Requirements and Analysis Document for “Tanks”

Table of Contents

Version 1.1
Date 2013-05-14
Authors Albin Garpetun, Johan Backman, Per Thoresson, Daniel Bäckström

This version overrides all previous versions.

1 Introduction

1.1 Purpose of application

We got contacted by DICE to produce a spin off to their Battlefield series. They wanted to make a game with just tanks. After a long discussion we agreed with the terms that we would get 50% of the profits.

We were given a more specific goal by DICE and it was to to create a fast paced three-dimensional vehicle oriented third-person multiplayer shooter, with a rich cartoonish graphical style.

1.2 General characteristics of application

This project will be a multi-platform desktop application, featuring multiplayer player versus player gameplay. We will use a technique called “split-screen”. The application will be controlled preferably with one to four XBox controllers, via an adapter plugged into the computer. If not possible though, controlling solely via keyboard will be an option, but due to hardware restrictions more than 2 players on one keyboard will not work.

The game starts up in a menu, where you can choose rules, number of players and such things. When the main game is started every player gets their Tank played out in the game map. The goal is to kill each other by shooting, and by utilizing the power-ups scattered around the map. Depending on the rules, you either win by amounts of kill over a timer, or if you get first to X amount of kills. If you die (by health dropping to zero) you either respawn or die permanently, depending on the rules you have set beforehand.

1.3 Scope of application

The game prototype will not feature online multiplayer gameplay or a version compatible with the mobile operating system Android.

1.4 Objectives and success criteria of the project

1. One to four players should be able to start the 3d game with xbox controllers, drive around with their tanks, pick up powerups and kill each other until someone wins the game.

2. In the final prototype the models should look cartoonish and simple to enable a normal gameplay without performance issues on normal modern computer hardware and without glitches.

1.5 Definitions, acronyms and abbreviations

JMonkey Game Engine, the game engine used in the development.
Java, the programming language used in the project.
Tank, the player controlled vehicle.

Projectile, object fired by the player.
Power Up, a special projectile or “super power” given to the player.
Xbox controller, a hand controller from the Xbox 360 entertainment system.
DICE, a swedish game producer owned by EA Games.
Spatial, graphical representation of an object.
FPS, frames per second.

2 Requirements

2.1 Functional requirements

The player should be able to:

1. Start a new Game.
   - The player can change with rules will apply from an options menu.
2. Pause the game.
   - When pause button is pressed, the pause menu shows.
3. Start new game.
   - From the pause menu.
4. Move around in the world.
   - The tank can collide with objects
   - The tank can move in any direction.
5. Fire a weapon.
   - The player can always fire a cannonball, if the cannonball hits another player, that player will lose health.
6. Pick up a powerup.
   - By driving over a powerup the powerup it either gets activated immediately or gets picked up and put in the powerup slot depending of the kind of the powerup.
7. Use a powerupslot.
   - If you have a powerup in your powerup slot you can activate it by pushing a button.
8. Exit the game.

2.2 Non-functional requirements

2.2.1 Usability

The players should easily and quickly be able to specify number of players and setup other settings for the game and then start it. When in game the players should easily be able to restart and access settings. The game should supply the player with all the necessary feedback needed i.e. when you gain score, die etc.

There should be short commands and easy access to toggle different settings of the game, i.e. toggling sound, pause and controlling sound volume.

2.2.2 Reliability
N/A.

2.2.3 Performance
1. The application should respond seemingly immediately to any player actions with proper graphics displayed.
2. On a modern computer the application should never drop lower than 30 FPS.

2.2.4 Supportability
The game will be very expandable, maps, various tanks and super powers to be added without any apparent effort. The view and the model should be well separated which means in this case a game engine is involved things that are associated with JMonkey Game Engine ports in controls, while all data and calculations ports in the models, spatials and graphical object should be changeable without hassle.

2.2.5 Implementation
The application will be implemented solely with Java, utilizing an existing library called JMonkey Game Engine. Java Runtime Environment will be needed to run the application.

