

Prime Research Paper

Pacific Rim Undergraduate Experiences program (PRIME) is an international summer internship program, launched in 2004 with support from the National Science Foundation in the University of California, San Diego (UCSD). Key components such as research experience, summer at an International host institution, dual mentors, and International and Cultural Experience delivers an intensive nine-week research opportunity to prepare students for the global workforce of the 21st century. This research document extends the previous undergraduate students' developments by covering additional hardware such as PlaceSticker and Android Development.

In 2010, Dr. Haga of the UCSD Bioengineering Department formerly mentored PRIME alumni; UCSD undergraduate student, Kevin Nguyen developed a Multi-Touch Table with Frustrated Total Internal Reflection (FTIR) and Diffused Illumination technology (DI). The advantage of the FTIR is that there is a strong contrast between a touch and the background due to the pressure approximation. If the user is not applying pressure to the surface, the Diffused Illumination will be able to detect the soft touch. By combining both technologies, the disadvantage of each technology will be reduced to minimum with thorough sensor requirements.

In 2011, the Deputy Director at the San Diego Museum of Art (SDMA), Dr. Haga, and Mrs. Haga, took interest in the technology to present their photography collection at Tokyo Photo 2011. In addition to that, Ideum, a leading company advanced in integrated large-scale multi-touch tables and displays for museums, educational institutions, government agencies, and Fortune 500 companies, offers an open-source

advance multi-touch framework called Open Exhibits. Open Exhibits is an Adobe Platform based on frameworks that were released in November of 2010. UCSD Undergraduate students, Wesley Hsu and Lance Castillo took the advantage of Open Exhibits and developed software called "*My Gallery*" based on the Open Exhibits framework in Adobe Flash environment. *My Gallery* allows the user to go through a collection of 50 photos representing MOPA's collection and submit them to be viewed by others.

In 2012, Dr. Shimojo of the Osaka University Cybermedia Center professor and the executive researchers at the National Institute for Information and Communications Technology (NICT), Dr. Haga, and Mrs. Haga reached a decision to use PlaceSticker, multi-touch table, and android technologies to present in San Diego Museum of Art's (SDMA) artworks. UCSD undergraduate student, Michael Yao and Scott Mo developed the *Mobile Gallery* and the *Puzzle Game*.

The *Mobile Gallery* is a project combined with hardware and software for the SDMA. Visitors can select exhibition artworks from the multi-touch table placed at the entrance of the SDMA. By placing their hand-held android mobile device onto the multi-touch table, it will synchronize and the user is able to drag the artwork icon towards the android mobile device.

Upon ready, visitors select "Create Tour" on the multi-touch table and "Update Tour" on the android mobile device to transfer selected artworks information to android

mobile device.

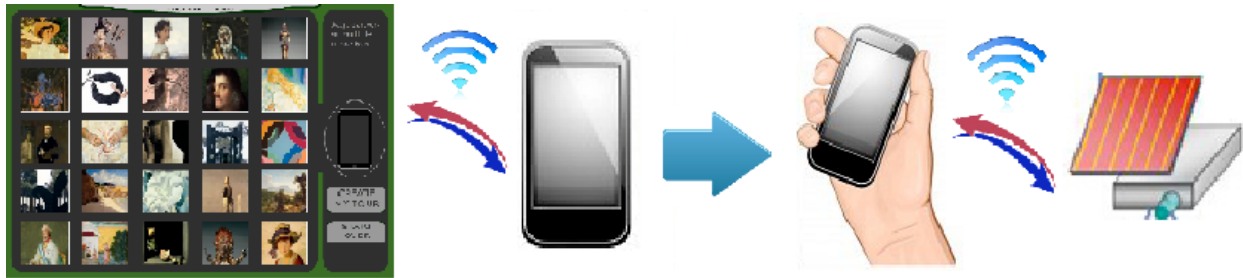


Figure 1: The flow of *Mobile Gallery*.

Android mobile device screen will provide a tour list displaying the artworks and locations for the visitors. Once the visitors are in the range of the artwork they selected, android mobile device will receive identification and distance update. Android mobile device will send out sound or vibrate notification based on visitors' default settings. After that, android device will bring out the artwork photo, video, and metadata. Once the visitors leave the appropriate radius for the artworks, the previously visited artworks information will be erased and the tour list will be re-updated. This is done based on the artwork's policy, which aims at preventing any unauthorized usage on the artworks.

