

Empowering Middle School Students through Mobile Applications and Social Networking

Melissa Serrano, Alain Edwards, Sifat Islam, Ravi Shankar, Iris Minor, Susanne Lapp

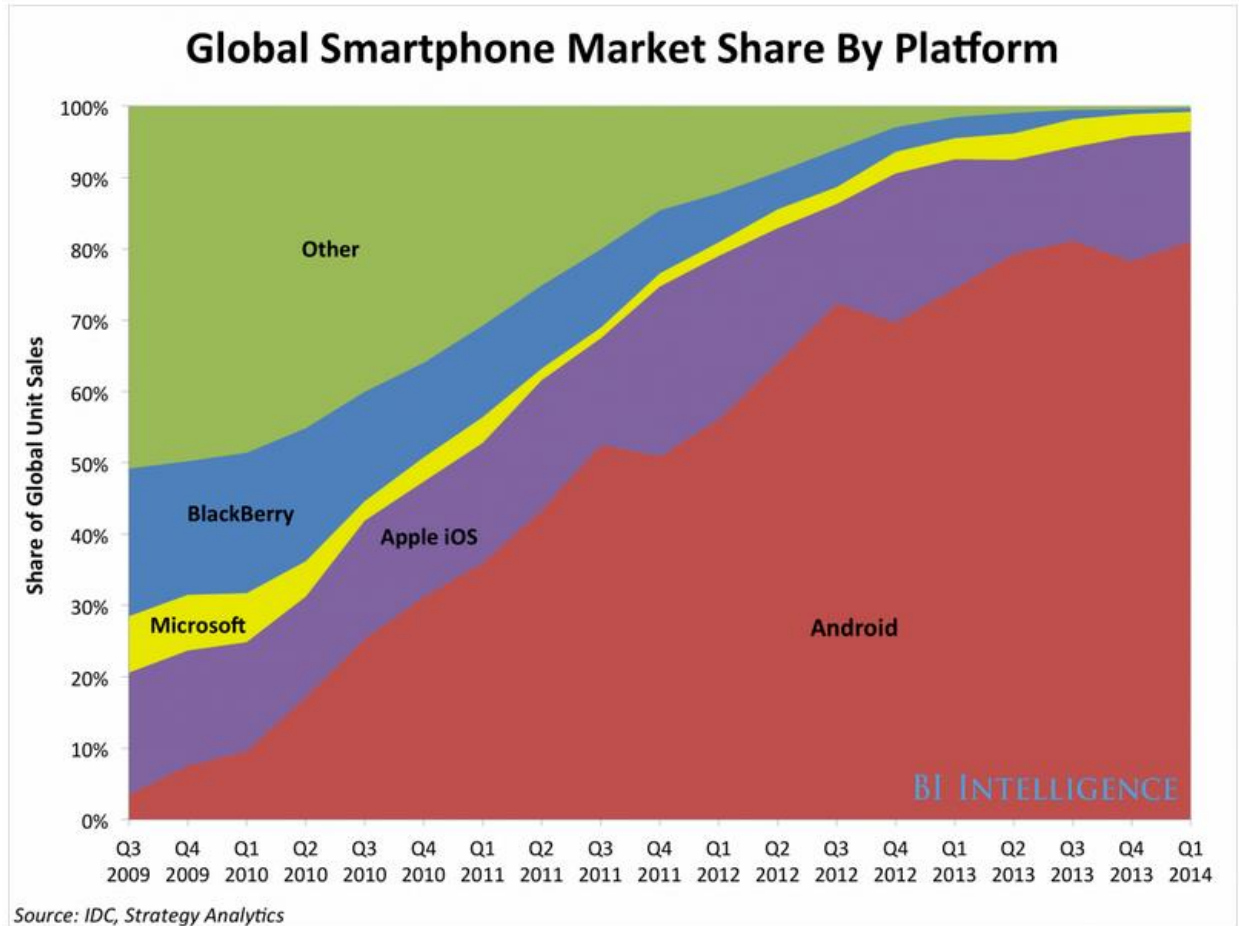
Abstract

During the fall 2014 semester at FAU a group of undergraduate students created an app to empower middle school students called Cityville, which had the highest ranking among judges at the end of the semester. The pilot version of the Cityville app allows users to add community events and report locations of concern in their community. We propose to enhance the pilot version of Cityville by allowing users to create a personal profile using the Processing programming language, integrate the app with their preferred social media account. We have included mockups of our proposed changes. We feel that the enhancements we have proposed are extremely important to obtain the amount of usage we are wanting out of the students with the app. Without a significant amount of usage it would make our research data very sparse and likely inaccurate. Upon implementation of our proposed enhancements we will have achieved an Android app which will empower middle school students as well as provide us with useful data that we can use in our analysis.

