LITERATURE REVIEW: MOBILE GAME DEVELOPMENT

Alex Koller
Department of Computer Science
Supervisors: Prof G. Foster and Mrs M. Wright
Rhodes University
26th June 2007
Grahamstown, South Africa

1 Introduction

Recent years have seen the rise of mobile users with PDAs, laptops and particularly mobile phones. Worldwide there are over 2 billion mobile phone users [1] – roughly one third of the world's population. In South Africa alone there are 30 million mobile phone users [1] out of a population of 47 million. This pervasiveness of mobile devices has created a substantial market for mobile applications and games that continues to grow extremely rapidly.

With so many mobile devices, it is crucial for a developer to target the right mobile platform and development environment, or a combination thereof. Portability across platforms will ensure the maximum target audience is reached. This is specifically important for mobile game development.

The mobile game industry has firmly asserted itself as a significant industry with major development houses such as EA Mobile driving it forward. As such, it is possible to make some observations about the mobile game industry. Primarily, we will analyse which games have succeeded and which have failed, and determine why. In addition to the nature of the game, user input and hardware limitations also affect the success of a mobile game.

A brief, yet concise, discussion will be conducted of two of the main, and current, environments for mobile game development will be conducted. The two environments under discussion are Java ME and Flash Lite. The APIs and features of these two development environments are explored in detail. Methods of deploying applications, as well as the future of Java ME and Flash Lite will be discussed.

Finally, means of comparing – quantitatively and qualitatively – the development of mobile games on Flash Lite and Java ME will be discussed. This section looks at both the development phase, and the final game product of each environment.

2 Mobile Environments

With over 2 billion mobile phone users worldwide [1], there is a plethora of different mobile devices available, which operate on different platforms and have support for various application environments. There is significant variety among mobile devices both in terms of hardware capabilities and software support.

According to Canalys [2], 64 million smart phones were shipped worldwide in 2006 – a 30% increase from 2005. This suggests the smart phone, or high-end device, market is growing rapidly and as a result the availability of high-end devices is increasing.

2.1 Mobile Platforms and Application Environments

On high-end devices, such as smart phones, there are various platforms or operating systems.

Common Smart Phone Platforms

Symbian OS has been shipped to over 80 million smart phones so far and has an overall 67% smart phone OS market share [3]. Morales and Nelson [4] state applications for this platform can be developed in C/C++ to be run natively on the phone.

Microsoft Windows Mobile is an operating system developed by Microsoft, which offer tight integration with Microsoft products such as Microsoft Office and Windows Media Player. However, Windows Mobile has a low market penetration, according to Canalys [2].

Palm OS is developed by Palm for smart phones and PDAs. Like Windows Mobile, it is licensed to several other manufacturers, according to Crooks [5].

Applications can be developed natively for these platforms. However, to achieve greater portability, many applications are developed for a specific environment, which runs on top of the mobile device's native platform. Morales and Nelson [4] support this and suggest that portability is of great concern to mobile game developers in order to reach a larger target audience.

Available Environments

Java Platform, Micro Edition (Java ME), previously known as J2ME, was developed by Sun Microsystems. It is a subset of the Java platform aimed at mobile devices. Java ME is covered at length in section 4.

Flash Lite was originally developed by Macromedia as a lightweight version of their Flash Player for mobile devices. Adobe acquisitioned Macromedia in 2005, and now continue to develop Flash Lite. Flash Lite is covered extensively in section 5 of this paper.

In 2005 Nokia released PyS60, which is a Python interpreter for Symbian S60 devices. PyS60 includes many of the modules from regular Python implementations, as well as additional modules for interacting with the Symbian operating system. Python is often used in conjunction with other environments, such as Flash Lite or Java Me, to achieve a greater level of interaction with the underlying operating system.

.NET Compact Framework is a compact version of the Microsoft .NET Framework that is designed to run on mobile devices. The .NET Compact Framework includes some of the class libraries from the full framework. Applications can be created in Microsoft Visual Studio, and written in C# or Visual Basic .NET. These applications are run on a mobile-device specific just-in-time compiler.

2.2 Market Shares

As table 1 below shows, Nokia holds the largest share in the smart mobile device market with over 11.1 million shipments in Q4 of 2006. Sony Ericsson has the largest growth at 946.2%. Both Nokia and Sony Ericsson are shareholders of Symbian, ltd. [6].

