Test report - Gradius

Table of Contents

1 Introduction

1.1 Purpose of application

The purpose of this project is to learn how to work and programme as a group in a large software project and how to requirement test, additionally we also want to learn more about GIT and to learn the basics in android programming. The project is developed within the course DAT255 “Software Engineering Project”.

1.2 General characteristics of application

The application will be a space game. In the end product the software will be a up-and-side-scrolling game or a game where the spaceship moves in a straight line with monsters and other objects moving towards the spaceship “attacking” it. The spaceship will collect points by flying into coins floating around in space and in that way increasing the life or force of the spaceship, in its resistance of “hits” from objects or monsters. The project is scalable meaning that it could be expanded with more functions later in the developing process but could be tested and played also in the early stages. The timeplan below is an estimation and will probably be revised and modified throughout the elapse of the project. The project was developed through an Agile work process, and will be held within the limitations that this process puts upon the project.

2 Test Environment

2.1 Hardware environment

To be able to run the application the system will only require two hardware components:

2.1.1 Android Device

An Android Mobile Device is required to serve as the user’s interface device for the interstitial viewer. Any Android-based mobile device is sufficient to run the interstitial viewer; given it has camera and accelerometer sensors.
For our development we will be using a Motorola XT860 smartphone running Android 2.3, which has the following:

- Dual-core 1 GHz processor
- 512 MB RAM
- 16 GB internal memory (expandable)

2.1.2 Computer

A computer is required to run Gradiuss. Gradiuss is a demanding software requirements and minimum specifications are the following

- 1.5+ GHz processor
- 1+GB RAM
- Hardware accerated OpenGL drivers

- 128 (256+ recommended)MB graphics card.
- Solaris/Linux: Nvidia graphics cards/drivers
- At least DSL-speed internet connection.

For code development we will be using a combination of machines with different set-ups (all meeting minimum requirements)

2.2 Software environment

Here is a short description of what software are needed/recommended to start developing in the project.

2.2.1 Eclipse

Eclipse is a multi-language software development environment comprising an integrated development environment and an extensible plug-in system. This application is used for developing in the Java.

2.2.2 Android SDK

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit. After installing Eclipse the developer will need the Android SDK extension and related tools to be able to program in the Android language.

2.2.3 Git

GitHub is a web-based hosting service for software development project that use the git revision control system.

2.3 Software settings

What the settings and versions should be on the software needed.

2.3.1 Android SDK

The Android version which is used to develop the project is Android 2.1 (API level 7). The minimum SDK version is also API level 7.

3 System Information

The information presented in this report is for the version v.1.0.

4 Known bugs and limitations

- The vibration is always on. It can't be turned off.
  - Req 2.16 -> TC 2.16.2
  - Req 6.2 -> TC 6.1.1

- The game is not paused when there is an incoming call.
  - Req 2.18 -> TC 2.18.1

- The user is unable to pause the game.
  - Req 2.17 -> TC 2.17.1

5 Requirements

5.1 Main screen (StartGameActivity)
Req 1.1

5.2 Playscreen (GameViewActivity, GameView and Game models...)
Req 2.1
Req 2.2
Req 2.3
Req 2.4
Req 2.5
Req 2.6
Req 2.7
Req 2.8
Req 2.9
Req 2.10
Req 2.11
Req 2.12

5.3 Game Over Screen (GameOverActivity)
Req 3.1

5.4 Highscore Screen (HighScoreActivity)
Req 4.1
Req 4.2

5.5 Options Screen (OptionsActivity)

5.7 Help - Instructions

5.1 Main screen (StartGameActivity)

Req 1.1

The screen shall display the Main Menu, where six options can be made. The options that could be made are to start the game by pushing the button Start Game, to make options, such as adjusting the volume, by pushing Options, to see what person has got the highest score by pushing Highscore, to find out more about how the game was developed by pushing About and to get instructions on how to play the game by pushing Help. By pressing Quit, the game should exit.

TC 1.1.1:

Description:
The user clicks the Start Game button the Play Screen opens up and the game starts.

Precondition:
The application icon has been pressed and the main screen is now in the front.

