Android Game Development
BalloonSaga

By

Chaitanya Patel

11BCE069

Vivek Patel

11BCE073

Savan Patel

11BCE084

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

AHMEDABAD-382481

November 2014

Android Game Development
BalloonSaga

Minor Project

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Engineering

By

Chaitanya Patel
11BCE069

Vivek Patel
11BCE073

Savan Patel
11BCE084

Guide

Prof. Dvijesh Bhatt

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

AHMEDABAD-382481

November 2014

Certificate

This is to certify that the Minor Project entitled **Android Game Development - BalloonSaga** submitted by **Chaitanya Patel (11BCE069), Vivek Patel (11BCE073) and Savan Patel (11BCE084)**, towards the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Engineering of Nirma University, Ahmedabad** is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Dvijesh Bhatt
Assistant Professor,
Dept. of Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Head of Department,
Dept. of Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. Malaram Kumhar
Assistant Professor,
Dept. of Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Acknowledgment

We would like to thank my Faculty Guide, **Prof Dvijesh Bhatt**, Assistant Professor, Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout the Minor project. We heartily thankful to him for his time to time suggestion and the clarity of the concepts of the topic that helped us a lot during this minor project.

We are thankful to all faculty members of Department of Computer Engineering, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

The blessings of God and our family members makes the way for completion of minor project. We are very much grateful to them.

The friends, who always bear and motivate us throughout this minor project, we are thankful to them.

Thanking you.

Chaitanya Patel
11BCE069

Vivek Patel
11BCE073

Savan Patel
11BCE084

Table of Contents

Table of Figures

Chapter 1

Introduction

1.1 About Android

1.1.1 Brief Note

Android is the operating system for smartphones and tablets. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users, power up their Android devices for the first time and start looking for apps, games, and other digital content.

1.1.2 Android Development

Android gives a world-class platform for creating apps and games for Android users everywhere, as well as an open marketplace for distributing to them instantly. Android gives everything that one need to build best-in-class app experiences. It gives a single application model that lets you deploy your apps broadly to hundreds of millions of users across a wide range of devices—from phones to tablets and beyond.

Android also gives tools for creating apps that look great and take advantage of the hardware capabilities available on each device. It automatically adapts UI to look its best on each device, while giving one as much control as one want over UI on different device types.

1.1.3 Android Version Revision History

Each Android version is named after a dessert. The version history of the Android mobile operating system began with the release of the Android^{beta} in November 2007. The first commercial version, Android 1.0, was released in September 2008. Android is under ongoing development by Google

and the Open Handset Alliance (OHA), and has seen a number of updates to its base operating system since its initial release.

Here is the list of Android version according to its release date.

- ✓ Android 1.0 (API level 1)
- ✓ Android 1.1 (API level 2)
- ✓ Android 1.5 Cupcake (API level 3
- ✓ Android 1.6 Donut (API level 4)
- ✓ Android 2.0 Eclair (API level 5)
- ✓ Android 2.0.1 Eclair (API level 6)
- ✓ Android 2.1 Eclair (API level 7)
- ✓ Android 2.2–2.2.3 Froyo (API level 8)
- ✓ Android 2.3–2.3.2 Gingerbread (API level 9)
- ✓ Android 2.3.3–2.3.7 Gingerbread (API level 10)
- ✓ Android 3.0 Honeycomb (API level 11)
- ✓ Android 3.1 Honeycomb (API level 12)
- ✓ Android 3.2 Honeycomb (API level 13)
- ✓ Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)
- ✓ Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)
- ✓ Android 4.1 Jelly Bean (API level 16)
- ✓ Android 4.2 Jelly Bean (API level 17)
- ✓ Android 4.3 Jelly Bean (API level 18)
- ✓ Android 4.4 KitKat (API level 19)
- ✓ Android 4.4 KitKat with wearable extensions (API level 20)
- ✓ Android 5.0 Lollipop (API level 21)

1.2 Project Profile

1.2.1 AndEngine – The Game Engine

AndEngine is a broad 2D game engine which allows game developers, both experienced and inexperienced, to develop games for the Android platform with ease. As easy as it is to "pick up and go," AndEngine includes enough functionality to bring any type of 2D game world to life. AndEngine library is used to connect a features of real physical world in the game. AndEngine library enables real world PhysicsBox2D extension to make the objects work like real world in the game.

1.2.2 Features of AndEngine

The features of AndEngine library are as follows-

- Creation of ultimate Android games with ease using recipes that take advantage of AndEngine's powerful framework and extensions.
- Making games playable across a vast range of devices by implementing multi-touch, performance-optimizations, and accurate, screen-resolution scaling.
- Construction of a customizable, front-end framework that simplifies menu and level creation.
- Use of the Box2D extension to generate realistic, physics-based gameplay and simulations.
- Advantage of source code for a full-featured game built with AndEngine.
- Vector-based graphics with AndEngine's SVG extension.
- Building of animated, responsive Live-Wallpapers for Android's home screen using the AndEngine's Live-Wallpaper extension.
- Control every aspect of interaction that players have with games by managing the Android application lifecycles.

1.2.3 BalloonSaga – The Game

BalloonSaga is android based mobile application. The application is basically an *Arcade Game* and developed by the members of team *RocketAppsStudio*. This game aims to bring out your thinking

skills and it is opened for people of all age groups. This is a game with new idea and developed and uploaded on Google Play store.

This game is of real fun and excitement. It makes the player to think logically and pass through various indulging levels by receiving the stars at each level to unlock the further levels. This is a one player game with ten levels implemented in the game. Help section is included to give user the information about how to play the game and reach to final stages.

1.2.4 Product Scope

The application runs on android platform and compatible with devices which have android version 2.2 or above. An optimized product is generated and more levels will be implemented and new updates will be uploaded on Play Store by the team *RocketAppsStudio*. This game runs smoothly without any specific hardware requirement.

1.2.5 Testing and Debugging Hardware

Computer

Machine	HP Pavilion dv6-6119tx
Processor	Intel Core i5-2410M
Processing Speed	2.3 GHz
RAM	4 GB 1333 MHz DDR3
Operating System	Windows 8.1 Pro

Mobile Device

Devices	1. Sony Xperia SP 2. Nexus 5 3. Moto X
Processor	1. Qualcomm MSM8960T Snapdragon 2. Qualcomm MSM8974 Snapdragon 800 3. Qualcomm MSM8960Pro Snapdragon
Graphics Processor	1. Adreno 320 (Sony Xperia SP, Moto X) 2. Adreno 330 (Nexus 5)
Processing Speed	1. 1.7 GHz (Sony Xperia SP, Moto X) 2. 2.3 GHz (Nexus 5)
RAM	1. 1 GB (Sony Xperia SP) 2. 2 GB (Nexus 5, Moto X)
Operating System	1. Android OS, v4.1 (Jelly Bean) (Sony Xperia SP) 2. Android OS, v5.0 (Lollypop) (Nexus 5) 3. Android OS, v4.4 (KitKat) (Moto X)

Chapter 2

Literature Review

Even for experienced java developer, to develop a game with AndEngine library it is required for developer to learn some basics of AndEngine. Various literature for learning AndEngine and extension libraries including e-book, internet tutorial etc. Though, these are not standard references from the original AndEngine library developer, they provide strong foundation for the basic understanding of it and use of the various functionalities.

The “AndEngine Cookbook”, provides lucid explanation of AndEngine library functions, game engine. The book teaches methods of crating scene resources and populating scene elements. CookBook also discusses one of the important aspect of the game development i.e. use of physicsbox2dlibrary. PhysicsBox2D library is required for simulation of physical world.

Figure 1 AndEngine Logo

Figure 2 Android Logo

The tutorial at matim-dev.org also is a good resource for the kick start of the game development learning. The entire tutorial step by step leads to sample game development. This excellent tutorial covers all aspect of what a beginner game developer requires.

Like all other programming language forum, for example, stackoverflow.com; matim-dev.com has also troubleshooting forums. At this place you can view or discuss various issues that you might face during game development. You can also discover possible solutions to problems you might be facing during development because as of now the forum contains almost all the posts that covers basic issues faced.

The above discussed sources are the only major sources for learning of AndEngine. But still they cover all the aspect of AndEngine. Same sources were referred for learning of the AndEngine and development of this game.

Chapter 3

Design and Methodology

3.1 Brief Introduction

The game BalloonSaga is developed is with the help of AndEngine Library, a popular game development library that is part of game like Angry Birds. The game features ten intuitive levels for user to play requiring to clear levels. BalloonSaga also features soothing music, handy controls and cheesy buttons.

3.1.1 Story-Plot of Game

Mr. Balloon has got trapped in a mysterious world through a portal. He is required to clear each level before getting into the next. Collect the stars, beware of traps, save balloon from spokes and get him to exit door.

