Alibi: an Android Platform Mobile Application Utilizing the Google Play Game Services SDK

By

Sofie L. Grant

December, 2014

Alibi: an Android Platform Mobile Application Utilizing the Google Play Game Services SDK

By

Sofie L. Grant

A project submitted in partial fulfillment of the requirements for the degree of

Master of Science in

Computer Information Systems

at

Grand Valley State University

December, 2014

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Hans Dulimarta Date**

Table of Contents

Abstract [4](#__RefHeading___Toc133549436)

Introduction [4](#__RefHeading___Toc133549437)

Background and Related Work 5

Program Requirements 6

Implementation 11

Results, Evaluation, and Reflection 12

Conclusions and Future Work 12

References 13

# Abstract

The Google Play Game Services SDK offers game developers a variety of features such as user accounts that can be readily adapted to their games. Using previous knowledge of the Java programming language and iOS platform mobile application development, the author developed Alibi, an Android platform mobile game application based on the popular word game Mad Libs. Mad Libs involves one player prompting the other to suggest nouns, verbs and other word types. These words are then inserted into a story, producing a nonsensical tale both players can enjoy reading aloud. Alibi utilizes Google Play Game Services to implement a turn-based multiplayer game structure that utilizes game logic to track word suggestions on behalf of the players, allowing both players to suggest words during the game. In this iteration of the game, one player initiates a new game of Alibi, takes a turn suggesting a word and ends their turn, thereupon passing his or her game to the next player. The game continues until both players have suggested enough words to complete a detective story, which both players then read in full.

# Introduction

Chetan Sharma Consulting reported that respondents to their 2014 Mobile Industry Predictions Survey believed Google was poised to be this year's most “important [player] of the mobile ecosystem” (2014, p.8). In line with those predictions, Google Senior Vice President of Chrome and Apps Sundar Pichai announced during a July 2014 Google event that over one million Android platform mobile applications had been published to the Google Play Store (Price, 2014, p.2). NetMarketShare's November 2014 report on the mobile/tablet operating system market found that Android had the largest market share of all mobile and tablet operating systems (2014).

Given the “shortage of skilled mobile developers” and the sizable market demand for Android platform mobile applications, acquiring fluency in the Android platform offers developers a compelling avenue to address consumer software demands (Engelsma, 2014, p.11). Google offers a variety of tools to interact with mobile application consumers. One significant sector in Android platform mobile development is that of games.

Chetan Sharma Consulting's respondents anticipate that entertainment will be the industry category fourth most impacted by mobile development over the course of the next five years (2014, p.23). The survey additionally reported that of consumer mobile applications, games were anticipated to be the fifth most popular in 2014 (2014, p.10). Popular mobile game applications, like the playful fruit demolishing game Fruit Ninja, can even sell millions of downloads (Sliwinksi, 2011). The Google Play Games Services SDK is particularly interesting in light of the popularity of mobile game applications.

The Google Play Game Services SDK offers game developers a variety of features that can be readily adapted to their games. These features tend to be based on frequently used concepts in mobile game applications such as player accounts, which serve to track player progress through a game. Use of these features benefits game developers by offering tested solutions to common game design needs. These solutions can expedite the game development and publication process (Android, 2014b, para. 2). The Google Play Game Services additionally offer developers access to the Google Play Developer Console. This web-based software publication tool supports publication to the Google Play Store, an access portal from which mobile application gamers can download game developers' Android platform products. Once published to the Google Play Store, the Google Play Developer Console allows game developers to control distribution of their product and monitor user reviews, crash reports and use statistics, among other benefits (Android, n.d., sections 10-12).

Recognizing the benefits of the Google Play Game Services SDK, the author developed an Android platform mobile game application named Alibi that featured key features from the SDK in order to improve upon the popular word game Mad Libs. Several versions of this game are currently available on the Google Play Store. However, none appear to use the turn-based multiplayer game setup offered by the Google Play Game Services SDK that was implemented in Alibi. It is common to see mobile applications that build a better mousetrap, or are iterations of a similar idea that may include improvements or new features that are not included in other iterations. For example, a brief search of the Google Play Store reveals that there are a dazzling number of available iterations of the popular card game Solitaire. Alibi is then a product that is in line with consumer demand but offered the inexperienced author the opportunity to become versed in the Android platform.

