Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Requirements and Analysis Document for (name of game)

1. Introduction

We want to create an enjoyable 3D game based in VR, with design similarities to the film “Tron Legacy”. The game will be played using a hover based spacecraft confined in a circular course. The spacecraft will be controlled using a seperate joystick and the visual elements will be shown on a phone connected to a google cardboard.
The game will be used as a recreational activity and the main target group will be game enthusiast with access to the necessary items required to run the application.
The game will be available for both mobile devices and computers.
The aim of the game is to generate as much points as possible by surviving in a course with increasing difficulty.

1.2 Definitions, acronyms and abbreviations

App - The application as a whole.
Game - The actual game that the user will play.
World (Värld) - The universe we define our game in.
Camera (Kamera) - A point from which the observers vision is based.
Libgdx - Our game engine that handles both functionality and 3D rendering.
Tron - A sci-fi film we base our design from. Known for its neon blue colors.
Rendering - Plotting of virtual elements from the perspective of our camera.
Enemy - AI controlled, often able to shoot. Can be in the form of a ship.
Obstacle - Objects that is in the way, when things that you lose life from when you get hit
Collidable - A combined name for both enemies and obstacles. Things the ship can get hit by.
Ship - The spaceship that the player will control in the game.
Course - The 3d track in the game that the player’s spaceship will fly in.
Entity - A renderable 3d model that will be visually shown and used in the course.
HUD (Head Up Display) - A layer with information on top of the game. Displays information such as amount of health, score and current power-up.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

2. Requirements

2.1 User interface

Pictures from already created games that we have used to come up with ideas for our own game.

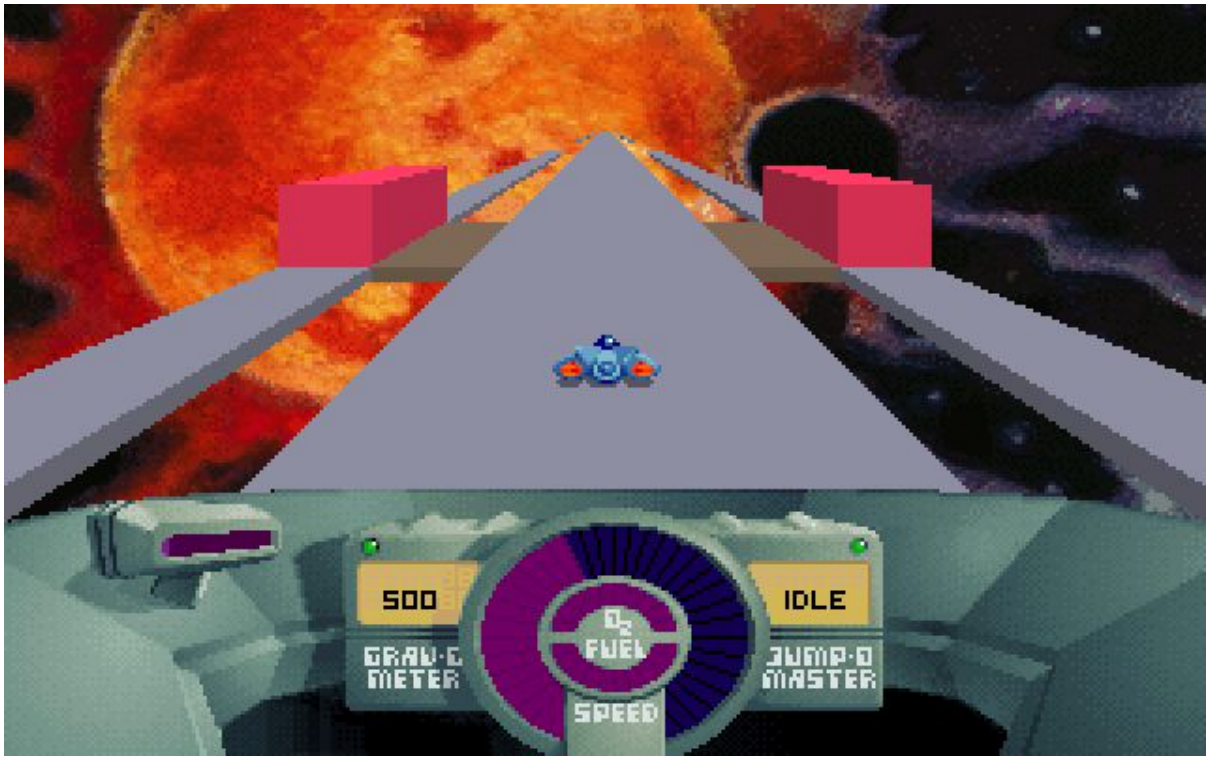
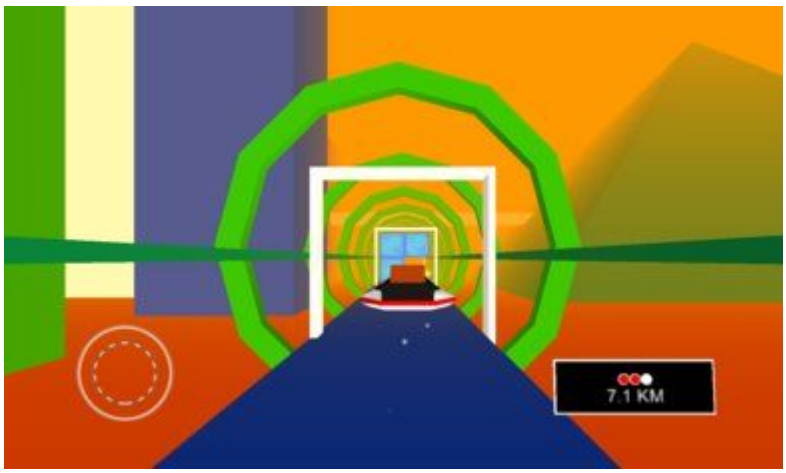


