

The future of the European mobile game ecosystem



European
Games Developer
Federation

nordicgame
IP SOURCES

NCC

JCPIC

**For further information,
please contact:**

www.mobilegamearch.eu

--

**Dr. Malte Behrmann,
Secretary General**

European Games Developer Federation (EGDF)
malte.behrmann@egdf.eu | www.egdf.eu

Katariinankatu 3 00179 Helsinki Finland
t. +358 (0)9 4289 1606
f. +358 (0)9 666 334

**Jean-Charles Point
CEO**

JCP-Consult
pointjc@jcp-consult.com | www.jcp-c.net

9, mail de bourgchevreuil 35510 Cesson-Sévigné France
t. + 33 2 23 27 12 46
f. + 33 2 99 27 77 82

**Maarten Noyons
Managing Director**

NCC PARTNERS
maarten@nccpartners.com | www.nccpartners.com/

25 Cours d'Estienne d'Orves 13001 Marseille France
t. +33 491 315 217
f. +33 491 143 199

**Erik Robertson
CEO**

Nordic Game Resources
erik@nordicgame.com | nordicgameresources.com

MEC, Stora Varvsgatan 6A SE 211 19 Malmö, Sweden

The Mobile Game Arch project

The penetration of mobile telephony had reached 95% by the end of 2010 in Europe and approximately 80% globally. Consequently, mobile content has emerged as the lead medium of communication. However, for the European mobile content industry, the lack of mobile standards (e.g. for mobile Application Programming Interfaces (API's)) is becoming one of the main factors hindering growth. At the same time, Europe is losing ground to non-European mobile technology manufacturers and mobile content developers.

The Mobile Game Arch project supports European actors from both the content and technology sides of the mobile games industry to join forces in tackling this barrier to growth. By making significant contributions to global standards, the project intends to create a base for a new generation of mobile games technology that will allow European content to flourish globally and to narrow the digital gap in Europe. The Mobile Game Arch project attempts to reinforce the position of the small and medium sized enterprises (SMEs) working in the European ICT and digital media industry and to increase the accessibility of digital media/service platforms' aggregators.

The main objectives of the Mobile Game Arch project are to:

- *Identify the current growth barriers in the mobile games industry;*
- *Identify a roadmap for growth of the mobile gaming industry and to identify specific European barriers;*
- *Support actions to provide recommendations on standards, specifications, actions, and practices in the area of mobile games technology.*

From this, the Mobile Game Arch project has produced a roadmap and a set of technological, commercial, and political recommendations. Mobile Game Arch solutions will also be applicable to other mobile applications as well.

The Mobile Game Arch project brings together the following partners:

- *JCP-Consult, <http://www.jcp-c.net/en/home.asp>*
- *European Games Developer Federation, <http://www.egdf.eu/>*
- *NCC Partners, <http://www.nccpartners.com/>*
- *Nordic Game Resources, <http://nordicgameresources.com/>*

I © Mobile Game Arch (2013)

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 288632. The outcomes of the project reflect only the author's views and that the Union is not liable for any use that may be made of the information contained therein.

WHO IS MAKING MONEY WITH MOBILE GAMES?

1. The main actors: GAME DEVELOPERS and APPLICATION STORES



2. PUBLISHERS / AGGREGATORS / MIDDLEWARE DEVELOPERS



3. METRICS PROVIDERS / ADVERTISERS / SOCIAL DISCOVERY COMPANIES



Evolution of the mobile games market

Mobile games have been available to consumers since 1997 when a team of Nokia engineers realised that the mobile platform was advanced enough to support games. The first downloadable content arrived about 2000. The early value chain was very operator-centric with developers typically working with publishers or aggregators, which in turn had relationships with operators. Operators were the main distribution channel to reach the consumer and the value chain was linear. Unfortunately, European operators typically had the worst revenue sharing agreements with mobile games companies in all the major mobile games markets.

From 2000 to 2008, in most cases operators had long-term agreements with a restricted number of publishers and aggregators; they generally did not sign up new parties. A publisher typically funds some or all of the development, taking some of the risk but the developer gets a lower (10% or less) share of the revenues. An aggregator typically takes completed games from developers and shares around 50% of the end user revenues. In most cases, operators requested that publishers provide versions ('builds') of mobile games for all the handsets the operator was supporting and selling and to test these builds with the operators' Q&A staff. The amount of builds could reach several hundreds and the process of testing could take up to 9 months.

The market for mobile games changed radically with the launch of the Apple App Store in 2008, giving a big boost to developer power in particular and broadening the market from a niche proposition to virtually every smartphone owner downloading mobile games. End users have a greater choice of where to download apps (application store, operator's store, and third party stores). And developers can also go directly to the consumer. The tight integration of the App Store with the device itself led many consumers to try out apps, and the games market received a considerable boost. Games are easily the most popular category of apps and as the market for apps has grown rapidly, it has opened up new frontiers for game developers.

The launch of the App store was like the rolling stone that starts an avalanche. Like the mountain the avalanche tumbles down, the market landscape was hardly devoid of interest – application stores for mobile phones have been around since 1999 – but these have been largely forgotten in the noise the avalanche has generated. As it rolled down the slopes, the avalanche grew, picking up momentum as the other handset manufacturers got caught up. Apple had an enormously positive impact on the market from a developer perspective, but there have also been a few negative aspects.

Positive influences:

- Success of hardware has generated a lot of hype
- Single SDK for all devices.
- Better revenue shares.
- User-friendliness drives uptake and penetration.
- Fast time to market,
- Smaller companies can distribute through app stores.

Negative influences:

- A larger developer community has led to intense price competition.
- Lack of search functionality.
- Lack of marketing tools.
- Europe-based developers did not profit from the investment hype.
- US-centric business and content.
- Gold rush mentality, again.

“ The aggregators took half the revenue, in the worst case scenario without adding any value at all, re-distributing through each other, assembled portfolios of hundreds, even thousands, of games and thus totally marginalised the developers, financially and in terms of the influence over content creation. Developer revenue shares could drop to 1/16th of the end-user”.

- Erik Robertson, Nordic Game Programme

BUSINESS

Business Issues: Value chain and ecosystem

Developers are the creators of games. Often small, innovative, and entrepreneurial companies, developers typically focus solely on games development and only rarely have marketing budgets to promote titles.

Publishers typically own IP and work with external developers to create games based on that IP. Developers work with publishers based on a fee to create a title, with possibly an additional revenue share based on performance. Many publishers have internal development to create games as well as using external development resources. Typically large companies, publishers maintain relationships with multiple distribution channels and typically have a marketing budget to help promote games.

Aggregators are similar to publishers but do not own IP and do not commission external developers. Rather, an aggregator role involves developers coming with a title already created to leverage the aggregator's relationships with distribution channels to reach the market. An aggregator role typically only involves a revenue sharing agreement.

Application stores can be divided into 3 broad categories: handset application stores (e.g. Apple's App Store), operator stores (e.g. Orange), and independents (e.g. Amazon, Barnes&Noble (Nook), GetJar). An interesting and regularly updated overview of all application stores can be found here: <http://www.wipconnector.com/appstores> .

