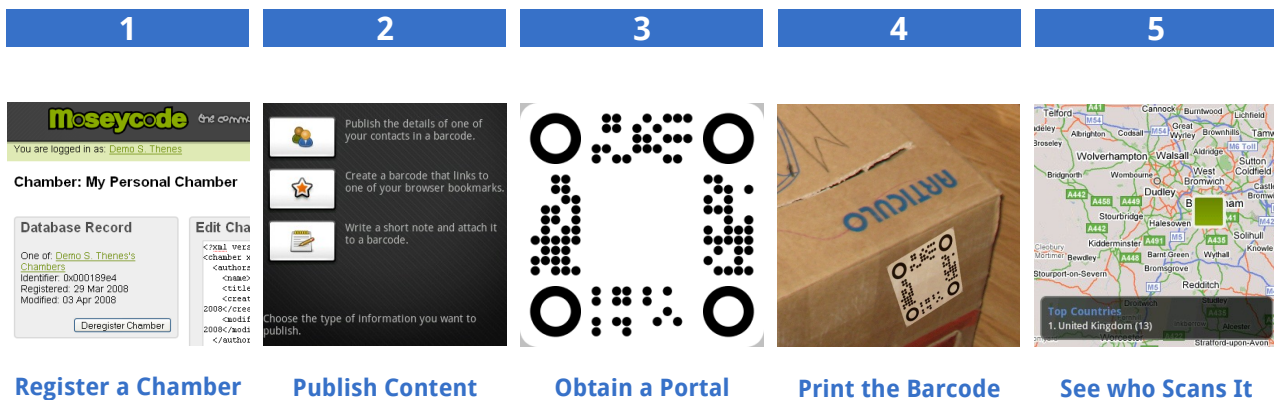


Moseycode

Introduction

Moseycode is a new barcode system for mobile devices. It was inspired by Android and began as a personal investigation¹ into multiple aspects of physical-digital interaction. It has since developed into a platform that is already being adopted by both enthusiasts and commercial organizations. Here's a brief overview of how the process works:

- 1 It is based on the idea of *portals* which are physical doorways into *chambers*. Chambers are digital repositories that can store a wide range of interactive digital content. Chambers need to be registered at the Moseycode website but registration is completely free for everyone.
- 2 Anyone can author their own chambers and fill them with photos, audio recordings and video. Information such as documents, web links and contact details can be included too. Publishing can be done directly from an Android phone.
- 3 They can then publish portals in the form of 2D barcodes. Others can view these barcodes using their Android phone and interact with them through a process known as *Augmented Reality* in which video imagery is combined with computer generated graphics in real-time.
- 4 Portals can be attached to anything, and provide the viewer with instant access to the chamber. In many cases they will contain 'live' data from the Internet. Obvious ideas include parcels (for delivery tracking), business cards (to supply contact details digitally), print adverts (for product information), but there are endless other possibilities.²
- 5 The original author can now track how often their portals are being opened and where it's happening. It's possible to obtain a live feed of this data for use with other applications.



Moseycode has been developed on the basis of providing a 'public good': all associated specifications may be freely implemented for any purpose and all key software components will eventually be released under a liberal open-source license.

As a result software developers are free to build applications around Moseycode in all sorts of ways. The Moseycode client application takes advantage of Android's model of activities and content providers to allow other developers to seamlessly integrate customized barcode scanning into their own applications. At the server end, the Moseycode server allows chambers to be dynamically populated by other author-delegated servers. These capabilities allow developers to build barcode-tracking applications in ways which (to my knowledge) have never before been possible.

¹ <http://www.tomgibara.com/android/moseycode/concept>

² I've already been contacted by people who are planning to use the system in ways I had never thought of.

The Moseycode System

The Moseycode system consists of several different elements. It includes:

- **A Barcode Symbology**
This specifies the way that data is encoded as a visual pattern. The barcode symbology is distinctive because, amongst other things, it has been designed to be extremely fast to scan and allows for the accurate registration of a barcode's position in 3D space. These are the main attributes that make it suitable for *augmented reality* applications.
- **A Chamber Markup Language**
Documents written in this language specify how a device should respond on scanning a barcode. The markup language is novel because this is the first time that an open language for barcode storage has been attempted in this way. It stores primarily semantic information so that a scanning application can adapt its behavior based on the capabilities of the device it's running on.
- **Web-Services**
A set of free public web-services is available for use by software developers to write their own applications that process Moseycode barcodes.
- **A Website**
The Moseycode website allows both casual users and developers to register their own Moseycode chambers that will store information about the portals they publish. It is expected that casual users will only use the tools available to them on their Android device (or in the future, third-party applications). Software developers will often go further and may allocate their own portals dynamically (by serving *chamber markup documents* from a web-server).
- **An Android Application**
The vital element that actually allows the barcodes to be read and interacted with. The application is exciting for a number of different reasons. The *Augmented Reality* nature of its interface is novel and opens up new possibilities for user interaction. It also makes effective use of Android's application model to serve as a component in other applications.

