Journal of
Information Technology & Software Engineering

Research Article   **Open Access**

Design and Implementation of an Android Game: Duelling Phone

Alaa Hassan*

University of Kirkuk, Kirkuk, Iraq

Abstract

This paper presents the design and implementation of a unique two person Android game, called 'Duelling Phone'. To the Author's best knowledge, the reported game has never been implemented before.

The Model-View Controller (MVC) design pattern has been applied, as it represents a pattern for the Graphical User Interface. This particular design pattern has been used in order to separate the data from the visual representation, since it consists of three separate parts. The 'model' represents the actual data in the program, while the 'view' only illustrates the user interface. The 'controller', finally, controls the process of the communication between the view and the model.

The process that has been followed in order to implement this application is 'Rapid Application Development' (RAD), which represents the process where the development cycle of any application is accelerated. It allows for the application to be developed faster, helping to retain a valuable resource.

Two modes have been implemented successfully: 'Single Player' mode allows the player to play against the Android device, while the 'Two Player' mode allows two players to play against each other, passing one Android device between them. The results were established following testing on the emulator, and by using an actual Android device.

Introduction

Mobile applications represent a relatively new concept in today's market, and although Personal Digital Assistants (PDAs) were the first devices capable of handling mobile applications, these devices never became mainstream, tending to be used only by computer experts. Over time, the functionality of the mobile phone has engaged with PDAs, resulting in the generation of new devices, known as 'smartphones'.

Currently, three different smartphone developer platforms are available, represented by: the BlackBerry by RIM, which is not suitable to be utilized as a gaming platform, due to the small screen size; and two others representing the technological peak of industry operating systems- the iPhone Operating System (iOS) by Apple, and the Android by Google, both of which are ideal platform choices for game programming. However, unlike the iOS, Android applications can be programmed under an 'open source' license, and no special equipment is required to build a game. On the other hand, to build a game on Apple devices, both a developer license, and a Mac computer, are required to compile a gaming application [1]. The Android platform is one such platform whereby application writing is supported by Java programming languages [2], Eclipse, Free SDK and Emulator(SDK website), [3] (Eclipse website); in addition, Android devices have four different screen sizes making it possible to adapt the game for more than one screen size. In these circumstances, the Google Android seems to be the most logical platform to choose for the project proposed in this paper.

A 'mobile game' is a video game that can be played on the mobile phone, played by using the technologies installed within the device itself. The majority of smartphones are equipped with the hardware necessary for more advanced and complex games, which may be either a single or multiplayer game, the latter whereby consumers play the same game on their phones while connected to a shared network.

Games on Android are already soaring in popularity, and as the Android increases in computational power, the demand for more challenging games increases. Although Android games are programmed using the Java programming language, it doesn't use some of its core libraries, instead, its own libraries are utilized which are crafted for the specific needs of the platform. Consequently, it was necessary to study the specific details of the Android platform, and how to use its libraries. To write a good program and implement a game for the Android, it was necessary to gain a familiarity with the specific device that was used to run the application, in terms of the version of Android, and the resolution of the screen.

The objective of the project presented in this paper is to design, implement and test 'Duelling Phone', a two person Android game. It is not a difficult game to implement, but the extensions required to make it a practical, easy-to-use game with a variety of options, makes the project building challenging and results in an interesting, two persons, Android game.

Although the proposed game is similar to other Android games designed in the past(such as 'Tic Tac Toe', commonly known as 'Noughts and Crosses' in the UK) in the way that it is a two-persons game, based on an old pencil-and-paper game, the aim of this game, however, is quite different. It is also worth mentioning that the main focus of this project has been applied to implementing all requirements required to make the game playable, while artwork was kept minimal.

The rest of the paper is organized as follows: Section II presents the description of the implemented game. Section III is devoted to the design process of the game, represented by the Unified Modelling language (UML) diagrams and the wireframe of the application. Section IV discusses the implementation and testing of the Duelling Phone game. Finally, conclusions are given in Section V.

Game Description

As previously mentioned, the aim of this research is to design and

implement the 'Duelling Phone', two person Android game, and this section has been devoted to the description of what the formal elements and the challenges in the game as well as the balance in this game and how it works, by demonstrating a simple example.

- Formal Elements- the formal elements in this game are represented by the following elements: 1. Board; 2. Steps; 3. Coins; 4. Stack of chips; 5. Player side goal boundary

- Challenges - Beating the opponent by reaching the goal line on player's end.

- Balance- Both players have equal chance of losing and winning based on the value of the chips.

- There is no dominant strategy.

