

WLAN-Opp-based Turn-based Strategy Game

Semester Thesis

Hagen Seifert

January 18, 2013

Advisors: Sacha Trifunovic
Supervisor: Prof. Dr. Bernhard Plattner

Computer Engineering and Networks Laboratory, ETH Zurich

Abstract

A turn-based strategy game for Android for two players, either on a single device or two devices over network using the *WLAN-Opp* Platform [1], was created. *WLAN-Opp* enables Android devices for opportunistic networking via open stationary and spontaneously created mobile Wi-Fi access points. It was developed at the Communication Systems Group at the ETH Zurich. The game not only demonstrates the usefulness of *WLAN-Opp* for local network games, it also helps to spread its use. This is important because opportunistic networking requires a critical mass of supporting devices to be useful.

The gameplay is inspired by the Heroes of Might and Magic game series on PC. Each player can move his heroes on the game map and gather resources, capture mines and start battles against neutral creatures and enemy heroes and towns. Gathered resources can be invested in towns owned by the player for new buildings, troops and heroes. The interface and controls of the game are specifically designed for easy use by touch on small screens.

Contents

1	Introduction	5
2	Related Work	7
2.1	WLAN-Opp	7
2.2	Similar Games on the Google Play Store	7
3	Design	9
3.1	A Short Explanation of the Game	9
3.2	Views of the game	9
3.2.1	The Map View	10
3.2.2	The Town View	10
3.2.3	The Battle View	11
3.3	Creating a Map	12
3.4	Multiplayer using <i>WLAN-Opp</i>	12
4	Implementation	15
4.1	Save Files	15
4.2	Multiplayer Service	15
4.3	Graphics	16
4.3.1	Rendering the Views	16
5	Evaluation	19
5.1	The Game	19
5.2	Save Files	19
5.3	Devices Tested On	19
6	Future Work	21
7	Conclusion	23
	Bibliography	25

Chapter 1

Introduction

Opportunistic networking can be used in various fields of application, such as local social networking, communication in remote areas or in disaster emergency situations [1]. The growing number of Wi-Fi capable smartphones offers a good base for such opportunistic networks.

WLAN-Opp [1] is an Application for Android smartphones developed at the Communication Systems Group at the ETH Zurich. It uses open stationary Wi-Fi access points to connect to nearby devices or spontaneously creates an access point if none are around. Despite its possible uses, opportunistic networking requires a sizable user base for communication to be effective. In this project, a game is created, that demonstrates the usefulness of *WLAN-Opp* for local communication and helps to spread its use.

Because of the growing hardware performance and wide spread of Android smartphones, Android has become a major gaming platform. This fact is emphasized amongst others by the immense success of the Kickstarter project of an Android powered video game console, OUYA [2, 3], as well as the recent announcement of NVIDIA to release such a console [4].

Going back to Android smartphones, a big challenge in game development remains the controls via touchscreen. Additionally, most people see games on their smartphone more as a short time filler, for example while waiting for a bus. This has led to a large amount of casual games, whose success is based on an easy to grasp game mechanics and short continuous play times.

The game implemented in this project is a turn-based strategy game. Against real-time games this has the advantage, that it doesn't require constant attention, making it feasible to be played almost anywhere. Also, the controls are less of an issue, since the player has time to carefully think what he wants to press. Nonetheless, this game will focus on intuitive touch controls.

To stick out from the masses of Android games, this game tries to deliver a more complex game mechanics, inspired by the older games of the Heroes of Might and Magic [5] series. The game is therefore called and will be referred to in this work as *Android Heroes*.

Chapter 2

Related Work

2.1 WLAN-Opp

The multiplayer functionality with two devices over network uses the *WLAN-Opp* Android application [1] that was developed by the Communication Systems Group at the ETH Zurich. This allows the game to automatically connect to other nearby Android devices via open Wi-Fi networks or dynamically created mobile hotspots. This project demonstrates the usefulness of the *WLAN-Opp* platform for multiplayer games and helps to promote and spread its use.

2.2 Similar Games on the Google Play Store

The Google Play Store [6] is the most common source for app installations on Android. As of October 2012 it offers 700 thousand different apps [7], that have been downloaded over 25 billion times in total [8]. When planning to create a game for Android, it should therefore be considered, what competitors there are already on the market.

