Requirements and Analysis of GradiuZ

DAT255 – Software Engineering

Rehan Butt	860722-4972
Martin Bäckman	881101-2718
Michael Jasinski	880423-5094
Patrik Nygren	821023-8518

Introduction

This plan is formed to give an overview of the organization and conceptualization of the project “GradiuZ”, and also what deliverables it will consist of and at what time these deliverables will be delivered. 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”.

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 developed through an Agile work process, and will be held within the limitations that this process puts upon the project.

1. Project Organization

The work is divided into a number of content areas. Each content area is lead by a committer which is responsible for the content area. The responsible comitter is also responsible for delegating the workload amongst the collaborators in the content area for achieving the completion of the tasks within each iteration.

The work is divided into the following content areas:

- Project management: Patrik Nygren
- Requirements: Michael Jasinski
- Change management: Martin Bäckman
- Development: Michael Jasinski
- Architecture: Patrik Nygren
- Test: Martin Bäckman
- General: Rehan Butt
- Design management: Rehan Butt

2. Project “GradiuZ” overview

The application will be a space game. In the end product the software will be a sidescrolling 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 layout of the game will be like figure 1.

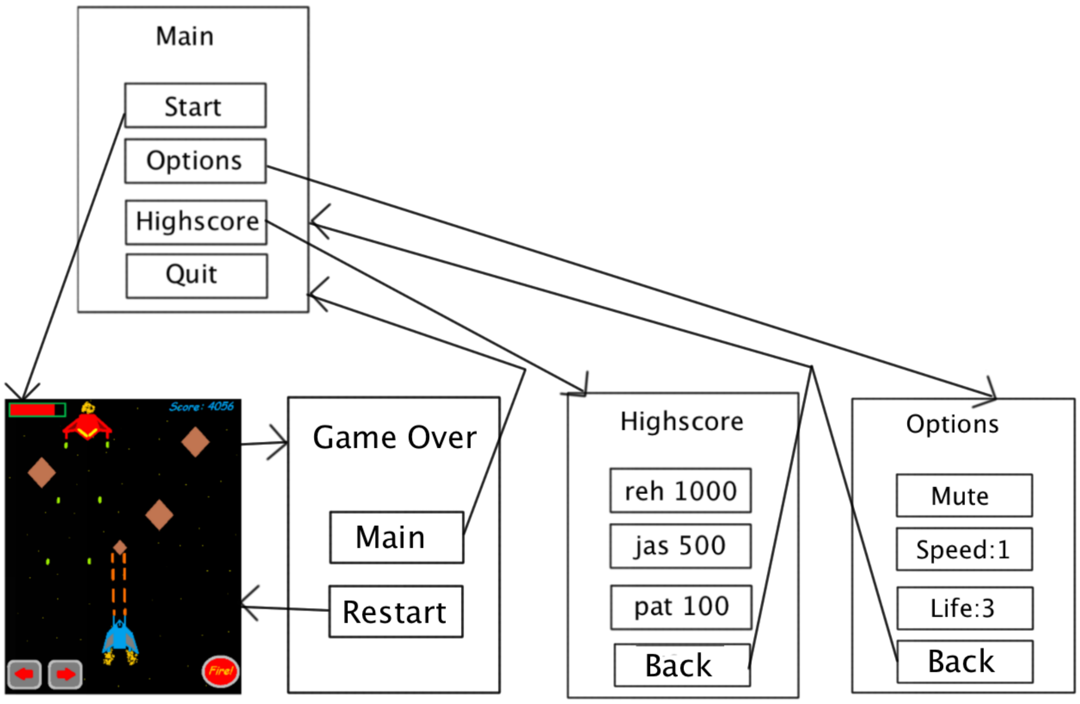

Figure 1

General specifications of the game on the Android plattform

Game Programming concepts

- Game loop
- Game Engine (update and draw)
- Sprites Coalitions detection
- Game resources: images and sounds

Android Specifics

- SurfaceView and holder events management
- Resource management in android = bitmaps and sound

Game Architecture

- The problem in simple and reusable parts
- Decouple the game logic from the android architecture

3. System Requirements

3.1 Main screen

3.1.1 Description and Priority

- The main screen in the game will contain five buttons: Start Game, Options, Highscore, How-to-play and Quit Game.
- This feature has a high priority, since it is the starting screen but some of the subfeatures will have lower priority. Start Game, Options and Quit Game will have the highest priority.