In keeping with the original system concept, every one of these elements supports 'location tracking'. This means that portals can be tracked from the moment of creation, through every individual scan and interaction. The intention is to track the portals and not the users; the Moseycode system *does not* build profiles of individual users. Since Moseycode is already a semi-live system the current Moseycode application implementation cannot use the mock location providers included in the SDK, so geo-location is currently inferred from the host IP address.

Uses of the Moseycode Application

Moseycode already provides a fully operational system to deliver information via a barcode to an Android phone in the form of URL links, contact details and text messages among others. Moseycode's distinction between portals (physical) and chambers (virtual) means that even after distribution, the contents of a barcode can be updated. This makes it ideal for the delivery of information that is subject to change (such as contact details).

However, distribution of information in this way is only one of many useful roles for Moseycode. The next page highlights some other uses.

Advertising

It is only recently that advertisers in Europe and the US have looked to augmenting print advertising with barcodes. The barcodes currently used for this purpose usually direct potential customers to related web pages and nothing more. Moseycode provides several novel benefits to advertisers that go well beyond that capability.

Firstly, the locational properties of the system can inform an advertiser where their barcodes are being read. This can be used to provide important regional information that can provide feed back for subsequent marketing decisions.

Secondly, with the flexibility of having an extremely large number of portals leading to a single chamber, advertisers can print barcodes that are unique to every magazine, or poster that the advert appears on. As with web-statistics, tracking which portals are used most frequently allows advertisers to pin-point their advertising spend on the most effective publications or hoardings.

Thirdly, and perhaps most uniquely, the advertising barcodes are more engaging. Rather than just directing a potential customer to a website, they are more immediate and more interactive. Imagine adverts in which the product 'pops out' in full 3D. You could look around the product by wandering around the advert and watching it through your phone. It might even be interactive, allowing you change styles and colors etc. as you examined the product.

Moseycode was designed to provide lots of 'white-space' that can be used to brand a portal's appearance. Combined with the features above, Moseycode provides advertisers with a unique ability to monitor print advertising more accurately and to deliver an interactive experience that sets their company apart.

Authentication

Moseycode uses author-designated web-servers to provide a unique degree of flexibility in the way that user/barcode interactions can be controlled. This, combined with data stored in the barcode, can be used to create a secure way for someone to log onto a website.

A visitor to a website simply associates their phone with the website (just once). When presented with a Moseycode portal at the site, they scan it with their phone, this in turn contacts the web-server and securely identifies the user. Using AJAX, the login process wouldn't even require a keypress. This approach could be used in addition to a user-name and password to provide increased security (since in principle, an attacker would need your user-name, password *and* phone).

There is even the possibility that the barcode could be 'location locked' so that using GPS, even if someone stole your phone, they would still need to view the barcode near to your machine to authenticate themselves.

Games

Imagine, as an example, a war based strategy game played on a tabletop where all the 'units' are represented by portals printed onto small square cards. Looking at them through your mobile phone you can see the 'virtual' pieces of the game, perhaps in full animated 3D. You can move around the tabletop battlefield to look at it from different angles. You can move your troops by physically moving the portals across the table.

A group of you might play a team game, sitting round the table, taking turns to move your own units. Each of you sees the game in 3D from your own perspective through your own phone. Each player might see different things within the arena. For example, players on the same team might see information about allied units. On the other hand opponents might not be able to see what your units are at all until they have 'line of sight' (this situation is commonly referred to as *fog of war* in real-time strategy games).

Even though you are all sharing the same game in the same room, it's possible to hide information from foes and allies alike in a way that isn't normally possible away from a computer screen. Though this particular idea is in its formative stages, Moseycode certainly provides the opportunity to introduce genuinely new game-play concepts.³

³ Game concepts may require better performance than is currently the case. This mostly due to the lack of a compiler and no proper camera device within the emulator. For more details see the closing comments of: <http://www.tomgibara.com/android/moseycode/releases/0.2.1/releasenotes>

Installing the Application

This application requires version M5-RC15 of the Android SDK. One apk file should have been received:

moseycode-0.2.1.apk

This contains the Moseycode Android application. This application allows a user to both scan barcodes and publish their own.