Many complex strategy games exist for PCs, but on the Play Store they are only scarcely available. Strategy games on Android are dominated by casual games, such as the Tower Defense genre. Well known examples are *Plants Versus Zombies* [9], *Fieldrunners* [10] and *Sentinel* [11]. There are only a few more serious strategy games such as *Great Big War Game* [12], *Battle for Wesnoth* [13], *Majesty: Fantasy Kingdom Sim* [14] and *Free Heroes 2* [15]. Except for *Great Big War Game* [12], all of them are ports from the PC platform, and are therefore not well adapted for small screen sizes and touch controls.

Therefore I believe, that a well implemented complex strategy game could be very successful on the Play Store.

Chapter 3

Design

3.1 A Short Explanation of the Game

Android Heroes is a turn based strategy game for two players. It is inspired by the Heroes of Might and Magic video game series for PC. The central element of the game is a map of arbitrary size. Each player can possess towns, heroes and mines on the map. The goal of the game is to capture all towns and defeat all heroes of the other player.

Towns are placed on a specific spot on the map from the beginning. If a town is owned by a player, this player can build various buildings in it, that allow the recruiting of new heroes or troops.

Heroes are the only objects on the map that can move. They can carry an army of up to seven troops of different creature types. They can start battles with towns and heroes of the other player, a neutral army or capture towns and mines.

Mines are, similar to towns, fixed on the map. A player can capture a mine by entering it with a hero. There are four different types of mines, according to the four different resources in the game: Gold, wood, stone and gems. A mine gives an amount of its resource to the owner each turn. Resources are used to build buildings in a town and buy troops or heroes. They can also be collected by heroes on the map.

The game can be played either on a single device, which is handed to the other player after the turn is finished, or on two devices using the *WLAN-Opp* Service.

3.2 Views of the game

The game is implemented in Android using one Activity which has three different views. The main view is the map view. From there the other two views, the town and the battle view, can be accessed.



Figure 3.1: Screenshot of the map view of the game.
 Map: 1) A hero owned by the red player, 2) a gold mine, 3) a neutral troop, 4) a town. Gray flags indicate that mine and town are not currently owned by any player. 5) Fog covers the parts of the map that the player hasn't explored yet.
 Fixed user interface: a) Information about the resources the player owns and in which round the game is. b) The owned towns and heroes of the player, which can be selected by tapping. c) Buttons to move the selected hero to its target coordinate and to end the turn.

3.2.1 The Map View

The map view, as seen in Figure 3.1, is where the player typically spends the most of his time. The map currently consists of 32 times 32 square tiles. When fully zoomed in, a tile has a side length of 100 pixels. The map is covered by two ground types, grass and sand. Movement of heroes on sand takes 50% more movement points. Movement is restricted by various objects like trees and mountains that act as barriers. Neutral armies block pathways and guard mines and resources. To pass them, a hero has to defeat them in battle.

The heroes, after which the game is named, are the only possibility of moving troops on the map. They can be moved for a specific amount of movement points each round to collect resources, start battles against neutral armies and enemy heroes and towns and capture mines and towns.

The control of the view is simple and user friendly. In accordance to many Android apps like for example Google Maps, the map can be dragged with one finger and zoomed using the two finger pinch gesture. A tap on a symbol of a player's hero or town in the top right corner of the screen selects it and focuses it on the map. A second tap either shows the army of the selected hero or enters the town screen.

3.2.2 The Town View

The town view (Fig. 3.2) can be accessed either by walking into the town with a hero or by tapping its symbol twice in the map view. In the upper part it shows the buildings that have been built in this town. They can be tapped to open



Figure 3.2: Screenshot of the town view of the game.

1) The buildings in the town (from left to right): Castle, Fairy Tree, Bear Cave, Elf House, Tavern. 2) The armies of the town (top) and visiting hero (bottom), each has space for 7 troops. 3) Buttons to enter the build menu, to split a troop into two, and to go back to the map view.

a) One of the dialogs is shown, recruiting creatures. b) The build menu.

their dialog, which is depending on the building either a simple information dialog, a dialog to buy creatures or a dialog to buy a hero.

In the lower part, the army of the town and, if there is any, of the visiting hero are shown. The troops can be rearranged, combined (if they are of the same type) or split via drag and drop.

