Requirements and Analysis Document for Not enough space, 20.

Version:
Date:
Author:

This version overrides all previous versions.

1 Introduction

Background explaining why this application is needed (besides mandatory in course). What's the problem addressed (use imagination)? What will it do? Who will benefit/use from this application? In what situation will the application be used? Define the application. General characteristics of application.
Not enough space

Ett arkadspel i två delmoment där spelaren styr ett ufo för att samla kossor till sin hemplanet.

Delmoment 1: Plocka
Ufot hovrar över planeten och använder sin tactor-beam för att stråla upp kossor. Som hinder kan spelaren råka plocka upp bråte och stenar som tynger ner farkosten. Kossor kan springa för sitt liv och gömma sig i ladugårdar.
Spelaren styr med fyra piltangenter och använder en knapp för att aktivera strålen. Spelaren har en begränsad tid på planeten.

Delmoment 2: Frakta
Flygning med 2D-navigering genom ett asteroidfält för att nå hemplaneten.
Ufot kan styra i alla fyra riktningar samt rotera för att undvika asteroiderna.
Vid liten krock tappar ufot en del av kossorna samt tar lite skada. Vid helkrock exploderar skeppet.
Vikten av lasten gör att styrningen blir segare.

1.2 Definitions, acronyms and abbreviations

Spel - en runda i spelet.
Program - själva programmet.
Fas/steg - används synonymt och syftar på första och andra delen av spelförloppet, under en runda.
Traktorstråle - “Tractor beam” från engelska, alltså en klassisk UFO-dragningsstråle som används för att kidnappa människor eller liknande.

2 Requirements

2.1 User interface

Sketches, drawings and explanations of the application user interface (possible navigation).

2.2 Functional requirements

Delmoment 1 Delmoment 2 Pausmeny Startmeny

1. Navigera 1 - S
2. Aktivera traktorstråle - P
3. Avaktivera traktorstråle - P
4. Stråla - P
5. Fånga - P
6. Klara spel 1 - P
7. Starta spel - F
8. Avsluta program - F
9. Ändra inställningar J
10. Pausa spel - F
11. Återuppta spel J
12. Avbryt spel J
13. Starta om spel J
14. Navigera 2 - S
15. Krocka - F
16. Förlora spel J
17. Klara spel 2 - S
18. Fasövergång - F
19. Skjuta - S
20. Maskhål - S

2.3 Non-functional requirements

Programmet ska gå att köra på datorer utan kraftfullt grafikkort.

3 Use cases

An UML use case diagram

3.1 Use case listing Use case texts (using the use case template)

Use Case: Navigera 1

Summary: The actor is navigating her ship in phase 1.

Priority: high
Extends: -
Includes: -
Participators: Actor, System

NOTE: The keys called “the arrow keys” below don’t have to be the actual arrow keys of the keyboard, but will correspond to the usual purpose of these.

Normal flow of events

Actor moves the ship sideways/back- and forwards with the arrow keys. (The movement described is relative to the surface of the planet, so each movement is actually along a sphere outside of the planet’s surface.)

	Actor	**System**
1	Presses one or more of the arrow keys.	
2		Ship object is moved in the (possibly combined) direction of the arrow(s) that is(/are) pressed. The ship is also tilted in this direction. It accelerates quickly up to top speed. The ship continues to move as long as the user keeps the key(s) pressed. While moving a gentle buzz sound is constantly played.

Normal flow of events 2

Actor stops moving the ship by releasing the arrow key(s).

	Actor	**System**
1	Releases all the arrow keys.	
2		Ship object tilts upright and slows down to a complete halt.

Normal flow of events 3

Actor rotates the ship (around a local y-axis) with the rotation keys.

	Actor	**System**
1	Presses one of the rotation keys	

	(LEFT or RIGHT).	
2		Ship object is rotated in the direction corresponding to the users key press. It continues to rotate as long as the user keeps the key pressed. While rotating a buzz sound (different to that playing while moving as described above) plays.

Normal flow of events 4

Actor stops rotation of the ship by releasing the currently pressed rotation key.

	Actor	**System**
1	Releases the rotation key.	
2		The ship object's rotation decelerates quickly until it comes to a complete stop.

Use Case: Activate beam

Priority: High
Extends: Play round
Includes: Use beam
Participators: Actual player

Normal flow of events

	Player	**System**
1	Presses spacebar	
2		Starts playing "sonic" sound and beam becomes visible
		See **Use beam**

Use Case: Use beam

Summary: The user presses spacebar to beam cows below the spaceships
Priority: High
Extends: Play Round
Includes: Activate beam, Deactivate beam
Participators: Actual player

Normal flow of events:

The beam is activated while no object is in range.

	Actor	System
1	Activate beam	
2		Check for objects below spaceship within reach of beam
3		No object found. Beam still active.

Alternate flow:
The beam is active while object is in range, object is picked up.