Background

The expansion of internet use on mobile devices and the vast accessibility to technology since the turn of the millennium has greatly empowered our society. Twenty years ago a child might have asked his dad how to fix the chain on his bicycle when it fell off, now that same child won't wait for dad to get home from work— he'll look it up on YouTube! Having an endless sea of knowledge at our fingertips is a great empowerment tool. But does everyone realize how empowering the internet and technology is?

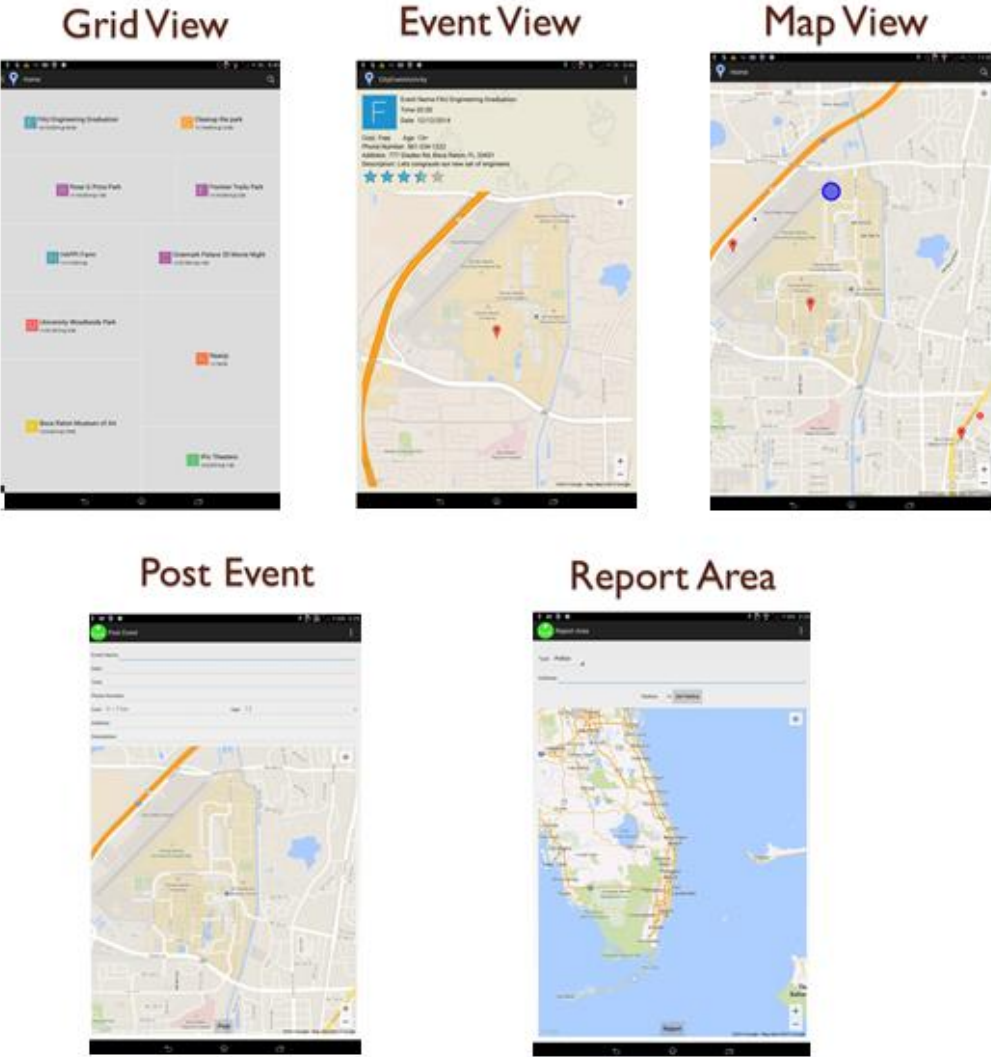
Bill Gates once stated, "As we look ahead into the next century, leaders will be those who empower others." During the fall 2014 semester at FAU, Professor Shankar's undergraduate Android Application Development class showed that all of us are leaders, because every one of us has the ability to empower others in some way. The development of Android applications (versus other platforms) offers a lot of development support and an active open-source community. Android also has the largest user community compared to other mobile platforms like iOS, Blackberry, and Windows.



With suggested topics from community youth counselor Iris Minor, the students developed many Android applications designed to empower students. <https://github.com/RShankar/Empower-App-Cityville-Fall-2014> Among the apps created the Cityville App was ranked the highest by a group of academic, high tech, and movie industry professionals at FAU, and therefore chosen to be used in a study to analyze STEM interest amongst middle school students. With our contributions, the Cityville App will empower middle school students by allowing them to add community events or safety reports to Cityville as well as, learn basic programming skills while creating and sharing their profile, and integrate Cityville with their preferred social media app. The usage of Cityville by the middle schoolers will then be analyzed by PhD candidate Sifat Islam to study the students' usage as it pertains to STEM interest and activities.