There are several developments in the *Mobile Gallery*. The multi-touch table placed by the entrance of the SDMA is developed in Adobe Flash with Open Exhibits SDK. The server and client development is developed in Java with Extensible Markup Language (XML) parser SDK and operates on the multi-touch table. The mobile application development is developed in Java with Android SDK, PlaceSticker SDK, and an open-source Android YouTube Player SDK.

During the development stage, the success to drive this project forward is to be on schedule. This is especially important in the new development where several backup plans are established in case the project falls behind the schedule. Due to the complexity of this project in relation to other software developments, this document will focus on the development process, encountered problems, alternative methods, and difficulties.

Nowadays, mobile applications using Global Positioning System (GPS) are considered state-of-the-art for location-based services (LBS). However, problems exist in environments such as indoors and underground. Wireless Local Area Network (WLAN) seems to solve these problems and provide large-scale indoor tracking. Information Services International-Dentsu LTD (iSiD), a Japan-based company engaged in the provision of information services provided a cutting-edge technology called PlaceSticker from their Open Innovation Laboratory.

PlaceSticker is a new technology that uses periodic radio signal with limited information content such as identification and location for position estimation. In addition, PlaceSticker can serve as a permanently active sensor with sensitized solar cell panel. Therefore, the PlaceSticker was used in our research, as well as, PlaceSticker Android Software development kit (SDK), which is also provided by iSiD. During the development process, a problem was encountered where the PlaceSticker Android API specification document was in Japanese. Consequently, necessary procedures were taken to ensure that the SDK was thoroughly understood. With background experience

on how to work with third party SDKs coupled with previous programming experiences, the interpretation was successful and comprehensive.

The PlaceSticker SDK is somewhat straight forward with Eclipse. Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and extensible plug-in system written mostly in Java. The android contains a code that is used to declare object and initiate PlaceSticker listener.

```
Private PlaceStickerReceiver receiver;
receiver = new PlaceStickerReceiver(this);
receiver.setPlaceStickerListener(this);
```

The *receiver* object concentrates on the PlaceSticker's radio signal and interprets the signal into PlaceSticker identification/distance. The PlaceSticker's identification and distance from a Device Position object can be determined using various methods.

```
DevicePosition position;
position.getNearestPlaceSticker().getId();
position.getNearestPlaceSticker().getDistance();
```

The getId method allows a program to label each PlaceSticker to an associate artwork and the getDistance method can specify and respond based on the distance between the PlaceSticker and the Android Device. To prevent unstable radio signal, hardware are tested and the distance for the artwork is maintained at 25 or less. It is noteworthy that, when the Android device is next to the PlaceSticker, it might return a negative distance.

PlaceSticker must implement onPositionChanged (DevicePosition position, int style) method to monitor the location of the android device. Additional methods such as

onResume and *onPause* must be overridden. The *onPause* method focuses on the situation where the user exits the application thus prompting the receiver to stop scanning. The *onResume* method prompts the *receiver* to start scanning for PlaceSticker. Unfortunately, PlaceSticker SDK has problems when restarting the receiver from *onResume* method. Nonetheless, this problem can be solved by recreating the receiver object and listener.

Android application was first designed to store artworks, video, and meta-data. It was easier to create a working model before stripping down any unnecessary codes and functionalities. At one point, the program reached 90MB and the installation process took up to 2 minutes. This level of efficiency was lower than expected. Nonetheless, as the project got closer to the beta version, we started to remove all the artwork pictures, videos, and meta-data. Instead of storing data in the phone, the application would receive data from the server and load pictures from the SDMA website, videos from YouTube, and meta-data from the server.

The flow of the mobile application is quiet simple to follow. The first phase has a welcome page with update and start buttons, as well as, a grid page inside the welcome page to display selected artworks.