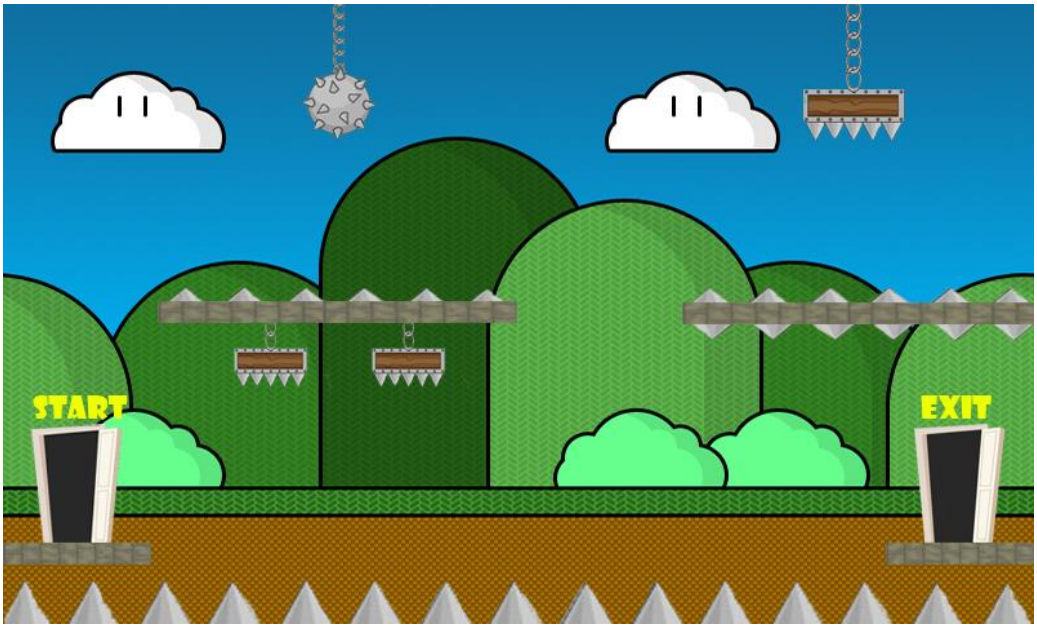

Figure 3 The Game Story-Plot

3.2 Overall Description

3.2.1 Product Perspective

The application is totally based on new ideas of gaming. This application is the puzzle/arcade game which starts with splash screen. It then resumes with main menu which includes control options like play, credits, quit and sound on/off etc. the game consists of a storyline which includes different levels with increasing difficulty at each level. It is a tricky game. This game is developed in order to entertain users. This game is developed on Android platform with AndEngine library extension

3.2.2 Product Functions

The major functions this software performs are:

- Main Activity Module will control transitions between splash screen and menu screen. This Activity will be executed first and start the game.
- Scene Manager Module will be used to control transitions among play scene, menu scene, quit scene and credit scene. In this module, various methods will be used of AndEngine library to control and load resources. The needed resources will be stored in folder called assets.
- Game Manager Module will be used to load resources for playing the game, selecting the levels and logic for playing the game. In this module, resources for different levels will be also defined and used.

3.2.3 User Classes and Characteristics

The various classes of users that may use this product is:

- User: User will have full access to all the game contents. User can open the game, play the game, select level, see help contents, see credit contents and quit the game.

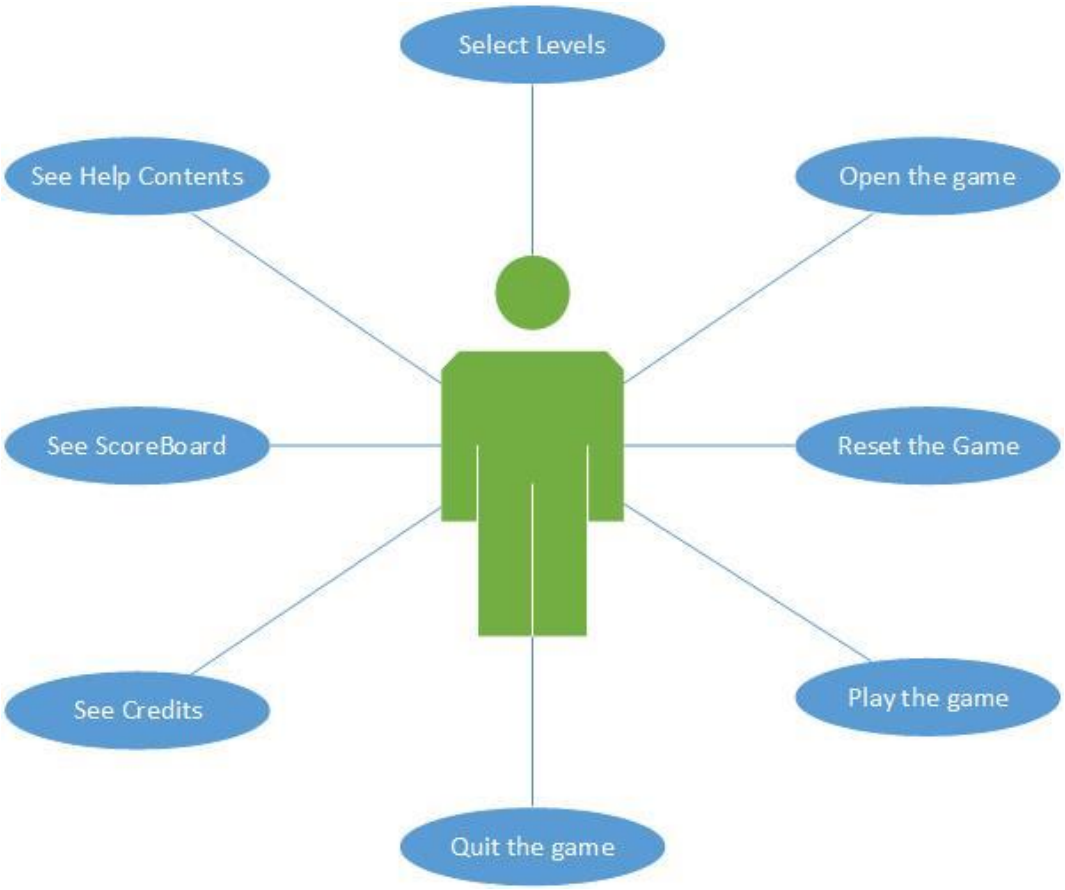

Figure 4 Use Case Diagram

3.2.4 Database Implementation

The game uses the database to store the progress of user in the game. The database in implemented on the basic SQL library that is provided by the Android. Initially, all the levels, except first level are locked when user installs the game for the first time. The level can be unlocked by completing previous levels. One cannot start the locked level without unlocking the level.

The database contains the information whether the level is locked or not. The database is initialized once the game is installed into the device. It is also accessed every time the game play button is pressed to check the current condition of database that whether the database is consistence with all the level values or not. The next level will be unlocked when the current level is completed with some objectives.

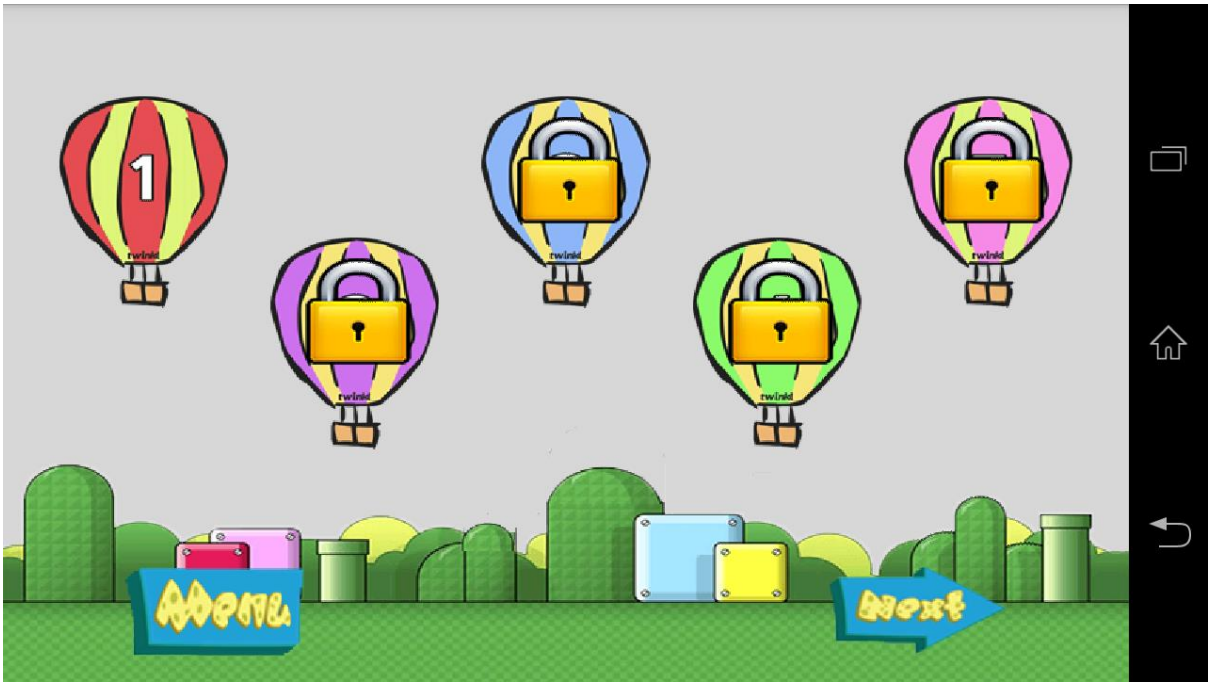

Figure 5 First Level Select Scene (with locked and unlocked levels)