Pilot method

Among the Android apps developed by Professor Shankar’s undergraduate class was ‘Cityville’ created by Alain Edwards, Adam Moulton, and Lance Williams. Cityville’s pilot method features a dynamically loading grid view, interactive Google Maps view to view events and safety advisories, a form to allow users to post a city event, and a form that allows users to report safety advisories. The Java programming language was used to develop the core functionality of the Android application and it is extended by the use of the Parse.com and Google Play Services plugins. Google Play Services API was used, specifically Google Maps API. Google Maps API is used to allow for an interactive map view. The interactive map view allows users to see locations of events and safety alerts such as locations of police, fires, and traffic posted within the Cityville app. The Parse.com plugin is used so that all the event and safety advisory data can be stored in the Parse database. Parse.com is also used for reporting and tracking the usage of the Cityville app. Some of the data the pilot version of the app currently tracks are the number of events posted within a specified time period, number of safety reports posted within a specified time period.



User Interface layout of CityVille’s pilot version

Proposed Enhancements

To enhance the pilot version of Cityville developed by the undergraduate students we propose to allow users to learn basic programming skills while creating and sharing their profile and integrate the app with their preferred social media account.

Personalization

The app will allow the user to create a personal profile using the Processing programming language. For the middle school students we will have the Android Processing App installed on the Nexus 7 tablets. Melissa, a masters student, created a tutorial series for them with a skeleton code showing them the basics of programming in Processing. The tutorial series is available on the [YouTube Channel EmpowerMe - Melissa Serrano](#). This Channel is linked to our [EmpowerMe GitHub repository](#) where all code and documentation we created that is related to this paper have been uploaded.

The lesson plan that Melissa created walks students through using the Android Processing Integrated Development Environment (IDE) on an Android tablet.

Tutorial Outline – Using Processing Programming Language to Create a Profile for CityVille

These video tutorials are based on the tutorials and examples on <http://processing.org>

Enclosed with this outline I have included sample code as we progress through each tutorial as well as the complete sample code.

- 1) What is Processing, Using the Android Processing IDE App:
 - a. Brief Introduction and Explanation of the programming language
 - i. <http://processing.org>
 - ii. <http://processing.org/reference>
 - iii. <https://processing.org/tutorials/>
 - iv. <https://processing.org/examples/>
 - b. IDE buttons, compile, build, save and find file location on tablet
- 2) Creating Your Profile – Coordinate System and Shapes:
 - a. Setup, draw, screen size, relative coordinate system
 - b. 2D primitive shapes: ellipse, rect, line, point
- 3) Creating Your Profile – Color:
 - a. RGB URL Resource: http://www.w3schools.com/html/html_colors.asp
 - b. Background, stroke, fill, random
- 4) Creating Your Profile – Text:
 - a. Name & Quote
 - b. Font, textAlign, textSize
- 5) Creating Your Profile – Interactivity & loadImage():
 - a. mouseX, mouseY, mousePressed
 - b. Image, loadImage
 - c. Talk about other capabilities of Processing code
- 6) Upload Processing code file:
 - a. Copy and Paste to text document
 - b. Drop file in Google Drive Folder: SUBMISSIONS

Tutorial Lesson Plan Outline

Over the course of 3 weeks, in 1 hour sessions we will show the students how to draw on a digital canvas by providing resources along with video tutorials and sample code for them to

build off of. We relate the coordinate system on a computer screen to the cartesian coordinate system they are used to in school. Then they draw shapes (rectangles, ellipses, lines, and points) and text.

```
//Tutorial #3
//point(x, y);
down = 120;
for(int i = 0; i < (boxHeight)/40; i++)
{
    down = down + 12;
    //rect on left side
    rect(boxStart + 5, down, 40, 50);
    //point on outside of box
    point(boxStart- 5, down);
}

//rect on right side
down = 120;
for(int i = 0; i < (boxHeight)/40; i++)
{
    down = down + 12;
    rect(boxWidth + boxStart -45, down, 40, 50);
}
```