Middleware includes cross-platform development tools, game engines, and other software designed primarily for developers to speed time to market. A discussion on when developers use external engines or when they build them themselves can be found in the technical section. The three leading cross-platform game engines are currently Unity3D, CryEngine, and UnrealEngine. Unity3D has grown very fast to become the tool many, if not most, mobile game developers currently use. Challengers of these three engines are Shiva (created by Stonetrip), who in partnership with Marmelade (created by Ideaworks3D) offers deployment to most mobile platforms and, last but not least, Delta Engine (an Open Source Game Engine). See also <http://www.gamemiddleware.org/>

The sale of **advertising** to media companies and brands is a specialist task, and developers and even the larger publishers lack the resources and expertise for this. Advertising companies sell advertising inventory for placement of ads in-game. Revenue is then shared with the publisher or developer (usually depending on which company owns the IP of the game). Advertising is one business model which allows developers to generate revenue from free games. Typically advertising companies keep 40% of revenue and share 60% with the developer.

Metrics: A number of companies have emerged providing the tools for developers to track download, usage, and other metrics. This has become of increasing importance to developers, allowing them a better understanding of the market and the audience. Examples include Flurry and Distimo.

Social and multiplayer tools to allow gamers to play with friends compete for high scores, and so on. Working with many smaller developers, each of which would only have a small audience, these companies can build up a large aggregate audience of gamers. Application stores have typically been poor at enabling the discovery of new games and a key challenge for developers when launching a new title is that they have a minimal marketing budget and simply appear in an app store as one of tens of thousands of other games. Social discovery companies include GREE and Scoreloop.

Getting to market: Generating revenue

The success of manufacturers in moving into the apps market and thus the diminishing role of operators in the value chain has challenged the publishers' addressable market and in the short term has diminished their importance. Developers can:

- Work directly with manufacturers for distribution onto smartphones;
- Work directly with independent stores;
- Operators are also opening doors to the smaller developers, although publishers still have a role to play;
- Work with publishers or aggregators for distribution;
- Go D2C (direct to consumer) although in the case of Apple, the Apple App Store is the only permitted distribution mechanism.

Most publishers are now super developers and most developers are self-publishing. Despite the fact that developers are now self-publishing, the publishing of mobile games has become more complex than what it was in the time of the operator dominated market: the role of social networks and other viral marketing tools is becoming more important and many techniques and services to boost the number of downloads of games are emerging. With the emergence of social discovery, developers can opt to work with these companies instead to find an audience.

Until recently, the main option for generating revenues was a simple payment on downloading a game. Today, a number of new business models have emerged which are often collectively referred to as "freemium". The game download itself is typically free and then revenue is generated after download. The freemium model, as opposed to the premium model is becoming dominant; some 80% of the top iPhone games today use "freemium" models to generate revenues. A general point to note is that these new business models are all reliant on usage of the game. This is one of the major reasons for the increasing importance of companies providing metrics for game developers. These new business models have been enabled in part by the new generation of application stores which allow for in-app transactions, and by advertising companies which have created technology to allow advertising to be delivered inside apps.

Also known as micro-transactions, in-game purchases typically allow users to play some (or all) of the game for free but pay small amounts either to open up new parts of the game or buy virtual items to use in the game itself. New level packs, upgraded items and cosmetic changes such as different outfits for characters are typical of the types of things available for purchase by players. Transactions' values are typically low. Payment is often tied to the user's account with the application store itself, so the user does not have to have a string of €0.50 transactions on their credit card. Revenue sharing for micro-transactions is usually the same as for pay per download business models (i.e. a 70/30 split in favour of the game developer). The delivery of advertising into games has been facilitated both by the new generation of application stores, by digital advertisers, and more recently by mobile operators as well. Integration is similar to the insertion of advertising into web pages.

Competition is fierce, discoverability is key. At the moment the ranking and featuring in app stores contribute to the visibility of a game. Ranking is determined by the number of downloads and features are controlled by the app store. The consumer comments and appreciation on the app store is another reference that is important. Unsurprisingly, Google has taken a major role in this space not only in its own Android Market but its advertising technology can be used in any app (Android or not) distributed in any application store. Apple also launched its own iAds program, but also permits developers to use third party advertising solutions.

TECHNICAL

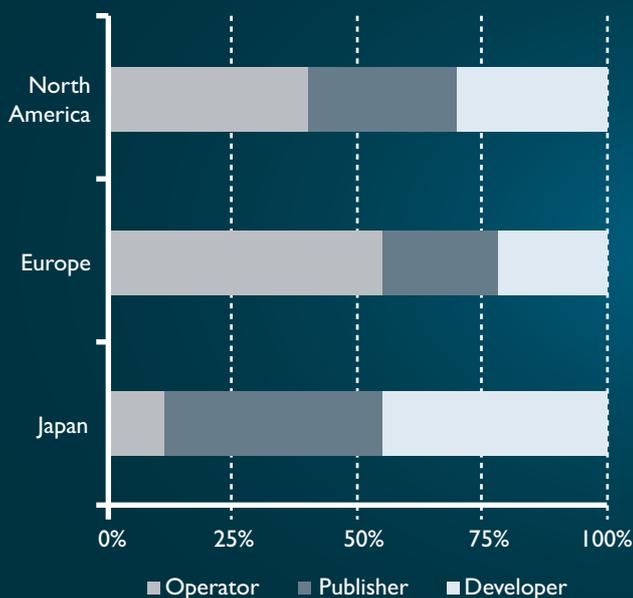
Although Apple had a dominant position from 2008 to 2010, the rest of the market has caught up and today the market is extremely fragmented. Within this market fragmentation, there are then subsequent, multiple additional layers of fragmentation. A developer creating an app for the iPhone will, in many cases, only have to produce one version of the app. A developer creating an app for distribution via Android, for example, must produce multiple versions for each Android handset.

For the game developer community fragmentation and the inherent costs it incurs are the major issues of developing applications and games for mobile. The costs of such fragmentation can often be over 50% of the total cost of the game project. Anything that can reduce or eliminate such cost can be directly ploughed back into better gameplay, enhanced user experience, and better testing to deliver a superior product. However, in the view of some, fragmentation is no longer the big problem for game developers, as the smartphone market is by far the largest market for mobile games in terms of revenues and that market is now dominated by iOS and Android. Some fragmentation worries persist, or are renewed, particularly regarding Android. Middleware is currently offering what is deemed adequate solutions to address the situation. The mobile environment is going through the same phases as the PC environment.

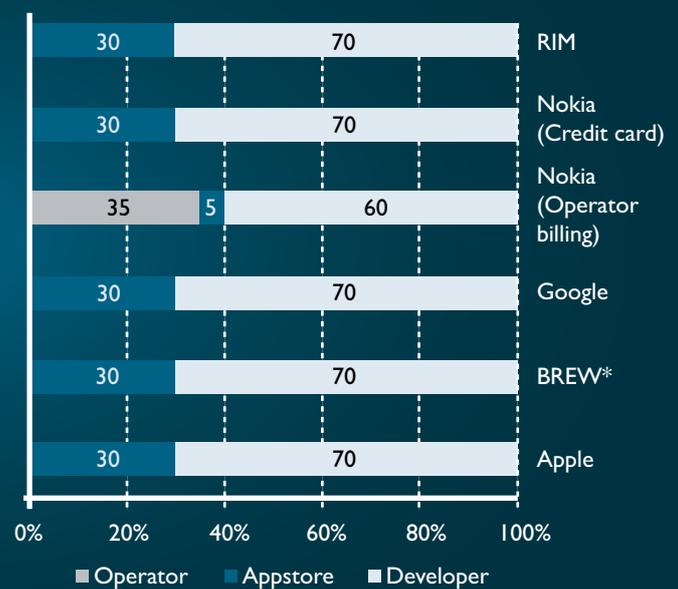
With the advent of the Apple App Store many developers have wholly embraced the homogenous and simple nature of the platform and the business model. It is clear how the money is made. The game platform capability is high so the user experience is very positive and the money flow is fair and allows for reinvestment in further game development. Many developers have dropped developing for other mobile platforms because of the success of the App Store as a channel to market. The question is whether this one-channel strategy is a good choice. Although there are a handful of studios which made several millions, there are hundreds of studios which were a lot less successful on the App Store.