2.2.6 Packaging and installation
Every personal computer with Java Runtime Environment can run this application. A single jar-file is all that is needed to run the application. If one would want to compile the application from the source code, one would need Java Development Kit 1.6 or later, as well as jMonkey Engine.

2.2.7 Legal
N/A.

2.3 Application models

2.3.1 Use case model
See Appendix for UML diagram and a list of Use Cases.

2.3.2 Use cases priority
1. Steer vehicle
2. Shoot

3. Start new game
4. Exit game
5. Pause game
6. Use powerup

2.3.3 Domain model

See Appendix for Domain Model.

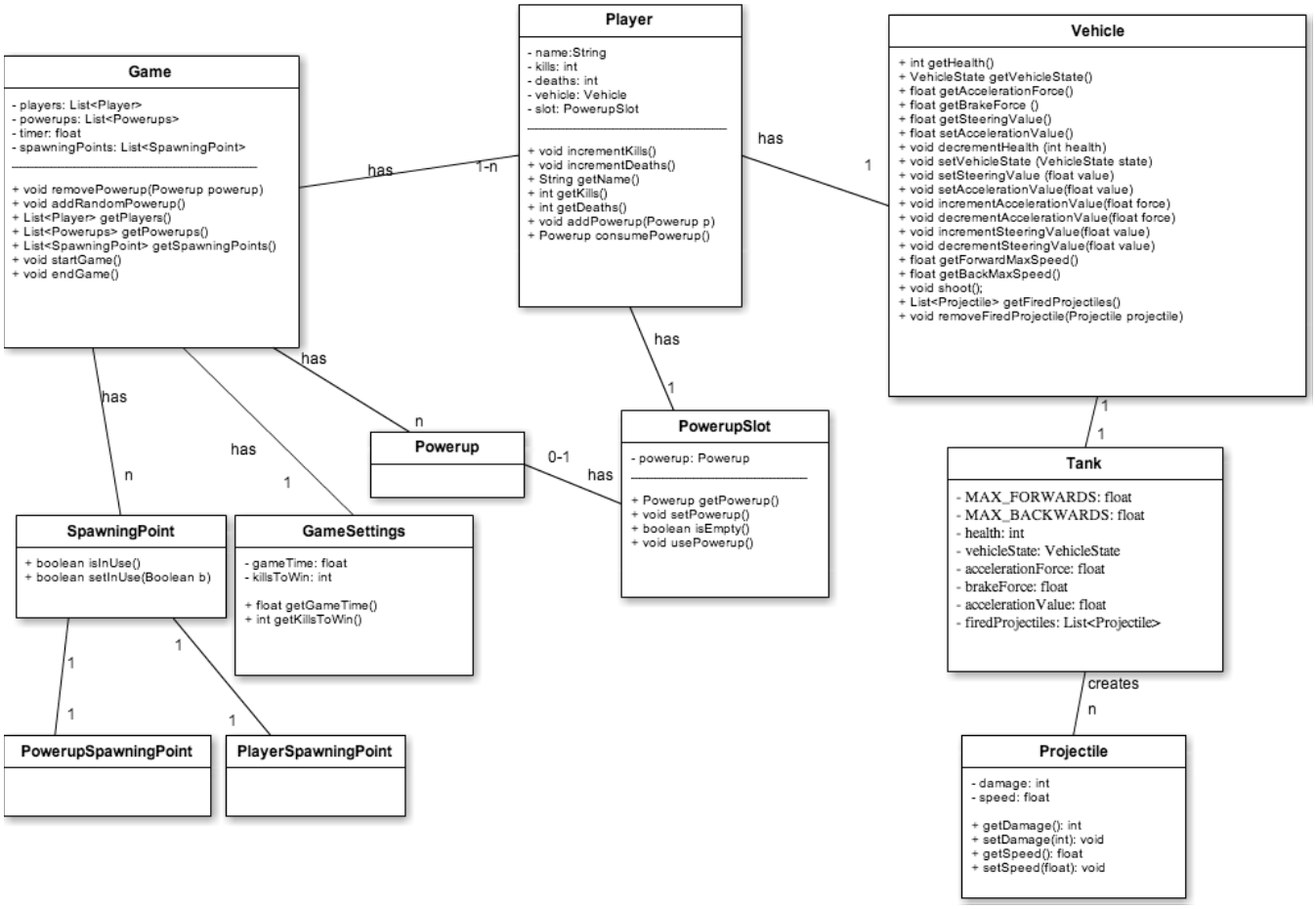
2.3.4 User interface

The game utilizes a cartoony graphical style so the user interface also utilize the same graphical style and feeling.

2.4 Appendix

GUI

Domain model



Use case texts

Use Case: Steer vehicle

Summary:
Steers the players vehicle in different directions depending on the keymap of the game.

Priority:
High

Extends:
N/A

Includes:
N/A

Participators:
The player of the game.

Normal flow of events
A simple movement in the game world with no consequences.

	Player	**System**
1	Presses any directional key.	
2		Vehicle moves in the corresponding direction.

Alternate flow

Flow 2.1 Collides with a powerup

	Player	**System**
1	Presses any directional key.	
2		Vehicle moves in the

		corresponding direction.
3		Vehicle collides with a powerup.
4		Player obtains the powerup.
5		Remove the powerup from 3d world

Exceptional flow
Player can't drive with the vehicle through an environmental object or another player's vehicle.

Use Case: Shoot

Description
The player shoots straight in front of itself.

Priority
High

Extends
N/A

Includes
N/A

Participators
N/A

Normal flow of events
Firing of a simple shot, which hits the environment.

	Player	**System**
1	Presses shoot button	
2		Fires projectile

3		Impact on environment
		Animation on impact

Alternate flow
Firing of a simple shot, which hits another player or a target.