# Background and Related Work

As discussed previously, the Google Play Store is a primary repository for Android mobile applications. Alibi is modeled on Mad Libs. In its original format, Mad Libs is a multiplayer game where one player takes a turn reading prompts from the game's booklet to another player. These prompts ask for types of words such as nouns or verbs. These responses are inserted into a story. Once enough responses have been collected to complete the story, the players read it aloud. The result is a nonsensical tale, since the player suggesting the words does not know in advance how they will be used in the story.

Searching the Google Play Store for the keyword “Mad Libs” returns approximately twenty-five mobile applications that appear similar to the original game. Of these iterations, there are a few notable trends. The first trend is that in most iterations, the game takes on the role of providing word prompts to the player. This format allows single players to play the game. It also requires multiple players to take turns taking dictation and suggesting words, collaboratively suggest words while sharing a single device or take turns playing the game on a single device. Several iterations have well-developed art assets and multiple well-reviewed, game enhancing features such as random word generators. But many iterations are somewhat crude and underdeveloped. Another trend is that while many iterations allow players to save or share their completed stories, there is no in-game means by which to collaborate on a story. Finally, most games are not well-reviewed overall due to a lack of features, bugs and lack of new content, which suggests there is some room for a new iteration such as Alibi to be successfully introduced to the marketplace.

# Program Requirements

Alibi implements several key features offered by the Google Play Game Services SDK in order to develop a turn-based multiplayer version of the popular game Mad Libs. One such feature was user accounts, which facilitates player access to multiplayer matchmaking. The Google Play Game Services SDK allows developers to bypass developing their own player account setup by asking players to sign into their existing Google Plus account and tracking their game progress under that account. This additionally saves players the difficulty of recalling extra login credentials.

Due to the link between games that utilize the Google Play Game Services SDK and Google Plus, players can use Google Play matchmaking services to locate their Google Plus contacts and invite them to play the game. The Google Play Game Services SDK offers two flavors of multiplayer game setup: real-time and turn-based. Real-time multiplayer games are setup and played immediately to completion after two or more players join a game. Turn-based multiplayer games set up an initial game between two or more players and are played out over a series of turns. Turns may be taken up to days apart.

The game's structure is as follows. Upon opening the game for the first time, a player is greeted with an initial Google Plus login screen. If the player's device only has login credentials for one Google Plus account stored to its memory, the login process attempts to login to this account. If there is more than one Google Plus account stored to the device's memory, the login process will prompt the player to select his or her desired account from a list of accounts currently associated with their device (Figure 1).

Once logged in, the player is greeted with the main menu screen (Figure 2). The main menu includes a small text view that displays greeting with the player's Google Plus account name. This serves to provide an additional confirmation that the player is signed in to his or her desired account. The main menu screen houses six buttons titled Tutorial, Play Alibi, Check Inbox, Sign In and Sign Out. By selecting these buttons, players may respectively read the game tutorial, proceed to a new game, check their inbox for new game play invitations and alerts to ongoing games, sign out of their account or sign back in to their account.

Pressing the Tutorial button diverts the player from the main menu screen to the in-game tutorial screen (Figure 3). This consists of a brief text description of the game's objectives and navigation buttons. It also includes a button that redirects the player back to the main menu screen.