The game consists of two opposing sides, one side for each player, with three 'steps' on each side. A coin(or similar marker) will be on the middle line(denoted '0'), that separates the two sides. Whichever player manages to get to the end of their three steps first will be deemed the winner.

The game starts with each player have the same number of chips (50) in a stack, with each player wagering on a number. Whichever player wagered the most chips in that turn take one step towards their goal. If both players bet the same number of chips, the coin will remain in its place. On each turn, the number of each player's remaining chips is reduced from the stack by subtracting the amount bet. Subsequently, the game will continue until one of the players either reaches the end, or the game ceases (by neither player reaching the end before their pile of chips is reduced to 0). The following is an example of the 'Duelling Phone' game, won by the player on the right hand side. It shows the design of the game and how it works, step by step:

In this example, firstly, Figure 1 shows the game in its initial state;
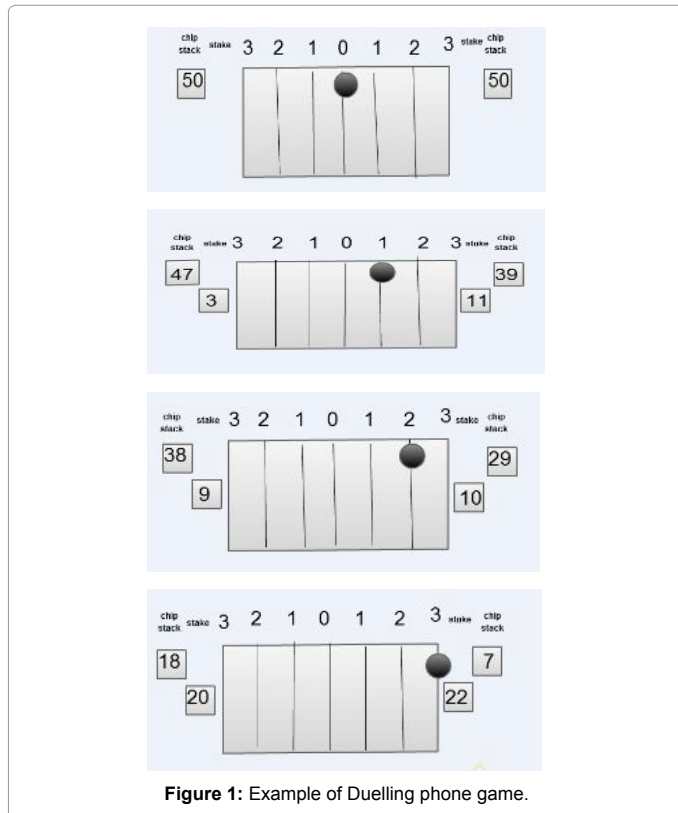

Figure 1: Example of Duelling phone game.

whereby no number has been bet, and the chip stacks contain the opening 50. The first turn of the game, in which the first player, located on the left hand side of the game boards, has bet 3 of their chips, so has 47 remaining in their chip stack (50-3 = 47). The second player, located on the right hand side of the game board, has bet 11 chips so has a remaining 39 in the chip stack (50-11). As a result, the coin (or other marker) moves one step forwards the goal of the second player, player 2 having bet a greater number than the first player. In the next turn, player 1 bets 9 of his/her 47 chips, so 38 now remain in the chip stack. Player 2, however, has bet 10 of 39 chips remaining leaving 29 in the chip stack, and, with a greater winning bet, again sees the marker moved another step towards his/her goal. In the next turn, the coin is in a critical position. In this situation, the player 1 should be aware, and critically think about choosing a number that will stop player 2 from winning, whether by a larger bet, or at a minimum, an equal bet to ensure the coin moves no nearer the goal of player 2. In our example, however, the first player chose only 20 for this bet, despite knowing that the second player could choose any number up to the 29 in their chip stack. In this situation, the player 2 chose to bet 22, ensuring the coin/marker moved the final step towards his/her goal, and in turn giving the win to the second player. In this turn, if the first player had thought more critically, and bet a larger number than the second player, the coin would have moved a step back towards the goal of player 1.

Functional Specifications and Technologies

Game specification

As mentioned earlier, the aim of this project is designing and implementing a Duelling phone-two person android game.

- The game should work perfectly with all the probabilities that may contain.

- The player of the Duelling phone game should be able to play the game either with the Android device (SinglePlayer mode) or with another player (TwoPlayer mode).

- In future, the developer should be able to extend the game and adding new feature and new forms of players without affecting the existing code of the game.

Technologies required for the implementation

These packages represent the software packages, chosen for the developing environment. Since it has been recommended by the official Android developer website, the game has been implemented using Eclipse. The available Android plugins simplify the building and running of the Android emulators also, making this choice preferable.