Table 1: Worldwide Smart Phone Market [2]

Worldwide Total Smart Mobile Device Market **Market Shares Q4 2006, Q4 2005**					
Vendor	**Q4 2006 shipments**	**% share**	**Q4 2005 shipments**	**% share**	**Growth Q4'06/Q4'05**
Total	**22,124,400**	**100.0%**	**17,053,930**	**100.0%**	**29.7%**
Nokia	11,114,630	50.2%	9,268,410	54.3%	19.9%
RIM	1,829,260	8.3%	1,185,340	8.0%	54.3%
Motorola	1,463,090	6.6%	777,580	4.6%	88.2%
Palm	1,211,930	5.5%	1,563,680	9.2%	-22.5%
Sony Ericson	1,137,360	5.1%	108,710	0.6%	946.2%
Others	5,368,130	24.3%	4,150,210	24.3%	29.3%

Table 2 shows the distribution of operating systems among smart phones. Symbian has almost 70% market share with 12.7 million smart phones being shipped with Symbian OS. According to Symbian [6], cumulative sales of smart phones will pass 1 billion units by 2011, and that by 2010 Symbian will still be leading the market with a 65% share.

Table 2: Worldwide Smart Phone OS Market [3]

OS vendor	Q2 2005	% share	Q2 2006	% share	Growth Q2 05/Q2 06
Symbian	7,648,920	75.19%	12,720,920	69.7%	66.31%
Linux	1,448,320	14.24%	3,541,870	19.41%	144.55%
PalmSource	496,310	4.88%	562,960	3.08%	13.43%
Microsoft	355,650	3.5%	898,440	4.92%	152.62%
RIM	134,540	1.32%	475,860	2.61%	253.69%
Others	89,490	0.88%	51,360	0.28%	-42.61%
Total	**10,173,230**		**18,251,410**		**79.41%**

2.3 Game Development on Mobile Devices

Symbian OS is by far the most pervasive platform for smart phones [table 2]. However, this excludes low-end mobile phones. Crooks [5] suggests that, unlike desktop game developers, mobile game developers should not concentrate on a single platform or device, but rather target several in order to maximise their potential marketplace. In order to reach the largest possible audience, most mobile games have been developed for Java ME as it has the highest market penetration among mobile devices, according to Morales and Nelson [4], and Java ME can run on most platforms, including Symbian OS.

Rhodes [7] claims Flash Lite offers the ability to create games that are cross-platform compatible. Flash Lite 2.1 players are available for Symbian OS, BREW 2.x/3.x and Microsoft Windows Mobile 5 [8]. However, Suomela [9] has encountered problems where Flash Lite applications behave very differently on different devices.

Mobile games are developed in C/C++ for Symbian OS. These games are not as portable however, and thus there is not as large a market for them. By far the most prominent environments for mobile gaming are Java ME, and now potentially Flash Lite

3 Mobile Gaming

Mobile gaming started with Snake in 1997 on the monochrome display of a Nokia 6110 [11]. Since then, improvements in hardware and game development have altered the user experience dramatically. An understanding of who currently plays and purchases mobile games is vital when designing and developing mobile games.

3.1 Mobile Gamer Profile

According to de Boer *et al.* [10], mobile gamers can be divided into two groups; hardcore gamers and casual gamers, stating that on mobile devices, casual gamers "make up the vast majority of the total gamers". Graft [11] agrees with this distinction, but notes that there is a dedicated mobile game fan base within the estimated 2 billion cell phone users worldwide. A survey conducted by the Interactive Digital Software Association [12] states that 87% of the people asked said the played games for fun, and 74% said they wanted to be challenged.

Jacobs [13] writes on casual gaming: "I think mobile games have come of age," Bruce Gibson, Research Director at Juniper Research, said. "The casual games sector is going to be the market driver, even though it may not be at the leading edge of mobile games technology. Casual games make most use of the inherent advantages of the mobile platform..."

3.2 Successful Mobile Game Genres

This categorising of mobile gamers is useful in determining what types of games are successful on mobile devices.

Games can be divided into genres, and certain genres are better suited to mobile devices than others. De Boer *et al.* [10] categorise games into various genres and describe their suitability for mobile devices, specifically with Flash Lite in mind.

De Boer *et al.* [10] found that strategy and logic games, such as *Minesweeper*, work well on mobile devices, as well as role-playing, sport strategy and management games. Games that fall into the first-person shooter genre, were found to be "almost impossible" to create in Flash Lite [10]. Action games and 2D racing games were said to be technically possible with Flash Lite, but the processing power required to achieve the desired speed was the limiting factor in these types of games.