Post-conditions:
The game is started with a countdown that prepares the user to start playing the game.

Normal Sequence:
The user starts the application
The user clicks on the “Start Game” button.
The countdown screen opens and counts 3, 2, 1, GO!
The PlayScreen starts and the game is on.

TC 1.1.2:

Description:
The user clicks the Highscore button and the Highscore Screen opens up.

Precondition:
The application icon has been pressed and the main screen is now in the front.

Post-conditions:
In the Hichscore Screen a list of the best results are displayed

Normal Sequence:
The user starts the application
The user clicks on the “Highscore” button.
The Highscore Screen shows and a list of the best results are shown

TC 1.1.3:

Description:
The user clicks the Options button and the Options Screen opens up.

Precondition:
The application icon has been pressed and the main screen is now in the front.

Post-conditions:
In the Options Screen some options can be made. The user can change the level of the volume and change spaceship

Normal Sequence:
The user starts the application
The user clicks on the “Options” button.
The Options Screen shows and a some options can be made

TC 1.1.4:

Description:
The user clicks the About button and the About Screen opens up.

Precondition:
The application icon has been pressed and the main screen is now in the front.

Post-conditions:
In the About Screen information about how the game was developed are displayed

Normal Sequence:
The user starts the application
The user clicks on the “About” button.
The About Screen shows and information about how the game was developed are displayed

TC 1.1.5:

Description:
The user clicks the Help button and the the Help Screen opens up.

Precondition:
The application icon has been pressed and the main screen is now in the front.

Post-conditions:
In the Help Screen instructions about how to play the game are displayed

Normal Sequence:
The user starts the application
The user clicks on the “Help” button.
The Help Screen shows and instructions about how to play the game are displayed

TC 1.1.6:

Description:

The user clicks the Quit button and the application shuts down.

Precondition:
The user has been playing and now returns to the main menu.

Post-conditions:
The application is not in the front anymore.

Normal Sequence:
The user plays the game.
The game end and the user returns to the main menu.
The application is turned off.

5.2 Playscreen (GameViewActivity, GameView and Game models...)

Req 2.1

The screen shall display a spaceship (the player) that can move back and forth in the x and y - directions by pressing the appropriate control buttons in the North, East, South, West directions, and a "firebutton" together with a "change weapon button".

TC 2.1.1:

Description:
The user will see a spaceship and see buttons in the North, East, South, West directions, a firebutton and change weapons button and by pressing the buttons move the spaceship in the direction pressed

Precondition:
The player have started the game by pressing "start game" in the start menu.
Lägg in relaterade test case.

Post-conditions:
The game is in progress.

Normal Sequence:
The user press start game.
The user sees the spaceship and controls in North, East, South, West directions.
The user moves the spaceship by pressing the buttons.

Req 2.2

The spaceship should be bound by the screen boundaries and not be able to move outside the screen.

TC 2.2.1:

Description:
The space ship moves against an edge of the screen and stops.

Precondition:
The game is in progress, in PlayView..

Post-conditions:
The spaceship is blocked from moving further in the blocked direction.

Normal Sequence:
The user is playing the game.
The user (space ship) collides with a screen edge.
The user is blocked from moving further.
The user can move in the other direction.

Req 2.3

The spaceship should be able to shoot projectiles

TC 2.3.1:

Description:
The user shoots several projectiles and by holding in the button the spaceship is able to shoot several projectiles in a row.

Precondition:
The game is in progress.

Post-conditions:
When letting the button go the projectiles ceases to come from the spaceship and the last projectiles move out from the screen if no obstacle is in the way.

Normal Sequence:
The user is playing the game.
The user (space ship) pushes the fire button.
New projectiles emerge from the spaceship moving in the upward direction.
The user lets the button go and the projectiles cease to be shot.

TC 2.3.2:

Description:
The user shots one shot by simply pushing and then immediately letting the button go, one shot emerges and moves in the upward direction

Precondition:
The game is in progress.

Post-conditions:

The projectile moves through the screen if no obstacle is in the way.

Normal Sequence:
The user is playing the game.
The user (space ship) pushes the fire button on time.
A new projectile emerge from the spaceship moving in the upward direction.