The Dalvik Debug Monitor Server (DDMS) is a tool which can be used for SMS and call spoofing. Boasting a plugin for Eclipse, it is able to access from within the IDE, and is used for debugging throughout the development process, finding any possible leaks in the memory usage and creating screenshots (these can then be posted on the Android market when marketing any application). On the other hand, Android Logcat shows information about user interactions, phone state, and the stack trace of any exceptions.

Quality assurance- lint warnings, emulators and test devices

There are many tools for testing: The Android Lint will be used to help minimize possible problems. This tool is capable of showing any warnings for any code in Eclipse that may cause accessibility problems, or fails to conform to best practice. The full list of checks that can be performed can be found on the Android Lint website.

Design

The functionality of the application (game), the way of using the interface, and how to play the game should be presented to the user in an easy-to-understand manner, while every player should be able to play the game without having to read the help guide first.

This section will discuss how the program was designed, represented by the use of a Case Diagram, an Activity Diagram, and the Wireframe that was used to design and understand the user flow of the application.

A design pattern

The design pattern is a template that shows the interactions between the program components, such as the classes and the objects. It is used as a reusable solution for software engineering problems.

Model-view-controller

The use of the MVC design pattern helped to separate the programming of the view and the model in two different classes, helping to extend and modify both much easier. The view provides a visual representation of the data model, while the controller links the model and the view, by 'listening' for events from the view, and carrying out the corresponding action on the model (Figure 2).

Design patterns in the duelling phone program design

The Model-View-Controller has been used in this program, where the Game.class represents the model containing all variables and methods employed in the game, and represented by the view. The work was started with a model class, Game.class, and one view class, 'DuellingPhone.class'. Following completion of this mode, which represents the SinglePlayer mode, the first extension of the game was made, represented by the Two Player mode. The first extension was implemented by adding another view for the Two Player mode, and the functions relating to the mode, simply added to the Game.class.

Use case diagram

One of the first steps in designing the game was to model it on its phases, where the Use case was designed first. Describing a set of actions the user can take while using the application, each use case should provide some valuable result to the users interacting with the system, but does not attempt to represent the number of times, nor the order, that the system's actions should be executed [4,5].

The following figure shows the Use Case diagram for the Duelling Phone game (Figure 3).

Figure 2: MVC pattern.

Figure 3: Use case diagram.
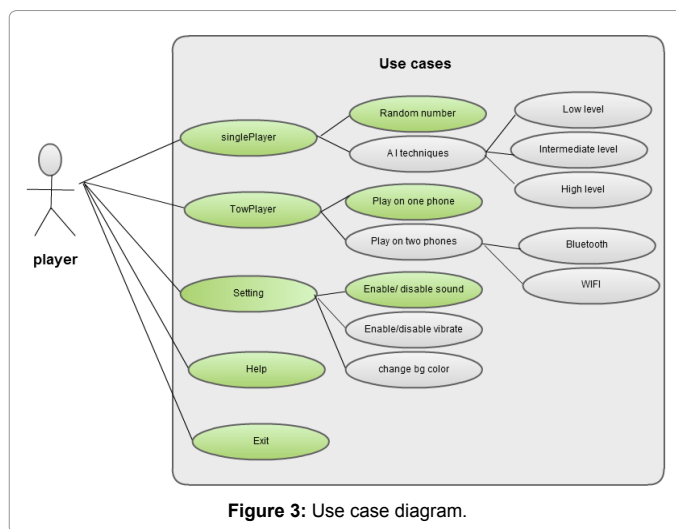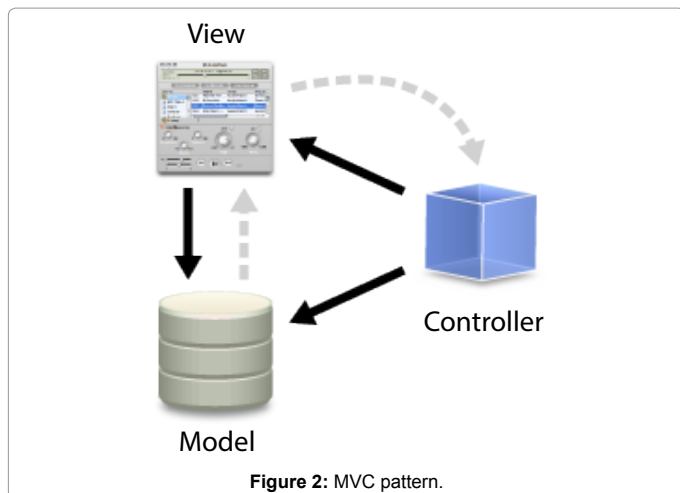
General use case description

The following are short descriptions of each use case illustrate in the use case diagram that will be used in the game (Table 1).

