Riaan Van Onselen - 97124275

31285 Mobile Applications Project

Planet Tanks

Table of Contents

[1 Introduction 3](#_Toc484445126)

[1.1 Application Vision 3](#_Toc484445127)

[1.2 Scope 3](#_Toc484445128)

[1.3 Document Version History 3](#_Toc484445129)

[2 Functionality Overview 4](#_Toc484445130)

[2.1 Feature Summary 4](#_Toc484445131)

[A playable game: 4](#_Toc484445132)

[An interactive virtual control layout: 4](#_Toc484445133)

[A level selector screen: 4](#_Toc484445134)

[A level editor menu: 4](#_Toc484445135)

[A level editor: 4](#_Toc484445136)

[Level saving: 4](#_Toc484445137)

[2.3 Application Flow Diagram 5](#_Toc484445138)

[3 Design 6](#_Toc484445139)

[3.1 Overall Components Overview 6](#_Toc484445140)

[3.2 Mockups: 6](#_Toc484445141)

[3.3 Used Application Programming Interfaces 8](#_Toc484445142)

[LibGDX game engine: 8](#_Toc484445143)

[AdMob: 8](#_Toc484445144)

[4 Application Structure 9](#_Toc484445145)

[4.1 Packages 9](#_Toc484445146)

[4.2 Activities 9](#_Toc484445147)

[6 Testing 9](#_Toc484445148)

[7 Conclusion 9](#_Toc484445149)

[8 References 9](#_Toc484445150)

# 1 Introduction

Planet Tanks is a game where you control a tank and must seek out and destroy other tanks to complete beat the levels. The game will have a set of base levels however, there is also a level editor that players can use to create their own unique challenges. Planet Tanks is designed to provide a varying degree of gameplay difficulty allowing the player to have countless fun experiences as they get better.

## 1.1 Application Vision

This game idea is based off pre-existing games, namely Wii Play Tanks running on the Wii and Tank Hero which runs on Android. However, Planet Tanks will have an additional two base features, a level editor and traps. A level editor can be used to create and change game levels, allowing the players to customise their game experience. Many different games have this feature and it has been shown to increase the players own fun. Additionally, trap tiles that perform various effects on gameplay will be added. These traps will alter the games mechanics causing the player to change their strategy as they progress through the levels. An example of a trap tile would be a mud tile. These tiles will slow the player down making it harder for them to doge enemy projectiles.

## 1.2 Scope

Initial Scope:

* A top down 3D game where a player controls a tank
* 5 levels base that the player can play through and learn the mechanics
* 2 Enemy tanks that shoot and behave differently
* 3 trap tiles that have different effects on game mechanics
* A level selection screen
* A level editor that the player can use to edit pre-existing levels or create their own.
* Connection to Google’s Play Games Services for leader board scoring
* Connection to Google’s Play Games Services for in game achievements

Optional Items:

* Additional levels
* Additional enemy tanks
* Additional trap tiles
* Online map sharing using either google firebase
* Online multiplayer where players play against each other.
* Advertisement support

## 1.3 Document Version History

|  |  |
| --- | --- |
| Version Number | Changes |
| Version 0.1 | Contained a brief overview of the game idea, potential APIs for use and some simple wireframes. |
| Version 0.5 | Added introduction, application vision, scope, feature summary, application flow diagram, application programming interfaces and activities |
| Version 0.6 | Added packages and updated some formatting |
| Version 1 | Attempted to finish up most of the required parts |

# 2 Functionality Overview

## 2.1 Feature Summary

### A playable game:

The core feature of this app is a 3D game. In this game, the user controls a tank and must navigate between walls and spikes to shoot at the enemy tanks. These tanks are stationary; However, they will attempt to shoot the player once they come in range. The aim of the game is to kill all enemy tanks in the level. This allows the player to progress onto the next level.

### An interactive virtual control layout:

The game presents two touchpads to the user that they can use to control the tank. There are also two buttons on the game play screen, one to fire and one to pause the game. These UI controls are slightly transparent allowing the user to see the level behind the controls.

### A level selector screen:

This screen allows the player to choose the level they wish to play. It also allows for levels to be locked forcing the player to play the earlier levels before they can attempt the later levels.

### A level editor menu:

This screen allows the player to see what levels they have created and choose to either play or delete them. This screen also gives them the option of creating a level, which then activates the level editor.

### A level editor:

This screen allows the user to create their own levels. The user is shown a grid that they can tap to change the tile or game object that is stored in that position. This allows the user to visually see what their level is like before they save it. Additionally, there is a play test feature which allows the user to play their level before they decide on saving it.

### Level saving:

This game allows the user to save the levels they have worked. The app automatically reloads the users created levels on start-up allowing for quick playing time.

## 2.3 Application Flow Diagram



# 3 Design

## 3.1 Overall Components Overview

The base project will need only once activity to link to the game engine. This will later be converted into a fragment to allow interaction with additional API.

## 3.2 Mockups:

|  |  |  |
| --- | --- | --- |
| Screen Index: | Screen: | Links and functionality: |
| 0 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-52- | Splash screen:  Used to load in all the assets. It then loads screen 1. |
| 1 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-52- | Main menu screen.  Play button -> 2  Level Editor button -> 7  Quit button closes the application |
| 2 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-50- | Level select screen:  Level buttons -> 3  Back button -> 1 |

|  |  |  |
| --- | --- | --- |
| 3 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-50- | Game play screen:  Pause button -> 4 |
| 4 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-50- | Pause screen:  Resume button -> 3  Quit button -> 1 |
| 5 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-50- | Game over screen:  Retry button -> 3  Quit button -> 1 |
| 6 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-50- | Level complete screen:  Next level button -> 3  Quit button -> 1 |

|  |  |  |
| --- | --- | --- |
| 7 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-51- | Editor menu screen:  Create level button -> 8  Play button -> 3  Back button -> 1 |
| 8 | ../../../../../../../Desktop/Screenshot_2017-06-05-13-51- | Editor screen:  Back button -> 7  Play button -> 3 |

## 3.3 Used Application Programming Interfaces

### LibGDX game engine:

This is the core API used in this game. It provides the game engine used for drawing and interacting with the 3d world. Almost all of the code for this application is contained within the game engine.

### AdMob:

This API allows you to include adds from googles admob service. This allows your app to request an app from googles add service and display it on the screen.

# 4 Application Structure

## 4.1 Packages

Packages:

com.riaanvo.planettanks

com.riaanvo.planettanks.Managers

com.riaanvo.planettanks.Objects

com.riaanvo.planettanks.Physics

com.riaanvo.planettanks.States

com.riaanvo.planettanks.LevelFramework

Assets:

assets.Levels

assets.Models.Tank

assets.Models.Wall

assets.Textures

assets.UI

## 4.2 Activities

* Game screens:
  + Main menu screen:
  + Level selection screen:
  + Game play screen:
  + Game over screen:
  + level complete screen:
  + Level editor menu screen:
  + Level editor screen:

# 6 Testing

The development of this game had two main testing stages. The first was creation and testing of the game using a desktop version followed by testing for android using both an emulator and a Samsung Galaxy S3. The game was built and given to people to test if the application was usable and the game playable. Adjustment to core game variables where then made to ensure users fun.

# 7 Conclusion

In conclusion, I was able to implement a simple 3d game with a level editor as described in the initial scope. However, there were only spikes and a simple enemy implemented as game objects and two levels that the player can attempt in the base game.

# 8 References

References of snippets of code are located in method java doc comments.

Game UI art from <https://github.com/czyzby/gdx-skins/tree/master/flat-earth>