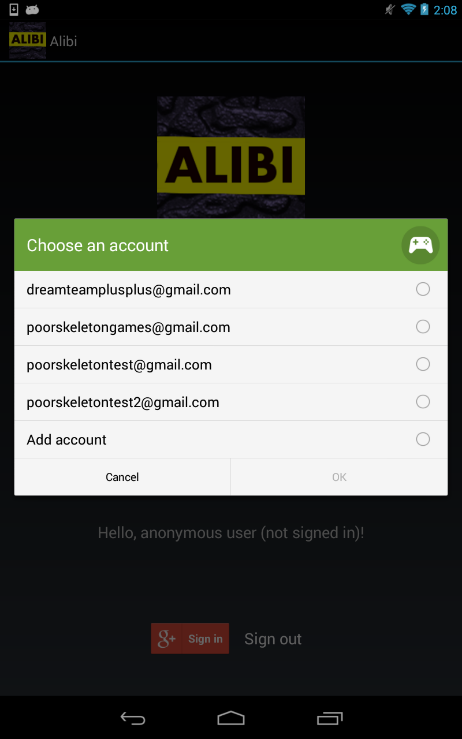


Figure 1: Account Selection

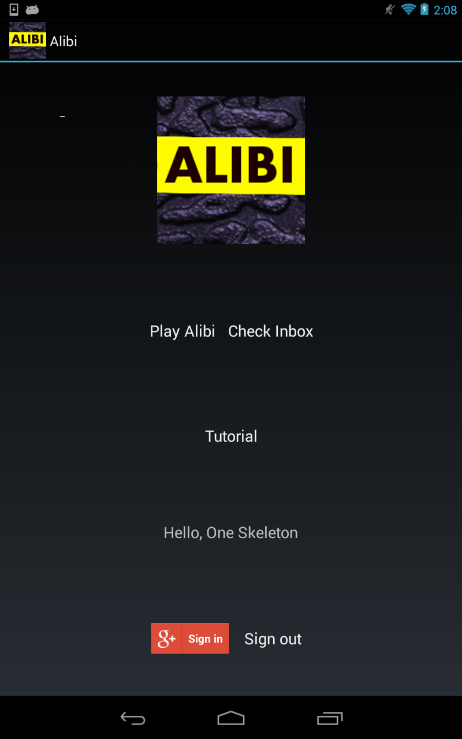


Figure 2: Main Menu Screen

Selecting the Play Alibi button directs players to the Google Play default turn-based multiplayer interface (Figure 4). The interface populates list of the players' Google Plus friends, making friend selection easy. Players may also use the interface to locate other people who have Google Plus accounts.

Upon selecting another person to play with, Alibi redirects the player to the Alibi game screen (Figure 5). The Alibi game screen consists of a simple edit text field and three buttons. The edit text field displays hint text asking the player for a specific type of word and records the player's input. The first button is titled Done and confirms that the player is satisfied with his or her input and wishes to pass the turn to the next player. After pressing Done, the player is redirected back to the main menu screen. The second button is titled Cancel and confirms that that the player no longer wishes to play the game. The player is redirected back to the main menu screen and an invitation is forwarded to the player's opponent's inbox that informs him or her that the other player has declined continuing to play the game. The third button is titled Leave and confirms that the player does not wish to submit an answer at the moment and redirects the player back to the main menu screen without forfeiting the player's turn.

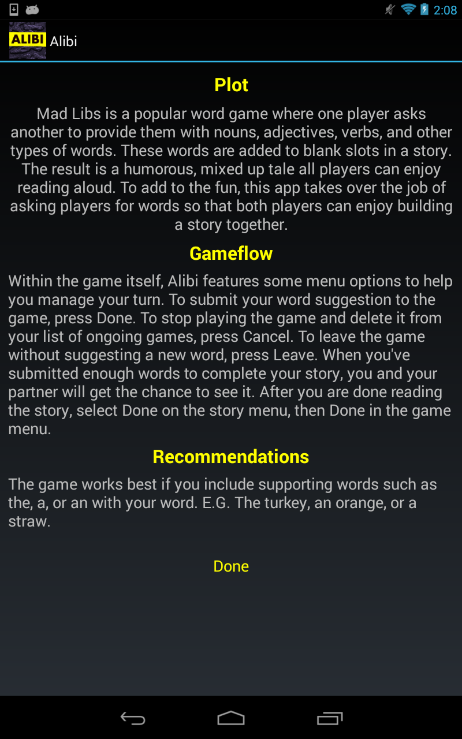


Figure 3: Tutorial Screen

After the player has successfully completed a turn, is their String response and an int which serves as a turn counter are persisted to a simple byte array. This byte array is passed along with a few other pieces of data such as the match's unique identification number into the takeTurn() method of the Google Play TurnBasedMultiplayer class. This allows the application to store the users' responses in a location that is mutually accessible on any of the participating players' devices. The data is then unpersisted by the second player upon beginning his or her turn. As each player contributes a new word, the number of collected String responses grows.