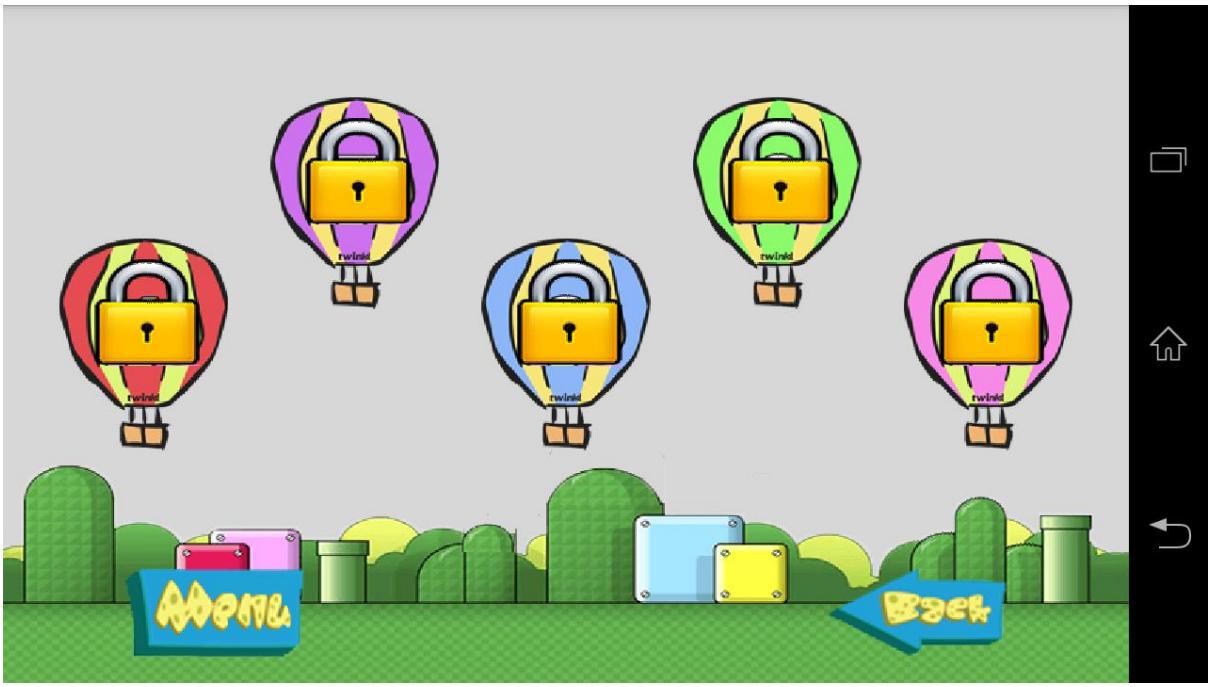

Figure 6 Second Level Select Scene (with all levels locked)

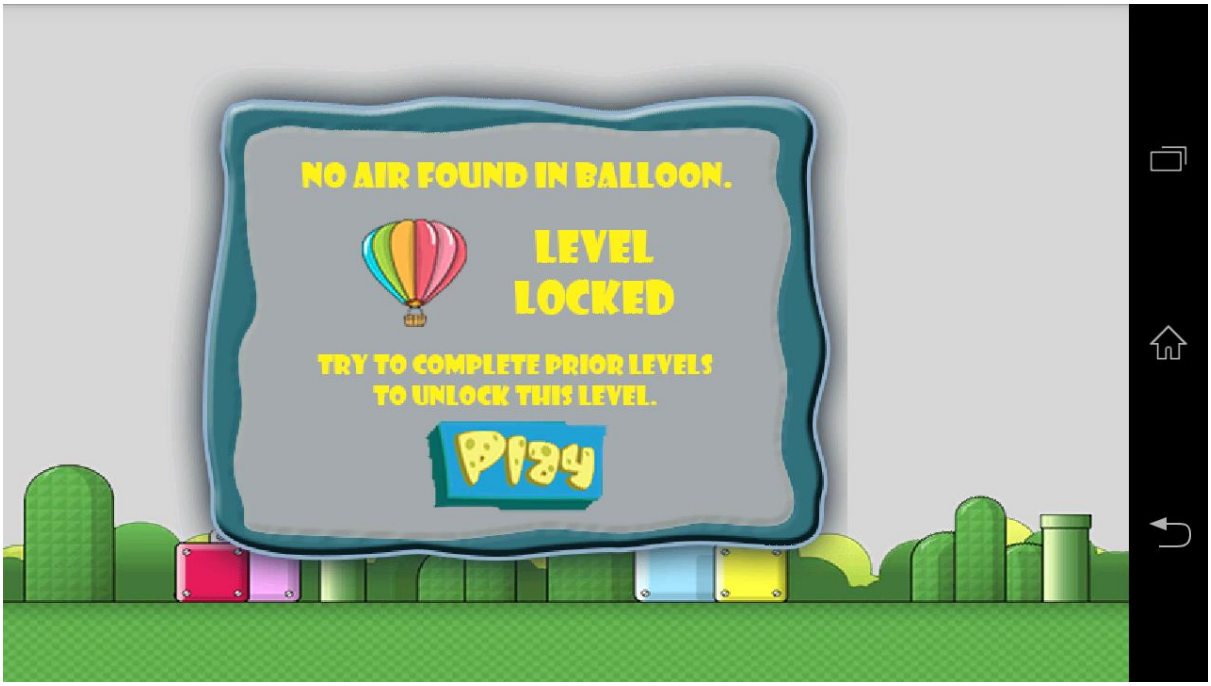

Figure 7 Level Locked Message

3.3 Functional Requirements

3.3.1 Main Activity Class

Description and Priority

This is the main class which can be termed as the main base class for the game. This class will provide the main framework for the game on which all the function will be invoked and also create main objects to handle AndEngine like- engine, camera and physicsworld objects.

Functional Requirements

REQ-1: User must open the application.

REQ-2: The RAM must be available at the time of invocation.

REQ-3: According to the class of the user, appropriate UI should be displayed to the user.

REQ-4: Splash resource should be loaded and the next activity will be displayed.

3.3.2 Scene Manager Class

Description and priority

This class will provide the basic management functionality to manage scenes and resources related to it. This class will also invoke game manager class.

Functional requirements

REQ-1: The main activity must have generated resources for this class.

REQ-2: According to the scene chosen by the user, appropriate UI should be displayed to the user.

REQ-3: User can make transition smoothly via buttons shown in UI.

3.3.3 Game Manager Class

Description and priority

This class will provide the main gaming environment. User can play various levels according to his/her gameplay.

Functional requirements

REQ-1: The scene manager activity must have generated resources for this class.

REQ-2: According to the level chosen by the user, appropriate UI should be displayed to the user.

REQ-3: User can play the game and see the score according to game-play.

3.4 Non-Functional Requirements

3.4.1 Performance Requirements

The objective of this project is to develop a game application for the android devices which can be played by the user in the arcade gaming section with some logics and physics basics. The game is running flawlessly into gameplay and it provides richer gaming experience.

3.4.2 Safety Requirements

During use of the game, players experiencing eye strain should take a break from playing to avoid further strain and/or possible damage. Some players may be prone to seizures because of flashing or bright animations. It is therefore not recommended people who have a history of epileptic seizures or photosensitive seizures play “BalloonSaga”.

3.4.3 Security Requirements

The game, running as an application on the Android device should need no additional information.

3.4.4 Software Quality Attributes

The application should be able to work on any graphics-compatible Android device with 256MB of RAM. Installing the application should be a simple process, ideally identical to downloading and installing any application from the Android App Market (PlayStore™). The application should run smoothly without crashing or freezing, regardless of any game parameters, i.e. stage, number of time played, player time duration, etc. It should have a very intuitive interface that is easy to learn so the player can focus mostly on the game itself.

3.4.5 Business Rules

Any individual may have use of the system for academic or personal purposes. As the project is part of minor project, code, documents, or other materials used for this project cannot be used for

commercial purposes. However, others wishing to further develop the code after the project's completion are required to take permission from the authors of the project.

Chapter 4

Coder's Perspective

4.1 Brief Note

Game is developed using AndEngine and physicsbox2d library. AndEngine library forms the core part of the game providing basic services like game camera settings, update intervals, game resolution etc. physicsbox2d library helps to simulate objects and their behavior in real world like collision effect, elastic property of objects, effect of gravity on objects, effect of force and velocity etc.

4.2 Game Class Implementation

4.2.1 MainActivity

MainActivity initializes and engine to various initial values which are required for setting up basic game environment. Some of key functionalities of it are:

- Initialize game camera.
- Setting screen orientation like LANDSCAPE MODE.
- Setting up audio options.
- Setting up game scene update frequency.
- Loading and generating splash screen.
- Back key press management

The MainActivity rely on following core functions:

- **onCreateEngineOptions() :** General initialization like camera object, view mode of game, audio options etc.
- **onCreateEngine() :** Setting up game FPS engine i.e. update frequency.
- **onCreateResources() :** creating scene resources.
- **onBackPressed():** management of back key pressing event in various game scenes

Figure 8 RocketAppsStudio Splash Screen

Figure 9 BalloonSaga Splash Screen

4.2.2 SceneManager

After the splash screen SceneManager class takes over the game handling of game. It is responsible for handling of Menu, quit, help and credits scene. Thus scene manager controls primary scenes of the game.

Key functions in SceneManager:

- **setCurrentScene()**: sets current scene of the game.
- Loading of level selection resources.
- **loadHelpResources()**: for help scene generation.
- **loadQuitResources()**: for quit scene generation.
- **loadCreditResources()**: for credit scene generation.
- **createMenuScene():** creating menu scene.