3.1.2 Use Cases

UC 1	Starting game
Description:	The user chooses to start the game by pressing “Start Game”
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.

3.1.3 Functional Requirements

Nr	Requirement	Priority	Status
1	There shall be 5 options to choose from on the main screen • Start Game: Starts the PlayScreenActivity. • Options: Starts the OptionsActivity. • Highscore: Starts the HighScoresActivity. • How-To: Starts the HowToPlayActivity.	High	
2	When the menu button is pressed on the phone a single menu option shall appear, “About Us”. This option will display a short presentation of the authors of the game.	Medium	

3.2 Play screen

3.2.1 Description and Priority

- The main feature in the game will be the game view. It will feature a view similar to the Space Invaders game and enable a spaceship shooting obstacles and moving according to x and y -coordinates of the screen.
- This feature has a high priority, since it is a main feature that will be implemented and it will be what the user will navigate while playing the game.

3.2.2 Use Cases

UC 1	The user collides with an obstacle such as a monster.
Description:	The user will lose “life power”, and if too much is lost the game will end.
Precondition:	The game is in progress, in PlayView.
Post-conditions:	The game continues if the “lifepower” is enough but if too low the game switches to GameOverScreen..
Normal Sequence:	The user is playing the game. The user (space ship) collides with obstacle. The user's loses life power or loses the game. The user can choose to restart game or save the result if high score has been achieved.

UC 2	Player shoots or kills monster/boss/obstacle
Description:	The obstacle looses life power or dies(disappears) from screen by shots from spaceship.
Precondition:	A game is in progress and a obstacle gets hit by shots from spaceship.
Post-conditions:	The game takes away life power in case of monster or boss, but takes away size in case of obstacle.
Normal Sequence:	The player fires shots. Obstacle gets hit. The obstacle looses size or life power, or disapear from screen.

3.2.3 Functional Requirements

Nr	Requirement	Priority	Status
1	The screen shall display a spaceship (the player) that can move back and forth in the x and y -directions.	High	
2	If the spaceship collides with the edges of the screen the movement shall be stopped.		
3	The spaceship should be able to shoot projectiles.	High	
4	There should be obstacles (e.g rocks and alien ships) on the screen that the spaceship can shoot at. If the projectiles collides with these objects they shall be destroyed.	High	
5	If the spaceship itself collides with the rocks or alien ships, the spaceship should lose lifepower and if enough is lost the “game over” screen shall be displayed. When all of the life power is lost, the game is over and the GameOverActivity shall appear.	Medium	
6	There shall be a life bar and a score counter at the top of the screen that displays the life left and the scores so far.	Medium	
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.	Low	
8	The rocks and alien ships shall be able to move towards the bottom screen and when they touch the bottom screen they shall get destroyed.	Low	
9	The rocks and alien ships shall be able to move down towards the bottom screen in a random manner. The movement will alternate randomly between moving left-diagonally and right-diagonally.	Low	

3.3 Game Over screen

3.3.1 Description and Priority

- The Game Over screen will pop up when the user gets killed. This screen will have the functionality to restart the game or return to the main menu.
- This feature has a high priority, since it is a main screen so that the functionality of the application will work.

3.3.2 Use Cases

UC 1	The user collides with an obstacle such as a monster.
Description:	The user chooses to restart the game by choosing “Restart” or returning to the main menu by choosing “Main”.
Precondition:	The application is running and the user dies.
Post-conditions:	The “Game Over”-screen appears.
Normal Sequence:	The user dies. The user has the options to choose between Restart and return to main.