The technical challenges for game developers are to keep up with the fast pace of technological changes in order to compete in a global market, and the European developers need to be represented in standard initiatives and to be represented at a relevant level in negotiations with platform holders. Today that representation does not exist. We think developers are those who can give valuable contributions to the specifications, the discussions and the implementation of new standards and tools. They can provide a vision that is not just looking at the fragmentation on the operator side or the handset side, but everything in between, including the tools that they are using and/or that are available on the market. Developers are looking for a knowledgeable representation in standardisation initiatives and in representation towards the main players in the mobile industry Google, Apple and Microsoft, Research in Motion, Samsung and Nokia.

BEFORE iPHONE: POOR DEVELOPERS



AFTER iPHONE: FAIR DEALS



*BREW = revenue sharing after operator has taken retail markup

Source: Strategy Analytics

The evolution of the handset

The first 'intelligent' handsets that were available to the majority of developers came with the advent of Java, Symbian and BREW technologies. These handsets were equipped with very simple hardware, small monitors and a keypad. Eventually, they were equipped with slightly better hardware, and developers got access to core features of the handset, such as the camera and file system. From here, hardware-wise there was something of a standstill, while software was booming. Handsets began supporting various 3D APIs (Application Programming Interface) such as Mascot Capsule and M3G together with hardware-specific technologies such as GPS tracking, Bluetooth, etc.

Simultaneously to the development of ordinary mobile phones, smartphones were beginning to appear. At first, they were only systems used by people wishing to combine a PDA and a mobile phone, but slowly they started to enter the broader consumer market. It was not until 2011 that Symbian was overtaken and lost the lead in smartphone penetration, to Apple's iOS's premier rival; Google's Android.

One of the biggest breakthroughs in mobile phone usage and its validity as a multimedia/gaming platform has been the screen. The quality, size, resolution and richness of graphics are of course central to the gaming experience and the developers' ambitions and challenges. However, it can be argued that the touch-screens made part of the mainstream through Apple's iPhone were of even greater importance; as by combining well-functioning touch controls, well-adjusted to screen size, with user-friendly and coherent interface design, a new paradigm for user interaction was introduced – a breakthrough not comparable with gradual graphics improvement.

We are seeing a trend towards high-power display hardware, in separate GPUs (Graphics Processing Unit – or the display hardware) in the newer phones, using ARM's Mali or nVidia's Tegra processors. This has not only enabled developers to do more but has once again widened the appeal of the multimedia aspect of smartphones. The CPU (Central

Processing Unit) of mobile phones was, in the early years, a limiting factor, and has now been replaced by today's reliance on GPUs. However, CPUs still stand for big performance progress and enable developers to create more and more advanced applications.

Recently, with the rise of powerful screens, display hardware and CPUs, battery life is again becoming a distinct problem. Most smartphone users today have come to terms with having to charge their handsets daily, something which used to be a weekly exercise. In gaming terms, a too-powerful gaming title might in fact drain the user's batteries, diminishing the use of the handset and also lowering the perceived quality of the game itself. Internet forums are filled with angry exclamations regarding games that "drain" battery life.

Geo-location

Most smartphones are expected to come with some kind of location/navigation software and hardware. Despite the fact that this trend has been around for quite some time, this segment is still fairly unexploited by game developers.

Network-based techniques utilize the service provider's network infrastructure to identify the location of the handset. The advantage of network-based techniques (from the mobile operator's point of view) is that they can be implemented non-intrusively, without affecting the handsets.

Handset-based technology requires the installation of client software on the handset to determine its location. This technique determines the location of the handset by computing its location by cell identification, signal strengths of the home and neighbouring cells, which is sent continuously to the carrier. In addition, if the handset is also equipped with GPS, significantly more precise location information is then sent from the handset to the carrier.



Geolocation level 2 is not there. Geolocation level two is going to give you street addresses and places beyond just numbers on the map.

- Charles McCathie Nevile, Yandex/W3C

Using the SIM in GSM and UMTS handsets, it is possible to obtain raw radio measurements from the handset. The measurements that are available can include the serving Cell ID, round trip time and signal strength. The type of information obtained via the SIM can differ from what is available from the handset.

Hybrid positioning systems use a combination of network-based and handset-based technologies for location determination. One example would be some modes of Assisted GPS, which can use both GPS and network information to compute the location.

Human machine interface

It is easy to debate that the most widely used feature of cell phones, apart from their communication capabilities, is the built-in camera. Coming as a de-facto standard, today's handsets can do anything from taking high-resolution images, to capturing panorama scenes and recording HD video. Several applications have been made that utilize the built-in camera for AR (Augmented Reality), including games where overlays most often serve as indicating the presence of features from an alternative or parallel reality. However, it cannot really be claimed that this type of features are yet part of mainstream games on mobile phones. Some applications, and a few instances of games, have actually used the camera as an input/control device.

However, the primary input device for mobile phones has of course been the keypad. Lately, the touch-screen has made a very big impact on how the user interacts with applications. The microphone has also been used as a software control device, but not very often.

Mobile phones have in recent years been shipped with advanced sensory equipment. Most advanced phones today have, for instance, accelerometer support. Plenty of games already use these kinds of sensors to deliver a richer experience to the player and also cope with the diminishing number of physical buttons available. Today, it is important for most game developers to make good use of the sensors, to simplify the game experience. This is mostly due to the fact of the non-existent physical buttons, and how emulation of keys on modern touch screens is far from optimal.

SOFTWARE AND PLATFORMS

Software has always evolved together with hardware, enabling developers to do more, faster. Developing for a certain range of phones is enabled by a set of base technologies available on a subset of the wanted platform.

Some of the technologies appear in more than one case (such as OpenGL, and Java libraries). These lower the porting effort of developers and offer a familiar environment for many. OpenGL itself is a very good example, as it is available on a wide range of high-end platforms. This lowers porting work slightly, but there are still plenty of hurdles to overcome going from one platform to another. Java is also a prevailing factor. A lot of the older platforms supported Java, and all the new Android handsets do as well. Similarities do arise, especially regarding baseline libraries available to the developer. In some cases, porting games from an old J2ME handset to a new Android one is thus easier than porting it to other platforms.

Smart development studios should aim to develop their titles in as many shared libraries as possible, thus simplifying the porting process later on. For instance, Android applications can also be developed natively, and thus sharing plenty of libraries between the iOS and Android versions, in this way a studio can win plenty of development time.

DEVELOPMENT FRAMEWORKS AND ENVIRONMENTS

Some development frameworks allow developers to create products that can be published on multiple platforms with minimal changes and adaptations. This kind of environment can boost productivity and shorten development time while QA (Quality Assurance) requirements are the same. QA time can never shorten, as your software still has to be tested on a (very) wide array of handsets, where all of them perform differently. However, time saved in development time can of course be put to good use in the QA cycle instead.