Figure 10 Menu Scene

Figure 11 Help Scene (Descriptive)

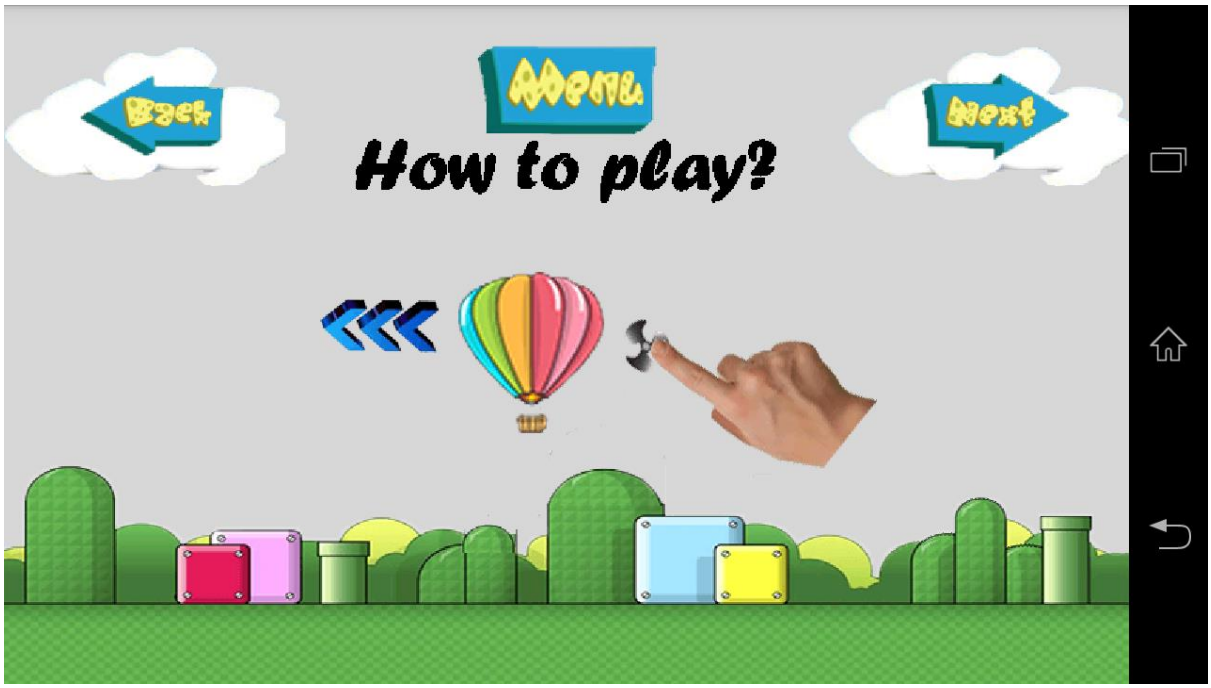

Figure 12 Help Scene (Illustrative)

Figure 13 Credit Scene

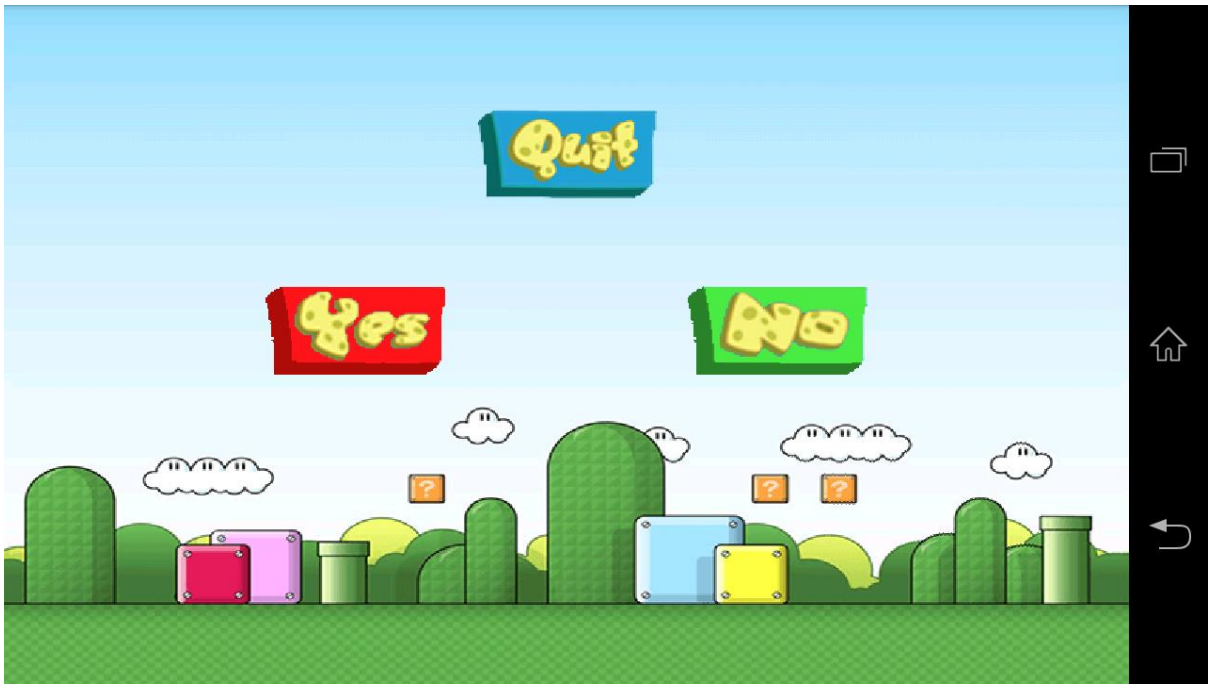

Figure 14 Quit Scene

4.2.3 GameManager

GameManager class implements various game levels as selected by player. It handles update of locks on various levels, changing of music in game. GameManager class also relies on database helper classes for level-lock implementation.

Key functions in GameManager:

- **loadMusicResources()**: loads music resources as required in game.
- **stopAllMusic()**: handling music played in game at each level.
- **loadLevelOneSelectionResources()**: for choice of levels 1-5.
- **loadLevelTwoSelectionResources()**:for choice of levels 6-10.
- **Loading of various level resources and creation of respective scenes**.
- **setCurrentScene():** setting current scene in game engine.
- **createGameOverResources() and createGameOverScene() :** for game over event and implementing based on win or lose condition.

GameManager class uses physicsbox2d library to simulate gravity effects on balloon, wall fixture properties etc.