Telephia [14] published the following table, showing the penetration rate of various game types among mobile game players in the U.K. These penetration rates of the various game genres support what De Boer *et al.* stated in terms of strategy and logic games being successful. However, arcade style games have been more successful, while sports and racing games have been less successful.

Table 3: Penetration Rate per Game Type [14]

Game Type	Penetration Rate
1. Puzzle/Strategy	61%
2. Retro/Arcade	45%
3. Action/Adventure	42%
4. Card/Casino	39%
5. Trivia/Word	32%
6. Sports/Racing	30%

Telephia also found that the mobile games that generated the most revenue in 2006 were *Tetris*, *PacMan*, *Ms. PacMan*, and *Bejeweled*. Morales and Nelson [4] point out that these games existed prior to being ported to mobile devices, and as such you could assume the mechanics of the game had proven themselves already. Morales and Nelson [4] also point out that people who like playing these games enjoy the mechanics of the game regardless of the platform the game was being played on. These games all have simple user interfaces and controls, which are easily performed on cell phones.

3.3 The Future of Mobile Game Development

The future of mobile gaming definitely looks bright. According to Gartner [15] the worldwide mobile gaming revenue will grow from $2.9 billion in 2006 to $9.6 billion in 2011, and is on target to reach $4.3 billion in 2007, a 49.9 percent increase from 2006. Gartner accredit this increase mainly to the pervasiveness of mobile phones in many markets and the ease of game play on mobile phones.

All sources are in agreement that the future of mobile gaming is spectacularly bright, and will continue to grow. However, the mobile game development community is divided on whether devices will become more powerful, and as Graft [11] suggests, be able to play more complex and

demanding 3D games with hardware acceleration, or, as Crooks [5] suggests, whether other limiting factors, such as user input, will see games that rely on captivating game play and user interfaces suitable for cell phones. Morales and Nelson [4] write that most successful mobile games in recent years have been 2D games with simple user controls, but are confident that 3D games will become the predominant type of game on mobile platforms.

4 Java ME

Java Platform, Micro Edition (Java ME) was developed by Sun Microsystems and is a "collection of technologies and specifications that are designed for different parts of the small device market." [16]. It is essentially a subset of Java APIs for development of applications on mobile devices with limited resources, such as cell phones.

Java ME has a particularly high market penetration; according to Morales and Nelson [4], approximately 68% of mobile phones are Java ME-capable, which equates to more than 350 million Java ME-capable mobile devices worldwide. Until now Java ME has been considered the de facto for mobile application development due to this high market penetration and relative maturity of the platform.

Java ME was designed to use profiles and configurations to enable devices of varying ability to be able to run Java ME applications on the Kilobyte Virtual Machine (KVM), which is the micro version of the Java Virtual Machine (JVM).

There are currently two configurations available: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). Figure 1 illustrates how the CDC and the CLDC together make Java ME. The diagram also shows the Java ME architecture, and how it fits in the overall Java model.

The CLDC targets devices with very limited resources, such as cell phones, while the CDC is designed to target devices with slightly more resources, such as set-top boxes. The minimum requirements for the CLDC 1.1 are a 16 bit CPU, 160 KB memory available to the virtual machine, and a limited network connection.