These types of development frameworks have often been shunned, as their versatility comes at the cost of performance. Even though this might not be quite as true in today's systems, the prejudice remains. Of course, very performance sensitive applications will always benefit the most from being written completely in the native development framework. Today, we see many different types of frameworks available, where many of them have had great success and seen a large number of titles successfully published. This, of course, is a sign of the highly-fragmented times, where a long turn-around time can completely destroy a project.

The different platforms can differ (sometimes wildly) with regard to how versions are pushed out to testers and customers. The Apple App Store has its own set of selection rules and deployment restrictions, compared to the fairly free-form Android Market of Google. The open nature of Android allows and encourages parallel deployment platforms, one example being Amazon's marketplace for Android. While this eases market access as a whole for developers, at the same time it comes with the price of more overheads, for administration and adaptation.

The network

Mobile device hardware performance is progressing along with Moore's law. At the same time, hardware accelerators and new cameras allow mobile devices to support full HD encoding. With LTE technologies, also the wireless bandwidth available in mobile devices will typically increase to several Mbps in wide area networks. Future evolutions of 3G and LTE networks, like LTE advanced, or currently discussed 5G evolutions, are expected to provide bandwidth in the order of 100 Mbps.

Notwithstanding these evolutions, applications that require significant processing power, such as high-quality rendering, lay out of reach when only local processing capacity is considered. To overcome this issue, specialized cloud computing facilities targeted at real time applications are being deployed.

Even though the remote processing capabilities offered by emerging cloud computing facilities are compelling, there are two major drawbacks related to this approach: large Wide Area Network (WAN) latency and the limited mobile energy budget for communications. Unlike the permanent increase in internet network bandwidth over recent years, a significant reduction of latency is unlikely to occur in the near future. For streaming content that is entirely generated on the cloud servers this one-way latency can be tolerated. However, since advanced immersive multimedia applications such as augmented reality (AR) rely heavily on rich sensing for real-time interaction with their surrounding environment, this would require two-way high-volume data transfers with strict end-to-end latency bounds between the mobile device and the remote cloud, which is unrealistic even for future networks.

A simplified value chain model adapted for mobile games



PURPOSE OF THIS GAP ANALYSIS

This chapter presents visions of desirable near-future situations, per area, thus indicating the present gaps between what is, and what should be, from the perspective of the European mobile games developers. It is important to note that it is this perspective, looking from the typical micro-SME viewpoint that is chosen for this paper, rather than a top-down industry-as-a-whole view, or as seen through the individual creators' and engineers' eyes.

Our main reason for this is that the situation, the access to information, technology, and marketplaces is primarily related to the situation for each company as an independent business entity. What is available of such resources for a large game development studio is not at all certain to be available for a small, but talented, start-up. So, seeing the gaps between what is and what would be desirable must be regarded in relative, quality or volume terms, rather than either/or absolutes. Put differently, this is the gap this project is all about; the position of European game developers in society and their own perception of this position.

To illustrate and position the gaps for the European mobile games industry, we have chosen the value chain model as our primary structure. We feel that it is better suited to broaden the understanding and seems to have more explanatory value for the situation of this industry, rather than stakeholder, financial, geographical market access, or other models.

Preconditions are here seen as a wide set of factors, ranging from public policy and applicable law, standards, infrastructure, platforms, operating systems, APIs, middleware, and tools, and the developers awareness and access to information about them. These factors are as a rule not under any influence from the game development company. The developers do have a varying degree of choice when it comes to APIs, middleware, and tools, although some of these are specific for the hardware and OS targeted. Certainly the conditions in the market, with user preferences, competition etc, are not controlled by the developer either, but the process of getting to market we see as an iterative process where the developer's choices will have a direct impact, even if marginal. The developer may choose payment systems and structures, but, again as a rule, have no influence over them as such.

PRECONDITIONS

Public Policy

A non-fragmented European digital single market area is one of the key requirements for the economic growth and new jobs generated by the mobile content creators. That means that we need more middleware, but not necessarily more regulation. However, it could be that a legally harmonized European digital single market area has to be created to not endanger the growth of the European mobile games industry.

There is no single age rating system for digital mobile content. Instead each mobile platform (App Store, Google Play etc.) has their own age rating system for the applications published in the platforms. As there is no major European mobile platform on the market anymore, these age rating systems are not based on European standards. The Commission tried to solve this problem by forming a CEO coalition on industrial self-regulation in this area, but almost no European game content creators were included in the process. There is clear gap between the European actors creating the mobile content and mainly non-European actors running the **protection of minors** system. Consequently, European Commission should launch a parallel initiative to the CEO coalition that focuses on engaging the European creators of mobile content to self-regulatory actions in the area of protection of minors. This initiative could also reach into the field of technical standards.

From the beginning of the year 2015, **VAT** will be paid in the European digital single market area based on the location of the consumer instead on the location of the company. According to the new regulation, the mobile platform holders, unless otherwise stated, should be responsible for paying the VAT. This will significantly strengthen the position of non-European platform holders in the mobile value chain. The Commission should secure that the way the regulation is implemented does not lead to double taxation and still makes it possible for European actors to bill their content directly through mobile internet as a web service, instead of using the closed mobile ecosystems.

At the moment, no cultural heritage institution in Europe focuses on preserving mobile games and applications partly due to strict patent and copyright legislation. Consequently, a large part of the early European **digital cultural heritage** might be lost during upcoming years. This might seriously hinder the understanding of the historic trends in the industry. The European copyright framework should be rewritten in a way that it clearly allows breaking the patents and anti-piracy measures for research and preservation purposes.

The current **copyright** legislation also seriously hinders the use of European music in European mobile content. It is often easier for European mobile content creators to access European music through a non-European licensing operator or by composing it by themselves than contacting each national collection society. Further from a general point of view the European games industry appreciated the initiative of the commission to develop a unified European license system. Pan-European licensing for music should be made much easier and flexible, so that more European music could be used in European mobile content.

As accessing venture capital is much harder in Europe than it is in the USA, the European mobile developers are depending much more on public funding as start-ups than their competitors in other regions. Unfortunately, there has been only a limited set of **European and national funding instruments** available for game developers and even their existence have been questioned due to economic challenges the member states are facing. Public funding for mobile game developers should be secured through industry specific programmes.

Standards

Standardisation has always been required in the network domain to ensure product interoperability. This is difficult to achieve in the middleware / API domain where consensus, or harmonization, is always almost impossible to reach as different incompatible implementations could coexist. This has resulted for instance in the fragmentation of game engines and required many different implementations of the same game. Android and Apple application stores have temporarily improved the situation but fragmentation starts again to appear according to different Android implementations. HTML5 could be a solution which would improve the situation, but it is not suited for AAA or mobile games yet.

In the mobile ecosystem, the leading standardisation bodies are World Wide Web consortium (W3C), European Telecommunication Standards Institute (ETSI), Open Mobile Alliance (OMA), Khronos Group, GSM Association (GSMA). It can basically be taken as a given that SMEs, and especially mobile game or application developer studios, have no discernible influence on standards. What is even worse, is that their limited resources for business intelligence makes standards, whether they are of the stipulated or more of the ad hoc harmonization kind, not an opportunity for excellence, but rather something that just happens to these small companies, with varying foreknowledge or early warning.

Some developers feel that **HTML5** could alleviate the fragmented situation somewhat. However, it is quite obvious that HTML5, especially the implementation thereof, still has a way to go. The obvious practical gaps for game developers are today in the differences among web browsers.