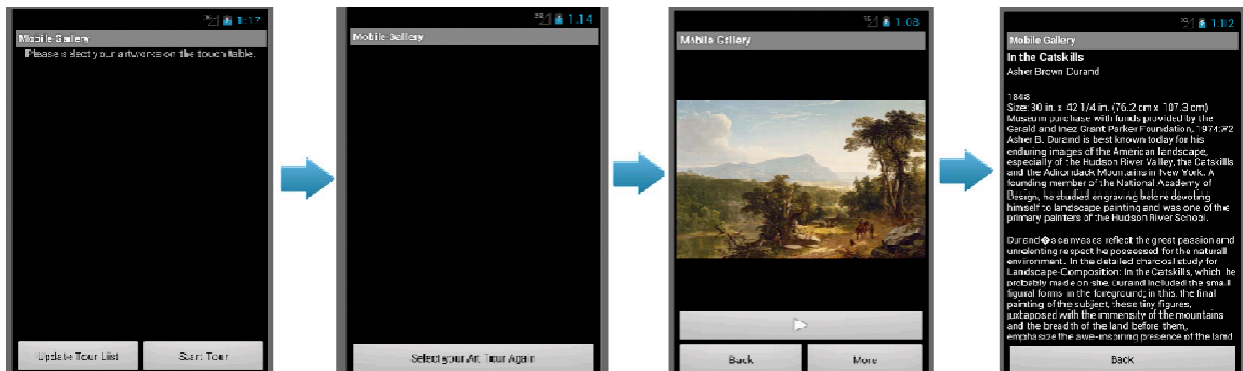


Figure 2: The flow of android mobile application.

The second phase is the start tour page to guide the user to their selected artworks with artwork locations. In addition, the second phase has a grid displaying artworks details and a reset button to restart the tour. The third phase displays the artwork under two conditions; the artwork is selected in the current tour list and the user is next to the artwork. The third phase also includes the video feed and meta-data. Artwork information is based on the radio signal from the PlaceSticker.

To display icons and artworks in the welcome page, the efficient method to use is gridView by extends BaseAdapter.

```

Private GridView gridView;
imgAdptr = new ImageAdapter( this, titleArr, authorArr, numOfArtArr );
gridView.setAdapter( imgAdptr );

```

Nonetheless, a minor problem was noted during the project in regard to the gridView update. Initially, it displayed 25 artworks with three repeated pictures. It was noted that the gridView updated the three visible columns and repeated these three columns. To solve the problem, updateGrid method was written to prevent this bug.

```

public void updateGrid();

```

The start tour page has a similar design as a gridView to display the current artworks in the tour list. It also provides the location of each artwork and notification. Notification feature is quiet useful in android device since it notifies the user about the artworks.

```

Private NotificationManager mNotificationManager;
Private Notification notifyDetails;
mNotificationManager =
(NotificationManager) getSystemService (NOTIFICATION_SERVICE);
notifyDetails = new Notification(R.drawable.icon, "Location Alert",
System.currentTimeMillis());

```

A custom method called onNotificationChanged (String ID, int Distance) is used to monitor different signals from the PlaceSticker. During the test phase, android phone runs the notification smoothly. But android tablet crashes when it reaches to notification. Since this is an android mobile application, notification feature is still used in the development.

The video page displays the artwork pictures, video, and meta-data. An open-source video SDK called OpenYouTubePlayer developed and shared on Google code was found.

```

String url = "http://www.youtube.com/movie";
Uri uri = Uri.parse(url);
String id = url.getQueryParameter("v");
Intent IVideoIntent = new Intent(Intent.ACTION_VIEW,Uri.parse("ytv://"+id)
,YouTube.this, OpenYouTubePlayerActivity.class);
startActivity(IVideoIntent);

```

This OpenYouTubePlayer SDK has been updated to work with the latest YouTube API changes. The main reason for preferring OpenYouTubePlayer SDK over YouTube SDK is because OpenYouTubePlayer is focused on creating Android activity that allows developers to play videos that are hosted on YouTube. The current project will not

involve recording, uploading, and sharing features. At this point, this is more appropriate under the current development until YouTube changes the API again.

Up to this point, the mobile application can run individually without communicating to the multi-touch table. This is to prevent the development running short on schedule or being unable to establish communication between the multi-touch table and the mobile application. The next step is to search for methods to run the multi-touch table, server, and client.