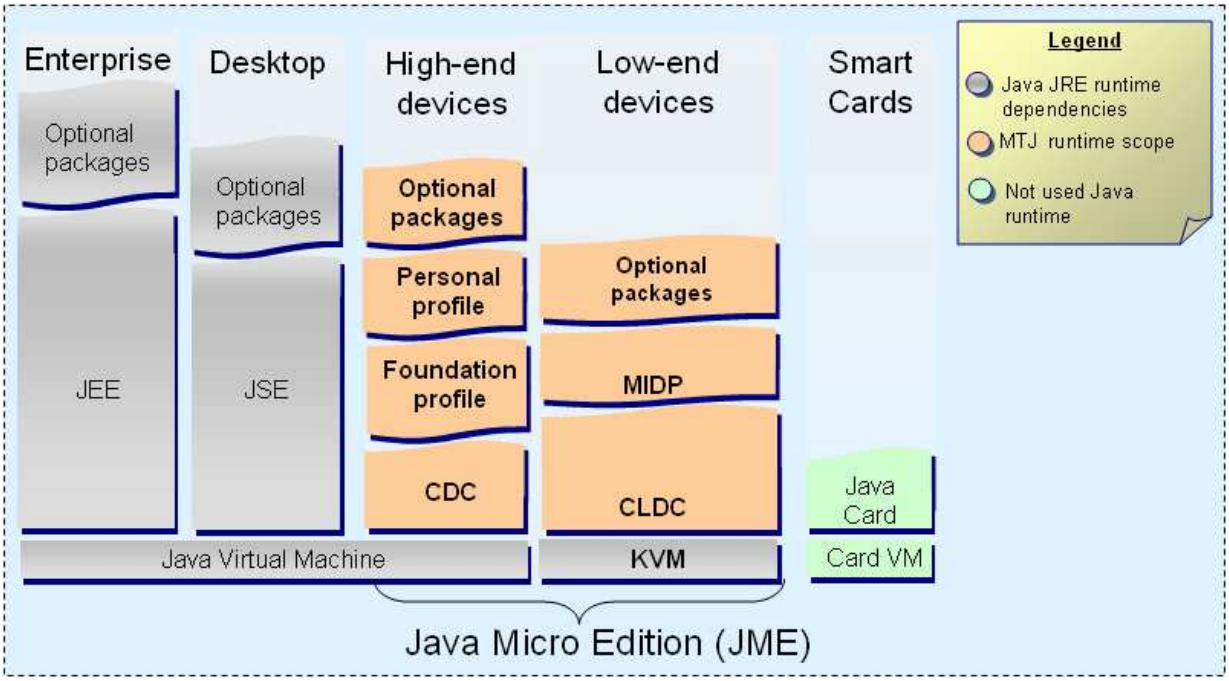

Figure 1: Java ME Architecture [17]

The most common profile is the Mobile Information Device Profile (MIDP), which is aimed at mobile devises such as cell phones. Currently MIDP 1.0 and MIDP 2.0 are available, while MIDP 3.0 is being developed. Java ME applications developed for an MID Profile are called MIDlets.

4.1 Java ME APIs

MIDP 1.0 offers the following APIs:

- **javax.microedition.io**

  Contains Java ME-specific classes used for I/O.
- **javax.mocroedition.lcdui**

  Contains Java ME-specific classes used for the GUI.
- **javax.microedition.rms**

  Provides a form of persistent storage for Java ME.
- **javax.microedition.midlet**

  Contains the base classes for Java ME applications.

MIDP 2.0 adds to following APIs:

- **javax.microedition.media**

  Contains classes used for multimedia playback.

- **javax.microedition.lcdui.game**
  A gaming API aimed at 2D sprite based game development. This API adds support for sprites, layers, and a tiled game canvas. These classes are used in most types of games.
- **javax.microedition.pki**
  Contains classes for authentication for secure connections.
- **javax.microedition.messaging**
  Wireless messaging API for sending SMS and MMS messages.
- **javax.microedition.pim**
  Personal information management API for accessing the address book.
- **javax.microedition.io.file**
  Provides access to the local file system.

Optional packages, such as Java APIs for Bluetooth, Mobile 3D Graphic, and JBDC can be added to MIDlets to offer greater functionality at the cost of being more device-specific. The ability to utilise these optional packages is where the maturity of Java ME is noticeable, with many APIs designed specifically for game development.

4.2 Deployment

A MIDlet is typically deployed as a single Java Archive (JAR) file containing all the class files and resources. A JAR file is used to contain compiled Java classes, associated metadata and resources pertaining to a certain application. Within the JAR file is the Manifest file which contains the metadata for the application. A Java Application Descriptor (JAD) file usually accompanies the JAR file. The contents of the JAD file is essentially the same as the Manifest file. The JAD file allows the user to view the details of a MIDlet without having to download the whole file.

The Over-The-Air distribution model allows MIDlets to be served from existing web servers, which can be accessed directly from cell phones though the device's built in web browser, if it has one, as shown in figure 2.