Cloud is emerging rapidly for mobile gaming applications, but today the cloud infrastructure is not able to meet the constraints introduced by mobile games. Cloud must be decentralised and linked with QoS provided by the network to fulfil the requirements of gaming. Decentralising the cloud definitely requires standardisation (at least between layers) as it involves the network layer. Thus standardisation “ad minimum” is going to be more and more necessary with the deployment of mobile cloud gaming architectures. In a more positive scenario, game developers are not only aware of, but consulted with and directly involved in standardisation processes.

At the same time, there is emerging and increasing demand for standards especially in the area of payment solutions and protection of minors solutions. In both of these issues there is an increasing demand for standardized procedures and requirements in addition to the harmonization of technological solutions.

Infrastructure

Cloud gaming is considered as a promising solution because it addresses several recent problems. With gaming no longer restricted to game console and PC, but expanding to non-game-specific platforms such as TV, tablets and smartphones, users are faced with the issue of running computationally expensive games on devices with no sufficient processing power. This is a relatively old problem, which seems to come back. Mobile and TV manufacturers choose in fact hardware configurations with relatively low computing and storage capabilities in order to address battery issues and to maintain low prices. By performing the intensive computation on the cloud, cloud gaming allows games that require heavy computation to be played on any device. From the game developers’ point of view, the proliferation of devices has made the creation of cross platform games even more complex, increasing costs and reducing control on the software.



I really believe that the cloud and the folk computing, the cloud close to you, will be the business model for the future.

- Bogdan Tudosoioiu, Mobile Heights

Current cloud gaming solutions, which we refer to as the first-generation cloud gaming systems, suffer from critical weaknesses: Today’s

- game engines are not designed for cloud gaming,
- cloud infrastructure does not match requirements of cloud gaming,
- cloud delivery systems are not designed for cloud gaming.

In general, decentralizing the cloud for mobile, and optimizing the QoE and energy parameters of the mobile terminals requires the definition of a framework, something that we could call “Resource Delivery network”, by analogy to CDN, where application components are hosted dynamically in an optimal way, according to the terminal requirements and location in real time. This includes the definition of the following features, such as Management of application components, Surrogate discovery and Security:

Ecosystems

It is highly important to note the growing importance of ecosystems in the mobile application space. For our purposes, a definition of an ecosystem is something that comprises a hardware platform, an operating system, and a distribution channel or platform.

The more of these components that are proprietary, in that they cannot be replaced with a parallel product or service, the more there is vertical integration from hardware to software application distribution to the end user, the more of a closed ecosystem is the case. The more that are readily available, as open source, or basically equivalent and interchangeable products from competing manufacturers, the more open is an ecosystem.

Choosing to work in a more closed ecosystem may mean a total absence of choice in the software components, tools, or marketplaces that may be utilised.



We all have to think how we are effecting on people's everyday life and how we can actually bring such a value that people are ready to pay more to you instead of some other company. And there all these new technologies like cloud computing come relevant, because you can actually save money by running a business over the cloud.

- Harri Koponen, Rovio

Hardware

The large majority of game developers have no idea what hardware manufacturers are planning for their next generation of smart phones. They instead need to wait until the phone is brought to market so they can buy one to find out for themselves. The development of a game often starts before the release of the hardware, basically operating in the dark, without knowing the actual specifics of that device. This can be especially dramatic if the hardware introduces totally new features, such as accelerometers or multi-touch interfaces.

From a game developer point of view, hardware manufacturers should agree on a certain number of basic standards, which would benefit gamers worldwide. A broad dialogue between developers and hardware manufacturers about future devices and the games that could run on them can lead to better products and thus more value for costumers.

Hardware manufacturers are trying to compete with each other by offering larger screens, high-performance chipsets, lighter phones and other special hardware features that could make their smartphones more unique and more competitive. However, this differentiation of devices has always resulted in fragmentation: developers need to address different screen-sizes, a variety of GPU's and CPU's super high-resolution cameras and all the other distinctive features. They have to pay the price for this fragmentation, since every different phone requires a dedicated version of the game.

Android divides the range of actual screen sizes and densities into a set of four generalized sizes: small, normal, large, and xlarge, and a set of four generalized densities: ldpi (low), mdpi (medium), hdpi (high), and xhdpi (extra high). The hardware fragmentation still poses high requirements on game developers. To support 80% of the documented user base, it seems that 13 of the 20 devices, or 65%, should be supported.

It is not only Android that suffers from fragmentation. At the time of writing, Apple has released six major versions of the iPhone smartphone hardware, most or all of which are still in use. The diversity and fragmentation of devices is not something that is desirable for mobile games developers. Evolution and introduction of new features enable new, creative games and new user experiences. But this is something that developers should be closely involved in, as they need to be better informed on and more influential in the hardware development processes.

Operating systems

It would be highly beneficial if it was easy for developers to follow operating systems (OS) developments, from features to timelines and market shares, in a unified environment, with notes, tips, peer reviews, comparison tables, ratings, discussions and the like. With clear benchmarks accessible to developers and users alike, force would be applied simultaneously from both content providers and end customers, a force that would hinder or at least put the light on negative fragmentation.

The market research firm Strategy Analytics reported in January 2013 that 68.4% ran Android as the operating system, while 19.4 per cent ran iOS. Thus, a developer seems to be able to reach around, or more than, 80% of devices by supporting these two operating systems.

At the moment Android seems to become the predominant Operating System (OS) for Smart Devices and Apple number two. Microsoft and BlackBerry (previously Research in Motion - RIM) follows in a currently undecided third and fourth place, while the newly announced Tizen of Samsung could become a fifth player and potentially a threat to Microsoft's and BlackBerry's positions.

However, in terms of revenue for game developers, iOS is by far number one, leaving Android and the other OS's behind. This is connected to the fact that the development sector considers iOS to be the most secure medium to monetize their content: The number of supported devices is relatively small, but growing with each new release. In the case of Android the variation is much higher as every handset manufacturer adds proprietary features on top.

The more fragmentation is allowed inside an OS, the higher the costs for developers to reach consumers. For consumers it is quite frustrating to buy a device that allegedly supports Android applications and games, but doesn't work properly in a variety of cases. It can be claimed that Android and Blackberry 10 OS are more or less open, whereas iOS and Microsoft are closed, not open to modification. We currently consider the fragmentation within Android as one of the biggest headaches for mobile game developers in the context of this discussion on operating systems. On the other hand it is relevant for game developers is the insight they may or may not have in the roadmap of the operating system. In general terms, this roadmap is more accessible in open platforms such as Android than iOS.

Given the current legal battles between OS and handset providers, it is clear that the real battle is about stopping competitors from catching up with your advanced technologies. Needless to say, these battles have little or no real meaning for consumers or developers, but for manufacturers a lot is at stake. What does interest developers is how the OS gatekeepers manage the fragmentation inside their operating system. Still, a developer needs to be aware of the developments in the operating system market, and to track the progress. Among developers there are of course highly differing opinions of the future potential and impact of the above. Not all predictions, positive as well as negative, will become reality. Just looking at recent history, the market shares of operating systems can change quite rapidly.

The more fragmentation is allowed, the higher the costs for developers to reach consumers, and the more frustrating for consumers that expect to, but do not get reached. Mobile game developers need insight into, and potential influence over, the development plans for operating systems. We need to document to what extent OS manufacturers consult with game developers on issues regarding OS development.