Figure 15 Instruction Scene

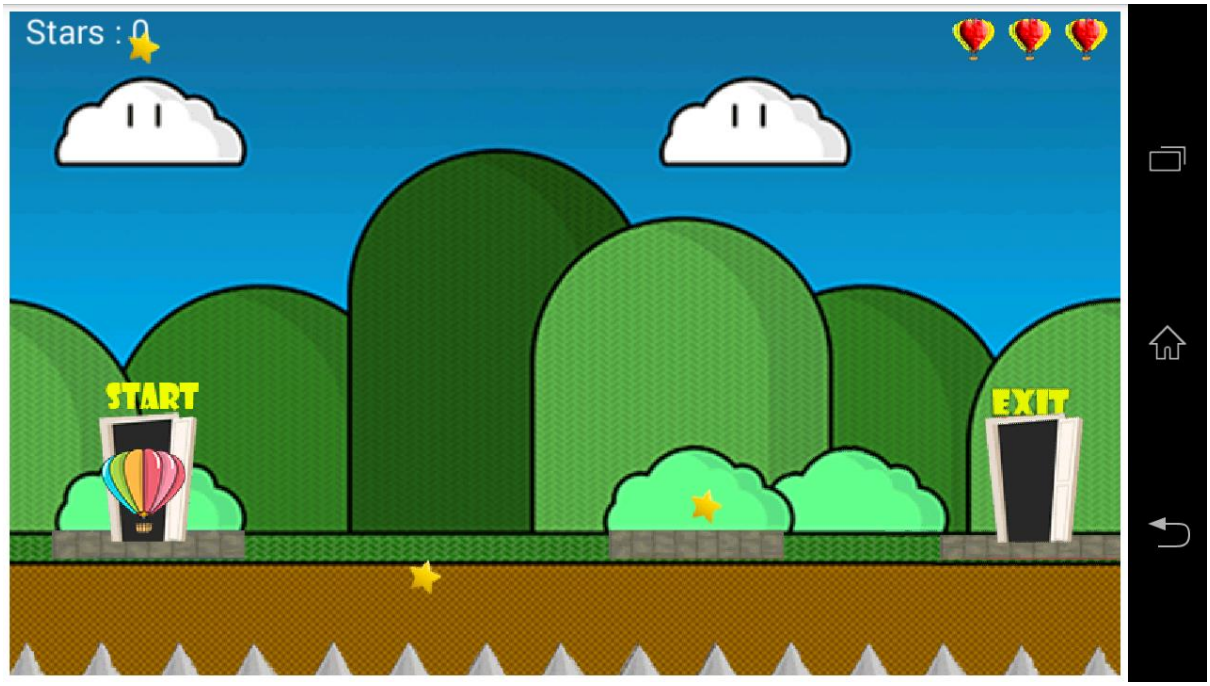

Figure 16 Starting Level

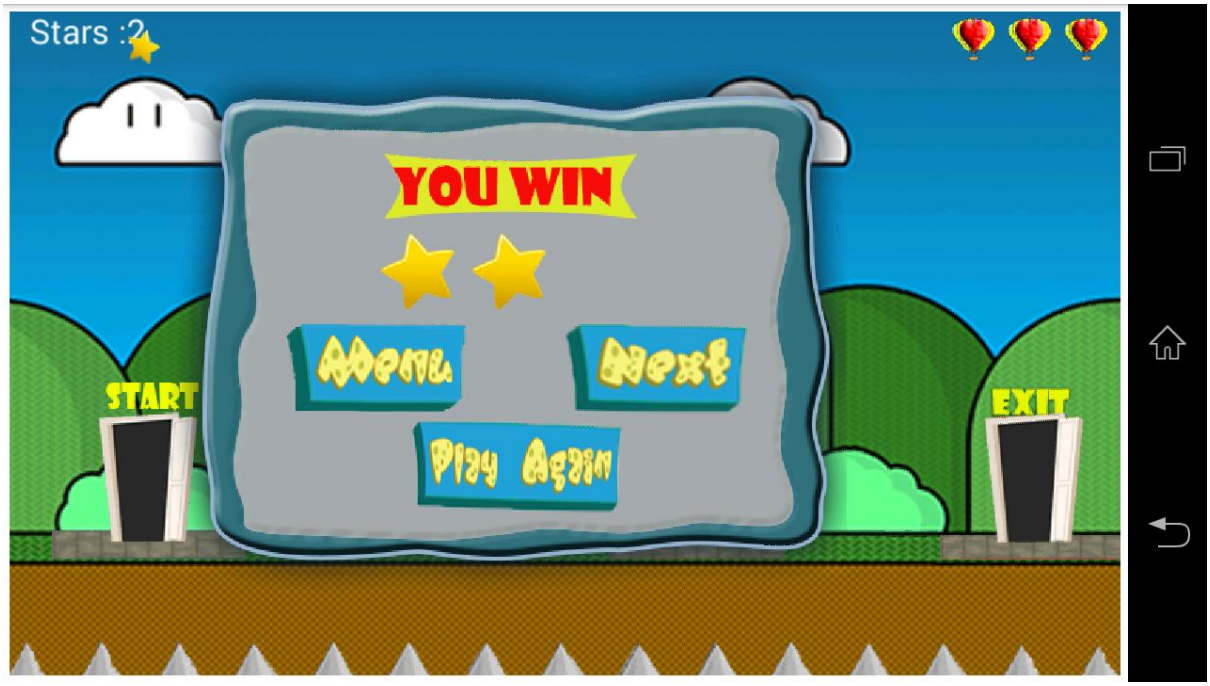

Figure 17 Level Completed Game Scene

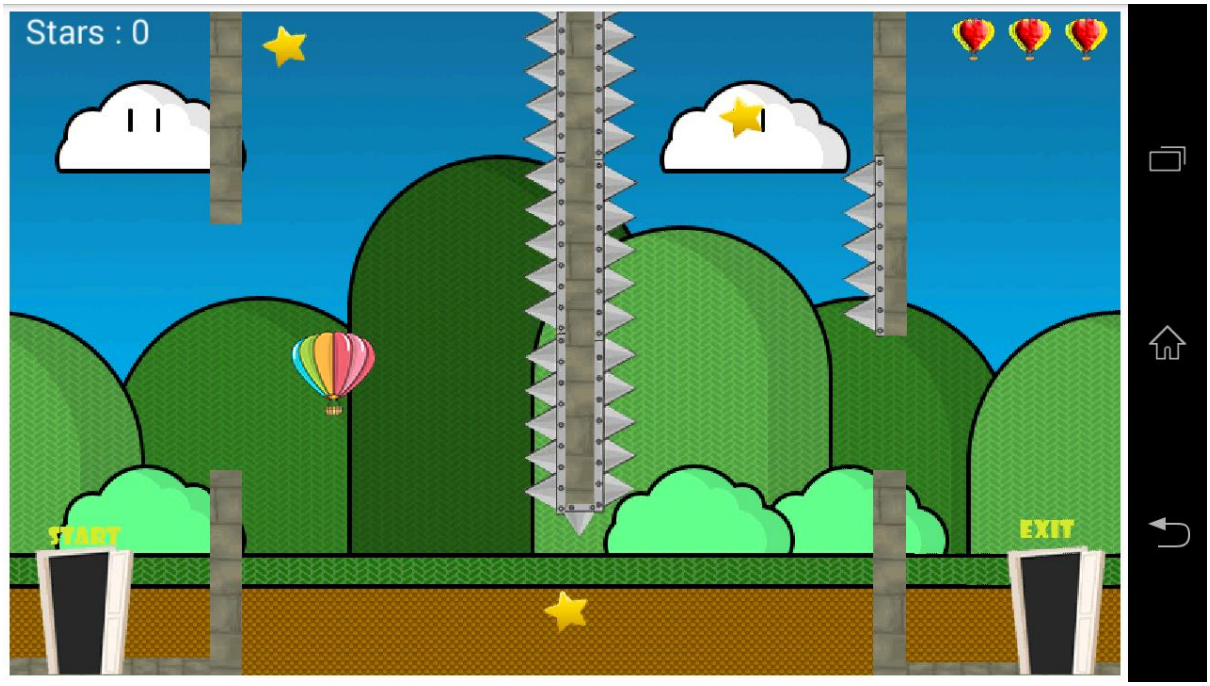

Figure 18 Level 4 Scene

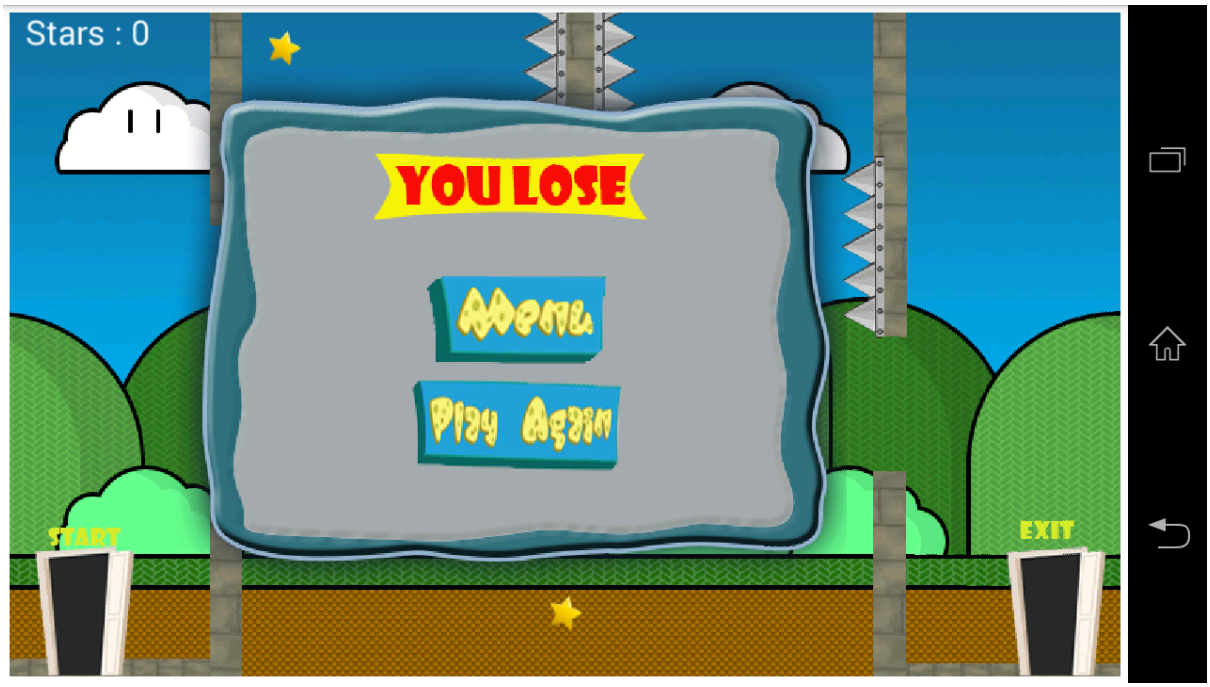

Figure 19 Level Lose Game Scene

4.2.4 DataBaseHelper

Besides these three classes that govern almost whole functioning of the game, the sometimes rely on a database helper class.

DataBaseHelper class maintains track of locked and unlocked levels. Game manager class updated levels with the help of database class based on the win or lose condition. LevelHelper.java is the class that provides various functions to add, modify level status.

Key functions in LevelHelper:

- **addLevel()**: adds levels to the game database with default locked status.
- **getLevelStatus()**: returns status of level i.e. locked/unlocked
- **updateLevelStatus()**: unlocks all levels below it , including itself