3.2.3 The Battle View

The battle view (Fig. 3.3) will come up whenever the player attacks either a neutral army or an enemy town or hero with his own hero.

The battle is, like the main game, turn based. Each troop of each player gets one turn per round. The sequence of turns goes from fastest creature to slowest. In a turn the active troop can either walk to an unoccupied space in its range or attack an enemy troop in its range. If a troop is attacked in close combat for the first time in a round, it will automatically retaliate. Alternatively the player can skip the turn or wait, to carry out the turn at the end of the round. All troops that waited will have their turns in the reverse order at the end of the round. When every troop either took or skipped their turn, a new round starts. When one team has no troops left (their count reached zero), the other team wins. A defeated hero or neutral army is removed from the map, a defeated town will change its owner to the winning player. The battlefield consists of fifteen times eleven hexagons. The attackers start on the Left side, the defenders on the right. Like on the map view, it can be dragged and zoomed with multi-touch gestures. It also features animations for moving and attacking of each of the four available creatures (pikemen, fairies, bears and elves).



Figure 3.3: Screenshot of the battle view of the game. Attacking troops are looking to the right, defending troops to the left.

1) When a troop is selected, its information is shown at the top of the screen. Here for example the selected troop represents 5 bears. 2) The active troop is marked with a green hexagon. In this example it's the fairies' turn and they are about to attack the bears. Hexagons with a dark overlay are the ones that can be reached by the active troop in one move. 3) The player has selected this opposing troop of bears and the possibilities of different attack directions are shown. 4) Buttons to either wait with this troop's turn for the end of the round or completely skip this round.

3.3 Creating a Map

Placing the 28 different objects, two ground types, heroes, resources and neutral armies on a map of 32 times 32 squares is impossible without the use of some kind of map editor. Therefore the open source map editor *Tiled* [16] was used. It supports two dimensional maps in a tile layout with arbitrary tile and map size. In *Android Heroes*, each tile has a size of 100 pixels, which is a large enough size to make sure it can be comfortably tapped with a finger even on very high pixel density screens. The *Tiled* map editor outputs TMX map files, which are in XML format.

3.4 Multiplayer using *WLAN-Opp*

WLAN-Opp offers connection to other close-by Android devices via open Wi-Fi networks or, if necessary, by creating a temporary hotspot. This has the potential of playing games with close-by friends without the need to worry about the connection details. Because of the unstable nature of the connection, *WLAN-Opp* constantly tries to find as many connections as possible, which may force it to disconnect from a network temporarily, the game must be tolerant to loss of connection to the other payer.

Because *Android Heroes* is a turn based game, it does not require continuous connection during the whole game. There are two types of interactions between

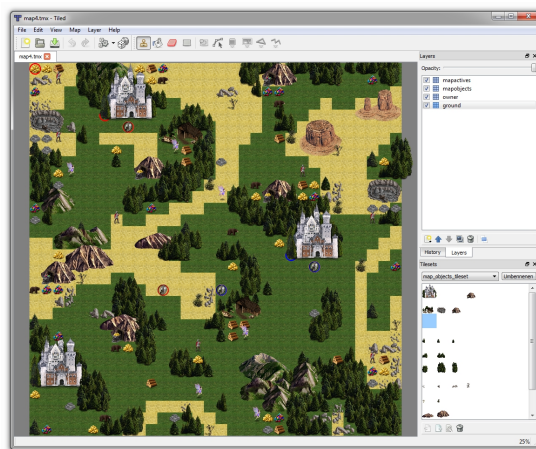


Figure 3.4: Screenshot of the *Tiled* map editor used to create *Android Heroes* maps

the two players. The first type is when one player ends his turn so the other one should continue. The second type is when one player attacks a hero or town owned by the other player.