Snippet of Sample Processing Code Provided to Students

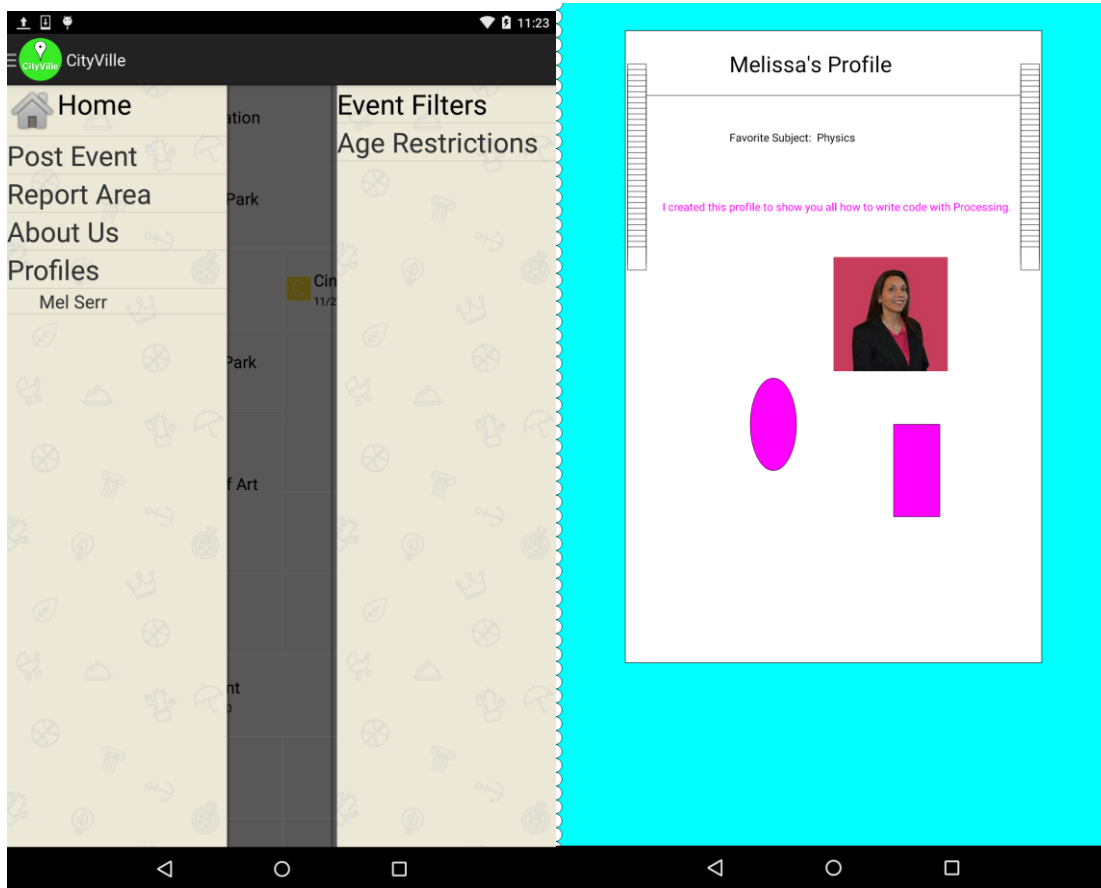
They also learn how to use a color picker and programmatically set colors using the rgb(red, green, blue) format commonly used in programming. The students even get exposure to more advanced aspects of programming such as setting permissions to allow programmatic access to web URLs, which they use to display an image on their personal profile; as well as interactivity programming using Processing keywords to detect finger touch locations using (x,y) coordinates and providing defined functionality when the user touches the screen.

```
//Tutorial #5
ellipse(horizontalCenter - 130, verticalCenter, 100, 200);
rect(horizontalCenter + 130, verticalCenter, 100, 200);
if(mousePressed)
{
    ellipse(mouseX, mouseY, 100,100);
}
```

Snippet of Sample Processing Code Provided to Students

This will empower the students to learn how to program and write their own profile in code. Once all the students complete their profiles and upload them to a shared Google Drive folder Melissa will package them with the CityVille code in an updated APK file to provide to Iris, this way the students can see their profile integrated with the CityVille app as well as those of their