Activity diagrams

The Activity Diagram is an important UML diagram, describing how activities are coordinated. It essentially represents the flowchart showing the flow from one activity to another, whereby the activity can be represented as an operation of the system.

The activity diagram is potentially used for modeling the logic, captured by a single use case. With respect to the Duelling Phone game, the activity diagram has been designed for the first and second use cases.

The activity diagram shown in Figure 4 defines the basic structure of the SinglePlayer mode, and represents the realisation of the SinglePlayer Use case. When a player wishes to play on his own against the Android mobile device, it is performed, and after selecting the SinglePlayer mode from the Menu screen, the application proceeds to take the player to the screen showing the game itself. The game starts with 50 chips in the stack. When the player bets a number, this bet number must be less than the number of chips remaining in the chip stack, and after pressing the 'Go' button, a random number appears to represent the bet of the 'second player',-the Android device. The coin will then move one step toward the side of the player betting the larger number. If both numbers were the same, the coin's positional value will remain constant and will stay where it was. The number in the chip stack is reduced by the number of chips bet, and the game subsequently continues. It does so until either one of the players wins (reaches their end first), or both players end up with zero chips remaining in their stack.

The activity diagram for the Two Player mode is shown in Figure 5, and in this mode, this flow occurs also (Figure 6) presents a high-level design of the activity diagram, whereby the activity diagrams of the SinglePlayer and TwoPlayer modes were combined in one. By following this activity diagram, the player should choose either to play the SinglePlayer mode or the Two Player mode, and depending on what is chosen, the process in Figures 4 and 5 will commence.

Wireframe

The next step in designing the game was to design the Wireframe, which represents a visual representation of the interface, designed to

Use case:	Single Player
Goal	Start a new game in single player mode.
Description	In this mode, the player will play against the Android mobile device. The Android acts as the second player and here, there are two options. The device gives a random number as a bet number or uses AI techniques.

Use case:	Random number
Goal	Play the game in single player mode
Description	In this mode, the player will play against the Android mobile device. The Android acts as the second player and gives a random number as a bet number.

Use case:	AI techniques
Goal	Play the game in single player mode
Description	In this mode, the player will play against the Android mobile device. The Android acts as the second player and playing in an intelligent way using AI techniques.

Use case:	TwoPlayer
Goal	Start a new game in two player mode
Description	In this mode, two players will play against each other either on one Android device or on two Android devices.

Use case:	Play on one phone
Goal	Start a new game in two player mode
Description	In this mode, two players will play against each other on one Android device by pass the device between the two players.

Use case:	Play on two phones
Goal	Start a new game in two player mode
Description	In this mode, two players will play against each other on two Android devices. These devices connected to each other using via Bluetooth or WIFI.

Use case:	Setting
Goal	Show the setting screen
Description	Show the setting screen where the player can change the setting of the game like enabling or disabling the background music, changing the background color or enabling/disabling the vibration.

Use case:	Help
Goal	Show the Help screen.
Description	Shows an explanation of how the game works.

Use case:	Exit
Goal	Exit the game.
Description	Exit the playing screen and Finishing the game.

Table 1: Short descriptions of each use case illustrate in the use case diagram.

plan the functionality of each activity in the game. Used as a technical guide to specify the structure of the interface, before creating the actual UI for the game, it shows how the interface is compiled, its content, and how the data is organised. It also shows how the interface works, how the user interacts with the interface, and how it travels between the different states.