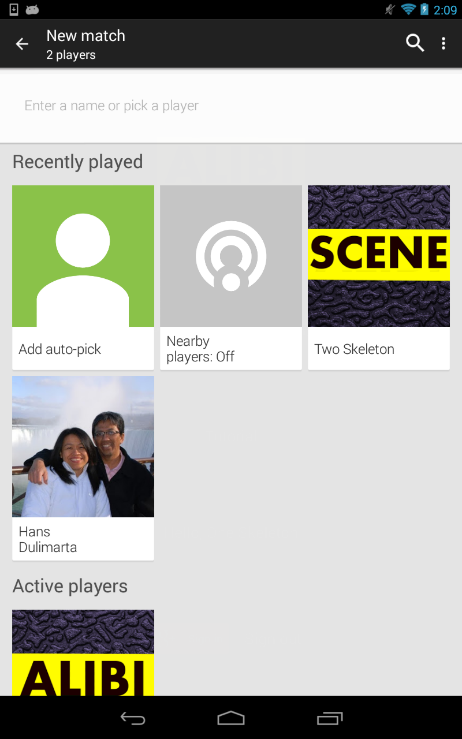


Figure 4: Google Play Default Turn-Based Multiplayer User Interface

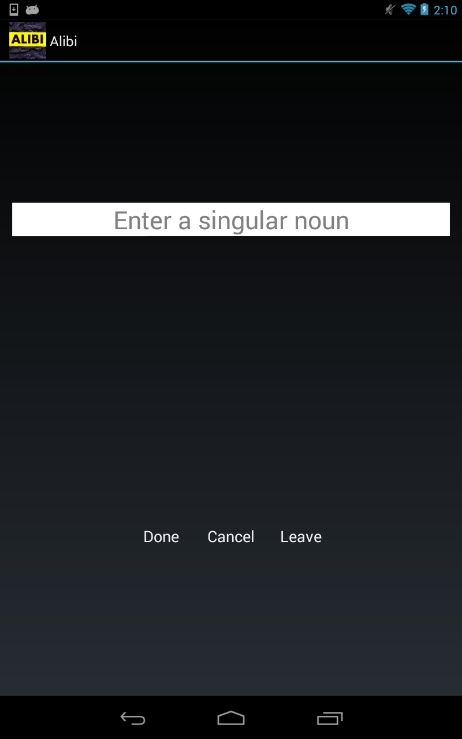


Figure 5: Game Screen

The Inbox button directs players to the Google Play default match inbox user interface (Figure 6). This screen lists the player's new game invitations or ongoing game statuses. If the player accepts a new game invitation, he or she is directed to the Alibi game screen where he or she completes the next turn of the game. The same data persisting process as described above commences. The first player shortly thereafter receives a notification in his or her inbox that it is again his or her turn. Should a player leave the game, the other player receives a notification that the game has been canceled. The inbox also provides feedback on which player currently needs to complete a turn.

Once sixteen words total (eight from each player) have been contributed to the response pool, enough words have been gathered to complete the story at the heart of Alibi. The game checks the progress of the story, finds that enough words have been completed and creates a new screen dubbed the win screen available to both players (Figure 7). The game adds the players' responses to predetermined empty slots in the story and publishes the final result to the win screen. Each player has one opportunity to view the story before the String responses and int counter data is cleared. This clearing process allows them to play the game together once again.