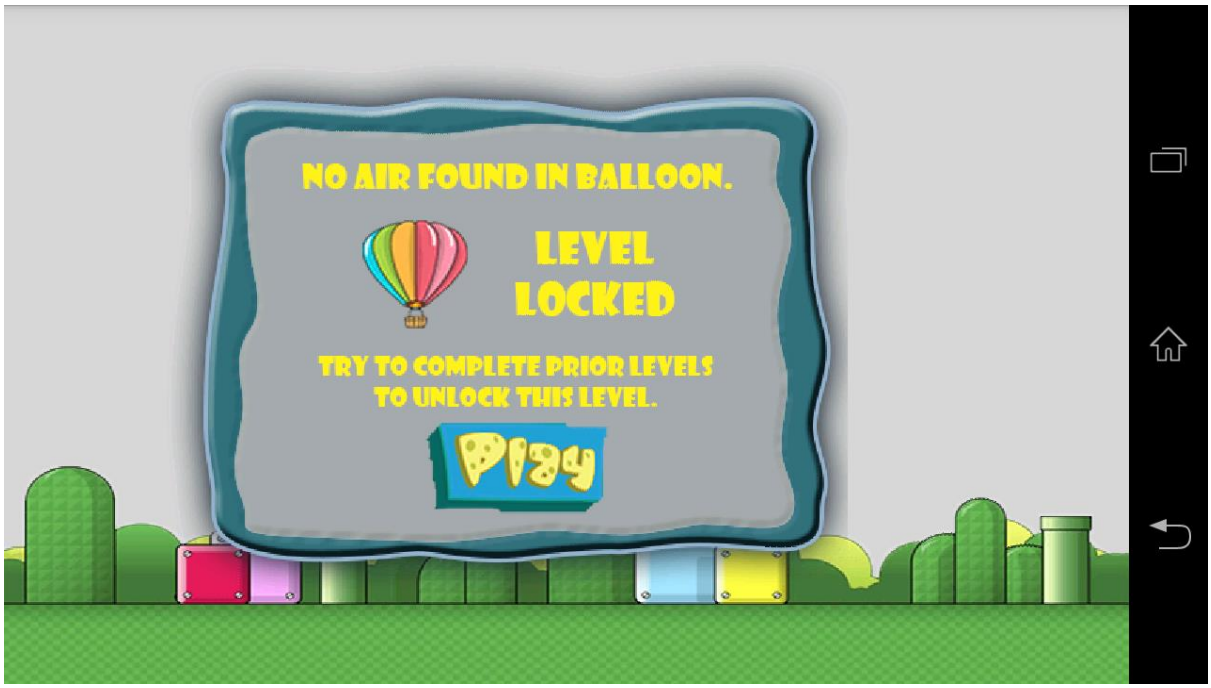

Figure 20 Level Locked Scene

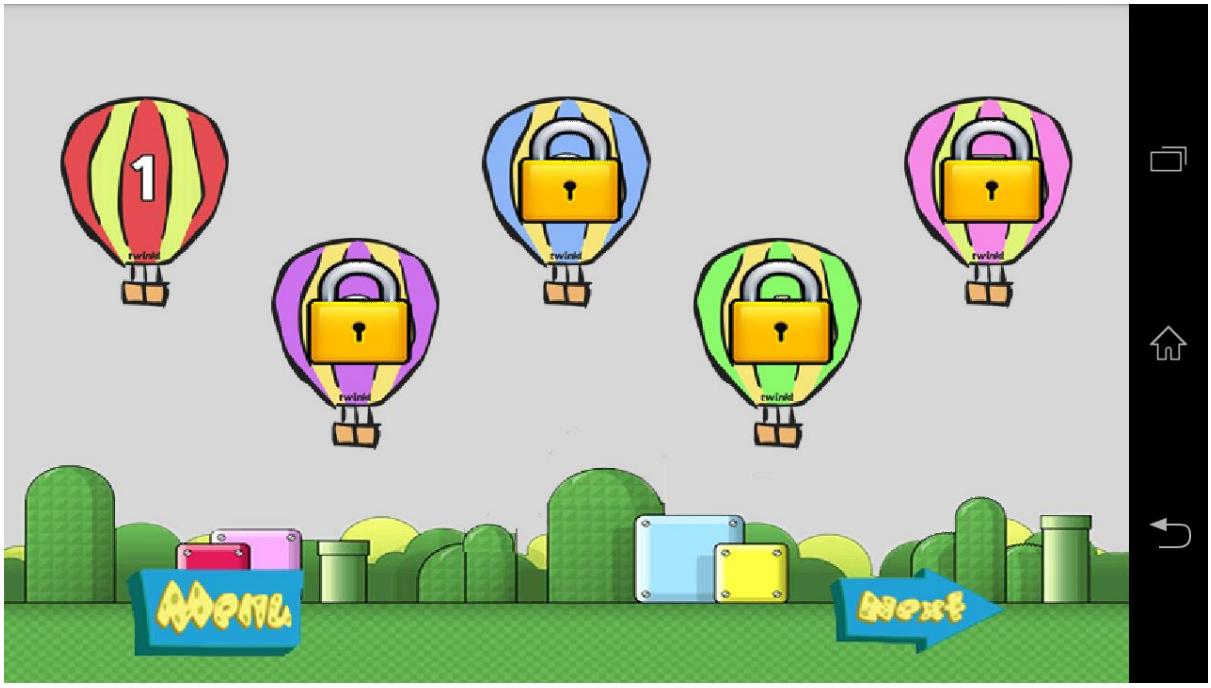

Figure 21 Level Select Scene (locked)

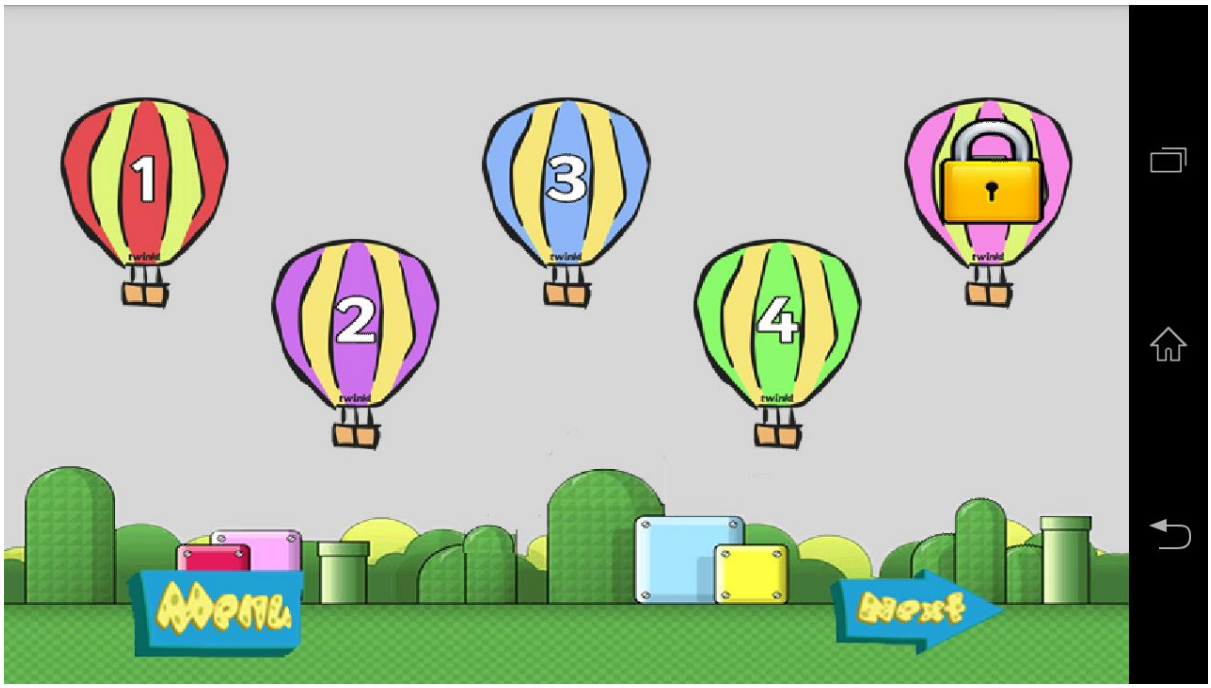

Figure 22 Level Select Scene (unlocked)

Chapter 5

Player's Perspective

5.1 Game Playing

The game starts with a splash screen and then displaying Menu options.

Figure 23 Splash Screen of Game

Menu Screen provides various buttons for exploring various game parts.

1) PLAY : button to play game.
2) SOUND ON OFF : to turn sound on/off
3) QUIT: to quit game
4) HELP: to view help menu.
5) CREDITS: to view credits.

Figure 24 Menu Scene with Buttons

On selecting PLAY button a new screen appears, that provides choice for various game level selections. Once a player completes a game level a win/lose window appears with various buttons for moving to next level, replaying level or going to menu scene etc. The back button press event is also handled which rolls back each scene that player has moved to get current level.

Chapter 6

Testing, Debugging and Release

6.1 Testing and Debugging

Following were the issues that were discovered during testing phase of the game prior to release of it into PlayStore.

- Sound toggle not working: In and engine platform it is required to explicitly take care of sound status. During initial phase we discovered malfunctioning of sound status of game when the scene was regenerated.
- Overlaying of images: Neglecting regeneration of resources may lead to garbled scene generation resulting in improper behavior of game.
- Game Music change problem: In AndEngine the music objects do not behave in the manner other objects do across the classes. A different approach was required to pause, resume or change game music at various game levels.
- Back button events: Explicit coding is required for back button is required for the back button pressed event with the knowledge of present scene in game.
- Game crashing: The possible factors included non-matching size between atlas map and image, null pointer exception i.e. unavailability of resource.

6.2 Release on Google PlayStore

After successful series of testing and debugging we uploaded a “BalloonSaga” game to Google PlayStore.

The game is available for free under Arcade category in the PlayStore. Current download count of the game is 200+ with 4.6/5 rating. Game is currently supported in around 8000 android devices.