Dr. Reiko Kadobayashi of the Osaka Electro-Communication University, the Department of Digital Games professor introduced multi-user application based on Processing with a costumed PX library. Processing is an open-source programming language in Java built for the electronic arts and visual design. The environment is similar to Linux Vim with functionalities from Eclipse and the main library is the Core.jar. Several meetings were organized to talk about possible developments with the aim of combining the multi-user application in the current project. The meetings were intense and generated useful information on the program design and user interface.

```
px = new PX( this, port ); //create connection
px.setClass("Artwork", Artwork.class); //use artwork class
zones = new TUIOzoneCollection(this); //drop object zone
```

The Processing development used TUIO SDK to develop a multi-touch application. TUIO is an open framework that defines a common protocol and API for tangible multi-touch surfaces including touch events and other tangible object states. The final

conclusion is to hold-on to this for future development and focus on single multi-touch table user.

The multi-touch table uses Adobe Flash Professional CS5.5 with ActionScript 3.0 and Open Exhibits SDK. This is due to the fact that previous PRIME project was developed with Open Exhibits. The initial strategy was to study previous program developed by Lance and Wesley. However, the program couldn't compile properly due to the newly updated Open Exhibits SDK 2.0. Ideum rewrote the SDK from ground up and made several dramatic changes. The Open Exhibits SDK 2.0 is not backward compatible.

Open Exhibits SDK supports floc. It is the perfect development for the FTIR table with Community Core Vision (CCV). CCV is an open-source/cross platform for computer vision and machine sensing software. Since the interface does not include 3D model, Open Exhibits is one of the best options with the floc multi-touch table. Programming must provide codes that are convenient to read and adjust and this is facilitated by the loadUI function. All the buttons and shaders are created in the loadUI function. Coding syntax is different as well.

```
private var resetBtnCtr:TouchSprite; //create resetBtnCtr
resetBtnCtr = new TouchSprite(); //assign TouchSprite
resetBtnCtr.addChild( resetBtn ); //add button icon
resetBtnCtr.addEventListener( TouchEvent.TOUCH_DOWN, resetDown);
resetBtnCtr.addEventListener( TouchEvent.TOUCH_UP, resetUp );
addChild( resetBtnCtr ); //finalize reset control
```

This is a snapshot of the code from the Main showing how Open Exhibits SDK works.

Gestures such as touch down, touch up, and drag are straight forward to implement.

In the course of the project, we noted that server development was significantly complex. Although there are various ways to implement, there was limited information from SDMA's IT department. In this respect, Java server emerged as the best choice. In an enterprise setting and for e-commerce, applications must protect user's information, using encryption, authentication, and secure communications protocols. The most popular protocol is the secure Hypertext Transfer Protocol (HTTPS). E-commerce applications already have been using the HTTP over the Secure Sockets Layer (SSL). The Java Secure Socket Extension (JSSE), which is a set of Java packages that enable secure internet communications, is a framework and 100% Pure Java implementation of the SSL. With JSSE integrated into the Java 2 SDK, Standard Edition, version 1.4 (J2SE 1.4), building a secure passage of data between a client and a server running application protocol such as HTTP, FTP, TCP/IP is easier than ever.

A different option would be the phpMyAdmin; an open-source tool written in Hypertext Preprocessor (PHP) intended to handle the administration of MySQL over the World Wide Web (WWW). It allows the administration to create, modify, and delete databases, tables, fields, or rows. It handles SQL statements and manages users and permissions. The intent is to store user data, history, and input for future developments.

Apart from these two, an alternative choice is to use Ad hoc server/client which is provided by a company called Elixir Technology Pte Ltd. It is a database similar to MySQL that can be accessed from the web UI depending on how the Repertoire Server has been integrated with the application. This decision was abandoned due to SDMA's environment that is not suitable for such application.