Bild 1: Spaceout,



Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

2.2 Functional requirements

The functionality for the application includes:

1. Start a new game.
2. During a game session a player can
    a. Move the ship around the course.
    b. Use different Power-ups(shield,cannon,nuke) that helps the player.
    c. Collide with different objects and depending on the object different outcomes occur.
    d. Get a score depending on how far the player has come and how many enemies defeated.
3. Pause the game where the user can
    a. Exit the game
    b. Restart the game
    c. Continue playing
4. Exit the application.

Uses cases ranked by priority

- Fly
- Move
- Collide
- Ship crash
- Look around
- Pick up power-ups
- Cannon
- Shield
- Nuke
- Laser eyes
- Extra life
- Mystery power-up
- Start menu
- Pause menu
- App start

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

2.3 Non-functional requirements

Game

Usability

“Galactica - space force of justice”(name in progress) is a computer/mobile game with VR functionality.

Reliability

Performance

Supportability

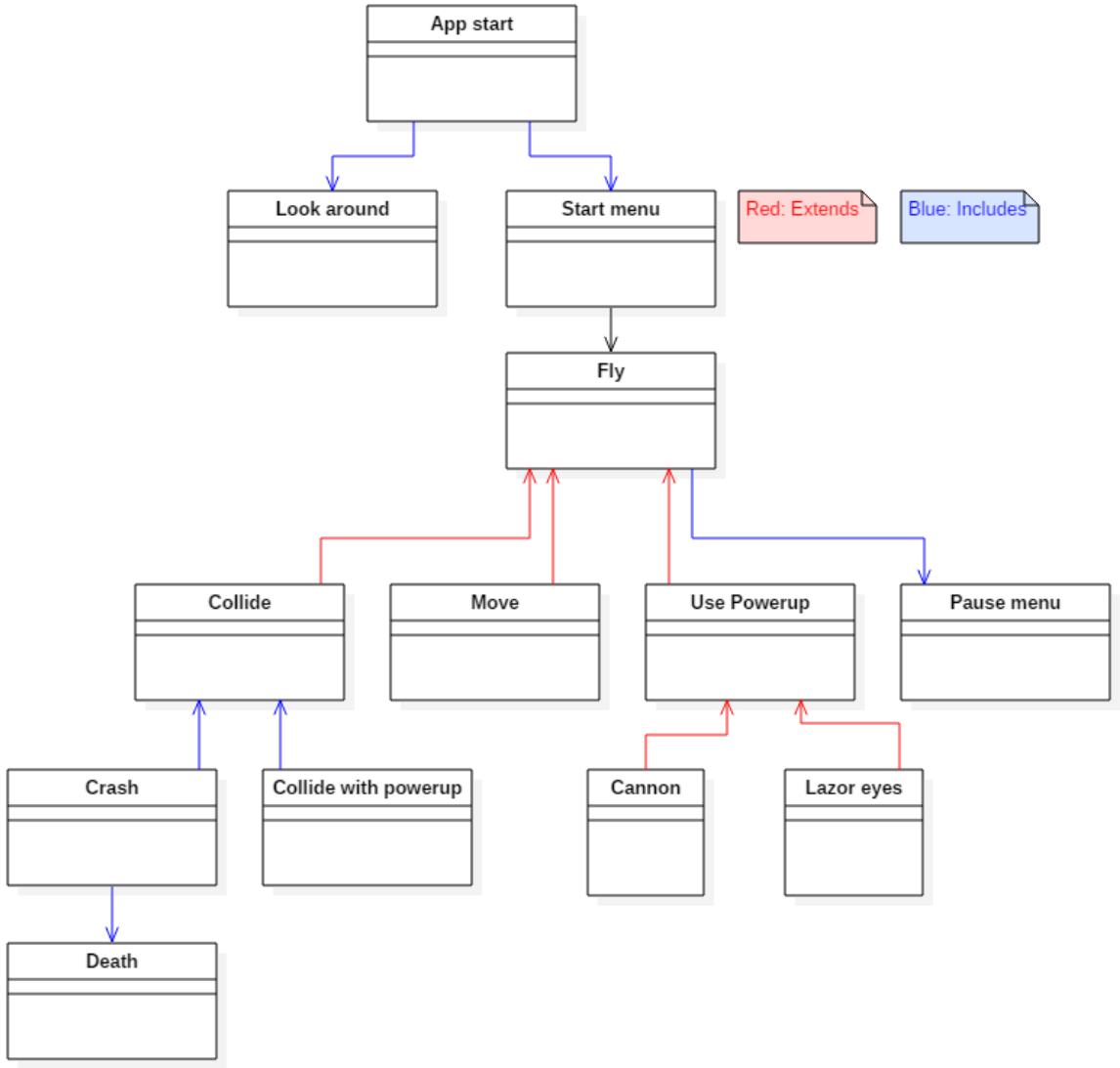
Legal

implementation

Testability

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

3. Use cases listing



Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

3. Use cases

1. Use case: App starts

Summary: Makes sure that user uses the necessary equipment
Priority: low
Extends: none
Includes: Start menu
Participants: Actual player

Normal flow of events:
Controller is connected

	Actor	System
1	User launches app	
2		System checks if controller is connected
3		Controller is connected. Shows prompt to put phone in cardboard.
4		Shows ”Start menu”
		See Start menu

Alternate flow:
Controller isn’t connected, but user has a controller

	Actor	System
1	User launches app	
2		System checks if controller is connected
3		Controller isn’t connected. Shows prompt to connect controller .
4	User connects controller	
5		Controller is connected. Shows prompt to put phone in cardboard.