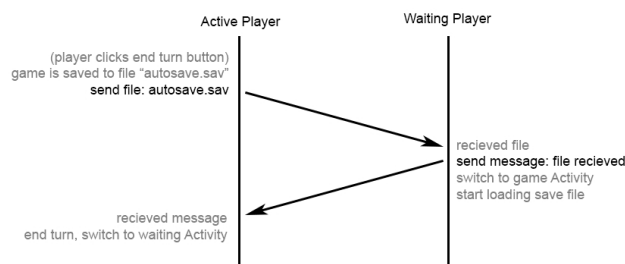


Figure 3.5: State diagram of the battle controller in a battle between two players

Passing the turn to the other player is the simpler type of interaction, since connection is only required once per round. The passive player's game is in a state of waiting, when the active player ends his turn. The active player's game then saves its state to an XML formatted save game file, which is transmitted to the waiting device. Upon receiving the file, a confirmation is sent, the save game is loaded and the second player's turn begins. Should an error occur during transmission of the file, the first player's round doesn't finish, so that he can either try again or, if the connection is lost for a longer time, save the game for a different time. This makes sure that no data is lost.

The second interaction, a battle against the other player, is a little more complicated. Instead of transferring the state of the whole game every few minutes, in the multiplayer battle every small change is transmitted as a string message. In the battle, turns belong to troops rather than to players (read more in section 3.2.3). The active player is the one to whom the active troop belongs. As can be seen in the state diagram in Figure 3.6, the device of the active player serves as the Master. It interprets the user input and takes action accordingly.

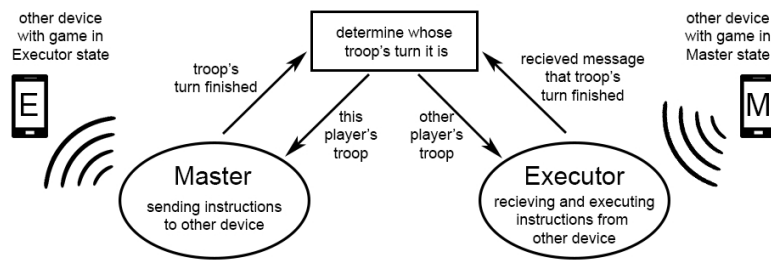


Figure 3.6: State diagram of the battle controller in a battle between two players

Possible actions are moving the active troop or attacking an enemy troop, in which case the the damage done is calculated and a possible retaliation and it's damage is calculated. Alternatively the player can skip the troop's turn or wait for a later turn. For each of those actions, a message is sent to the other device, which is in the Executor state. The device in Executor state simply replicates every action that it receives by string message from the Master device so that exactly the same thing happens on both devices. At the end of the turn, both devices check whose troop's turn it is next and accordingly switch to the Master or Executor state.

Chapter 4

Implementation

4.1 Save Files

Save files are used for two purposes in this game. Firstly for the player to save a game if he wants to exit and continue playing later. Secondly, save files are used to transfer the state of the game to the other player at the end of each turn in a network game.

The files are encoded in XML. They follow the syntax of TMX map files [17], which are used by the *Tiled* map editor. Both map and save files can be read by the same parsing function. A save file always contains all the information on the map, they can therefore be seen as an extended map file. This has the advantage, that in a network multiplayer game only the player that starts the game has to be in possession of the map.

4.2 Multiplayer Service

In order to play a multiplayer game over the network, *Android Heroes* has to bind to the Service [18] provided by the *WLAN-Opp* application. This allows the game to keep track of other players in the same network (also called *neighbors*) and communicate with them. For the basic functionality of the game it would therefore suffice to just bind the Activity [19] to that Service.

However, a user may exit the game by pressing the *Home* button, via the *Recent Apps* window or because he receives a call [20]. Especially the player, that is in the waiting state until the other player finishes his turn, may want to do something else with his phone. In such a case, the Activity is in the stopped state. The system may then destroy the Activity because it hasn't been used in a while or the foreground Activity requires more resources. In such a case, also the *WLAN-Opp* service would be unbound. If the other player then for example finishes his turn, the first player can not be notified.

Therefore, an additional Service is needed, that upholds the connection to the *WLAN-Opp* Service. This service is started as a foreground Service, so that the system it will not destroy it.

4.3 Graphics



Figure 4.1: Some of the over 280 images used in *Android Heroes*



Figure 4.2: Shooting animation of the elf in battle

One of the most time consuming tasks of creating this game is creating all the graphics. Currently, there are over 280 image files in the resources folder of the project. Many of them, like trees, mountains and grass are modified from public domain images [21, 22, 23], others are hand drawn on the computer, first with a mouse, later using a Wacom Bamboo [24] graphics tablet. Attack and walking animations of the creatures in battle are done in frame by frame animation with 8 frames each.