Implementation and Testing

Implementation

This application is made up of two, equally important parts- the user interface part, and the application logic part. It is worth noting that the programming environment of Android comes with great tools for localisation and internationalisation. All text, as a result, has been written to appear on the Android game's screen, and is originated from separate language files that make the game easily translatable. All activities, as well as the design for the (GUI), and the layouts, have been implemented in this application, and will be discussed in this Section.
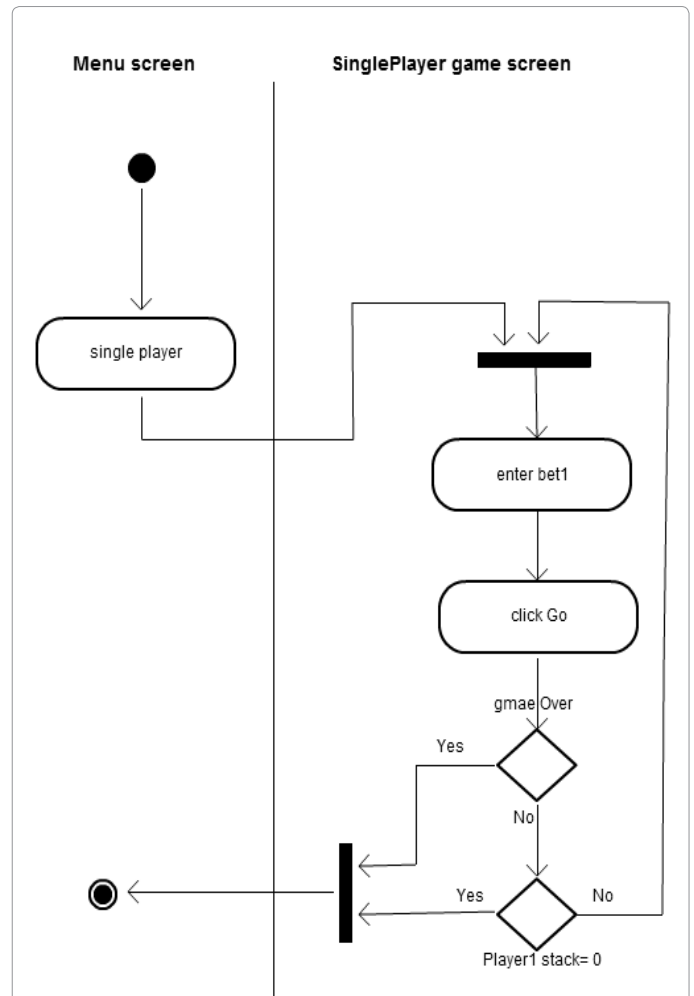
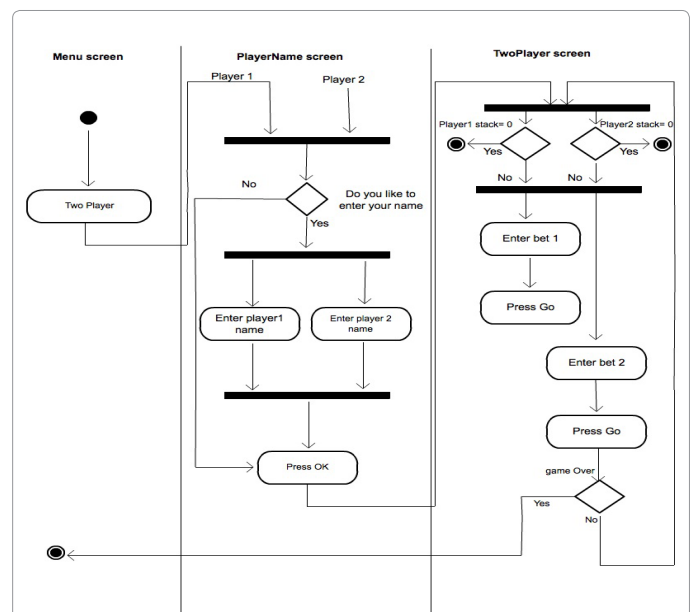**Figure 4:** The Activity diagram for the Single player mode.

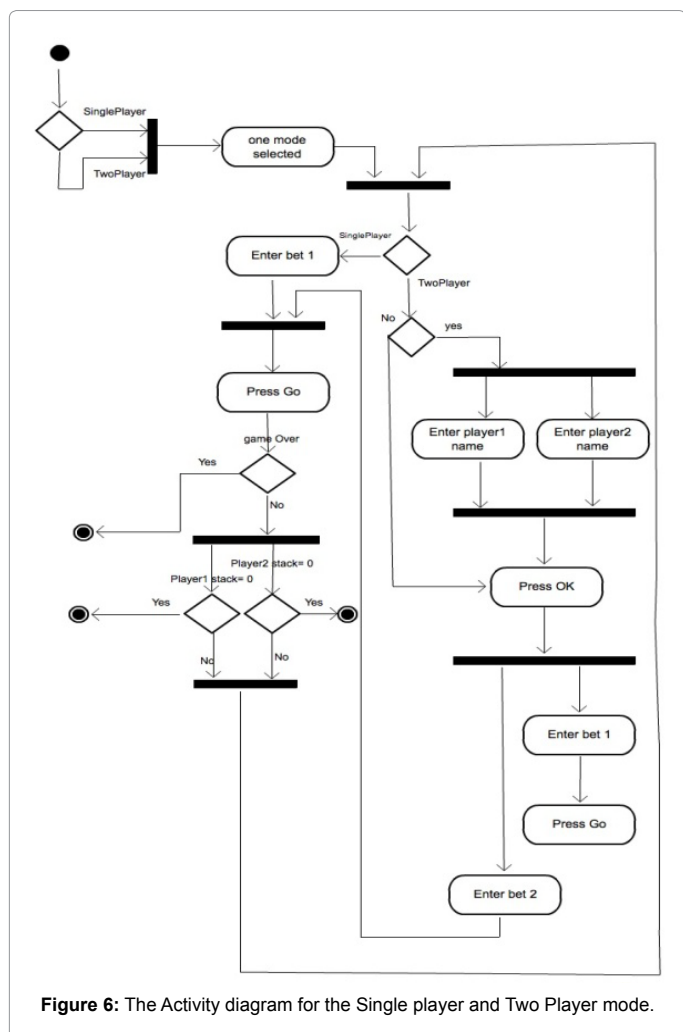**Figure 5:** The Activity diagram for the Twoplayer mode.