To install the Moseycode application simply use the command below.⁴ Together with an Internet connection, this should be all that is needed to allow the Moseycode application to run successfully.

```
adb install <path to the moseycode-0.2.1.apk file>
```

A very significant portion of the Moseycode Android application is the augmented reality (AR) nature of barcode interaction. This can only truly be judged on a physical device but can otherwise be mimicked using a webcam. Unfortunately the SDK does not at this time support emulation of the phone's camera. To overcome this I have produced some code that can transmit a webcam feed to an Android emulator over a TCP-IP connection.⁵ This is slow, but to-date remains the only way of developing Android applications that require real-time camera imagery.

Full details on how to set up a camera feed in this way are supplied in the documentation included in the application itself.

Using The Application

The instructions provided below are intended to help you to start using the application quickly. The application includes full end-user help which is available from the menus of most activities (click the **Menu** button with the mouse or press the **Page Up** key and choose **Help**). It is also available, as a convenience, through the following web page:

<http://www.tomgibara.com/android/moseycode/releases/help>

The application has two broad functions:

Publishing *which also includes*

Browsing Published Portals
Distributing Portal Barcodes
Tracking Existing Portals

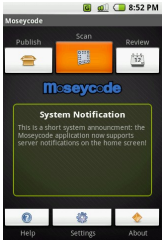
Scanning *which also includes*

Interacting with Portals
Reviewing Scanned Portals

The application's functionality has been structured into *activities* as encouraged by the Android platform. The result is that each of the functions listed above is supported by a single activity (together with a small number sub-activities as necessary). In the abridged documentation below, each activity is listed in the approximate order that you might typically encounter them.

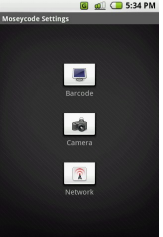
⁴ Please refer to the SDK documentation for full details about installing and launching the emulator.

⁵ This is openly available for all other android developers to use, see:
<http://www.tomgibara.com/android/camera-source>



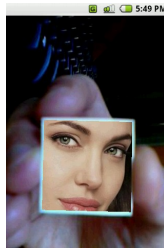
Home

This is the opening screen of the application. From here the main actions of “Publish” and “Scan” are available. You can also review scanned barcodes . A server status message is also displayed.



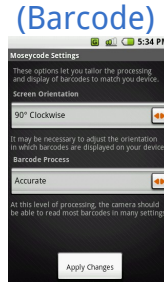
Settings

There are three groups of settings: Barcode (scanning and display) Camera (video sources) Network (web-service addresses) Consult the complete end-user help for full details.

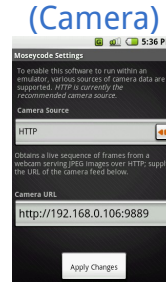


Scan*

This is the activity that allows you to interact with any Moseycode portals you find. Activating a portal requires pressing the center of the direction pad.



(Barcode)



(Camera)

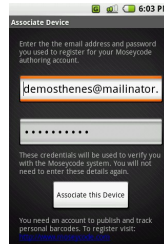


(Network)



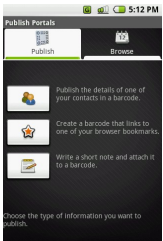
History

Allows you to review the barcodes you have previously scanned. Menu options are available for removing barcodes from the history and for clearing it entirely.



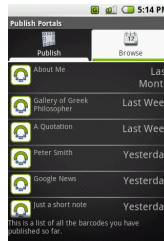
Associate/Disassociate

Enter valid credentials to associate this device with the Moseycode server. This must be done before you are permitted to publish or track any portals.



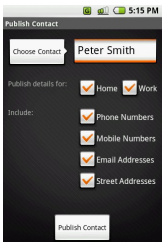
Publish (Options)**

Lists all of the sources of publishable information on the device. At present there are three; select one to publish a new portal. Click on the tab labeled “Browse” to list the existing portals.



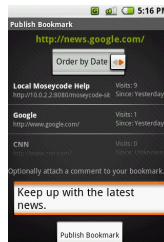
Publish (Browse)

Displays a list of the all portals you have already published, ordered by date. Click on a portal for further details. Click on the tab labeled “Publish” to see the publishing options.



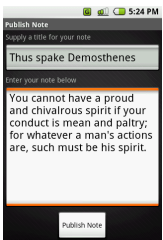
Publish Contact

Choose a contact to publish. Select the information to be included and then click “Publish Contact”. The new portal should appear in your list of published portals.



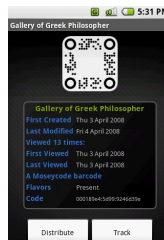
Publish Bookmark

Sort the list of bookmarks to help you find the one you want. Click on it, then optionally add a comment before clicking the “Publish Bookmark” button.