These three options are not similar. In fact, each method serves a different purpose. After careful consideration and testing on the suitability of each alternative, the preferable option was to use Java server with TCP/IP. The application does not require the user to transfer and store any personal data or information. Therefore, it doesn't even need to use SQLite, which is an ACID-compliant embedded relational database management system contained in a relatively small C programming language (~275 KB).

```
ServerSocket sSocket = new ServerSocket(8880);
Socket sockets = sSocket.accept(); //Wait for a client to connect
ClientThread cT = new ClientThread(sockets); //thread for each user
new Thread(cT).start(); //Start the thread
```

This is the snapshot of the server application on the multi-touch table. The server collects user's input from the multi-touch table application before transferring selected information to mobile application. Each transfer item has information stored in the xml file by using parser SDK.

```
File fXmlFile = new File( xmlPath );
DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(fXmlFile);
doc.getDocumentElement().normalize();
```

Given the convenience of SDMA, the exhibition artworks can be updated, rearranged, or removed without any programming skills. For simplicity purposes, the administrator is only required to update artwork information from the multi-touch table's xml file.

Since the multi-touch table application and server have been developed, it is now time to revise the mobile application to allow communication over wireless connection. To perform this simple task, the client's code can be written within the mainActivity.java.

Constantly checking the server from mobile device is considered a heavy load to the mobile device. A separate thread is therefore created for that purpose.

```
Thread fst = new Thread(new ServerThread());
public class ServerThread implements Runnable {
    public void run(){...};
}
```

This is, however, not a heavy game application that needs to constantly acquire the server. An alternative process is to create a method, extend `AsyncTask<Void, Void, Void>`, override `onPreExecute`, `doInBackground`, and `onPostExecute` method. The `onPreExecute` and `onPostExecute` methods determine if the program is successfully connected and catch any `unknownHostException`.

```
protected void onPreExecute(){...};
protected void doInBackground(Void param){...};
protected void onPostExecute(Void...param){...};
```

The final decision is to allow the user have the option to respond to the mobile application. Once the user confirms the selections, the mobile application will run the necessary steps to provide a friendly user environment.

The second project is the *Puzzle Game*. This multi-touch table placed by the entrance of the SDMA is developed in Adobe Flash with additional open-source PaperVision3D SDK, Caurina SDK, and open-source PurePlayer.



Figure 3: The flow of *Puzzle Game*.

This is an application with options of adding artworks up to 36 artworks. The user selects an artwork to play puzzle game with the artwork user selected. Once the user is done with the puzzle, the application will display full image of the artworks with options of reading meta-data and watching a short video.

The process is to develop the 3D interface and convert into Open Exhibits to work with flosc and CCV. The 3D interface consists of 36 tiles based on the mouse's location to determine x, y, and z coordinates whereby, the 36 tiles will rotate to face the user.

```
static private var toDEGREES :Number = 180/Math.PI;
static private var toRADIANS :Number = Math.PI/180;
var position :Number3D = new Number3D( this.x, this.y, this.z );
var target :Number3D = new Number3D( targetObject.x, targetObject.y,
targetObject.z );
var zAxis :Number3D = Number3D.sub( target, position );
zAxis.normalize();
```

The 3D calculation is quiet complex and is divided into different methods. Papervision is a 3D engine built with ActionScript that enables developers to start working in 3D. The snapshot is not enough to explain the complexity of the camera and the rotation of tile objects. Nevertheless, with Papervision, the application became easier to develop.

The conversion from mouse events to Open Exhibit's touch events was harder than it anticipated. There were problems during the conversion that took days to resolve. The main problem was the coding requirement from Open Exhibit SDK. The application works on the multi-touch table without the Open Exhibits and floc. This is because the floc is replaced by native gesture and is no longer available on SDMA's multi-touch table. For this reason, it is unnecessary to convert from native to floc. However, multi-touch feature that allows multiple users to play the same puzzle still need to be implemented.

```
mouseOrTouch = img_xml.inputmethod[0].@input.toString();
Multitouch.inputMode = MultitouchInputMode.TOUCH_POINT;
evt.target.startTouchDrag( evt.touchPointID );
evt.target.parent.addChild( evt.target ); //bring forward
```

Adobe Flash ActionScript 3.0 provides multi-touch features similar to Open Exhibits. It is called "touch point" event. Flash runtimes also support an additional set of event handling API called gesture events to fit in the current development needs. The xml parser read the xml file and load mouse/touch control, file name, title, and description. When the touch control is "touch", the multi-touch feature is enabled.

There are few important notes in regard to the Multi-touch object from Adobe Flash. It is not possible to create a Multi-touch object directly from ActionScript code. If the developer attempts to call new Multitouch (), an exception is thrown. This multi-touch feature is not supported by SWF files embedded in HTML and running on Mac OS.