Figure 6: The Activity diagram for the Single player and Two Player mode.

Implementing the activities

This section focuses on detailing the functions and methods that have been implemented in each activity. The following diagram shows all the activities that have been implemented to run this game (Figure 7).

Splash activity

The 'splash' is the first activity to start within the Duelling Phone game, and represents the launcher activity of the game. The activity is coded using a timer task to show the splash screen first, to display the Duelling Phone Logo to the user, and then to move on to the Menu activity of the application. The amount of time allocated to the display of the splash screen was 3 seconds. It is recommended in the literature that it would be good to have the splash screen display for a minimum of two seconds, as; anything shorter would not give the player a proper chance to view it. On the other hand, if it remains on the screen for more than 5 seconds, it violates the '5 second rule', which specifies that the interactive application should respond after a maximum of 5 seconds whenever possible [6]. Finally, this activity also uses a Media player object, in order to start the background music.

Menu activity

It represents the second activity that can be accessed in this game. Normally, the On Create() method was used to set the layout from the menu.xml file, and the activity also creates the button listeners that
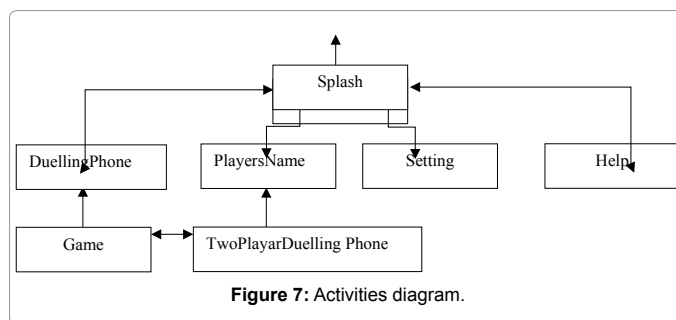

Figure 7: Activities diagram.

are used to start the SinglePlayer mode, Two Player mode, the game Settings, the Help, and to Exit the game [7].

Duelling phone activity

This activity represents the view for the first mode the game, the SinglePlayer mode, which is devoted to playing against the device. The Duelling Phone activity includes all objects for initialising the Edit Text, Text View and buttons. It contains a listener for the 'Go' button, allowing state of the game to change after clicking it. The(Math.ceil(Math.random()*(game.PLAYER2STACKChIPS())*0.15) equation have been used to allow the mobile device gave a random number as a bet number. This random number multiplied by game.PLAYER2STACKChIPS() to allow the Android device bet a number of chips equal or less than the number of chips in his stack, while it multiplied by (0.15) to add simple level of intelligence to the game. From the equation, it is clear that a simple level of Artificial intelligence techniques have been applied. However, with more time available, more advanced artificial intelligent level can be implemented [8].

Players name activity

This activity includes the initialisations of buttons, the Text Views and the Edit Texts, which allow the players to enter their name. By pressing the OK button, a new intent will be called to start a new screen, the Two Player Duelling Phone screen.

Two player duelling phone activity

This activity represents the view for the second mode of the game, the Two Player mode, where two players play against each other using one Android device. The Two Player Duelling Phone activity includes all objects for initialising the Edit Text, Text View and buttons, and contains a listener for the 'Go' buttons to hide the first player's bet number and to change the state of the game in order to start another turn. After the player presses the first 'Go' button, the Edit Text for bet1 will be disabled until the second player finish their turn, meaning that the first player cannot play again before the second player finishes [9].