[Special focus should be on] hardware in emerging markets [with] mostly low-end and cheap devices. In Asia, there are lots of no-name devices with really really weak technical platform, but you still can play games on them...

- Andrei Dementjev, Fortumo

MAKING THE GAME

It would be highly beneficial if it was easy for developers to look up API's, middleware tools in a unified environment, with notes, tips, peer reviews, comparison tables, ratings, discussions and the like. This would enable mobile game developers to make more informed choices, and thus improving both their production efficiency and the end user's benefit.

APIs

An API allows a developer to access services provided by the operating system or other components running on the system. The API exposes these to the application, but of course requires the developers' knowledge of the location, names, functions and methods to call, constants and variables to pass to these, and the results to expect. Using APIs may make applications both more conforming in appearance and more efficient to make. As a rule, APIs pose no restriction on the programming language or development environment used by the developer.

APIs come documented with SDKs - software development kits - from OS vendors, but also from other sources, for exposing components but also as other means of accessing OS services. Often OS version-specific, they may also be tied to a single mobile operator's services, such as location information provisioning. Something of a special case within APIs are analytics tools.

Many APIs exist (<http://www.wipconnector.com/apis>) There seems to be no obvious, central and trusted resources that provide an overview and can support game developers in evaluating APIs to use for their projects. Developers, as a rule, lack insight into how analytics providers process and retain their data. Your analytics provider being acquired by your competitor is probably not an ideal situation.

Middleware

A very important aspect of middleware is limitations. Aggressive marketing obscures facts from the developers and hides the weaknesses of a system. A middleware solution might promise code to be run-able on a wide spectra of platforms and devices, but in reality is a very constricted platform when the developer aims for such broad market availability. True write-once-run-everywhere systems do not exist and the complexity of development and difference between platforms (both technical and ideological) makes this a very difficult goal to achieve.

An ideal information resource for developers would shine a light on both advantages but also drawbacks of different middleware, thus enlightening the developer to and enabling informed choices, thus ensuring a higher-quality end-product to the consumer.

Middleware refers to a professionally produced product used in development. Today, companies thrive in software development by just producing middleware for other companies to consume. This practice is vigorously pursued in game development, as games by nature are complex and combine enough components to make middleware a lucrative solution to cut production time and costs. It is hard to find a product today that ships without any form of middleware, as many components are regarded as de facto standards and used for any triple-A production.

The rise of middleware as an important factor in game development comes from the equally expansive rise in game development complexity. With new systems, consumers' technological demands and market segregations, the costs of pure development skyrocket. Developers look for ways to cut costs and middleware solutions are thus built around this premise.

Over the last few years, development in cross-platform game engine middleware has expanded fiercely. Developers are today offered several different solutions to cut costs and deploy on many platforms at once, with the write-once-run-everywhere paradigm. As mentioned several times in this document, segregation of the market and walled-garden architectures drive developers towards these solutions. Sometimes the distinction between middleware and game engines is made, on the account that game engines offer a complete solution for an entire production, while middleware is regarded as having a specialist role covering a single system, or group of related subsystems. We regard game engines as middleware as well.

As a rule, middleware enables both easy access to lower-level systems services, often replacing - and abstracting - a number of APIs, but typically also the same functionality across different platforms and OSs, allowing the developer to code once, but deploy in many configurations. Middleware may very well encompass several to basically all the fields of user-entry processing, sound playback, 2D and 3D graphics, memory handling, physics and AI processing, including higher-level scripting. The middleware suites providing the most of these are called "game engines". Below is a table listing a number of middleware solutions and their features:

Referring again to the Mobile Game Engines website (<http://mobilegameengines.com> as accessed on 2013-03-18), we find that of the 62 game engines for iPhone, 31 also support Android, four support both Android and HTML5, and two support iPhone and HTML5 but not Android. This site seems to provide quite a lot of what is expressed in our vision, but the number of reviews and references are limited - more than half of the engines currently have neither.



“ It is important to understand that these regulators [all over the world] do come together, they do observe what is going on and they will affect different markets [in the area of protection of minors].

- Annie Mullins

Today, a developer can find middleware for any part of production, making gaps in the traditional (segregational) sense virtually non-existent. Some platforms might not be supported by your choice of middleware, or specific functionality on a specific platform might have been ignored to supply broad-spectra support.

For developers, this means that their product may have to co-exists in several middleware systems (and even in several game engines). Since most systems use different paradigms for development, switching from one to the other can in many cases be an outright painful process. These types of gaps are very hard to work around or reduce, as it is the middleware manufacturers right to create a system catering to their development ideology and choice of language, platform, feature-set, etc. Regardless, it still is a gap for developers to overcome.

It should also be noted that along with all the advantages that middleware and game engines bring, particularly as efficient and fragmentation-addressing parts of the development process, comes also the risks of lock-in and lock-out. Committing to a certain game engine that is subsequently not maintained and developed by the producer is certainly a disadvantage. There are also cases of commercially available game engines, in the wider game industry, being acquired by publishers and then made not only proprietary, but internal and private, closing off other users.

Tools

Like any industry, game development uses a series of tools to facilitate production. By tools we generally mean “productional” software but it is extended to hardware as well in some cases, such as motion-capture equipment, cameras, audio-recording, etc. The two main categories of tools are Integrated Development Environments (IDEs) and graphical tools such as 3D-modelling software (Maya, 3D Studio Max, Blender) and 2D-graphical software (Photoshop, Gimp).

Beyond the two main tool chains there is a subset of smaller tools with the purpose of creating anything from a bitmap font to physical models. In-house-developed tools are very popular in game development. Larger game development studios create entire multi-purpose tool chains that are proprietary but can be shared with other studios owned by the same publisher. Some studios also create full-fledged game engines, with much of the production pipeline and tool chain in place, for use by other developers at a considerable license fee.

Tools today affect game development in a technological gap. The choice of tools might enforce a certain paradigm or technological choice that cannot be reverted. Many tools have implementations and connected APIs/libraries are restricted to a single platform, making multi-platform development much harder and might force smaller studios to disregard multiple platforms entirely, at a cost for both the studio and the unknowing consumer.

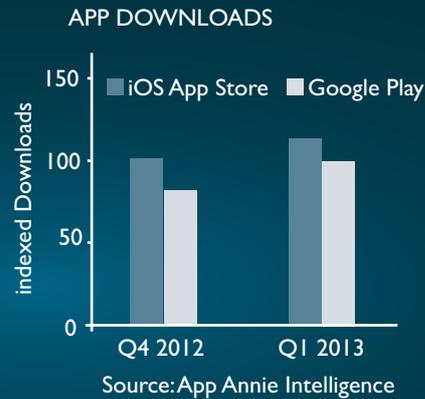
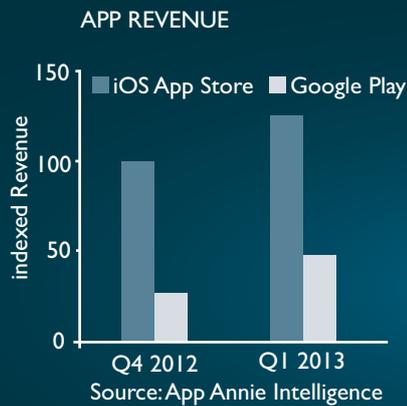
Finally, for proprietary tools there is no way for game developers to be active in the development of the tools themselves, other than plain suggestions. There are some initiatives in the tools sector aiming to streamline and unify different production formats so that tools and APIs can be synchronized over a myriad of platforms, but still there is much to be done. Involving more of the game development community in a community-based process can speed up the relatively slow standards process and also highlight actual needs of actual game developers.