6		Shows ”Start menu”
		See Start menu

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

2. Use-case: Look around

Summary: The player plays on a device that has access to a gyroscope and that lets' the player to be able to look around in game.
Priority: Medium
Extends:
Includes:
Participants: Player

Normal flow of events

	Actor	System
1	Moves head	
2		The view will rotate towards the user's head direction.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

3. Use case: Start menu

Summary: The user starts the app and then selects an action from the menu
Priority: Mid
Extends: None
Includes: Options menu, Fly
Participants: Actual player

Normal flow of events:
The player starts the game

	Actor	System
1	User marks the “Start” button (either by using the controller or by looking at it)	
2		Highlights ”Start” button
3	User selects the ”Start” button with action button on controller	
4		Starts game
		See Fly

Alternate flows:
The player opens options

	Actor	System
1	User marks the “Options” button (either by using the controller or by looking at it)	
2		Highlights ”Options” button
3	User selects the ”Option” button with action button on controller	
4		Opens ”Options” menu
		See Options menu

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Alternate flows:
The player closes the app

	Actor	System
1	User marks the “Exit game” button (either by using the controller or by looking at it)	
2		Highlights ”Exit game” button
3	User selects the ”Exit game” button with action button on controller	
4		Application closes

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

4. Use-case: Fly

Summary: The player is now piloting the spaceship.
Priority: high
Extends:
Includes: pause menu
Participants: User

Normal flow of events:

	Actor	System
1.1		System initializes the game and displays it for the user. The user starts flying the ship for a brief moment.
1.2		The system hands over the controls of the ship to the user.
1.3	The user now controls the ship and decides what will happen next in the game.	