Game activity

Representing the backbone of this application, this activity contains the main methods needed to be implemented in order to run the game, since it represents the home to the model, and the controller, of this game. It contains the on Draw() method, which uses the bitmap Factory necessary to draw the image on the screen. Two other important methods have been created here- the CalculatePlayer1Stack() and the CalculatePlayer2Stack(), methods used in order to calculate the remaining numbers of chips in the stacks after each bet. For the Two player mode, similar methods were implemented. Correspondingly, the player1Stack_IsEmpty() and player2Stack_IsEmpty() were used to ensure that the stack is not empty, before allowing the player to bet a number. The Game Over() method is used to stop the game by

returning a Boolean value= true, if both players stacks=0, and neither have reached the final step of the game. Other methods for checking the position, and changing the state, have been used to make the game work properly, with all probabilities that may be included [10].

Setting activity

This activity represents the standard style menu as seen throughout the operating system, which is familiar to Android users. The methods to change the settings of the game are included, such as the enabling and disabling of the sound, vibrations, alteration of the volume, and modifying the background colour, etc. In the case of this project, only the setting to enable/disable the sound has been implemented. In this case, the MediaPlayer object is used to start the background music, an object that can be controlled using a checkbox. A checkbox, rather than a wedge, was chosen, after reviewing a number of other Android applications, it became clear that the majority of them use this type of wedge to represent the sound on the Setting screen.

The get Preference object is used to help obtain the state of the checkbox, and whether it is checked or not. It can then save the state, using the Save() method, or loading the state using the Load() method, but, again due to lack of time, the other setting options were unfortunately left incomplete. For future development, the same methodology could be followed to apply them.

The 'Settings' activity also uses the 'Toast wedge' which allows a message to be shown, notify the player when the sound is being enabled or disabled [11].

Help activity

To display information to help the player during the game, and to give further information about the application (such as the version code), a standard Android activity initialises a text view.

Implementing the interface

The splash screen: This screen contains the logo that has been overwritten from iSmashPhone.com, after adding the name of the game (Duelling Phone) using PhotoShop software. The Logo was aimed at representing the application icon, and the splash screen. The reason for selecting this logo over others was that it was found to have some relationship to duelling, so the aim of the application and what it represented was made easier to understand (Figure 8).
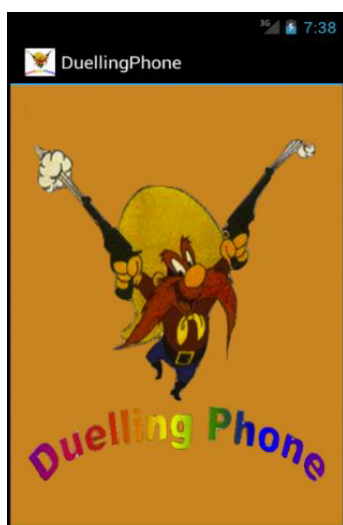

Figure 8: The splash screen.

The menu screen: It is the second screen of the game; a number of buttons used to perform different tasks are included. The first two buttons are the most important, as they take the player to the 'map' screen(the game screen) where the game itself begins. They represent the path to the game screen, where the user should choose one of two paths, either by deciding whether to play with the Android device, or to play with another person using the same handset (Figure 9).

The map screen for the SinglePlayer mode: This mode is the simplest mode of the game where the player plays against the Android phone. Two images are included: Board: that represents the board that represents the steps where the coin should move on and the Coin (a movable image), which should move to the left or to the right depending on the bet numbers (Figure 10).

The players name screen: This screen appears after choosing the Two Player mode from the Menu screen, showing two Edit Texts where the player can enter their name in order to appear on the main game screen. Otherwise, if the player does not wish to enter their name, player 1, player 2 is entered as a default name (Figure 11).
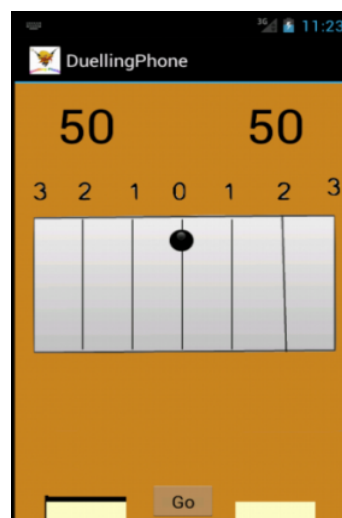

Figure 9: The menu screen.

Figure 10: Single player mode screen.

The map screen for the two player mode: This interface has the same design as the one of the SinglePlayer mode except that the text view of player 2 bet is now an Edit text in order to allow the player to make the bet. In addition, another "Go" button has been added under the edit text of player 1. On clicking this "Go" button, the bet of player 1 will be hidden and the Edit Text will be disabled until player 2 places a bet. This is so that player 1 could not place two consecutive bets without player 2 having paced any [12] (Figure 12).

Testing

The Testing process was used to test any smaller divisions of the code, to ensure it is performing as expected. Two different types of tools have been used to test the Duelling Phone game: the emulator, and device testing.