Each puzzle piece used is created manually though it can be done automatically. Automatic generation of puzzle pieces can reduce development process. The process

begins by loading the puzzle template, artwork, and resizing the puzzle template to be the same as that of artwork. This is followed by placing the artwork at the appropriate location on the setup frame. Lastly but not least, the individual pieces and mask are loaded.

```
imgLoader.width = fullImg.width;
imgLoader.mask = msk;
msk.alpha = 1;
var img: MovieClip = new MovieClip();
img.addChild( imgLoader );
img.addChild( msk );
imgArr.push( img ); //update img array
```

This is a high level description and sample code of how to automate puzzle pieces. The puzzle pieces scrambles to random location for user to play.

The last part of this application is the video using Colibri. It's an ultra-lightweight AS3 video player class based on pure ActionScript with no library assets required. It maybe lightweight but it has extensive features such as streaming preloader and thumbnail options.

```
var player:Colibri = new Colibri(videoWidth, videoHeight, videoURL,
autoPlayVideo, resizeMode );
```

The open-source Colibri is convenient and easy to customize. In addition, the stream values can be controlled by using instance.video.stream and setting the video buffer time.

In this research project, the software development is the key to success, especially when it comes to meeting specific needs of specific business. The key accomplishment of this research project was the combination of several stages of

software development. These included, analyzing the problem, gathering requirements for the proposed business solution, designing for the software-based solution, implementing the software, testing the software, deployment, and maintenance. These stages may collectively be referred to as the software development lifecycle.

The waterfall software development methodology also referred to as linear framework was used in this development process. In the research project, the waterfall software development methodology provided several benefits such as, disciplined process, forces to have complete requirements prior to start, and forces for analysis and design first. To overcome the drawbacks, daily meeting was conducted and weekly feedback was constantly provided from SDMA. Some of the drawbacks that were eliminated include prototyping, slow response to change, and high cost for unclear requirements. The methodology slowly shifted from waterfall software development methodology to incremental and iterative (IID) approach where additional functionality is implemented in each increment/release and repeat cycle of the design. This is built and tested under different version of android mobile device until the desired functionality is complete.

The research project was finished on time. However, one common problem that was noted is that, clients generally find it hard to give precise expression of their needs in a form that developers can understand. In this respect, it was recommended that for any future client, it is important to provide the general project overview, specific software development tasks, time frame, software development technologies, and project

documentation. This is the start of a successful software development and great experience for any future software project manager.

Work Cited

Android Developer. "WifiManager". 1st Aug. 2012

<<http://developer.android.com/reference/android/net/wifi/WifiManager.html>>

BETA ActionScript 3.0 Reference for the Adobe Flash Platform. 5th Aug. 2012. 21st July 2012.

<http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/package-detail.html>

Colibri. "lightweight AS3 Video Player". 7th July 2012.

<<http://blog.chrometaphore.com/2010/04/20/colibri-open-source-pure-as3-video-player/>>

Community Core Vision. 15th July. 2012 <ccv.nuigroup.com>

Elixir Technology. 10th July. 2012 <<https://sites.google.com/a/elixirtech.com/tutorial/>>

Jwei. "Incorporating Socket Programming into your Application" 27th Mar. 2010

3rd Aug. 2012

< <http://thinkandroid.wordpress.com/2010/03/27/incorporating-socket-programming-into-your-applications/>>

Keys. "android-youtube-developer." <<http://code.google.com/p/android-youtube-player/>>

Martin Kalttenbrunner. "PQ Labs", TUIO.org. 13th, Aug. 2012 www.tuio.org

Papervision3D, "Papervision3D on Lynda.com" 12th July 2010. 3rd June 2012.

<<http://blog.papervision3d.org/>>

Qusay H. Mahmoud. "Secure Internet Programming with Java 2" 27th July 2012

<<http://java.sun.com/developer/technicalArticles/Security/secureinternet/>>

SQLite. 29th July 2012. <<http://www.sqlite.org/>>

Wilkiewicz, Jarek. "YouTube." *Resources for Mobile Development with*. 6th, Aug. 2012
<https://developers.google.com/youtube/articles/youtube_mobileresources?hl=fr-FR>.