3.3.3 Functional Requirements

Nr	**Requirement**	**Status**
1	There shall be 2 options to choose from on the Game Over screen • Restart: Starts the PlayScreenActivity. • Main: Start the Main screen.	**1**

3.4 Highscore screen

3.4.1 Description and Priority

- The highscore screen will display the top scores that users have made when they have been playing the game. The highscore board stores the top 10 scores.
- This feature has a low priority, since it doesn’t contribute to the gameplay.

3.4.2 Use Cases

UC 1	The user collides with an obstacle such as a monster.
Description:	The score gets added to the highscore screen.
Precondition:	The application is running and the user dies.
Post-conditions:	The highscore screen appears if the user is in the top 10 players
Normal Sequence:	The user dies. The highscore screen appears if the user is the top 10.

3.4.3 Functional Requirements

Nr	Requirement	Priority	Status
1	The highscore screen has a top 10 list	low	
2	To be added to the highscore list, the user has to have a score better then the 10th score.	low	

3.5 Options screen

3.5.1 Description and Priority

- The Option screen will contain different options e.g. mute on/off, speed setting and life.

3.5.2 Use Cases

UC 1	The user wants to change settings.
Description:	The user want to mute the sound.
Precondition:	The sound is on.
Post-conditions:	The sound is off.
Normal Sequence:	The user is in on a lecture. The teacher eyeballs him. The student turns off the sound.

3.5.3 Functional Requirements

Nr	Requirement	Priority	Status
1	There shall be 3 options to choose from on the Options screen: • Mute on/off • Speed • Life	low	
2	Mute will turn off and on the music	low	
3	The speed option will change the speed of the obcstacles coming towards the user.	low	
4	The life setting will change how many times the user can be hit.	low	

3.6 How to - Instructions

3.6.1 Description and Priority

- The players do not understand the controls, rules, or objective of the game.
- This feature is of low priority because the players need to learn how to play the game if they cannot figure it out on their own. However, the game controls should be intuitive to learn.

3.6.2 Use Cases

UC 1	Player does not understand how to play the game or the objective of the game.
Description:	The game will display the user manual documentation.
Precondition:	The help document must exist and the user must request to access it.
Post-conditions:	The help document or user manual is displayed.
Normal Sequence:	The user asks for help. Game loads help document. Help information is displayed and the user can read it. The user closes the help document and continues on with the game.

3.6.3 Functional Requirements

Nr	**Requirement**	**Priority**	**Status**
1	The game shall provide support for accessing and displaying a user manual or help document from the main menu.	low	

4. External Interface Requirements

4.1 User Interfaces

- UI-1: All user interfaces shall be designed following the guidelines agreed upon by collaborators.
- UI-2: The main user interface shall consist of a window that will contain the actual game view and additional displays for providing status information, such as high scores and options information.
- UI-3: A toolbar shall appear on every screen, which will include buttons for frequently performed tasks (help, save/load, etc.).
- UI-4: Consistent pop-up windows or a defined message area shall display all error messages.
- UI-5: Additional interfaces shall be implemented for configuring the initial game setup, viewing detailed property management information, and for accessing the unique features such as life power and points scored.
- More specific details of the user interface design shall be documented in a separate user interface specification in the future.

4.2 Communications Interfaces

- There are no specific communications interface requirements for the game since online multiplayer support will not be implemented. However, game updates will require downloading and running the new executable from the Internet. Therefore, it is recommended that Internet access be available.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- PR-1: The game should be able to run efficiently on any android mobile or tablet that fulfills the hardware and software interface (api 2.1) requirements.
- PR-2: Within three seconds of starting the game the spaceship and game view should be visible.
- PR-3: Within 15 seconds, the game should be able to load in its entirety so that it is fully playable.

5.2 Reliability Requirements

- RR-1: The game shall be able to accurately keep track of previous high scores to adjust to actual games score properties, saved position in game from previous saved instance.
- RR-2: The game should not crash under normal or standard operation.