4.3.1 Rendering the Views

Android's XML based layout allows quick and easy creation of all sorts of menus, lists and buttons. *Android Heroes* makes use of this for the simply structured main menu, load and save game menu and multiplayer menu. However, to render the contents of the game itself, a higher amount of customization and control is needed.

Using a SurfaceView with a separate Thread

Each of the three views, the MapView, the TownView and the BattleView, are extending the `android.view.SurfaceView` class [25]. Other than the View class [26] that it extends, the SurfaceView class provides access to a surface that can be drawn on by a secondary thread. To make sure the surface is only accessed while it is valid, the `android.view.SurfaceHolder.Callback` interface [27] is implemented, and its functions `surfaceCreated(SurfaceHolder holder)` and `surfaceDestroyed(SurfaceHolder holder)` are overridden. `SurfaceCreated(...)` is called when the View is first shown on the device, and its job is to start the rendering thread. `SurfaceDestroyed(...)`, on the other hand, is called when the View is no longer shown and stops the rendering thread.

The rendering thread extends `java.lang.Thread` [28]. Its main purpose is to repeatedly call the `doDraw(Canvas canvas)` method of the associated view. To do so, first `SurfaceView.getHolder().lockCanvas()` is called. This returns a Canvas object through which the pixels of the surface can be set and makes sure, that the surface isn't destroyed or changed during that process. After drawing to the canvas is finished, `SurfaceView.getHolder().unlockCanvasAndPost(Canvas canvas)` shows the pixels of the canvas on the screen and releases the surface.

Handling of Bitmaps

Most of what is shown on the screen in-game consists of bitmaps. Bitmaps store per pixel red, green, blue and alpha channels of an image. In android the class used for this is `android.graphics.Bitmap` [29]. Bitmaps are loaded using `android.graphics.BitmapFactory` [30], typically from the application resources, although also files, byte arrays and `InputStreams` are possible.

Once a bitmap is loaded it can be drawn to a Canvas, using the various `android.graphics.Canvas.drawBitmap(...)` methods. A problem is though, that bitmaps use a lot of memory, a 2048×1536 image uses about 12MB [31]. Android limits the amount of memory each application is allowed to use for bitmaps, even if the device's RAM still has sufficient free space, which can lead to a `java.lang.OutOfMemory` exception. To avoid this happening, Bitmaps should not be loaded to memory in a higher resolution than they will be actually drawn and they should not be kept in memory when they are no longer used. To free the allocated memory for a Bitmap, the `Bitmap.recycle()` function has to be called. Trying to draw a recycled or even null Bitmap to the canvas will also throw an exception. Because of all these factors careful handling of Bitmaps is needed. For that purpose the `ScaledBitmap` and `ScaledBitmapObserver` classes were created.

The `ScaledBitmap` class is basically a wrapping class for a Bitmap. It provides several different functions to draw the Bitmap to a Canvas. They facilitate the drawing process by keeping track of the current size of the Bitmap and re-decoding it when a different scale or resolution is needed. This not only keeps memory usage low, it also prevents exceptions with recycled or null bitmaps. A Bitmap is loaded to the memory when it is first used. To make sure the memory is freed once a Bitmap is no longer used, an instance of `ScaledBitmapObserver` keeps track of all the `ScaledBitmaps`. When a `ScaleBitmap` loads a Bitmap, it registers to the observer. When it is drawn to the canvas, a counter is set to a

small number (the number can be adjusted, 5 seems to work well). Each time the view's `doDraw(...)` method is called, the observer decreases each Scaled-Bitmap's counter and, when zero is reached, the Bitmap is recycled to free the memory. This means, when a Bitmap isn't drawn a few frames in a row it will be removed from the memory, and reloaded once it is needed again.

Chapter 5

Evaluation

5.1 The Game

In the current state, the game is fully playable, either on one device or over network with two players via *WLAN-Opp*. However, it is far from complete. It features one town type with six different buildings, four different creatures and one 32 times 32 squares sized map that is covered by two ground types and 27 different objects. A complete game takes about 30 to 60 minutes. Due to the lack of content, the game does not yet achieve the complexity it is aiming for. However it sets a good foundation for far more creatures, towns, buildings, maps and content in general to be added.