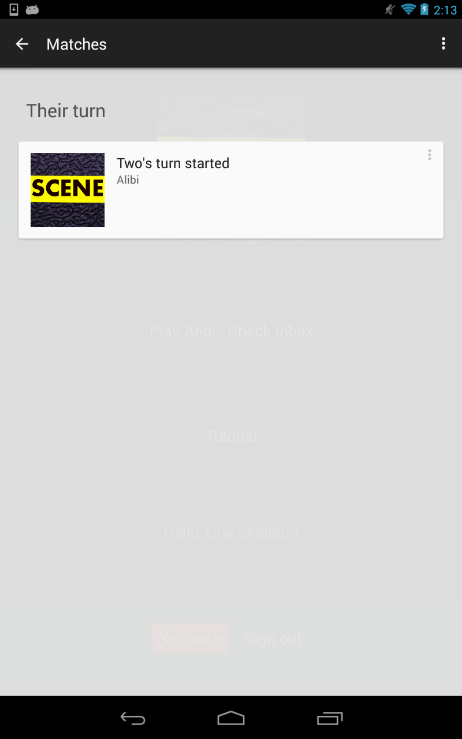


Figure 6: Google Play Default Match Inbox User Interface

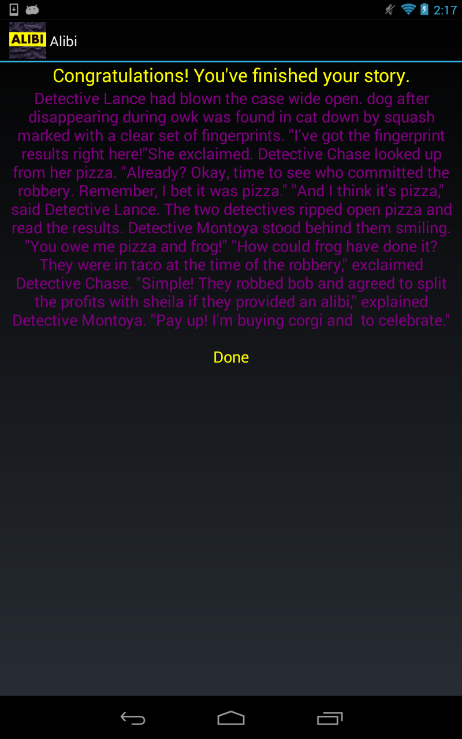


Figure 7: A Completed Story

# Implementation

A number of tools and technologies were utilized in the implementation of Alibi. Android Studio, the Java programming language and the Google Play Game Services SDKs were the project's core resources. The author was previously experienced with the Java programming language and iOS platform mobile development, but Android platform mobile development, Android Studio and the Google Play Game Services SDKs were all new topics. To remedy this, the author studied *Android Programming: The Big Nerd Ranch Guide* by Brian Phillips and Brian Hardy (2013). The author additionally studied the Google Play Developer's Guide, which provides extensive examples on implementing the Google Play Game Services SDKs (Google Inc., 2014c). The author also asked a third party designer, Jacob Munford, to create an original icon suitable for the game. Icons are typically used to represent the game in the Google Play Store as well as in the player's telephone, tablet or other device application directory.

Google offers access to several sample applications that developers can implement in order to learn more about the Google Play Game Services SDKs (Google Inc., 2014e). The author implemented the example application TypeANumber and the library BaseGameUtils to practice using Android Studio, the Google Play Developer Console and the Google Developer Console. The Google Developer Console is an interface that provides some additional application publication options unavailable in the Google Play Developer Console.

The BaseGameUtils library and select code from TypeANumber and another sample application, SkeletonTbmp, were implemented in Alibi in an effort to reduce the author's duplication of existing solutions. The BaseGameUtils library offers a set of methods related to authenticating player sign in (Google Inc., 2014e, para. 8). TypeANumber demonstrates Google Play sign in, leaderboards and achievements (Google Inc., 2014e, para. 8). It was selected as an overall game template for its clean and effective user interface design and straightforward implementation of Google Plus sign in. SkeletonTbmp demonstrates how to implement a small turn-based multiplayer game (Google Inc., 2014e, para. 9). It allows one player to create a String and pass it to a second player, who takes a turn editing it. The players continue to pass the editable String back and forth until they tire of the game. This sample provided matchmaking methods that were augmented for use in Alibi. Its byte Array creation and storage class, SkeletonTurn, was augmented to hold the String responses and int counter required for Alibi. Additional code was developed by the author to track turns, restrict the game to sixteen total turns for both players, store multiple String responses, format the story to include the String responses and display the game tutorial and win screens.

# Results, Evaluation, and Reflection

Alibi is currently a prototype. Due to the number of new skills the author was required to acquire before implementing the game, the development schedule of Alibi became compressed. The author had had hoped to implement a more fully developed game concept, but a longer production schedule would be required to do so. The game also appears to have a few repairable bugs that have not yet been weeded out but should be addressable with a little added time and effort.

While imperfect, the game does successfully meet its base objective: to create a marketable improvement to an existing game using the Google Play Game Services SDKs and turn-based multiplayer gaming. Moreover, the author has developed a foundation in Android mobile programming that will allow her to begin designing and implementing future applications to meet customer needs. Given the number of other features available in the Google Play Game Services SDKs that could enhance Alibi and future games, Alibi and the author have a bright future in the Android marketplace.