GAME TO MARKET

It would be highly beneficial if it was easy for developers to look up market channels in a unified environment, with market penetration, business models and terms, free-to paid conversion rates, notes, tips, peer reviews, comparison tables, ratings, discussions and the like. It would be desirable if something corresponding also was available from the end users' perspective, highlighting European mobile games and their producers. This would not only improve their production efficiency, but also the end user experience.

In the first 10 years of the mobile games history, mobile operators were the main gatekeepers of the industry and sometimes handset manufacturers allowed game developers or publishers to pre-load their games on certain devices. Today every developer can become a publisher on iOS, Android, Microsoft, Nokia or RIM or the stores of GetJar, Amazon, eBay and other stores.

However, this new situation has led to an extremely competitive industry, where those who have the means to boost their visibility and use the available tools to optimize the monetization of their games can reach unprecedented high returns on investment. Traditional marketing, such as advertising and PR is not really efficient in the app store world. Talking to journalists can increase the discoverability of an app and raise the chances for being featured in an app store.



Understanding social media has become a key asset in promoting a game. First of all, the promotion of a game cannot be done without a good social media campaign. This would give players the possibility to build a deep relationship with the game and the company who made it. Some services allow publishers and self-publishing developers to buy users. The more money they spend, the higher they will be in the charts. Success on the app stores can now be bought to a certain extent, since a game which is listed in the top ten gets noticed by a large number of consumers. And this fact alone generates more downloads and in time revenues. Key issues include customer care, localisation and emerging markets, App Store specifics (specific rules, gatekeepers) and piracy.

With an estimated 200 new games being launched on iOS, the discoverability of apps becomes more difficult every day and ends the dream of developers who think they can launch their game without any form of marketing. Most developers are now self-publishing and decide themselves what to deliver and what not to do. For most of them the decisions regarding the production of additional builds of a game are based on the expected return on investment. Visibility, or discoverability, is a main issue for developers.

“[...] Flurry calculates that the difference in revenue generated per active user is still 4 times greater on iOS than Android. [...] In short, Android delivers less gain and more pain than iOS”. From a game developers point of view only one thing really counts: Return on Investment. Any such ROI calculation must take the following issues in consideration:

- Revenue share for the developer. This is mostly 70% for the developer.
- Installed base and fragmentation of installed base. The more fragmented, the higher the development costs.
- Piracy and what does the OS owner do against it?
- Average pricing and consumer behavior on that specific OS.
- Competition: On iOS only, about 200 games are launched each day.
- Publishing strategy. Does the developer or publisher want to be present on all platforms and reach as many players as possible?
- Time to market: how long does it take to get your game through the selection procedures?
- Any censorship for specific games: violence, pornography, etc.

To make this calculation, and then basing an informed choice on this, it is of course essential to have unambiguous information on all the issues in question. Unfortunately small and medium-sized developers fall behind their larger competitors due to lack of marketing power. A system that ensures the European mobile consumer's awareness of the problem would benefit both the consumer (as high-quality content can instead be browsed on an interest basis) while developers can focus on creating games that target niche markets with quality productions.

In general, the markets are too diverse, and are changing too fast, even for our highly creative and technologically skilled micro-SMEs. Especially, in order to access the emerging markets they should acquire more expertise on how to enlarge their activities to different regions. This concerns marketing, payment, cultural issues, but also other related issues related to being truly local and global at the same time. They just cannot make informed decisions on what markets to address and how to reach them.

GETTING PAID FOR THE GAME

For game developers it is crucial that their customers, the players, can make payment for and within their games with the payment solutions they prefer to use. As games are sold globally there should be a universal joint payment solution uniting existing payment solutions in one system that could be used in different platforms. Ideally game developers would be able to sell everything to all platforms under one and the same banner, and keep one and the same revenue share. Of course the store owners are able to stop this the same moment it measurably hurts their business. It should be easy for developers to look up payment solutions in a unified environment, with market penetration, terms, comparison tables, ratings, discussions and the like.

Today, there exists a wide diversity of payment systems in the mobile games field. Currently, mobile distribution platform holders are taking stronger and stronger control over what payment solutions can be used in their systems. From the beginning Apple has limited the payment solutions that can be used in Appstore to credit cards. While Google has opened its system for carrier billing, besides that it only accepts payment solutions provided by major credit card companies.

Consequently non-European credit card companies are taking over the mobile payment markets, working with non-European mobile platform holders. This means that European payment solutions like operator billing and direct debit payments are facing new challenges in the mobile field.

One of the main challenges hindering the integration of European payment systems to the mobile platforms is the fact that there is no standardized way to do that. Each payment solution requires different information and often also uses different technological solutions for transferring the payments.

There is a need for single solution uniting the different payment solutions in a way that they can be easily integrated to different mobile platforms. Furthermore, a more unified European pressure is needed towards non-European mobile platform holders forcing them to integrate the European payment solutions in their systems. In addition to already existing European payment models the focus should especially on introducing eSEPA payments in mobile platforms.

PRECON-DITIONS

POLICY GAPS

European creators of mobile content should be encouraged to self-regulatory actions in protection of minors.

At the moment, all relevant mobile platform holders own proper protection of minor systems. These systems are usually based on different classification mechanisms. European creators of mobile content should thus be encouraged to self-regulatory actions in protection of minors.

Actions

- **Creation of a working group**
- **Definition of technical standards and development of API**

VAT regulation must not lead to double taxation.

The double taxation has emerged from different approaches on the taxation of virtual goods within games. The new VAT regulation will be implemented in the European Digital Single Market Area from the beginning of 2015 and EGDF is currently discussing different stakeholders about the key methods to tackle the issues related to the double taxation.

Actions

- **Discussions with EC and member states**

The Pan-European licensing for music should make it easier to use European music for European mobile content.

The Pan-European licensing for music should make it easier to use European music for European mobile content. Although the differences among the game markets within Europe cannot be 100 percent lifted EGDF supports the license for Europe processes started by the European Commission focusing on lifting of legal barriers and complications.

Actions

- **Legal study & regulatory actions**

European actors should be able to bill for their content as a web service, not be forced to use Non-European closed ecosystems for payments.

Although billing could be a possibility for European Telcos to re-enter the value chain we must admit that current closed payment ecosystems do have a better impact on payment methods from a developer's point of view (Apple). European actors should be able to bill for the content as a web service and not forced to use non-European ecosystems. It thus seems to be the right time to get into contact with SEPA for the next round, especially concerning micro payments.

Actions

- **Information to game developer community about SEPA**
- **Discussions with SEPA**

The European copyright framework should secure that we do not lose early European digital cultural heritage published through mobile stores.

The current legal framework for the preservation of European digital cultural heritage does not fully acknowledge the existence of mobile games. Consequently, the legal barriers hindering the preservation of digital content in memory institutions should be cleared.

Actions

- **Legal study & regulatory actions**
- **Research to be done**

PRECON- DITIONS

STANDARDS GAPS

Game developers should follow standards development and be ready to contribute

The game developer community is composed mainly of small SMEs which have limited knowledge of standardization and limited resources to spend on this issue. However, we have had many examples of damages and costs due to lack of standardization or monopoly of proprietary solution, e.g. in the game engine domain. Moreover, if cloud solutions become real, game developers interest will be confronted with operators and cloud provider strategy. Therefore it is necessary that the game community unify their forces to define and defend their common interests in standardization.