5.2 Save Files

The save files are encoded in XML with no compression, which is not the most memory-efficient solution. The file size depends greatly on the map size, since the map is included in each save file. The currently only map has a size of 32 times 32 tiles. This leads to file sizes ranging between 50KB and 100KB, which is small enough to only make an insignificant delay when sending over WLAN. However, increasing the map size to 128 times 128, leads to a proportional rise in file size. Sixteen times as many tiles leads to sixteen times the file size, making about 1MB, which is not negligible anymore. To improve this, a simple ZIP compression could be applied. In a test, a 60KB save file occupied only 2KB after compression, a 1MB file even only 6KB corresponding to a 97% and 99% compression ratio respectively.

5.3 Devices Tested On

The game was tested on three different devices. The main development device was a Samsung Galaxy Note [32] with a screen size of 5.3 inches and a resolution of 1280 times 800 pixels. It was tested on Android versions 4.0 Ice Cream Sandwich and 4.1 Jelly Bean. Tested were also the HTC One X [33] (4.7", 1280

x 720, Android versions 4.0 and 4.1) as well as the Samsung Galaxy Nexus [34] (4.65", 1280 x 720, Android versions 4.1 and 4.2). The game runs without problems on all of those devices. Furthermore it was tested on a ZTE Blade [35] (3.5", 800 x 480, Android version 4.0, Cyanogen Mod 9). It also runs on this device, although the interface of the game is not adapted to this lower resolution. Therefore the game can not currently be recommended for this phone.

Chapter 6

Future Work

As mentioned in the evaluation, the basic features of the game are implemented, but there is still a lot of content to be added. The goal in the far future is for the game to be released on the Google Play Store. For the release version, it is planned for there to be multiple different fractions with seven creature types each. The main reason that this is lacking now is the immense amount of time it consumes to create the graphics for the creatures, their animations, their buildings and the town itself.

Also, since *Android Heroes* is a turn based strategy game, players will probably expect more complexity. For this, several features are planned. In the present state of the game, heroes' only purpose is to carry an army around the map. In the future, every hero should be unique, gain experience and rise in level through battles. A magic system will allow spells to be casted during the battles. Items will be able to enhance the attributes of heroes and grant them advantages in battle.

Not everyone has someone to play this game with all the time, therefore, a single player mode should be implemented. This will require an artificial intelligence opponent, which will be a huge challenge to create.

On the technical side, the game needs to be optimized for different screen resolutions and sizes.

Chapter 7

Conclusion

Motivated by the lack of complex games on the Google Play Store [6], a turn-based strategy game named *Android Heroes* was implemented. It is a multiplayer game for two players that can either be played on one device or over network using the *WLAN-Opp* [1] platform. The game takes place on three different views, the map, town and battle views. On the map view, a player can move his heroes, which each command an army. Heroes are used to attack the other player and collect resources. The town view is where a player can recruit new armies and heroes. The battle view is reached when a hero attacks an enemy. A battle ends when one army has no more troops left.

For multiplayer with two devices over network, two types of interactions between the two players were implemented: Ending one player's turn with transfer of the complete state of the game and the battle between two players. In the battle each change to the battlefield, such as a moving or attacking a troop, is transferred to the other player.

Because *Android Heroes* is a turn based game, where interactions between the two players only occur once in a while, no continuous connection between the two players is required. Therefore it is possible to use opportunistic networking, which is provided by the *WLAN-Opp* application.

The game is designed in a way that additional content can easily be added. Additional maps can be created using the *Tiled Map Editor* [16]. Furthermore, the game has one town type and four creature types. In the future there will be three town types which each have five unique creatures. In short, the current state of the game is a good foundation. All major features are implemented, but it still needs more content to unfold its full potential.