# Conclusions and Future Work

There are a number of other Google Play features that could be added to this project. One key feature is that of achievements. Achievements are goals that players can work towards by playing the game. For example, an achievement might be playing the game ten times. Once that goal is reached, the player may earn a special badge that is displayed in their player profile or points that could be spent in an in-game store. The goal of achievements is to encourage players to replay the game. This increases a game's value to its customer base. The Google Play Store so values achievements to marketing games that all games must incorporate at least five achievements in order to be published (Google Inc., 2014d, para. 13).

Another popular Google Play feature is leaderboards. Leaderboards collect player scores and rank their progress on private leaderboards viewable only through the player's account or to public leaderboards viewable by any players who access the game. Leaderboards could offer a great deal of added game replay value, which may help address some of the complaints from reviewers unhappy with the other iterations of Mad Lib games available on the Google Play Store.

Although not included in the Google Play features, added stories and social media tie-ins would also improve the replay value of the game. Players could select from an entire library of short detective stories before beginning a new game. Tying in to Facebook or Twitter to allow players to share their completed stories or invite friends to join a new game would add more ways for the player to interact with the game, thus increasing the game's play value. The link to social media would also provide added publicity for Alibi.

Monetizing the game is also a possible future avenue. It is possible to offer one version of the game for free and another, more developed version for a small fee on the Google Play Store. By monetizing the game, the author would be able to acquire a small stream of revenue that could help offset the time and resources required to develop future products.

The possible avenues for the further development and improvement of Alibi are nearly as extensive as the demand for new, exciting and innovative mobile games. The author expects to shortly begin the process of editing Alibi and studying the business side of mobile application development. Thanks to resources such as the Google Play Game Services SDKs, Android platform mobile application users should have many future well-tested, feature-laden applications to which to look forward, both from Android platform mobile application developers in general and the author.

# References

Android. 2(014a, October). Android developers: Android Studio Beta (Version 0.8.6) [Software].

Available from https://developer.android.com/sdk/installing/studio.html

Android. (2014b, October). *Android developers: Google Play Services*. Retrieved September 13, 2014

from http://developer.android.com/google/play-services/index.html

Android. (n.d.). *Android Developers: Developer Console*. Retrieved September 13, 2014 from

https://developer.android.com/distribute/googleplay/developer-console.html

Chetan Sharma Consulting. (2014, January). *2014 mobile industry predictions survey* [SlideShare slides].

Retrieved from http://www.chetansharma.com/MobilePredictions2014.htm

Droid @ Screen. (2014). Droid @ Screen (Version 1.1) [Software]. Available from

http://droid-at-screen-ribomation.com/download

Engelsma, Jonathan R. (2014). *Introduction to mobile app development* [PowerPoint slides]. Retrieved

from https://mybb.gvsu.edu/.

Genymobile. (2014). Genymotion (Free version) [Software]. Available from http://www.genymotion.com/

Google Inc. (2014a). Google Developer Console [Web-based software publication tool]. Available from

https://console.developers.google.com/

Google Inc. (2014b). Google Play Developer Console [Web-based software publication tool]. Available

from https://play.google.com/apps/publish/

Google Inc. (2014c, July 1). Google Play Game Services. Retrieved September 13, 2014 from

https://developers.google.com/games/services/

Google Inc. (2014d, December 2). Google Play Game Services: achievements. Retrieved September 13, 2014 from https://developers.google.com/games/services/common/concepts/achievements

Google Inc. (2014e). Google Play Game Services: Android Samples [Software]. Available from

https://github.com/playgameservices/android-basic-samples

NetMarketShare. (2014, November). *Mobile/tablet operating system market share*. Retrieved December

6, 2014 from

http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1

Parse. (n.d.). Android SDK (Version 1.7.1) [Software]. Available from https://parse.com/docs/downloads

Phillips, B., & Hardy, B. (2013). *Android programming: The big nerd ranch guide*. Atlanta: Big Nerd

Ranch, Inc.

Price, Emily. (2014, July 24). Live from Google's San Francisco event. *Mashable.com*. Retrieved from

mashable.com/2013/07/24/google-sf-android-event

Sliwinski, Alexander. (2011, February 19). Fruit Ninja hits six million sales, sponsors a fruit orchard.

*Joystiq.com*. Retrieved from

www.joystiq.com/2011/02/19/fruit-ninja-hits-six-million-sales-sponsors-a-fruit-orchard