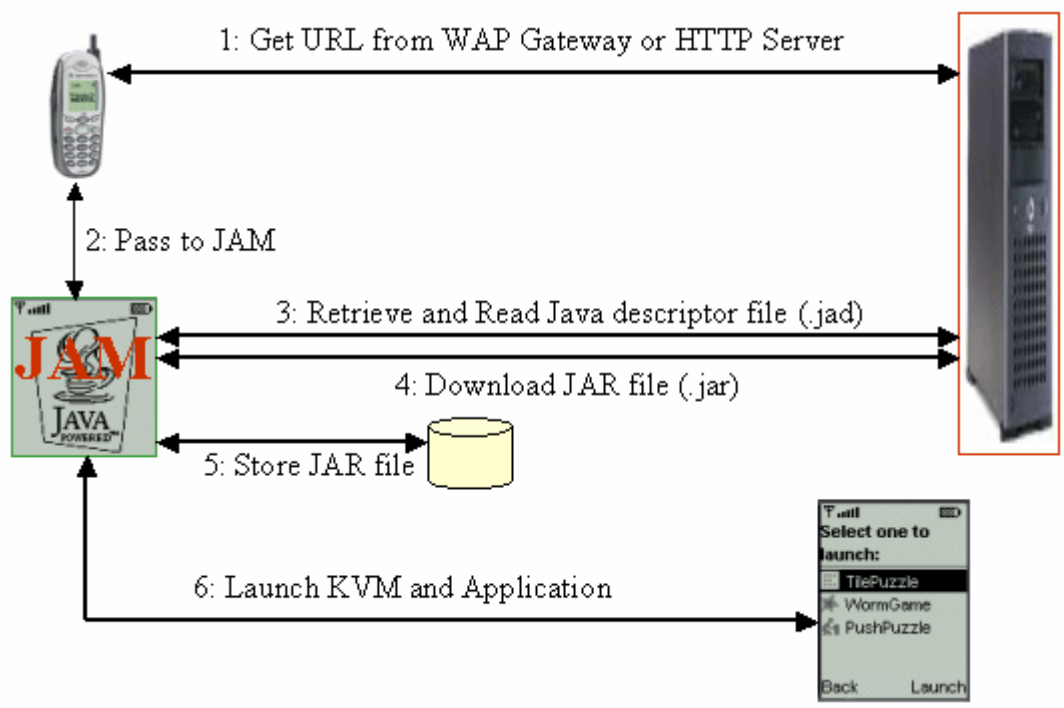

Figure 2: Over-The-Air Deployment [18]

4.3 Future of Java ME

According to Java Specification Request 271 [19], MIDP 3.0 plans to build upon the success of MIDP 2.0 by enhancing certain aspects of the profile and adding new features. Some of these enhancements include:

- Better support for devices with larger displays.
- Enable MIDlets to draw to secondary display(s).
- Enable richer and higher performance games.
- IPv6.
- Multiple network interfaces per device.
- Enable shared libraries for MIDlets.
- Enable background MIDlets (e.g. UI-less).
- Enable 'auto-launched' MIDlets (e.g. started at platform boot time).
- Enable inter-MIDlet communications.

With these enhancements and continued interest and support for Java ME, it is likely that Java ME will continue to be one of the main mobile game environments.

5 Adobe Flash Lite

Flash Lite is a lightweight version of Flash Player specifically developed for mobile devices. Flash has been an extremely successful technology for online games, which can be embedded in HTML web pages.

Flash Lite was originally developed by Macromedia before it was acquisitioned by Adobe in 2005. At that time Flash Lite had been available in Japan and Europe [20] as either Flash Lite 1.0 or Flash Lite 1.1. Later in 2005 Flash Lite 2.0 was released. Flash Lite 2.1 has been released, and version 3.0 is currently being developed.

5.1 Flash Lite APIs

Flash Lite 1.1 is based on a hybrid of Flash 4 and Flash 5, and supports Flash 4 ActionScript, which is a pre-ActionScript 1.0 version [21]. Flash Lite 2.1 is based on Flash 7 and has support for ActionScript 2.0. The main features of Flash Lite 2.1 are: [8]

- **ActionScript Extensions**
  Allow access to device specific features, such as battery level and signal strength. It also offers the ability to send SMS messages and dial numbers.
- **Network Access and Connectivity**
  This allows resources to be loaded via HTTP(S) requests, and email to be sent. It is also possible to load SWFs from web servers, allowing seamless updating of Flash Lite content.
- **Additional Audio Support**
  Support for MP3, PCM, and ADPCM audio. Hardware support for these formats is utilised when available.
- **Scaleable Rendering Engine**
  The core rendering engine of Flash Lite is well suited for rich interactive content as it supports vectors, gradients, bitmaps, user input, audio, and scripting.
- **Static, Dynamic Text and User Input**
  Flash Lite supports static and dynamic text fields. Dynamic text can be updated during runtime. User input is also supported during runtime.
- **Navigation and Key Events**
  Within an interactive movie, navigation can be achieved with the Up, Down, and Select buttons. The keys 0 to 9, * and # are also supported.