	Player	**System**
1	Presses shoot button	
2		Fires projectile
3		Impact on enemy
		Animation on impact
		Enemy takes damage

Exceptional flow
N/A

Use Case: Exit game from gameplay

Summary:
This is how the player exits the game.

Priority:
High

Extends:
N/A

Includes:
Pause game

Participators:

Player

Normal flow of events
Normal exit from actual gameplay

	Player	**System**
1	Presses the escape button	
2		Locks keybindings and shows the menu, the game continues in the background
3	Presses the exit button	
4		The game shutdowns

Alternate flow:
N/A

Exceptional flow:
N/A

Use Case: Start new game

Summary:
How player chooses to start a new game in startmenu.

Priority:
High

Extends:
N/A

Includes:
N/A

Participators:
Player

Normal flow of events

	Player	**System**
1	Player chooses to start a new game in main menu.	
2		Loads a new game.
3		Displays the 3d world with a tank in center.
4		Enables the player to control the tank.

Alternate flow

	Player	**System**
1	Presses any directional key.	
2		Vehicle moves in the corresponding direction.
3		Vehicle collides with a powerup.
4		Player obtains the powerup.
5		Remove the powerup from 3d world

Exceptional flow
N/A

Use Case: Pause game

Summary:
How the player can pause the gameplay.

Priority:
Medium

Extends:
N/A

Includes:
N/A

Participators:
Player

Normal flow of events

	Player	**System**
1	Presses keys “P” or “F10”	
2		Disables the control of the vehicle, game still runs in the background.

Alternate flow

	Player	**System**
1	Presses key “Esc”	
2		Disables the control of the vehicle, game still runs in the background.
3		Shows the main menu

Exceptional flow
N/A

Use Case: Use powerup

Description
A player uses the powerup he has in his slot.

Priority
Low

Extends
N/A

Includes
N/A

Participators
Player

Normal flow of events:

	Player	**System**
1	Presses button for powerup-usage	
2		Powerup removed from powerup-slot
3		Uses powerup on player

Alternate flow
N/A

Exceptional flow
N/A