Req 2.4

There should be obstacles (e.g asteroids and alien ships) on the screen that the spaceship can shoot at. If the projectiles collide with these objects they shall loose lifepower. If the lifepower reaches 0, the object should be destroyed and explode.

TC 2.4.1:

Description:
The spaceship shoots projectiles at a rock which disappears and the rock explodes in the appropriate size.

Precondition:
The spaceship is able to shot and the rock can minimize in size and lose lifepower. Lägg in relaterade test case.

Post-conditions:
The rock is replaced by an explosion and is not visible anymore.

Normal Sequence:
The user is shooting projectiles at the rock
The rock looses lifepower and its size gets smaller
The lifepower of the rock reaches 0 and the rock is no longer visible, and is instead replaces by an explosion.

TC 2.4.2:

Description:
The spaceship shoots projectiles at a alienship and it looses lifepower and eventually explodes in the appropriate size and disappears

Precondition:
The spaceship is able to shoot and the spaceship can lose lifepower.
Lägg in relaterade test case.

Post-conditions:
The spaceship is replaced by an explosion and is not visible anymore.

Normal Sequence:
The user is shooting projectiles at the alienship.
The alienship looses lifepower.
The lifepower of the spaceship reaches 0 and the spaceship is no longer visible, and is instead replaced by an explosion.

Req 2.5

If the spaceship collides with the rocks or alien ships, the spaceship should lose lifepower and the lifebar decrease, and if enough is lost the “game over” screen shall be displayed.

TC 2.5.1:

Description:
The user collides with an obstacle and loses lifepower and sees the lifebar decrease.

Precondition:
The game is in progress. There should be obstacles (e.g rocks and alien ships) on the screen that the spaceship can shoot at Req 2.4

Post-conditions:
The game continues if the “lifepower” is enough but if too low (reaching bottom) the game switches to Game Over screen.

Normal Sequence:
The user is playing the game.
The user (space ship) sees an obstacle coming towards the spaceship
The user gets hit by the obstacle
The user looses lifepower and if it loses too much the player sees the Game Over screen

TC 2.5.2:

Description:
The spaceships lifebar is low, an obstacle hits the spaceship and makes the lifebar reach bottom which causes the Game Over screen to show.

Precondition:
The game is in progress. There should be obstacles (e.g rocks and alien ships) on the screen that the spaceship can shoot at Req 2.4.

Post-conditions:
the game switches to Game Over screen.

Normal Sequence:
The user is playing the game and has low life bar.
The user (space ship) sees an obstacle coming towards the spaceship
The user gets hit by the obstacle
The user looses lifepower and sees the Game Over screen

Req 2.6

The obstacles shall be destroyed when they reach the borders of the screen.

Req 2.7

The Alien ships shall be able to shoot back. If the spaceship is hit by an alien projectile then the spaceship loses lifepower and the lifebar should go down.

TC 2.7.1:

Description:
An alienship shoots back at the spaceship

Precondition:
An Alienship is created somewhere on the screen. There should be obstacles (e.g rocks and alien ships) on the screen that the spaceship can shoot at (Req 2.4). If the spaceship collides with the rocks or alien ships, the spaceship should lose lifepower (Req 2.5).

Post-conditions:
The alienship shoots projectiles

Normal Sequence:
The user is playing the game
An alienship is created
The alienship starts to shoot at the spaceship

TC 2.7.2:

Description:
The spaceship is hit by an alien projectile

Precondition:
An Alienship shoots projectiles that hits the spaceship

Post-conditions:
The spaceship looses lifepower

Normal Sequence:
The user is playing the game
An alienship is created and starts shooting projectiles
An alienprojectile hits the spaceship
The lifebar goes down

Req 2.8

The alien ships should follow the movement of the SpaceShip and shoot at it when their x-positions aline. The Alien ship should not be shooting when this condition is not true.