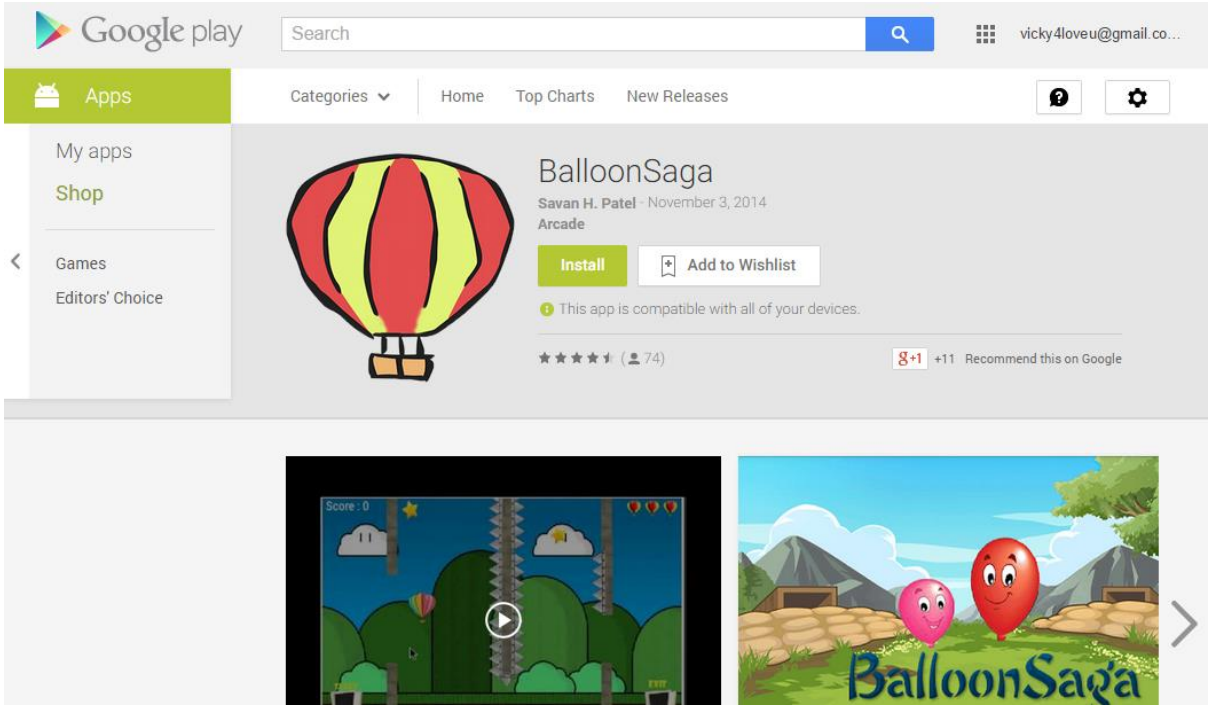

Figure 25 Google PlayStore (Web)

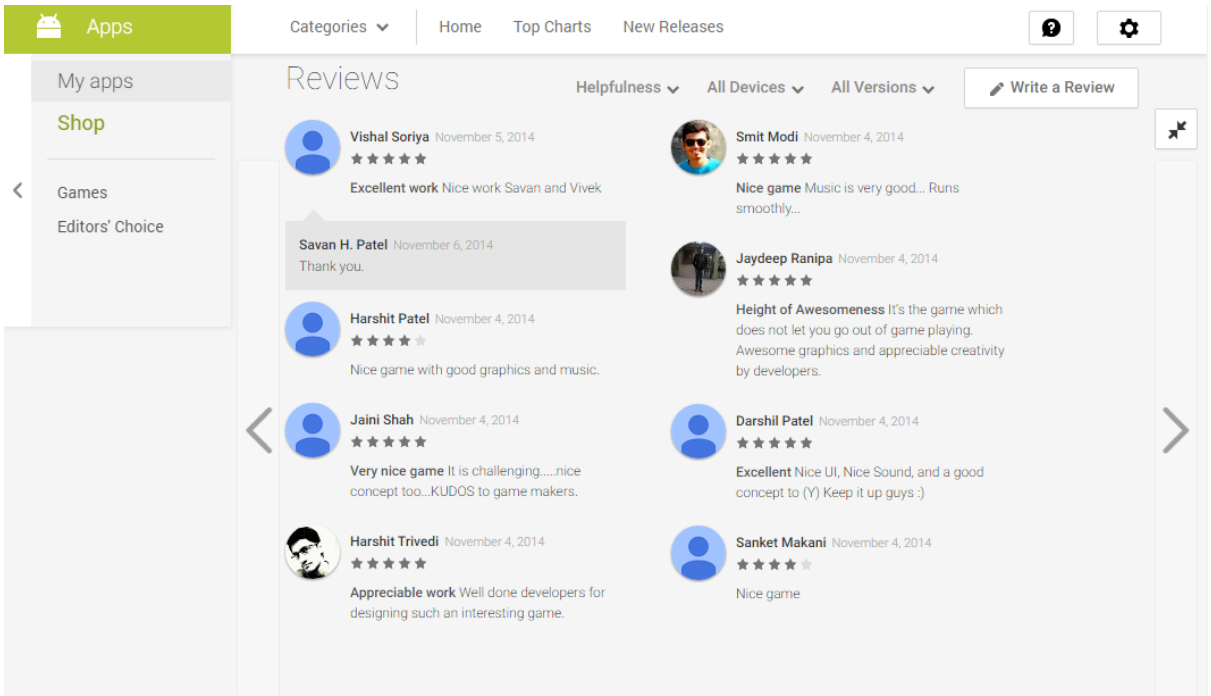

Figure 26 Google PlayStore Reviews (Web)

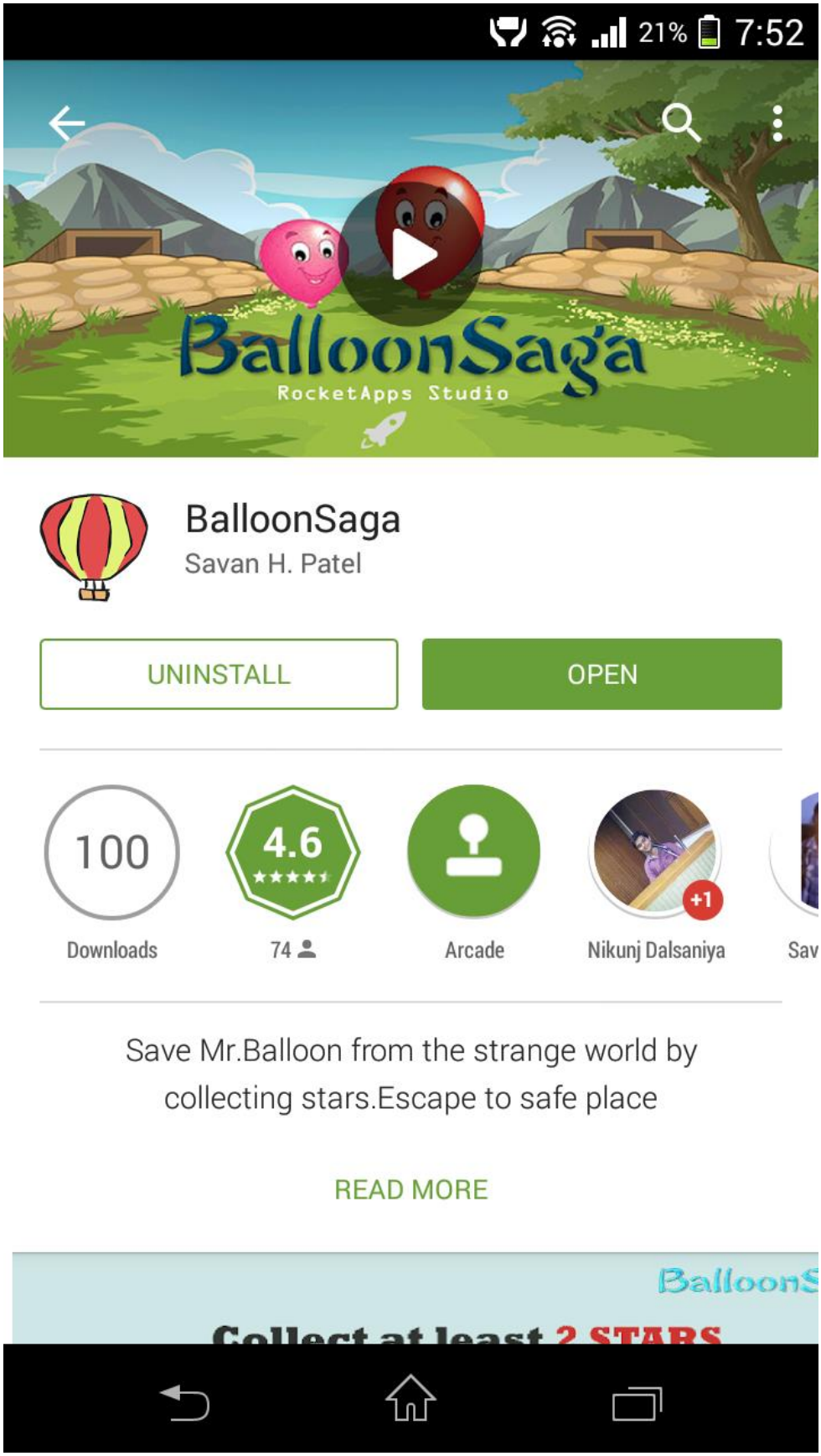

Figure 27 Google PlayStore (Mobile)