The emulator, provided by Android for developing Android applications, has many benefits and drawbacks. Although it allows the
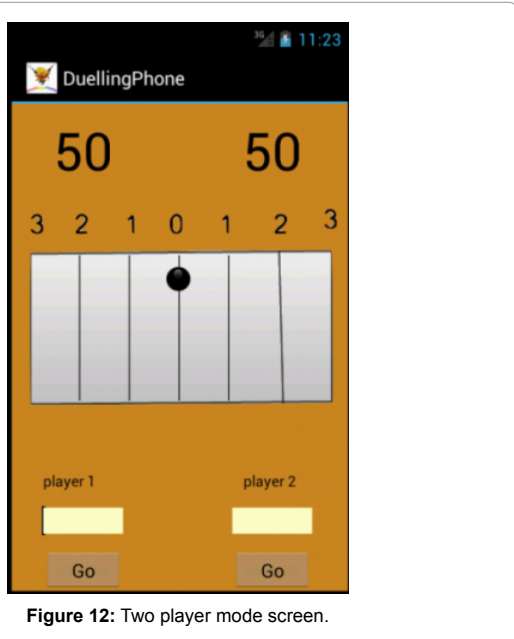

Figure 11: Player name screen.

Figure 12: Two player mode screen.

Device	Android Version	Screen Size	Resolution	Processor	Memory
Samsung Galaxy Nexus	4.0.3	4.65-inch	720 x 1280 pixels	Dual-core 1.2 GHz	1 GB
Samsung Galaxy Nexus	4.0.4	4.65-inch	720 x 1280 pixels	Dual-core 1.2 GHz	1 GB
Samsung Galaxy Nexus	4.1.1	4.65-inch	720 x 1280 pixels	Dual-core 1.2 GHz	1 GB

Table 2: Various devices that have been used to test the application.

developer to test the application across a full range of devices (no extra hardware is required), it does not give a perfect representation of real world usage since there are many differences between the emulator and the actual hardware. There are differences in terms of security, network performance, and user experience, and while the button may be easily selected with the on screen cursor, it could be much more difficult to be selected using an actual touch event. Hence, it is of prime importance to test the application on an actual device. The following table shows the devices that have been used to test the application (Table 2).

It should be noted that in order to fix the problems found in the code, two main methods have been used to test each part of the code upon completion. The first method is represented by the Log class in Android.Util.Log. While running the game, this class creates and shows the log data, which represents the result of feedback from the code, while running it without any need to stop the program. On the other hand, the second method used for testing the project is by creating a breakpoint in the code. While running the program, the code stops at the breakpoint, going through the code, step by step. This provides many details about what the code is doing at any given line.

Conclusion

The goal of this research project was to design and implement the Duelling Phone game on open platforms for mobile devices. The Android has been chosen as the development platform for this game, since its API is open source, and its license is available free of charge which makes it accessible by commercial and experimental developers. Beside the SinglePlayer mode of the game, the game also provides a Two Player mode, whereby two people may compete against each other. The project works perfectly on Android devices with large screen sizes, which represents the vast majority of Android devices on the market.

It should be noted that as with most applications, there are updates and modifications to make the project even better. With new devices emerging in the electronics market every day, continuous support is an important issue to consider with Android applications, as, in the future, some changes may be required to support the different platforms. Although the project is not clear from several limitations, the objective of the project has been successfully achieved, with the creation and implementation of the Duelling Phone game in the SinglePlayer and the Two Player modes.

References

1. Charles A (2011) Nokia and RIM bleeding smartphone share while Android cleans up.

2. Virtual Technology Summit (2016).

3. Eclipse Homepage (2015).

4. Android Homepage (2015).

5. Siebers OP (2012) Object Oriented Modelling, Lecture 03. G64 Object Oriented System. The University of Nottingham.

6. Burnette Ed (2010) Hello, Android: Introducing Google's Mobile Development Platform. (3rdedtn), United States of America : Pragmatic Programmers.

7. Rogers R, Lobardo J, Mednieks Z, Meike G (2009) Android Application Development. O'Reilly Media. (1stedtn).

8. Cheng J (2007) iPhone in depth: the Ars review. Ars Technica.

9. Haselton T (2007) Rumor: 6G iPod manufacturing to begin in August. Ars technica.

10. Murphy Mark L (2009) The Busy Coder's Guide to Advanced Android Development. United States of America: CommonsWare.

11. Meier R (2010) Professional Android 2 Application Development. Wiley Publishing, Ed. Indianapolis, Indiana, US.

12. Application Fundamentals (2015) Android Developers.