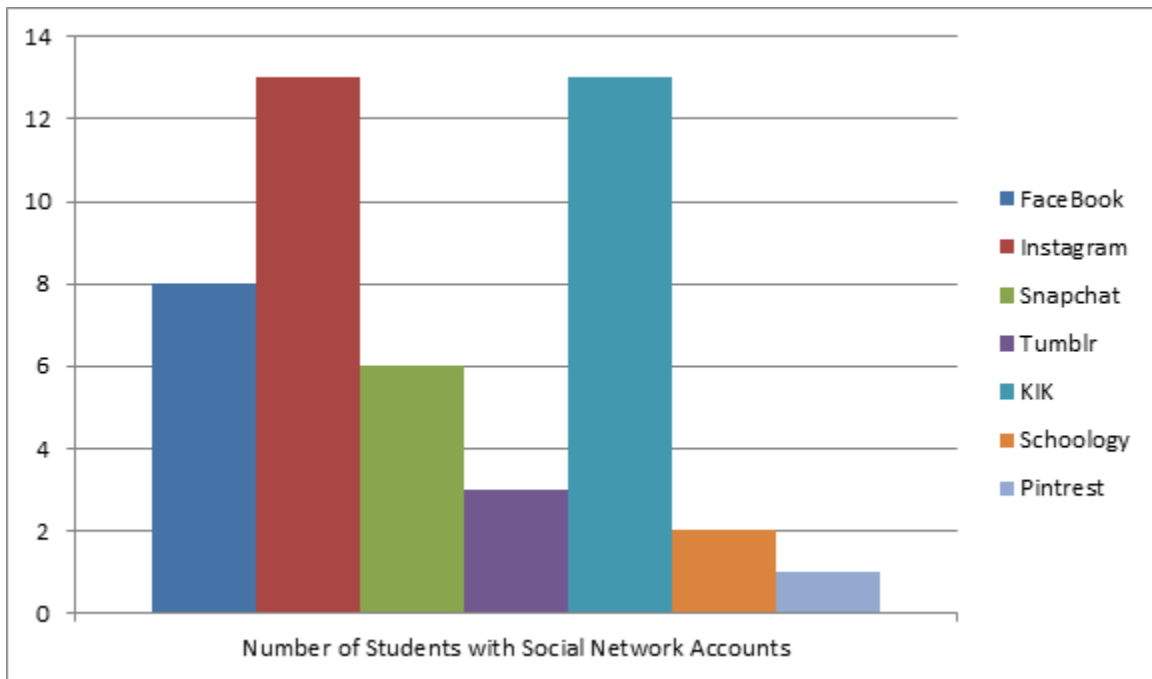
classmates. Giving the students the opportunity to see what they created with the code they wrote should get them excited about programming and interested in learning more. Sifat will use data retrieved from the students profiles to analyze how much of the provided sample code they changed, thus measuring how empowered they are through coding. Aspects of the students profile code that will be measured consist of how many lines of code the student programmed for their profile, which API functions did they use for it, and any textual information they include in their profile. We asked students to textually write in code the answer to a question, "Why are you interested in STEM?"



CityVille Menu with New Profile Feature

Social Media Integration

Most importantly, we will enhance the app to allow students to integrate the Cityville app with their preferred social media account. By doing this the students will continue using the social apps they always use and more readily upload events they are chatting about in their social networks to Cityville. With this feature we will be able to analyze users' connections in their social networks and relevant conversations they are having by recording the event that the student accessed this social network and allowing access to it. The graph below illustrates the results of a survey given to Iris's group of 34 middle school students. Each student indicated that they have an account for each applicable social network.



Survey Results of Student’s Social Media Usage

The results of the survey show that there was a tie for the majority of students using Instagram and KIK. Since we need to collect textual data to analyze students interest in STEM, we decided that Instagram would provide a limited amount of data for analysis. Based on these results we decided to integrate CityVille with KIK. The goal is to enable the student to share and add a comment related to the profile page they created with the Processing programming language, as well as other communications which can be used to analyze STEM interest. Kik is a smartphone messenger based on usernames, not phone numbers. Over 200 million people use Kik, including 40% of US youth. The KIK API provides a pop-up conversation functionality. Implementing this feature within the CityVille will encourage the students to use CityVille more and share events, reported areas, and profiles with their connections on KIK. This way each of the shares and text can be stored in the Parse.com database so that we can later analyze it.

To measure how empowered the kids are we will track connections made within their integrated social media account, events and reports uploaded, as well as the extent of their programming in the profile they create.

Results

When Melissa began to use the pilot version of CityVille, she realized that Android had made significant updates to it’s operating system since CityVille was developed and last used (approximately three months prior). CityVille was original developed for Android Version 4.4.2

(API 19), by the time she imported CityVille from GitHub into her Eclipse workspace, the Nexus 7 tablet that we were using to develop with (and the middle school students would be using) was running Android Version 5.0.2 (API 21). When CityVille ran on the tablet it would not display any maps because of the change in versions and updates to the Google Maps API. To solve this issue Melissa added a method `getMapFragment()` to `HomeMapFragment` class. By implementing this method we execute a different Fragment Manager function depending on the version. This resolved the issue of the app not displaying any maps.

```
private MapFragment getMapFragment() {
    FragmentManager fm = null;

    System.out.println("sdk: " + Build.VERSION.SDK_INT);
    System.out.println("release: " + Build.VERSION.RELEASE);

    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.LOLLIPOP) {
        System.out.println("using getFragmentManager");
        fm = getFragmentManager();
    } else {
        System.out.println("using getChildFragmentManager");
        fm = getChildFragmentManager();
    }

    return (MapFragment) fm.findFragmentById(R.id.mapHome);
}
```