TC 2.8.1:

Description:

An alien ship appears and starts to follow the spaceship in the x direction

Precondition:
An Alienship shoots projectiles that hits the spaceship

Post-conditions:
The spaceship looses lifepower

Normal Sequence:
The user is playing the game
An alienship is created and starts shooting projectiles
An alienprojectile hits the spaceship
The lifebar goes down

Req 2.9

There will be a fine tune playing in the background if the sound option and the music option is checked and the volumes are not zero. When the player pauses the game or when the Game Over screen shows up, the music shall stop playing. The music shall loop so that it does not end.

TC 2.9.1:

Description:
If the options "sound" and "Sound->music" are checked, the music shall start when the game starts

Precondition:
The "sound" and "music" options are checked and the game starts

Post-conditions:
The music is playing in the background

Normal Sequence:
The user presses Start Game
The music starts together with the game

TC 2.9.2:

Description:
If the "sound" option is checked but the "music" option is unchecked, and the game starts, then the music should not start playing

Precondition:
The "sound" option is checked and "music" option is unchecked and the game starts

Post-conditions:
The music is not playing in the background

Normal Sequence:

The user presses Start Game
The music does not start together with the game

TC 2.9.3:

Description:
If the "sound" option is unchecked and the "music" option is checked(but disabled), and the game starts, then the music should not start playing

Precondition:
The "sound" option is unchecked and "music" option is checked(but disabled) and the game starts

Post-conditions:
The music is not playing in the background

Normal Sequence:
The user presses Start Game
The music does not start together with the game

TC 2.9.4:

Description:
The music shall loop around. When the song ends, it shall start again from the beginning.

Precondition:
The song ends

Post-conditions:
The music starts again

Normal Sequence:
The music is playing
The song ends
The music starts from the beginning

Req 2.10

There will be soundeffects for when the spaceship shoots projectiles and for when the different explosions occur on the screen.

TC 2.10.1:

Description:
The user shoots projectile and hear an appropriate soundeffect.

Precondition:
The game is in progress and the fire button is pushed, and the “sound”

and “soundeffects” are checked. The spaceship should be able to shoot projectiles Req 2.3.

Post-conditions:
The soundeffect ends when the fire button is let go

Normal Sequence:
The user is playing the game.
The user (space ship) shoots a projectile and a sound is heard

TC 2.10.2:

Description:
An exposion is “heard” when object on the screen is hit and an explosion occurs.

Precondition:
The game is in progress and the fire button is pushed, and the “sound”
and “soundeffects” are checked. The spaceship should be able to shoot projectiles Req 2.3. An Explosion occur when enemies, obstacles and spaceship is destroyed Req 2.12.

Post-conditions:
The soundeffect ends a short while after the explosion

Normal Sequence:
The user is playing the game.
The user (space ship) shoots a projectile and hit an obstacle, or obstacle kills the spaceship
An explosion occurs and a sound is heard together with the explosion.

TC 2.10.3:

Description:
The user shoots projectile and does not hear any soundeffects.

Precondition:
The game is in progress and the fire button is pushed, and the “sound”
and “soundeffects” are unchecked. The spaceship should be able to shoot projectiles (Req 2.3)

Post-conditions:
No sound is played

Normal Sequence:
The user is playing the game.
The user (space ship) shoots a projectile and a sound is not heard

TC 2.10.4:

Description:
An explosion is silent when object on the screen is hit and an explosion occurs.

Precondition:
The game is in progress and the fire button is pushed, and the “sound” and “sound effects” are unchecked. The spaceship should be able to shoot projectiles (Req 2.12).

Post-conditions:
No sound is played.

Normal Sequence:
The user is playing the game.
The user (space ship) shoots a projectile and hit an obstacle, or obstacle kills the spaceship
An silent explosion occurs.

Req 2.11

The background image will move downwards during the whole game. When the game is paused the background should be stopped. The background should appear to be continuous without edges showing.