- **Fonts and Text**
  Flash Lite has support for both embedded fonts and device fonts. Embedded fonts allow for more control over the content, but increase the size of the final SWF file.
- **Multi-Platform support**
  Flash Lite 2.1 offers support across various platforms, including: Symbian S60 v2/v3, Qualcomm BREW 2.x/3.x and Microsoft Windows Mobile 5.
- **Dynamic XML data**
  Support for loading and parsing of external XML data.
- **Persistent data**
  Offers the ability to store and retrieve relevant, application-specific information locally. This provides a more robust development environment. User preferences, high scores, etc. can be stored.
- **Shape drawing ActionScript API**
  Enables developers to easily create sophisticated vector graphics and animated shapes, at runtime, using ActionScript 2.0.
- **Compressed SWFs**
  Flash Lite 2.0 supports the rendering of SWFs that have been compressed by the Flash authoring tool. Flash Lite decompresses the SWF before it starts processing it.

Unlike Java ME, none of these APIs are designed specifically for game development. Notable APIs that are missing include support for camera operation, Bluetooth communication, and 3D graphics.

Figure 3 shows the architecture of Flash Lite 2.1, and how the different components communicate with each other, as well as with the underlying operating system.

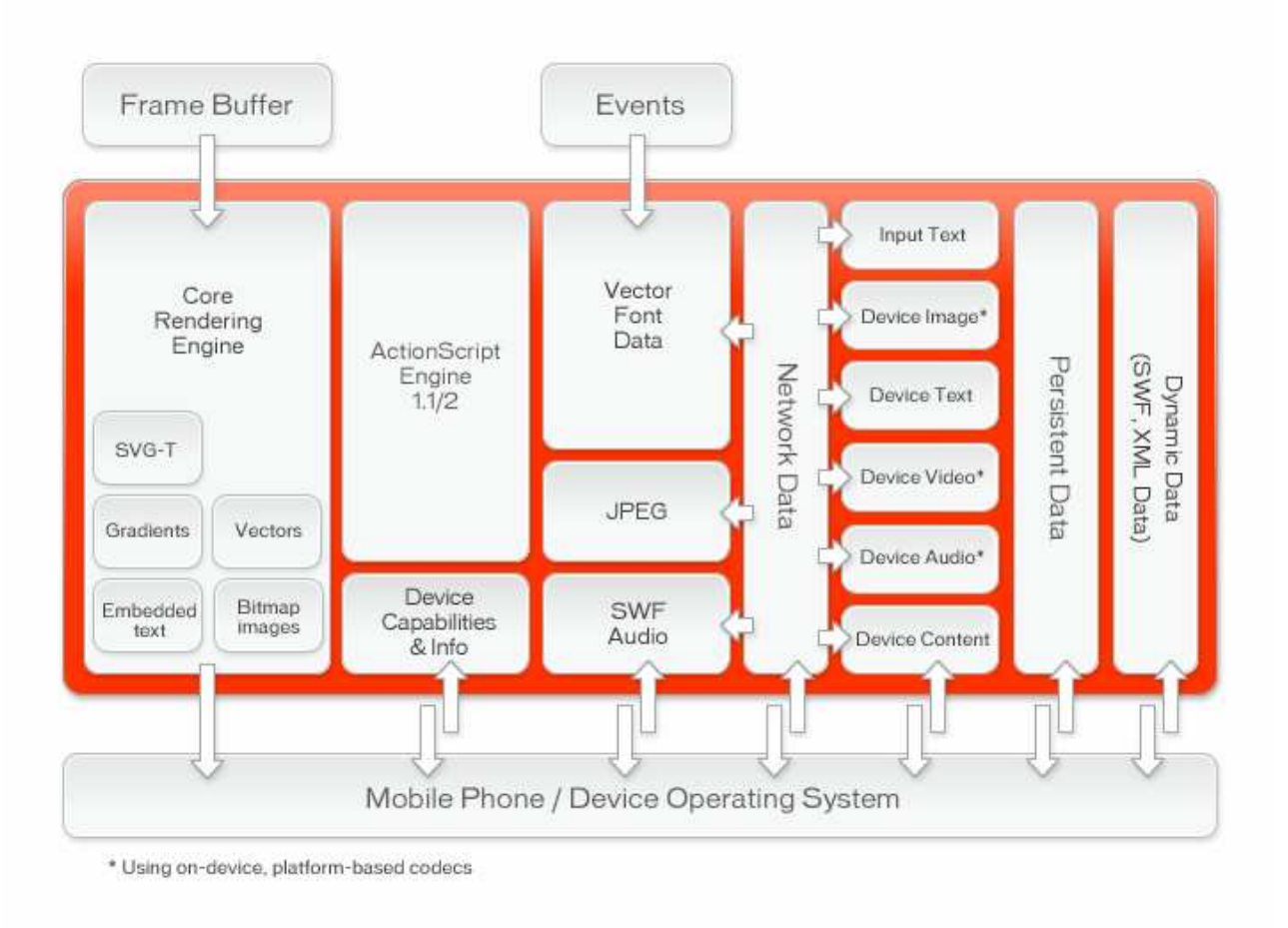

Figure 3: Flash Lite 2.0 Architecture [22]