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

5.Use case: Move

Summary: The user drags the joystick to any direction and the ship moves in that direction
Priority: high
Extends: Fly
Includes:
Participants: Actual player

Normal flow of events:

The player moves the ship.

	Actor	System
1	Drags the joystick in a direction	
1		Turn animation plays.
1		Checks if ship can fly in that direction
		Ship moves in that direction and stops t nd stops when the user stops moving t e joystick.

Alternative flow of events:
The player can't move the ship.

	Actor	System
1	The use drags the joystick in a direction	
2		Turn animation plays. Ship is slightly turned in th e direction of the ship.
3		Checks if ship can fly in that direction
		Stays where it is

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

6. Use case: Collide

Summary: The Player collides with an object.
Priority: Mid
Extends: Fly
Includes: Crash, Power-up
Participants: Actual player

Normal flow of events:
The player collides with an enemy ship.

	Actor	System
1	The user moves the ship into an an enemy using the joystick.	
2		See crash

Alternative flow of events:
The player collides with an obstacle.

	Actor	System
1	The user moves the ship into an an obstacle using the joystick.	
2		See Crash
3		
4		

Alternative flow of events:
The player collides with a power-up that does not require activation.

	Actor	System
1.1	The user moves the ship into an a power-up using the joystick.	See shield/extra life

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Alternative flow of events:
The player collides with a power-up that requires activation.

	Actor	System
1.1	The user moves the ship into an a power-up using the joystick.	See Use Power-up

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

7. Use-Case: Crash

Summary: Player's ship crashes into an obstacle.
Priority: High
Extends:
Includes: Death
Participators: User

Normal flow of events

	Actor	System
1		A life is removed from the ship’s lives.

Alternate flow of events

Flow 2 (Ship has no lives left)

	Actor	System
2.1		The player dies and the game ends. Death user case is shown.

Alternate flow of events

Flow 3 (Ship has shield powerup)

	Actor	System
3.1		Ship loses shield powerup.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

8. Use case: Pick up power-up

Summary: The user picks up a power up after colliding with a power up item

Priority: Medium

Extends: None

Includes: None

Participants: Actual player

Normal flow of events:
The spaceship has collided with a power-up item that doesn't contain a shield or an extra life. The ship isn't already equipped with another power up that isn't a shield.

	Actor	System
1		Power-up animation plays
2		Displays power up in upper right corner (HUD). Spaceship is now equipped with a power up.

Alternate flow of events:
The spaceship has collided with a power-up item that doesn't contain a shield. The ship is already equipped with another power up that isn't a shield.

	Actor	System
1		Power-up animation plays.
2		The previous power up is removed from spaceship. Icon disappears from HUD
3		Displays power up in upper right corner (HUD). Spaceship is now equipped with a power up.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Alternate flow of events:
The spaceship has collided with a power-up item that contains a shield.

	Actor	System
1		Spaceship is now equipped with shield.
		See Shield power-up

Alternate flow of events:
The spaceship has collided with an extra life.

	Actor	System
1		The spaceships gets an extra life.
2		The new life level is displayed up in upper left corner (HUD).

Alternate flow of events:
The spaceship has collided with a mystery power-up.

	Actor	System
1		System randomly selects a power up.
		See *Pick up power up* for this new power up

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.1. Use case: Cannon

Summary: The player uses the Cannon Power-up
Priority: medium
Extends: Use Power-up
Includes:
Participators: Actual player
Normal flow of events:

The player attacks and misses the target.

	Actor	System
1	Clicks the attack button	
2		Fires projectile in the direction of the ship. cannon icon disappears from HUD
3		projectile past the target into the distance.

Alternative flow of events:
The player attacks and hits the target.

	Actor	System
1	Clicks the attack button	
2		Fires projectile in the direction of the ship
3		Hits enemy and destroys it
		Gets points

Alternative flow of events:
The player attacks and hits the wall.

	Actor	System
1	Clicks the attack button	
2		Fires projectile in the direction of the ship. Cannon i on disappears from HUD.
3		The projectile hits the wall and a small explosion happens.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.2. Use case: Shield power-up

Summary: The user has a shield power-up activated, preventing one collision with either a bullet or a part of the terrain.
Priority: Medium
Extends: Use power-ups
Includes:
Participants: Actual player

Normal flow of events:
The player has an active power-up in the form of a shield..