TC 2.11.1:

Description:
The background is moving downwards during the whole game

Precondition:
The game starts

Post-conditions:
The background has moved during the whole game, and the game is over

Normal Sequence:
The user presses Start Game
The background starts moving downwards automatically

TC 2.11.2:

Description:
When the game is paused the background should be stopped, and started again when the game is unpaused

Precondition:
The background is moving and then the game is paused (req 2.17, req 2.18)

Post-conditions:
The background has started moving again when the game is unpaused

Normal Sequence:
The user is playing and the background is rolling
The user pauses the game
The background stops moving
The user starts the game again

The background starts moving

TC 2.11.3:

Description:
The background should appear to be continuous without edges showing

Precondition:
The background starts rolling down

Post-conditions:
The background shows no edges during the whole game

Normal Sequence:
The user presses Start Game
The background starts moving downwards automatically
No edges have appeared during the game

Req 2.12

An Explosion occur when enemies, obstacles and spaceship is destroyed. The Explosion appears in the appropriate size and in the appropriate position.

TC 2.12.1:

Description:
An Explosion occur when enemies, obstacles and spaceship is destroyed

Precondition:
The object is destroyed Req 2.4. If the spaceship collides with the rocks or alien ships, the spaceship should lose lifepower and the lifebar decrease Req 2.5.

Post-conditions:
The Explosion disappears.

Normal Sequence:
The player is shooting at a obstacle
The obstacle looses lifepower and when the lifepower reaches zero it gets destroyed.
An explosion appears at the position of the obstacle destroyed.
The explosion fades out.

TC 2.12.2:

Description:
The explosion appears in the appropriate position

Precondition:
The object is destroyed Req 2.4. If the spaceship collides with the rocks or alien ships, the spaceship should lose lifepower and the lifebar decrease Req 2.5.

Post-conditions:
The Explosion disappears.

Normal Sequence:
The player is shooting at a obstacle
The obstacle looses lifepower and when the lifepower reaches zero it gets destroyed.
An explosion appears at the position of the obstacle destroyed.
The explosion fades out.

TC 2.12.3:

Description:
The explosion appears in the appropriate size.

Precondition:
The object is destroyed (Req 2.4). If the spaceship collides with the rocks or alien ships, the spaceship should lose lifepower and the lifebar decrease Req 2.5.

Post-conditions:
The Explosion disappears.

Normal Sequence:
The player is shooting at a obstacle
The obstacle looses lifepower and when the lifepower reaches zero it gets destroyed.
An explosion appears in the same size as the destroyed obstacle.
The explosion fades out.

Req 2.13

The user shall be able to toggle between weapon-types by pressing a button.

TC 2.13.1:

Description:
User changes weapon-type and the SpaceShip shoots different projectiles.

Precondition:
SpaceShip is shooting projectiles (Req 2.3) of type 1

Post-conditions:
SpaceShip is shooting projectiles of type 2

Normal Sequence:
The player is pressing the shoot-button SpaceShip is shooting projectiles of type 1.

User presses the toggle buttons (change-weapon-button).
The SpaceShip is shooting projectiles of type 2.

Req 2.14

There shall be a lifebar that the user can see. The lifebar shall decrease in size when the SpaceShip gets hit by enemies or enemy-projectiles. When the lifebar is zero the game shall end and the GameOver-screen shall be displayed.

TC 2.14.1:

Description:
There shall be a lifebar that the user can see.

Precondition:
The user starts the game.

Post-conditions:
The user is playing and a lifebar is displayed.

Normal Sequence:
The user presses “Start game”
The user sees a lifebar on the screen.

TC 2.14.2:

Description:
The lifebar shall decrease in size when the SpaceShip gets hit by enemies or enemy-projectiles.

Precondition:
The user gets hit by an alienship or an alienprojectile or an asteroid.

Post-conditions:
The lifebar is decreased

Normal Sequence:
The lifebar is 50%
The user gets hit by an enemy or an enemy projectile
The lifebar is < 50%

TC 2.14.3:

Description:
When the lifebar is zero the game shall end and the GameOver-screen shall be displayed.

Precondition:
The spaceship collides with an enemy or enemy-projectile and the life bar decreases to zero

Post-conditions:
The game over screen is displayed

Normal Sequence:

The spaceship gets hit by enemies or enemy projectiles
The life bar goes to zero
The game over screen is displayed

Req 2.15

There shall be a score-counter that the user can see and that maps the amount of score-points that the user has at the moment. The counter shall increase when enemies are destroyed. The amount of score that the score-counter increases with is based on the type of enemy that is destroyed.