`getMapFragment()` in `HomeMapFragment` class of `CityVille` allows us to view Google Maps on Android Lollipop

When Melissa began to test `CityVille` and confirm with Sifat the data he was wanting to collect, they realized that all user login and registration information was being stored in another location - not in the Parse.com database. Since Sifat would like to capture how often a particular user logs in, posts events, or reports areas Melissa had to make changes to the pilot version of the `CityVille` code. She also had to make changes to the Parse.com database. New tables were created in Parse.com to capture all `CityVille` login and registration information and a new `userId` field was added to existing Parse.com tables to be able to identify which user posted events or reported areas of concern. Now `CityVille` will capture and store in the Parse.com app usage relative to its user.

Adding the Profile Feature to CityVille

When Melissa integrated profiles created with the Processing programming language into `CityVille` some changes had to be made to the file. Processing is very similar to Java but it is not an exact match to just put it right into an Android project that requires java classes. Processing produces a `.pde` file, and can be opened as a normal `.txt` file. The Processing profile must first be saved as a `.java` file. She added to the top of the file the `CityVille` package name and `"import processing.core.*;"` as required with all java files. She then encapsulated the Processing code into a java class by surrounding it with curly braces and heading it with `"public class <profile_filename> extends PApplet"`. `PApplet` allows us to run this class as a Processing Applet with access to all the Processing programming functionality. Now that she created the

java class a few more pieces of code needed to be added to comply with the java syntax. An access modifier must be added to all methods within the class. Processing has a much more relaxed syntax than java (which is what makes it great for kids learning to code), for example Processing did not require type-casting of variables between double and float, but to meet the java syntax requirements in our Android project the type casting is required.

```
//rect(x, y, width, height);
//rect(30, 10, 40, 40);
float boxStart = horizontalCenter - (width*.75)/2;
int boxWidth= (width/4)*3;
int boxHeight= (height/4)*3;
//big frame box
rect(boxStart, 60, boxWidth, boxHeight);
```

Processing code snippet without type-casting

```
//rect(x, y, width, height);
//rect(30, 10, 40, 40);
float boxStart = (float) (horizontalCenter - (width*.75)/2);
int boxWidth= (width/4)*3;
int boxHeight= (height/4)*3;
//big frame box
rect(boxStart, 60, boxWidth, boxHeight);
```

Processing code converted to Java syntax with required type-casting

An additional method had to be added to this java class so that when a user is done viewing the profile they can use the back button on their device to get back to the menu. Without this method the application will crash when a user presses the back button.

```
@Override
public void onBackPressed() {
    // fixes force close that occurs when using standard back button
    // this causes the activity to be destroyed and fall back to previous activity without null pointer FC
    finish();
}
```

onBackPressed() method added to Processing code within a Java class

Once Melissa made the changes outlined here the Processing code complies with Java syntax and it works seamlessly within the Android project.

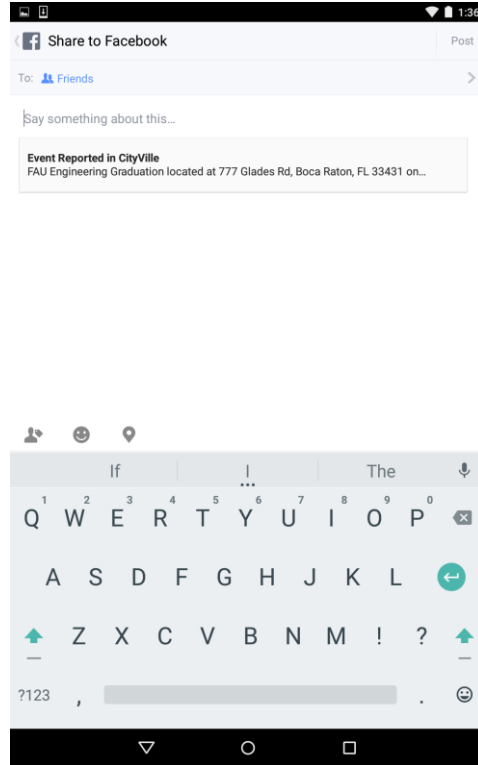
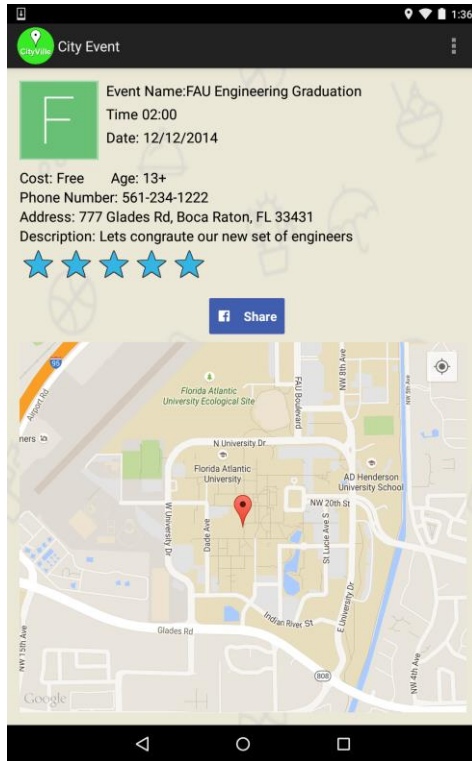
When implementation of our lesson plan began. Iris quickly realized that we should have allocated more time. Only one hour per week for five weeks is very limited timeframe when trying to teach middle school students something new to them. Iris also noticed that with a one week gap between sessions, students had forgotten what they did the week before. She suggested included a short refresher activity each week so the students could remember what they have learned so far.