5.2 Deployment

Flash Lite applications are deployed as single SWF files. As of Flash Lite 2.0, compressed SWF files may be used. The SWF file can be transferred to the mobile device using Over-The-Air, Bluetooth, or any other means of transferring data to the device. The SWF file is then launched from the Flash Lite player on the device. This is different to Java deployment in so much that Java ME applications consist of a JAD file which describes the application, and contains the location of the application JAR file. While this offers the ability for users to screen Java ME applications easily without downloading them, it is more complex than the single SWF file for Flash Lite applications.

5.3 Future of Flash Lite

According to Adobe [23], Flash Lite 3.0 is currently under development and is expected to be available “in the first half of 2007”. Adobe announced that support for video will be integrated into the next generation of Flash Lite.

Nokia Forum [24], refers to Flash Lite as "an emerging as a global phenomenon, enabling developers to deliver rich multimedia content to Nokia devices with less effort." The ease-of-use, short development time, and high return on investment are key factors that set it apart from other development platforms, specifically Java ME. According to Adobe [25], there are over 1 million Flash Lite developers, and the community is growing rapidly. This indicates a large scale and quick uptake of Flash Lite as a development platform for mobile devices.

6 Comparison Matrices

Quantitative and qualitative comparisons must be made between both the development phase, and the final product on both platforms.

6.1 Comparing Development

During the development phase comparisons should concentrate on the development environments, emulators, and development time. While comparing various runtimes on Symbian S60 devices, Suomela [9] compared the tools available and the development time for each runtime. Development time is dependant on many factors such as the developer's experience developing on the given platform. However, the tools available, and the development workflow used effect the development time. Suomela found, as a general rule of thumb, development is quicker on Flash Lite than Java ME, especially for non-programmers. Table 4 shows Suomela's comparison of available development tools.

Table 4: Comparison of Development Tools. Adapted from [9]

	Java ME	**Flash Lite**
Developer platforms	Windows	Windows, Mac
Tool cost	Free	$700 (free 30-day trial)
Getting started	JDK, IDE (Eclipse + Carbide.j or NetBeans), platform SDK	Adobe Flash CS3 Professional
Debugging and testing	SDK/Emulators, On-Device-Debugging	Flash/CS3 Device Central Emulator
Optimisations, etc.	Obfuscation	Decompilers

A closer comparison of the development environments can be done, contrasting the development workflows used, documentation available, and integration with other tools such as compilers and emulators.

6.2 Comparing the Final Product

Comparing the performance of the final products can be done in a range of ways. Depending on the type of game, the frame rate may be an important factor [26]. To benchmark the performance of each platform, an application that solves the same problem using the same general algorithm should be developed for each platform, according to Marne [26]. Using these two applications, it is possible to test the performance in a number of ways, including:

- **Lines of code**
  The total number of lines will be indicative of the amount of coding effort that was required to develop the project.
- **Speed**
  Measured in frames per second. For applications that require fast rendering, such as 3D games, this will give a good indication of how well the platform draw frames. A low frame rate makes games "jerky", or even unplayable. A high frame rate creates a smooth game play experience for the user.
- **Controls**
  Which controls are available to the user will determine how easy it is to navigate or control a game. This will have significant affects on the overall game play experience. The control set can be limited by the platform, such as Flash Lite only allowing two "soft buttons" to be used, or by the device itself.
- **Ease of use**
  If a game is difficult to use, understand, play, or navigate, this immediately has a negative effect on the user's experience. Ease of use also applies to the development environment and process.

These tests should be conducted on the same mobile device, and under as similar as possible conditions. This will result in the most useful quantitative comparison data. Comparisons of the controls and ease of use will result in qualitative comparison data, which can have major ramifications on the overall experience offered to the user by the game.

7 Conclusion

In this paper, we discussed mobile environments – the platforms and application environments – and how, for mobile game development, portability is the key to reaching the largest possible market. Java ME and Flash Lite are both well suited to mobile game development as they are both portable, and both environments are supported on many platforms, specifically Symbian OS.