The recommendation from this project is to create a forum of interest for mobile game developers in charge of defending common interests. The forum would be active at least on level I activities, i.e. based on medium term market interests, to define use cases and scenarios, and produce commercial and technical requirements according to these scenarios. The forum will be active with dedicated staff in SDOs to introduce these requirements and follow up the specification process according to those requirements.

Actions

- **Development, forum creation, input requirements to standards**

Developers need to be better informed on and more involved in HTML5 implementation.

The idea that developers need to be better informed on and more involved in HTML5 implementation has been one clear finding of the Mobile GameArch project. In practice, this should mean that developers need to be much more engaged in browser development, to be active and vocal, ensuring that features vital to their production actually make it into the latest browser versions.

However, without clear mutual benefit and support for discoverability, HTML5 games carry no greater promise than those of Flash games. One imaginable way of supporting game developers' input to browser implementation is to try to build consensus on sets of minimum browser features and performance.

Actions

- **Surveys, information aggregation and dissemination**

Standards on payment solutions integration are needed.

In the area of payment solutions, the specific gaps are related to the procedure according to which payments are processed and to the information required for these processes. Furthermore, there is a demand for technological standards on how to integrate in a mobile app different payment solutions

Actions

- **Research**

Common, good practice for in-game micro-transactions for children is needed.

The introduction of free-to-play business model is one of the main corner stones behind the recent boom of the European mobile ecosystem. However, a common, good practice for in-game micro-transactions for children is needed. To monitor an industry-led initiative which could attract attention on a European scale and lead into a recognizable discussion process, information should be gathered both from member states and internationally. Concerning the legal comparative approach it needs to be indicated, that Europe is strong enough in supporting the protection of minors on a legal level. Therefore the application of the regulations of protection of minors in general civil laws could be a doable solution to the problem.

Actions

- **A research project, and ongoing discussions and contributions**

PRECON- DITIONS

INFRA- STRUCTURE

Mobile cloud-based games need definition of management of application components, surrogate discovery, and security.

We identify three steps in cloud development for mobile games: optimizing server resources to share common assets between gamers, and adapting game engines accordingly; game engine evolution, evolving the cloud for mobile games, including all required technologies and architecture, and optimizing game engines for cloud. These steps would require/benefit from: setting up a forum representing the interests of game developers in standardization, interaction between this forum and other stakeholders and standardization of some interface and APIs.

Actions

- **Adapting game engines to cloud, provide requirements to standards**

HARDWARES

Developers need to be better informed on and more influential in hardware development processes.

In our gap analysis, we found that game developers need to be better informed on and more influential in hardware development processes.

Actions

- **Information aggregation and dissemination, establishment of manufacturer - game developer contact node**

Knowledge of hardware manufacturers' relations with game developers is needed.

We found that too little is known - even for central actors and organizations in this field - about hardware manufacturers' relations with game developers. As we are arguing that the lack of links between manufacturers and European mobile games developers are holding both parties back, it could be that one or more of the organized manufacturer clusters in Europe could not only support or take part in the survey work, but also serve as central parts of the coming, strengthened links.

Actions

- **Survey**

OPERATING SYSTEMS

Developers need to be better informed on and more influential in development paths for operating systems.

Game developers need to be better informed on and more influential in development paths for operating.

Actions

- **Developers need to be better informed on and more influential in development paths for operating systems.**

“Once you talk to us, as developers, you need to [remember]... that whatever we can change, it will be in the market much later in the future.

- Gemma Paris, ARM

**MAKING
THE GAME**

**PRODUC-
TION**

Overviews for evaluating APIs are needed.

That overviews for evaluating APIs are needed has been a clear finding leading of this roadmap. Already a lot of volunteer resources exist, as has been shown and exemplified in our earlier publications.

Actions

- **Information aggregation and dissemination**

Developers lack insight into how analytics providers process and retain their data.

To close the gap in knowledge, whitepapers could be published. Also, universities could start providing insights to game developers in the form of partnerships. Several faculties, such as econometrics could launch post-master programs to train professionals as analysts.

Actions

- **Whitepaper research**
- **Courses and training proposal**

Middleware and game engines brings the risks of lock-in and lock-out.

As use of middleware for part of a game production becomes the de facto standard, the threat of lock-in and lock-out risks grows immensely. Thus European game developers should be well-informed about the risks and these risks should be minimized by favoring open formats, multi-platform availability and selecting a common ground of must-have features

Actions

- **Information aggregation and dissemination**

Unified production formats are needed for tools and APIs to be synchronized across platforms.

Unified production formats are needed for tools and APIs to be synchronized across platforms. The effect of having to juggle multiple production formats is a multiplication of game data that quickly becomes unmanageable.

Actions

- **Actions towards adaptation of common formats to tool, middleware and API producers**

**SELLING
THE GAME**

MARKETING

The European mobile consumer's awareness of available games is low.

There is currently no consumer event informing the general audience about games, nor is there any trustworthy source about the history and the culture of mobile games. Furthermore there is no comprehensive methodology today, which allows parents to evaluate the content of games.

Actions

- **Feasibility study for center for mobile games**

Developers must make informed decisions on what markets to address and how to reach them.

Optimizing distribution and developing partner networks in large non European market areas can be instrumental to the success. Therefore, exchange of information on reliable partners and success stories can be very beneficial for the European mobile games industry.

Actions

- **White-paper and website documentation on international distribution**



“ Only people designing and developing games can express these needs [characterisations of infrastructure, security, portability, performance etc] to influence the providers!

- Patrick Guillemin, ETSI

GETTING PAID FOR THE GAME

PAYMENT

A single payment solution does not exist.

The first step in any solution to this large problem is of course to remove any monopolization of payment procedures. Monopoly-based systems are only detrimental to creativity and competition, thus creating an unhealthy climate. Removing all enforced payment requirements from the locked-in and monopolized stores must therefore be prioritized as highest.

The second step is then to provide a European payment standard available to developers that offer a high degree of flexibility and stability regarding a multitude of different types of payments (micro transactions, subscriptions, etc). To facilitate such a solution, a large development project must be started that would aggregate many different payment solutions and make them easily available to game developers. As traction is acquired, this system should be lobbied to current holders of payment systems and digital storefronts, to be used in their system as well. Of course, with every proprietary storefront the owner must receive a portion of the sale, as it is today, and this system should facilitate that as well.

Actions

- **Development of standard European cross platform payment solution**

Non-European mobile ecosystems and distributors should integrate European payment solutions.

A dialogue on mainstreaming the operator billing in platform devices should be started between European telecommunication operators and global mobile platform holders. Furthermore, the mobile platform holders should be strongly encouraged to accept SEPA payments in addition to credit card payments.

Actions

- **Development of standard European cross platform payment solution**

The European Games Developer Federation (EGDF) is committed to the stimulation and development of a stable, vibrant and creative European games development sector that is competitive globally and recognized culturally.

The EGDF will act to advance the political and economic interests of the European computer and video games industry by providing a platform for collaboration and discussion between European institutions and game developers.

www.egdf.eu | Katariinankatu 3 00170 Helsinki Finland
t. +358 (0)9 4289 1606 | f. +358 (0)9 666 334