	Actor	System
1		Shows a glowing aura around ship to indicate temporary invincibility.
2	Moves around with ship	
3		The aura, and icon in upper right corner starts blinking before fading out entirely
4		Power-up effect stops after a set amount of time (8s).

Alternative flow of events:
The player has an active power-up in the form of a shield that is used when a player hits an object.

	Actor	System
1		Shows a glowing aura around ship to indicate temporary invincibility.
2	Moves around with ship	
3	Gets hit by an object	
4		Removes both shield, and icon in the top right corner.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Alternative flow of events:
The player has an active power-up in the form of a shield that is used when a player hits an object.

	Actor	System
1		Shows a glowing aura around ship to indicate temporary invincibility.
2	Moves around with ship	
3	Gets hit by a projectile	
4		Removes both shield, and icon in the top right corner.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.3. Use case: Nuke

Summary: The user uses a “Nuke” power up
Priority: Medium
Extends: Use power-ups
Includes:
Participants: Actual player

Normal flow of events:
The player launches a nuke with enemies and/or obstacles on screen

	Actor	System
1		Nuke animation plays.
2		Checks if enemies and/or obstacles on screen
3		All enemies and/or obstacles are destroyed
		Gets points
4		Nuke icon disappears from HUD

Alternate flow of events:
The player launches a nuke without any enemies or obstacles on screen

	Actor	System
1		Nuke animation plays.
2		Checks if enemies and/or obstacles on screen
3		Nuke icon disappears from HUD

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.4. Use case: Laser eyes

Summary: The user interacts with surrounding using laser connected to eyes
Priority: Medium
Extends: Use power-ups
Includes: Look around
Participants: Actual player

Normal flow of events:
The player has an active power-up.

	Actor	System
1		Continuously shoots laser beam that travels in a straight line from the actor's head's position.
2	Moves head to look at target that should be blown up.	
3		The icon in the top right corner starts blinking to indicate that power-up is about to disable.
4		Power-up effect stops after a set amount of time (8s),removing laser effect from head position

Alternative flow of events:
The player has an active power-up and hits an enemy.

	Actor	System
1		Continuously shoots laser beam that travels in a straight line from the actor's head's position.
2	Moves head to look at target that should be blown up.	
3		Laser beam collides with target
4		System removing target, leaving behind debris that is indicating a it
5		he icon in the top right corner starts blinking to indicate that power-up is about to disable.
6		Power-up effect stops after a set amount of time (8s), removing laser effect from head position

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.5. Use case: Extra life

Summary: The user picks up a power-up that gives the player an extra life.
Priority: Medium
Extends: Use power-ups
Includes:
Participants: Actual player
Normal flow of events:

The player has an active power-up in the form of a shield..

	Actor	System
1		Shows a small but noticeable heart flashing next to the life-counter to indicate an extra life..
2		Adds a life to the life-counter

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

9.6. Use case: Mystery power-up

Summary:
Priority: Medium
Extends: Use power-ups
Includes:
Participants: Actual player

Normal flow of events:

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

10. Use case: Pause menu

Summary: The user pauses the game and selects an action from the pause menu
Priority: Mid
Extends: None
Includes: Start menu
Participants: Actual player

Normal flow of events:
The player pauses the game and then continues

	Actor	System
1	User presses the ”pause” button on controller	
2		Game pauses
3		Pause menu is displayed
4	User marks the “Resume” button (either by using the controller or by looking at it)	
5		Highlights ”Resume” button
6	User selects the ”Resume” button with action button on controller	
7		Game resumes