TC 2.15.1:

Description:
There shall be a score-counter that the user can see and that maps the amount of score-points that the user has at the moment. The counter shall increase when enemies are destroyed.

Precondition:
The score counter shows some value X and an enemy is destroyed

Post-conditions:
The score counter shows a value > X

Normal Sequence:
The spaceship shoots down an enemy, that gets destroyed
The score counter goes from the value X to X + Y, where Y > 0.

TC 2.15.2:

Description:
The amount of score that the score-counter increases with is based on the type of enemy that is destroyed.

Precondition:
The score counter shows some value X an Asteroid is destroyed

Post-conditions:
The score counter shows X + 100

Normal Sequence:
The spaceship shoots down an asteroid, that gets destroyed
The score counter goes from the value X to X + 100.

TC 2.15.3:

Description:
The amount of score that the score-counter increases with is based on the type of enemy that is destroyed.

Precondition:
The score counter shows some value X an Alien ship is destroyed

Post-conditions:
The score counter shows X + 50

Normal Sequence:
The spaceship shoots down an alien ship, that gets destroyed
The score counter goes from the value X to X + 50.

Req 2.16

When the SpaceShip is hit the phone should vibrate if that options is chosen in the options screen. If the option is set to false it should not vibrate.

TC 2.16.1:

Description:
When the SpaceShip is hit the phone should vibrate if that option is chosen in the options screen.

Precondition:
The vibrate options is turned on (Req 6.2) and the spaceship is hit by an enemy or an enemy-projectile

Post-conditions:
The phone vibrates

Normal Sequence:
The spaceship is hit by an enemy or an enemy-projectile
The phone vibrates

TC 2.16.2:

Description:
When the vibrate-option (Req 6.2) is set to false it should not vibrate.

Precondition:
The spaceship is hit by an enemy or an enemy-projectile

Post-conditions:
The phone does not vibrate

Normal Sequence:
The spaceship is hit by an enemy or an enemy-projectile
The phone does not vibrates

Req 2.17

The game should be paused if the user taps somewhere on the game-screen. When the user touches the screen when the game is paused, the game should continue where it left off.

TC 2.17.1:

Description:
The game should be paused if the user taps somewhere on the game-screen.

Precondition:
The user taps the screen while the game is on.

Post-conditions:
The game is paused.

Normal Sequence:
The user is playing
The user touches the screen
The game is paused and a pause screen is shown

TC 2.17.2:

Description:
When the user touches the screen when the game is paused, the game should continue where it left off.

Precondition:
The user taps the screen while the game is paused.

Post-conditions:
The game continues where it left off.

Normal Sequence:
The game is paused
The user touches the screen
The game starts again where the player left off

Req 2.18

The game should be paused automatically if there is an incoming call. When the user comes back to the game after the call, the game should be at the pause-screen.

TC 2.18.1:

Description:
The game is paused when there is an incoming call.

Precondition:
The user gets an incoming call while playing.

Post-conditions:

The game is paused.

Normal Sequence:
The user is playing
The user gets an incoming call
The game is paused

TC 2.18.2:

Description:
The game is resumed from paused state.

Precondition:
The game is paused

Post-conditions:
The game continues where it left off

Normal Sequence:
The user is talking on the phone and the game is paused
The user finishes the call and touches the screen
The game starts again where it left off

5.3 Game Over Screen (GameOverActivity)

Req 3.1

The screen should display the Game Over Screen, where the user can insert his/her player name and store a high score when pressing the save score button which will open up the high score screen. Addition to this button there are two more buttons, the Main Menu button which will return the user to the main menu screen and the restart button which will start a new game.

TC 3.1.1:

Description:
The user adds his/her player name to store the players high score

Precondition:
The user dies and the Game over screen appears with a high score.
(Req 2.14)

Post-conditions:
The high score is added and the high score screen appears.

Normal Sequence:
The user dies.
The game over screen appears.
The user adds his/her player name.
The user presses the save score button to add the score.

TC 3.1.2

Description:
The user ends up at the game over screen and wants to return to the main menu

Precondition:
The user dies and the Game over screen appears.
(Req 2.14)

Post-conditions:
The user ends up at the main menu.