It was found that certain game genres have achieved more success than others in mobile gaming. This can be attributed mainly to the limited resources available on mobile devices, as well limited user input through the small keys on mobile phones. Games that have achieved success have been easily controllable and relatively slow in nature, e.g. puzzle and strategy games.

The specifics of both Java ME and Flash Lite were discussed. Details of the APIs available in each development environment were listed. Methods of deploying applications, and future developments of both environments were also considered. The game development-specific optional packages available to Java ME demonstrate the maturity of the platform, as well as the large community support base for Java ME game development, while Flash Lite only has APIs for general application development, and has no APIs specifically for game development.

Methods for comparing mobile game development on Java ME and Flash Lite were investigated. Means of comparing the development phase and the final products – both quantitatively and qualitatively – were determined. This will enable us to give a meaningful comparison of mobile game development on Java ME and Flash Lite.

References

[1] Cellular Online, http://www.cellular.co.za/stats/stats-main.htm, 2007.

[2] Canalys, *64 million smart phones shipped worldwide in 2006*, http://www.canalys.com/pr/2007/r2007024.htm, 2007.

[3] Wei, C., *Mobile 2.0*, http://www.coachwei.com/blog/_archives/2006/11/4/2474357.html, 2007.

[4] Morales, C. and Nelson, D., *Mobile 3D Game Development: From Start to Market*, Charles River Media, Boston, MA, USA, 2007.

[5] Crooks II, C. E., *Mobile Device Game Development*, Charles River Media, Boston, MA, USA, 2007

[6] Symbian, *Fast Facts*, http://www.symbian.com/about/fastfacts/fastfacts.html, 2007.

[7] Rhodes, G., *Macromedia Flash 8 Professional Game Development*, Charles River Media, Boston, MA, USA, 2007.

[8] Adobe, *Flash Lite 2.x Features*, http://www.adobe.com/products/flashlite/productinfo/features/, 2007.

[9] Suomela, H., *Runtime Environments on S60 Platform*, Forum Nokia, May 2007.

[10] Leggett, R., de Boer, W., Janousek, S., *Foundation Flash Applications for Mobile Devices*, December 2006.

[11] Graft, K., *Analysis: History of Cell Phone Gaming*, Business Week Online, http://www.businessweek.com/innovate/content/jan2006/id20060122_077129.htm, 2006.

[12] Theesa, *Essential Facts About the Computer and Video Game Industry*, http://www.theesa.com/archives/files/Essential%20Facts%202006.pdf, 2006.

[13] Jacobs, D., *Causal and Female Gamers to Drive Mobile Gaming Industry*, International Business Times, http://www.ibtimes.com/articles/20061005/mobile-gaming-casual-female.htm, October 2006.

[14] Telephia, *Puzzle/Strategy and Retro/Arcade Mobile Games are the Most Popular Among U.K. 3G Subscribers*, http://www.telephia.com/html/insights_071206.html, July 2006.

[15] Gartner, *Gartner Says Worldwide Mobile Gaming Revenue to Grow 50 Percent in 2007*, http://www.gartner.com/it/page.jsp?id=507467, June 2007.

[16] Sun Microsystems, *Introduction to Mobility Java Technology*, http://developers.sun.com/mobility/getstart/, 2007.

[17] Eclipse Foundation, *Mobile Tools for the Java Platform*, http://www.eclipse.org/dsdp/mtj/, 2007.

[18] Mahmoud, Q., H., *Deploying Wireless Java Applications*, Sun Microsystems, http://developers.sun.com/mobility/midp/articles/deploy/, October 2002.

[19] Java Community Process, *JSR 271: Mobile Information Device Profile 3*, http://jcp.org/en/jsr/detail?id=271, February 2007.

[20] Wikipedia, *Flash Lite*, http://en.wikipedia.org/wiki/Flash_Lite, June 2007.

[21] Adobe, *Flash Lite 1.1 Features*, http://www.adobe.com/products/flashlite/productinfo/features/flashlite_1_1.html, 2007.

[22] Adobe, http://www.adobe.com/products/flashlite/images/fig01_lg.html, 2007.

[23] Adobe, *Adobe Flash Lite To Support Video for Mobile Handsets*, http://www.adobe.com/aboutadobe/pressroom/pressreleases/200702/021207FlashVideo.html, February 2007.

[24] Nokia Forum, *Flash Lite for S60 - An Emerging Global Ecosystem*, May 2006.

[25] Adobe, *Online FAQ*, http://www.adobe.com/products/flashlite/productinfo/faq/, 2007.

[26] Marne, J., *Evaluating Java for Game Development*, University of Copenhagen, Denmark, March 2002.