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

Alternate flows:
The player pauses the game and then restarts it

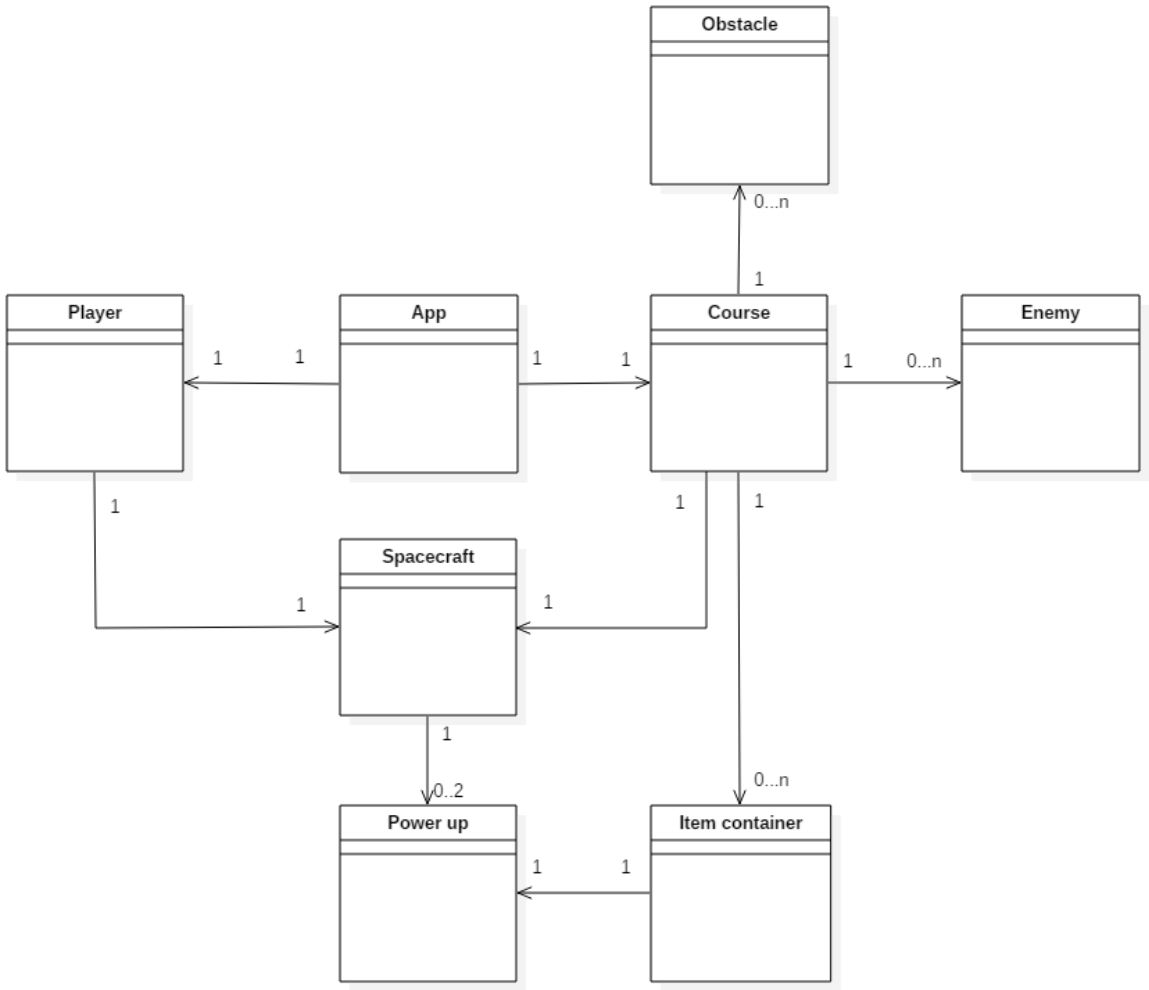
	Actor	System
1	User presses the ”pause” button on controller	
2		Game pauses
3		Pause menu is displayed
4	User marks the “Restart” button (either by using the controller or by looking at it)	
5		Highlights ”Restart” button
6	User selects the ”Restart” button with action button on controller	
7		System ends the game and starts a new one
8		See fly

Alternate flows:
The player pauses the game and then quits it

	Actor	System
1	User presses the ”pause” button on controller	
2		Game pauses
3		Pause menu is displayed
4	User marks the “Quit” button (either by using the controller or by looking at it)	
5		Highlights ”Quit” button
6	User selects the ”Quit” button with action button on controller	
7		System ends the game and return to the start menu
8		See Start menu

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

4. Domain model



4.1 Class responsibilities

“Player” is the actual user.
“Course” handles the tubular track that the spaceship can fly in and all of the other objects that exists in it.
“Obstacle” is a 3d object that will exist in the course that the spaceship can collide with.
“Enemy”(‘s) are computer controlled entities in the course that will try to inflict damage on the player’s spaceship.
“Spacecraft” is an entity in the course that will be controlled by the player.
“Item container” is an entity in the course that holds a powerup.
“Power up” is an ability that the player’s spaceship can use once they the spaceship have picked one up.

Version: 1.0.0
Date: 2017-03-22
Author: Ludvig Andersson

5. References