Normal Sequence:
The user dies.
The game over screen appears.
The user presses the main menu button.
The user ends up at the main menu screen.

TC 3.1.3

Description:
The user ends up at the game over screen and wants to restart the game.

Precondition:
The user dies and the Game over screen.
(Req 2.14)

Post-conditions:
The user starts playing the game.

Normal Sequence:
The user dies.
The game over screen appears.
The user presses the restart button.
The game starts again.

5.4 Highscore Screen (HighScoreActivity)

Req 4.1

The High score screen appears should appear and show the top 10 scores if there are registered top 10 scores.

TC 4.1.1:

Description:
The user wants to see the top 10 scores.

Precondition:
That there are some scores added in the high score list.

Post-conditions:
The scores are viewed.

Normal Sequence:
The user views the scores

Req 4.2

On the high score screen there is a button that should return to the main screen.

TC 4.2.1:

Description:
The user has entered the high score screen and wants to return to the main menu.

Precondition:
The user is in the high score screen.
(Req 1.1)

Post-conditions:
The user ends up on the main menu screen

Normal Sequence:
The user is in the high score screen
The user presses the main menu button
The user ends up on the main menu

5.5 Options Screen (OptionsActivity)

Req 6.1

There shall be two seek bars. One that control the volume of the in-game-music and another one that controls the in-game-sound-effects. The volume shall be saved in the applications shared preferences and the settings shall be remembered even if the application is restarted.

TC 6.1.1:

Description:
There shall be two seek bars. One that control the volume of the in-game-music and another one that controls the in-game-sound-effects.

Precondition:
The user hears no music and no sound effects and dies and ends up in the options screen and turns both volumes to maximum.

Post-conditions:
The user plays the game and hears music and sound effects

Normal Sequence:

The user hears no music and no sound effects and dies.
The user goes to the options screen
The user turns both volumes up to maximum
The user starts playing, and now hears music and sound effects

TC 6.1.1:

Description:
The volume shall be saved in the applications shared preferences and the settings shall be remembered even if the application is restarted.

Precondition:
The volume goes from minimum to maximum and the application is killed

Post-conditions:
The options screen shows maximum volumes on both seek bars.

Normal Sequence:
The user is in the options screen and turns both volumes from minumum to maximum.
The application is killed.
The application is restarted.
The user goes to the options screen.
The seek bars are both at maximum.

Req 6.2

There shall be an options to turn on/off vibrations when the spaceship gets hit by enemies or their projectiles. The button shall toggle between these options.

TC 6.1.1:

Description:
There shall be an options to turn on/off vibrations when the spaceship gets hit by enemies or their projectiles.

Precondition:
The vibrate option is on (true)

Post-conditions:
The phone vibrates when the spaceship gets hit by enemies or projectiles.

Normal Sequence:
The user goes to the options screen
The user turns on the vibrate function
The user plays the game and the spaceship get hit by an enemy
The phone vibrates

6 Automatic Test

6.1 Unit test

The testing framework used is Junit. It has been used with Eclipse as an Android Junit test.

6.2 Unit Testing Exceptions

We will not handle methods that doesn't return any values. The activity classes will not be handled because they are tested in test cases. The main reason is also that the GUI is troublesome to handle in unit testing.

6.1 HighScores

The high scores class will not be unit tested because all of its methods are tested with the test cases for HighScoresActivity and GameOverActivity.

6.2 GameView

Will not be unit tested because it is tested by test cases which show that the update methods work and draw methods work.

6.3 GameLoopThread

This is tightly connected with the gameview so it is confirmed that it works.

6.4 GameMusic

It works because the music plays. It is not unit tested because the file can not be compared to a value.

6.5 SoundEffects

It works because the effects are heard. It is not unit tested because the file can not be compared to a value.

6.6 Level

The majority of components in this class are get and set method and some are abstract. Which we have confirmed that they work by seeing it on the playscreen.

6.7 LevelOne

This will not be unit tested because of the same reasons as gameview.

6.8 GameObject and MovingObject

These two classes are super classes for all the model classes. They have been tested so the majority of the model classes will not be tested because their functionality has been tested in GameObject and MovingObject.