobile Phone Sales to end users by Vendor in 2011, 2012, and 1Q/2013 [39] [40] [41] [38] [47].

CONCLUSIONS AND FUTURE WORK
We have discussed a survey on the software engineering frameworks. Statistics on smart phone platforms, application categories, brands and some others.

The mobile multimedia application development frameworks are not mature neither well-defined. Some work is needed to well define a mobile software engineering. We will go on defining a clear software engineering for the mobile application development, even if it is a modification on a software engineering paradigm.

We will concentrate on the multimedia multiplatform application development framework.
7. REFERENCES


Appendix Table 1: Good features in Mobile Development Frameworks.

<table>
<thead>
<tr>
<th>Frameworks</th>
<th>The good feature</th>
</tr>
</thead>
</table>
| Biegl & Cahill [2004]       | • Allowing the developers to collect and manage data from sensors, context, and reason about context.  
|                             | • Achieve ease of context-aware application designing, prototyping and testing. |
| Buthpitiya et. al. [2010]   | • Reduces communication costs.                                                
|                             | • The widgets ensure high level of security.                                    
|                             | • Addresses the mobile environment challenges.                                |
| Wissen et. al. [2010]       | • Helps programmers to develop applications that deal with multiple context-aware.  
|                             | • Reduces development time, and puts more attention on the efficiency of the battery life. |
| Simon & Fröhlich [2007]     | • Reduce the inaccuracy of the GPS.                                             
|                             | • Encourages developers to exploit the device capabilities to provide an enhanced user experience.  
|                             | • Provides a single, unified data output format that is suitable for all mobile devices. |
| Sz´ant´o [2010]             | • Helps developing context aware applications.                                
|                             | • Solves part of the communication problems.                                   |
| Cugola et. al. [2011]       | • Allows the mobile applications to be adaptive to cope with the hardware.   
|                             | • Providing an effective self-healing behavior.                                |
|                             | 37. “Corona SDK – The World #1 Mobile Development Platform.”,                  |
|                             | 39. “Gartner Says Worldwide Sales of Mobile Phones Declined 2 Percent in First Quarter of 2012; Previous Year-over-Year Decline Occurred in Second Quarter of 2009”,  
|                             | 42. “iOS development news and information for the community, by the community”,  
|                             | 43. “jQuery Mobile: Touch-Optimized Web Framework for Smartphones & Tablets”,  
|                             | 46. “Take the pain out of developing mobile apps”,  
|                             | 48. “Xamarin - Create iOS, Android and Mac apps in C#”,  
Intel [2012]  
- Enables developers to provide feedback about what guidance worked best.
- Offers technical and project management tasks.
- Enables mobile devices to connect to the enterprise services.
- Helps navigate the unique aspects of developing for the mobile environment.

Cheng & Yuan [2007]  
- Is capable to fit different devices according to mobile UI formats, computing power and functionality.
- Supports weak devices by a server to function jobs they were not able to do before.

Oracle [2011]  
- Makes developers quickly develop applications for multiple mobile platforms.
- Developers only need to write the code once and then deploy it across multiple platforms.
- Ensures consistent application performance.

Sierra Systems [2011]  
- Provides guidance on how to select mobile development frameworks.
- Helps reuse of existing enterprise assets in web applications.
- Saves time at the outset of mobile application projects.

Smith [2012]  
- Enables developers to implement browser-based applications quickly.
- Developers do not need to write code to address the nature of the Web.

Microsoft [2011]  
- Silverlight and XNA are for building GUI applications.
- The applications are portable in both Silverlight and XNA.
- Silverlight uses very rich user interface markup.
- XNA is more graphics-focused framework.

Raj & Tolety [2012]  
- Classification cross platform development approaches into four approaches to bring awareness for vendors to choose the right approach.

Kim [2012]  
- Measures capability in terms of external effectiveness and efficiency of an organization.

### Appendix Table 2: Cross-Platform Mobile Development Tools Comparison.

<table>
<thead>
<tr>
<th>Cross-Platform Tools</th>
<th>Supported Platforms</th>
<th>Programming Language</th>
<th>Development Environment</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhoneGap</td>
<td>Android, iOS, webOS, Windows Phone, Symbian, BlackBerry and Bada</td>
<td>HTML5, CSS3 and JavaScript.</td>
<td>Eclipse, and XCode.</td>
<td>Open-Source</td>
</tr>
<tr>
<td>DragonRad</td>
<td>Android, Windows Mobile, and BlackBerry.</td>
<td>D And D</td>
<td>DragonRad Designer</td>
<td>Paid</td>
</tr>
<tr>
<td>MoSync</td>
<td>iOS, Windows Phone, Android, JavaME, BlackBerry and Symbian.</td>
<td>C, C++, and Java</td>
<td>Eclipse-based</td>
<td>Open-Source</td>
</tr>
<tr>
<td>Appcelerator Titanium</td>
<td>iOS, Android and BlackBerry.</td>
<td>HTML, CSS and JavaScript.</td>
<td>Eclipse-based Titanium Studio.</td>
<td>Open-Source</td>
</tr>
<tr>
<td>Sencha Touch 2</td>
<td>iOS, Android, BlackBerry, Kindle and Bada.</td>
<td>HTML5, CSS3 and JavaScript.</td>
<td>Eclipse, and Netbeans</td>
<td>Free license for development, and a paid license for OEM.</td>
</tr>
<tr>
<td>jQuery Mobile</td>
<td>Android, BlackBerry, Windows Phone, Bada, palm WebOS, Symbian and MeeGo.</td>
<td>HTML5, CSS.</td>
<td>jReply</td>
<td>Free license.</td>
</tr>
<tr>
<td>Xamarin</td>
<td>iOS, Android, and Windows Phone.</td>
<td>C#</td>
<td>MonoDevelop IDE</td>
<td>30 days trial.</td>
</tr>
<tr>
<td>Unity 3D</td>
<td>iOS, Android, and Windows Phone</td>
<td>C#, JavaScript and Boo.</td>
<td>Maya, 3DS Max.</td>
<td>30 days trial.</td>
</tr>
<tr>
<td>Corona SDK</td>
<td>iOS, Android, Kindle Fire and NOOK.</td>
<td>HTML5.</td>
<td>Lua scripting language.</td>
<td>Free unlimited trial, paid to publish.</td>
</tr>
</tbody>
</table>