Social Media Integration

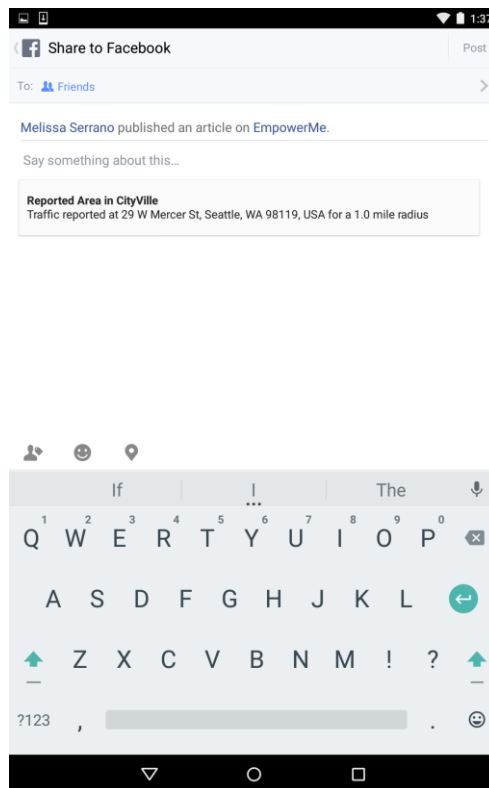
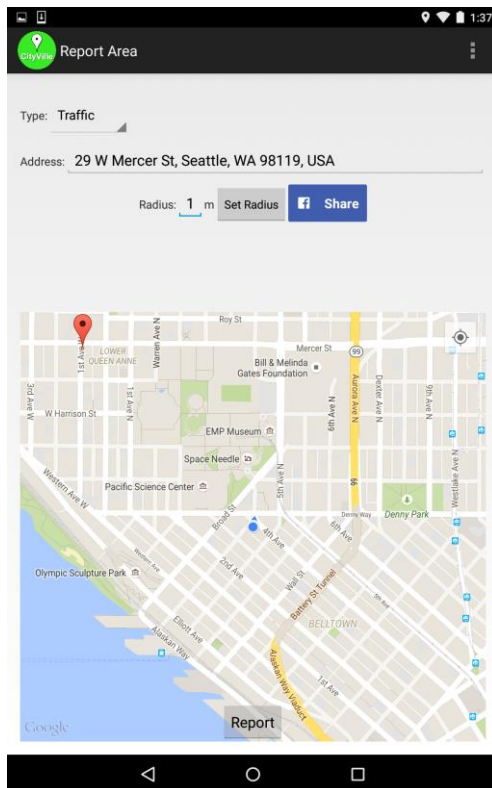
To give CityVille users the ability to send messages or share items with others, we considered integrating CityVille with KIK via the KiK API (Application Programming Interface). In our case, we would utilize the KiK API to access KiK users' profiles data or to send messages to other KiK users. We realized that the KiK API is very limited in the functionality that it provides to access data, especially for an Android App like CityVille. For example, the KiK API would not be able to provide us with any data related to the user's KiK contacts; we wanted to collect this data to track the user's interaction in their social network. However, the KiK API does provide a way for the user to send text message through KiK from the CityVille App. The user could type a message, but when they send the message they would be redirected to the KiK App to choose the recipients of the message. This would provide us with only a limited amount of data to gather for our analysis, because we would only be able to collect data regarding the text message and the Cityville user that sends the message. We would not be able to collect data about the KIK users that receive the text message and their reply. Therefore, we will be unable to measure the impact of the social network for empowerment. Since at the time of this writing, the KiK API doesn't support the functionality that we deem necessary for the data we would like to collect, we decided to evaluate another social network from the students' social media usage survey results.