5.3 Security Requirements

- SR-1: There shall not be any ways to "cheat" in the game, including any sneak paths.

5.4 Software Quality Attributes

- SQA-1: The game should be able to be played on any mobile that meets the minimum requirements for running “GradiuZ” by simply launching its executable.
- SQA-2: A player with basic computer skills shall be able to quickly master the game play controls.
- SQA-3: The game should be intuitive and fun to play.
- SQA-4: The game should give the experience of a classic retro game.

6. Project Objectives and Milestones

This section covers objectives for the entire “GradiuZ” project.

- Learning/Setting up Git repository and setting up the project in the repository.
- Learn Android, and android specific programming for game development within the relevant area.
- Learn Agile methodology for development in group, and other relevant methodology from the website/book.
- Learn requirement engineering.
- Learning how to manage a software project.
- Learning and applying requirements testing throughout the project.

This section covers the Milestones of the “GradiuZ” project.

- Creating an architectural design for the project, and then implementing the design.
- Creating the starting activity for the game with the option of starting the game and making those clickable.
- Setting up the “frame” for the actual “game area” and making the spaceship movable with basic buttons (back/forward and sides).
- Creating a simple object moving up and down on the frame.
- Implement the physics into the movement of the spaceship so that it accounts for acceleration and such.
- “Throwing” in some basic objects, in the form of simple boxes and triangles into the game and implement “collision handling” between the objects and the spaceship.
- Making the objects move (fly through space) in some predetermined path.
- Implementing the accelerometer for movements of the spaceship.
- Deploying on Android market!

Phase	Iteration	Primary Objective (risks and use cases scenarios)	Scheduled start or milestone	Duration Estimate (calendar days)
Inception/ Warm-up	1	• Learning git and the basic workflow of the VCS. • Learning Android and the common practices of game development. • Setting up git repository. • Setting up android SDK on the Eclipse IDE. • Creating a simple architectural design, with a class diagram of the project. • Meeting with tutors. Project organization and plan agreed upon.	2012-03-23 - 2012-03-30	7
Elaboration	2	• Learn Agile methodology for development in group • Creating the starting activity for the game with the option of starting the game and making those clickable. • Breaking up project into units and researching resources to evolve each unit. • Organize the main tasks for each collaborator of the project.	2012-03-30 - 2012-04-06	7
Elaboration	3	• Agree on naming conventions • Learn requirement engineering. • Learning and applying requirements testing throughout the project. • Setting up the “frame” for the actual “game area” and making the spaceship movable with basic buttons (back/forward and sides). • revision of milestones and objectives (this) to add or remove features	2012-04-06 - 2012-04-13	7
Construction	4	• Agree upon general structure of the project, and revision the original plan (this)	2012-04-13 - 2012-04-16	3

		• Creating a simple object moving up and down on the frame.		
Construction	5	• Implement the physics into the movement of the spaceship so that it accounts for acceleration and such. • Agree on a management process • revision of milestones and objectives (this) to add or remove features	2012-04-15 - 2012-04-22	7
Construction	6	• Finalize structure • “Throwing” in some basic objects, in the form of simple boxes and triangles into the game and implement “collision handling” between the objects and the spaceship. • Improve usability and first impression by blackbox testing the game, “user” testing, and letting the user review the game. • Establish a functioning management process for the project • revision of milestones and objectives (this) to add or remove features	2012-04-29 - 2012-05-06	7
Construction	7	• Improve usability and first impression by blackbox testing the game, “user” testing, and letting the user review the game. • Making the objects move (fly through space) in some predetermined path. • revision of milestones and objectives (this) to add or remove features	2012-05-06 - 2012-05-12	7
Transition / End Game	8	• End game • Implementing the accelerometer for	2012-05-12 - 2012-05-end	7

		movements of the spaceship. • Finalize launch collateral • Deploying on Android market!	of course	

7. Deployment

N/A.

8. Lessons Learned

N/A.