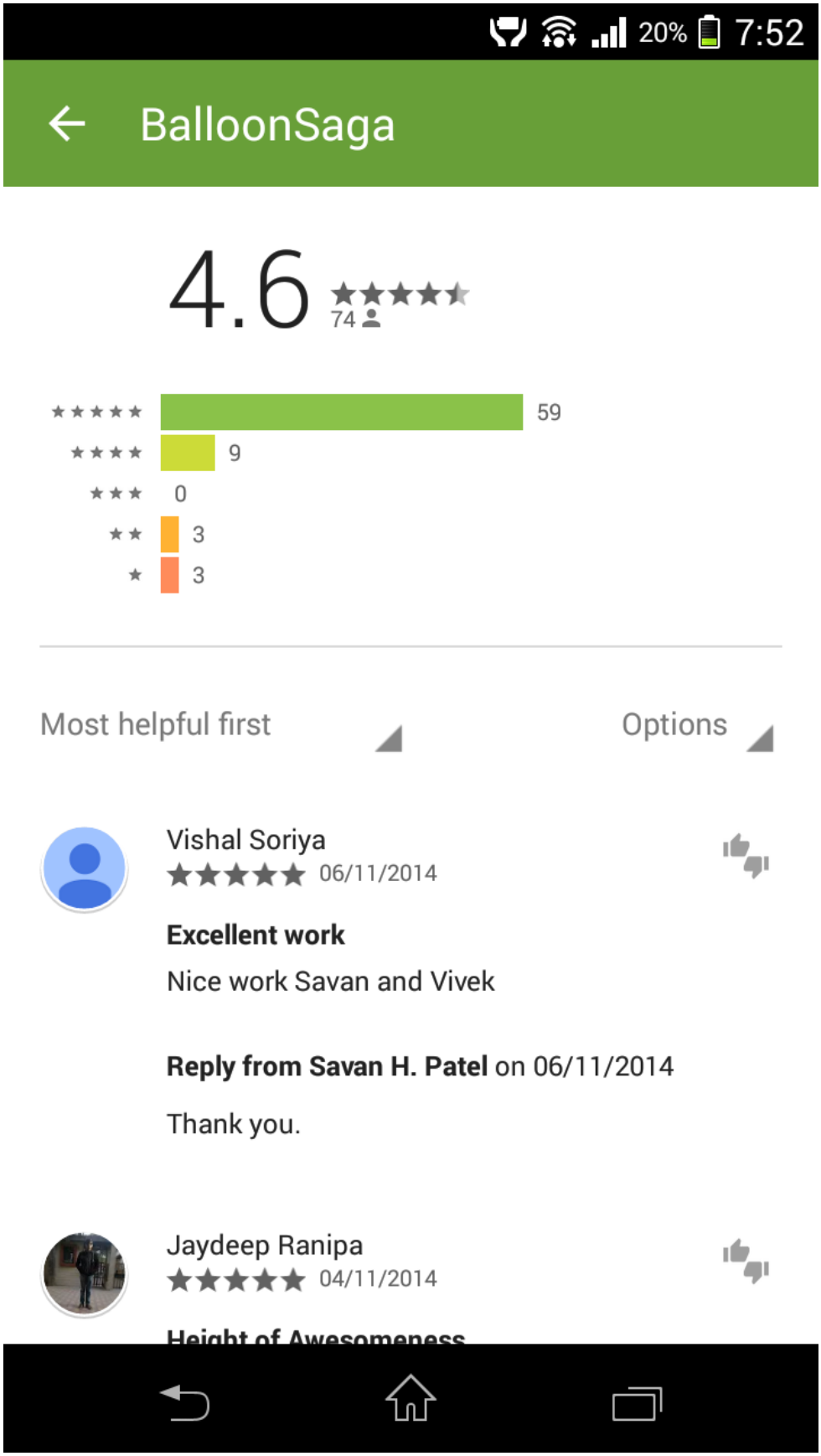

Figure 28 Google PlayStore Reviews (Mobile)

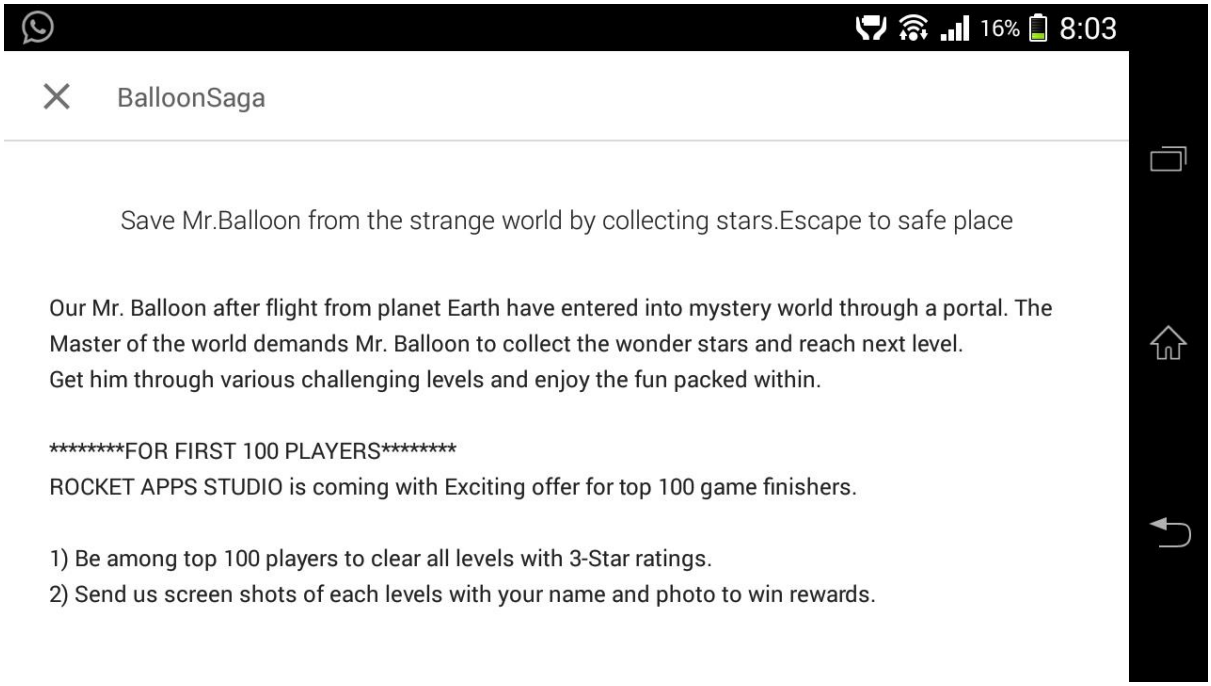

Figure 29 Google PlayStore Description (Mobile)

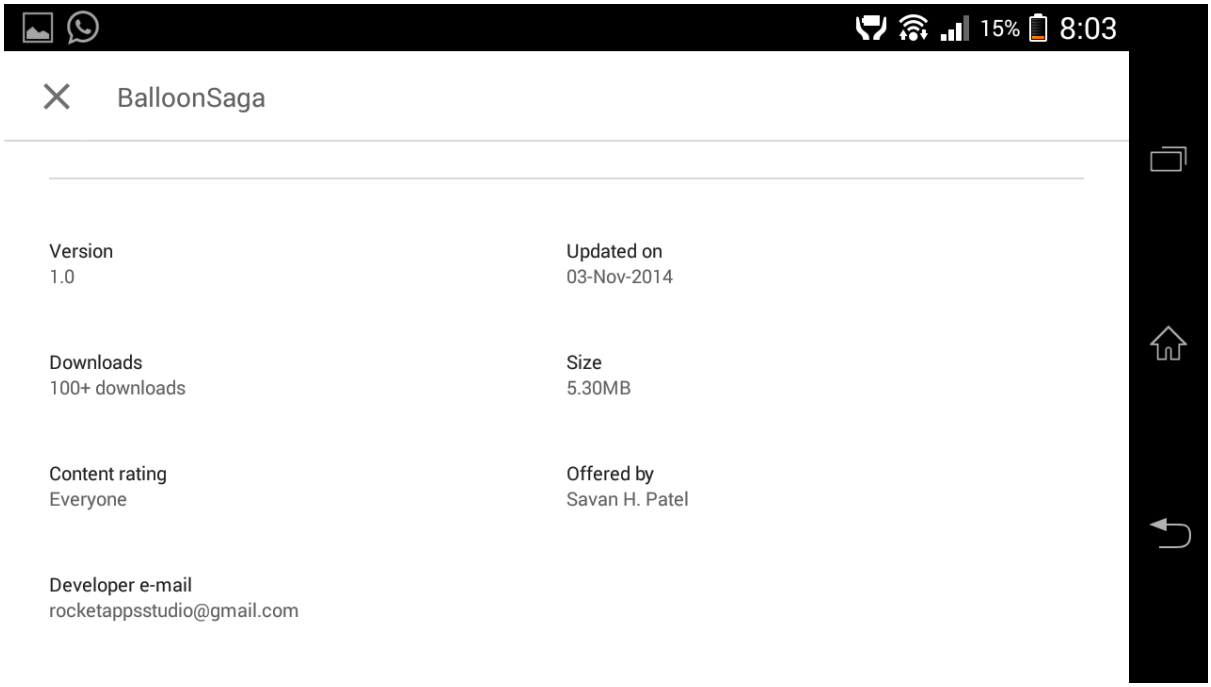

Figure 30 Google PlayStore Information (Mobile)

6.3 Future Work

We wish to expand the game with more fun packed levels with more furnished graphics in upcoming days under the group name “RocketAppsStudio” and release it on PlayStore. We also will try to incorporate google ad module in the game for monetization.

Goals for the future:

- New levels implementation.
- Adding more features – like game pause, game background messages etc.
- Incorporating with Google Games Services.
- Adding achievements and rewards for players.
- Next edition with all new theme.

References

[1] Nicolas Gramlich, “Founder of AndEngine,” *www.andengine.org*, 2010.

[2] Nicolas Gramlich, “nicolasgramlich/AndEngine,” *AndEngine Library Code at GitHub*, *github.com/nicolasgramlich/AndEngine*, 2013.

[3] Jayme Schroeder and Brian Broyles, “AndEngine for Android Game Development Cookbook,” *Packt Publishing, 2010*.