Bibliography

- [1] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, and Franck Legendre. WiFi-Opp: Ad-Hoc-less Opportunistic Networking. In *ACM MobiCom Workshop on Challenged Networks (Chants 2011)*, Las Vegas, NV, USA, September 2011.
- [2] OUYA: A New Kind of Video Game Console, Kickstarter, January 2013. www.kickstarter.com/projects/ouya/ouya-a-new-kind-of-video-game-console.
- [3] OUYA, January 2013. www.ouya.tv.
- [4] Portable Handheld PC Gaming - NVIDIA Shield, January 2013. shield.nvidia.com.
- [5] Valera Koltsov. Age of Heroes, January 2013. www.heroesofmightandmagic.com.
- [6] Android Apps on Google Play, January 2013. play.google.com.
- [7] John McCann. Google Play store matches Apple's as it hits 700,000 apps, Techradar, January 2013. www.techradar.com/news/phone-and-communications/mobile-phones/google-play-store-matches-apples-as-it-hits-700000-apps-1108577.
- [8] Zachary Lutz. Google Play hits 25 billion app downloads, Engadget, January 2013. www.engadget.com/2012/09/26/google-play-hits-25-billion-app-downloads/.
- [9] Plants vs. Zombies - Android Apps on Google Play, January 2013. play.google.com/store/apps/details?id=com.popcap.pvz_row.
- [10] Fieldrunners - Android Apps on Google Play, January 2013. play.google.com/store/apps/details?id=com.subatomicstudios.
- [11] Sentinel 3 - Android Apps on Google Play, January 2013. play.google.com/store/apps/details?id=com.Origin8.Sentinel3.
- [12] Great Big War Game - Android Apps on Google Play, January 2013. play.google.com/store/apps/details?id=com.rubicon.dev.gbwg.

- [13] Battle for Wesnoth - Android Apps on Google Play, January 2013.
play.google.com/store/apps/details?id=com.androthsoft.battle.
- [14] Majesty: Fantasy Kingdom Sim - Android Apps on Google Play, January 2013.
play.google.com/store/apps/details?id=com.herocraft.game.majesty.
- [15] Free Heroes 2 - Android Apps on Google Play, January 2013.
play.google.com/store/apps/details?id=net.sourceforge.fheroes2.
- [16] Tiled Map Editor, January 2013.
www.mapeditor.org.
- [17] Thorbjørn Lindeijer. TMX Map Format, GitHub, January 2013.
github.com/bjorn/tiled/wiki/TMX-Map-Format.
- [18] Service, Android Developers, January 2013.
developer.android.com/reference/android/app/Service.html.
- [19] Activity, Android Developers, January 2013.
developer.android.com/reference/android/app/Activity.html.
- [20] Managing the Activity Lifecycle, Android Developers, January 2013.
developer.android.com/training/basics/activity-lifecycle/index.html.
- [21] Public Domain Images, January 2013.
www.public-domain-image.com.
- [22] 5000 Free Stock Photos, January 2013.
www.public-domain-photos.com.
- [23] Google Advanced Image Search, January 2013.
www.google.com/advanced_image_search.
- [24] Bamboo Pen Tablets, Wacom, January 2013.
www.wacom.com/products/pen-tablets/bamboo.
- [25] SurfaceView, Android Developers, January 2013.
developer.android.com/reference/android/view/SurfaceView.html.
- [26] View, Android Developers, January 2013.
developer.android.com/reference/android/view/View.html.
- [27] SurfaceHolder.Callback, Android Developers, January 2013.
developer.android.com/reference/android/view/SurfaceHolder.Callback.html.
- [28] Thread, Android Developers, January 2013.
developer.android.com/reference/java/lang/Thread.html.
- [29] Bitmap, Android Developers, January 2013.
developer.android.com/reference/android/graphics/Bitmap.html.
- [30] BitmapFactory, Android Developers, January 2013.
developer.android.com/reference/android/graphics/BitmapFactory.html.

- [31] Loading Large Bitmaps Efficiently, Android Developers, January 2013.
developer.android.com/training/displaying-bitmaps/load-bitmap.html.
- [32] Samsung GALAXY Note - Samsung Mobile, January 2013.
www.samsung.com/global/microsite/galaxynote/note/spec.html?type=find.
- [33] HTC One X Overview - HTC Smartphones, January 2013.
www.htc.com/www/smartphones/htc-one-x/.
- [34] GALAXY Nexus - Tech Specs, Samsung, January 2013.
www.samsung.com/hk_en/consumer/mobile/mobile-phones/smartphone/GT-I9250TSATGY-spec.
- [35] ZTE Blade - Full phone specifications, GSMArena, January 2013.
www.gsmarena.com/zte_blade-3391.php.