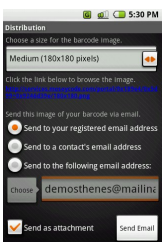
Publish Note

Simply type in a title followed by any text that you want to attach to a new portal before clicking the “Publish Note” button. The new portal will become available to browse.



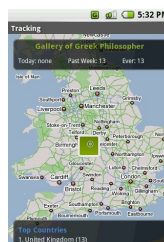
Details

Shows the details of a published portal. From here you have the option to “Distribute” or “Track” the portal.



Distribute

Allows you to send images of your portal barcodes via email for printing or alternative transmission. *Absence of email support in the current SDK means that no emails are sent in this release.*



Tracking

Viewing statistics for the selected portal will be marked on the map. Scans with known coordinates will be overlaid onto the map which can be panned and zoomed.

* Initially, directs you to “Settings”

** If not associated, this will start the “Associate” activity

Known Issues

There are a number of known issues, this is after all a pre-release application running on a pre-release operating system. Nevertheless, every effort has been made to minimize the likelihood that you will encounter an error.

The most probable risk is that the web-based Moseycode service might suffer an interruption. If this occurs, you will most likely receive a notification of the problem when the application starts-up in the form of a red error notification. A fault of this sort (whether you are notified or not) will manifest itself as barcodes not appearing to scan, or errors occurring during publishing related activities.

In addition, there are rare instances where packet loss or some other temporary glitch occurs in Internet transmission. If this occurs during publishing, then it may result in a 'lingering' progress meter. This is nothing more than an inconvenience, simply clicking the **Back** button should remedy the problem.

There are several other issues of less significance, including a number of documented shortcomings that arise from bugs in the current Android SDK. A full list can be obtained from the end-user help.

The only functionality that is truly absent from the application at the present time is the ability to delete and edit published portals (though this is possible via the Moseycode website). Distributing portals via email will fail due to the absence of email support in the current Android SDK.

Further Information

As stated in the introduction, Moseycode aims to be an open and public system that empowers individuals to both publish and consume barcodes, and which gives software developers an effective platform for building collaborative, location-aware barcode applications. Consequently there is already a growing volume of resources available to Moseycode adopters, some of which may prove interesting:

YouTube hosts some early demonstration videos:

http://www.youtube.com/profile_videos?user=tomgibara

Documentation for Android developers using the Moseycode portal history:

<http://www.tomgibara.com/android/moseycode/releases/0.2.0/content-providers>

Documentation for Android developers integrating Moseycode scanning:

<http://www.tomgibara.com/android/moseycode/releases/0.2.0/intents>

The fundamental Moseycode symbology specification:

<http://www.tomgibara.com/android/moseycode/symbology>

A specification for the encoding of Moseycode data:

<http://www.tomgibara.com/android/moseycode/encoding>

Documentation for the web-services on which Moseycode relies:

<http://www.tomgibara.com/android/moseycode/webservices>

A URL form of the Moseycode data for compatibility with other barcode systems:

<http://www.tomgibara.com/android/moseycode/url-form>

Preliminary documentation for the chamber markup language:

<http://www.tomgibara.com/android/moseycode/markup>

Future development

It's just the beginning for the Moseycode Android application. Here are just some of the things planned for future releases.

Reinstating full 3D rendering of barcodes

This was disabled for Android M5 release due to changes in OpenGL support. Development is also awaiting the availability of standard model data formats for Android.

Cross-encoding Moseycode barcodes

This is to allow the information in Moseycode barcodes to be published in barcode systems which are broadly adopted but have simpler URL based semantics.

Exposing more content providers

Allowing other applications to publish barcodes on behalf of the user, and obtain all relevant details will enrich the integration opportunities for other applications.

Touch based barcode interaction

When any portal is being displayed on the device (within the History or Publish activities say), the user should be able to launch them and view them interactively using motion sensing and touch.

Publishing a wider range of information

This is awaiting the availability of more publicly available Android applications.

Time series tracking and fuller analytics

Making more barcode scanning statistics available will make the system more useful.

Improving barcode detection

It is possible to improve several aspects of the current detection algorithms to provide more faster and more robust detection.

Fully scriptable barcode interactions

Authors will be able to control how their barcodes react to movement and to the adjacency of other barcodes. Requires scripting language support (eg. JavaScript) in Android.

Customizable Portal Designs

The Moseycode symbology was designed to be customizable by adding decals, messages and color. Users need the tools to make this easy.

Tom Gibara May 2008