When we reevaluated the social media usage survey completed by the students we noticed that a total of twenty-one students use FaceBook or Instagram. For our data analysis we were not concerned with collecting photos, as in Instagram. However FaceBook owns Instagram, and a user posting to either social media site has the option to also share their post on the other. Since our focus for data analysis is text we decided to integrate FaceBook with the CityVille app.

The integration with FaceBook allows a user to share an event or reported area in CityVille. When the user clicks the Share button from an event or reported area the details are included and get posted on the FaceBook wall of the FaceBook users they choose to share it with. A user can choose to make the post public, only for friends, only for himself, or customize who they would like to view their post.



A user can click the Share button and the details of the City Event will be included in their Facebook post



A user can click the Share button and the details of the Reported Area will be included in their Facebook post

To accomplish this, initially Melissa tried to use a new version of the Parse.com API that included a version of the FaceBook API. Since FaceBook recently acquired Parse.com and we were already using Parse for our database structure she thought this would be a good solution. However, because of the newness of the integration of the two APIs it still had some issues which other developers had yet to find solutions. So she decided to continue to use the original Parse API we were using from the pilot version of CityVille and use the standalone FaceBook API with it. Below are snippets of the code that was added to the CityVille app to accomplish this.

```
shareButton.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        SoundManager.playSound(6, 1);

        if (ShareDialog.canShow(ShareOpenGraphContent.class)) {

            //set all the reported data to be shared on FaceBook
            String fbDescription = eventName.getText().toString() + " located at "
                + eventAddress.getText().toString() + " on " + eventDate.getText().toString() +
                " at " + eventTime.getText().toString();

            ShareOpenGraphObject object = new ShareOpenGraphObject.Builder()
                .putString("og:type", "news.article")
                .putString("og:title", "Event Reported in CityVille")
                .putString("og:description", fbDescription)
                .build();
            ShareOpenGraphAction action = new ShareOpenGraphAction.Builder()
                .setActionType("news.publishes")
                .putObject("article", object)
                .build();
            ShareOpenGraphContent content = new ShareOpenGraphContent.Builder()
                .setPreviewPropertyName("article")
                .setAction(action)
                .build();

            shareDialog.show(content);
            shareButtonFlag = true;
        }
    }
});
```

The Share Button implements the `onClickListener()` to show the Share Dialog with the Event or Reported Area details

A new column was added to the CityEvent and ReportedArea tables in the Parse.com database so that Sifat could collect data related to the user sharing events or area. The `shareButtonFlag` boolean variable is set to true in the `onClickListener()` so that when we put the event or area in the Parse.com database we can record if the user shared it as well. Sifat will use this in his data analysis to determine how empowered the students are.

```
if(shareButtonFlag == true)
{
    newCityEvent.put("sharedOnFB", true);
}
else
{
    newCityEvent.put("sharedOnFB", false);
}
```

When the shareButtonFlag boolean variable is set to true we will record the share in the database

Discussion

To give the app more personalization and allow the students to learn basic programming we will introduce them to the Processing programming language. We are adding this aspect because it will allow the students to show empowerment and individuality by self-expression, while providing another metric to measure student empowerment. In order to increase usage of the app we think that it is extremely important to allow students to integrate the Cityville app with their preferred social media account. If this app is just another app they download they'll probably take a look at it a few times and go back to their usual social apps whether it's Facebook, Instagram, Snapchat, Tumblr, KIK, or any others. By giving the students this option, the app will work more closely with what they are already doing. This also allows for us to collect more data connected to each student to analyze later.

Conclusion

Upon implementation of our proposed enhancements we will have achieved an Android app which will empower middle school students as well as provide us with useful data which we can use in our analysis by making the app easy to integrate with what the students are already doing and giving it the personalized feel they are accustomed to with many other social games and apps.

In this paper we documented app and lesson plan development, implementation, and data collection. Future papers by this group will include details about data analysis of the data we collected during the 5-week program we implemented.

References

<https://github.com/RShankar/Empower-App-Cityville-Fall-2014/wiki>

<https://github.com/EmpowerMe/Melissa-Serrano>

EmpowerMe - Melissa Serrano YouTube Channel

https://www.youtube.com/channel/UCx_MZ6lfjTZcyk50pa9snYA

Global Smartphone Market Share By Platform

<http://www.businessinsider.com/iphone-v-android-market-share-2014-5>

Processing Programming Language Reference

<https://processing.org/>

KIK API Reference

<http://dev.kik.com/docs/>

KIK Native Android API Reference

<https://github.com/kikinteractive/kik-api-android>

FaceBook API Reference

<https://developers.facebook.com/docs/sharing/android>