	Actor	System
1	Activate beam	
2		Check for objects below spaceship within reach of beam
3		Object within range of beam is slowly lifted towards spaceship
4		Object reaches spaceship and is added to “storage”. HUD is updated with flashing new values of items and total weight. Beam still active.

Use Case: Deactivate beam

Priority: High
Extends: Play round
Includes: Use beam
Participators: Actual player

Normal flow of events
Empty beam is deactivated

	Player	System
1	Releases spacebar	
2		Stops playing “sonic” sound and beam turns invisible.
3		Checks for any object still hanging in mid-air.
		No objects in mid-air.

Alternate flow:
Beam carrying objects is deactivated

	Player	System
1	Releases spacebar	
2		Stops playing “sonic” sound and beam turns invisible.
3		Checks for any object still hanging in mid-air.
		Objects in mid air is released and fall to ground

Use Case: Klara spel

Priority: Mid
Extends: Play round
Includes: -
Participators: System only

Normal flow of events

	Player	**System**
1		Time reaches zero.
2		Navigation and beam is deactivated.
3		Play “homeworld” animation where cows are piled on alien planet.
		Score is displayed on High-Score list and endgame GUI is presented. (Replay, options, exit)

Use Case: Starta spel

Summary: The user clicks the “Start” button in the main menu.
Priority: mid
Extends: -
Includes: -
Participators: User

Normal flow of events

	Actor	**System**
1	Clicks “Start” button.	
2		Main menu is removed and stage 1 is displayed.

3		Countdown?
4		Timer starts and actor gains control of the ship.

Use Case: Avsluta program

Summary: The user clicks the “Exit” button in the main menu.
Priority: mid
Extends: -
Includes: -
Participators: User

Normal flow of events

	Actor	**System**
1	Clicks “Exit” button.	
2		Program shuts down.

Use Case: Pausa spel

Summary: The user is in-game and presses the pause key (usually Esc).
Priority: mid
Extends: -
Includes: -
Participators: User

Normal flow of events

	Actor	**System**
1	Presses pause key.	
2		Actor loses control, timer pauses and AI/Physics are stopped.
3		Pause menu is presented in the middle of the screen. Screen around the pause menu is dimmed.

Use case: Fasövergång

Summary: Stage 1 timer reaches 0:00
Priority: low
Extends: Starta spel
Includes: -
Participators: System?

Normal flow of events:

	Actor	**System**
1		Actor loses control of ship.
2		“Alert” sound plays.
3		Ship moves and rotates in an arc and ends up facing the asteroid field. Camera follows normally.
4		Actor regains control of ship.

Use case: Exit game

Summary: The player chooses to exit the game in the middle of a round
Priority: High
Extends: Start game, Pause game
Includes: -
Participators: Player, System

Normal flow of events:

	Actor	**System**
1	The player presses exit or the cross in the top right corner	
2		A menu becomes visible with the alternative “exit, resume game, restart game”.
3	Player presses exit	
4		Dialogue saying “Are you sure you want to exit?”
5	Player presses “yes”	

6		Game exits and the last screen with high score and option to restart appears

Alternate flow:

	Actor	**System**
1	The player presses exit or the cross in the top right corner	
2		A menu becomes visible with the alternative “exit, resume game, restart game”.
3	Player presses exit	
4		Dialogue saying “Are you sure you want to exit?”
	Player presses “no”	See use case for resume game or restart game.

Use case: Resume game

Summary: The player chooses to resume the game after pausing
Priority: middle
Extends: Start game, Pause game
Includes: -
Participators: System, Player

Normal flow of events:

	Actor	**System**
1	Player presses “resume game”	
2		The game screen where the player paused the game appears
3		Some seconds delay
4		The play resumes from where it was paused

Use case: Restart game

Summary: The player chooses to restart the game from the beginning
Priority: Middle
Extends: Start game, Pause game, Exit game
Includes: -
Participators: System, Player

Normal flow of events:

	Actor	**System**
1	Player presses “Restart”	
2		Dialogue: “Are you sure you want to restart?”
3	Player presses “Yes”	
4		The start screen of the game appears
		Some seconds delay
		The game starts

Use Case: Navigera 2

Summary: The actor is navigating her ship in phase 2.
Priority: low
Extends: -
Includes: Krocka
Participators: Actor, System

NOTE: The keys called “the arrow keys” below don’t have to be the actual arrow keys of the keyboard, but will correspond to the usual purpose of these.

Normal flow of events

Actor moves the ship sideways/up- and downwards with the arrow keys.

	Actor	**System**
1	Presses one or more of the arrow keys.	
2		Ship object is moved in the (possibly combined) direction of the arrow(s) that

		is(/are) pressed. It accelerates quickly up to top speed. The ship continues to move as long as the user keeps the key(s) pressed. While moving a gentle swoosh sound is constantly played.

Normal flow of events 2

Actor stops moving the ship by releasing the arrow key(s).

	Actor	**System**
1	Releases all the arrow keys.	
2		Ship object slows down to a complete halt.

Normal flow of events 3

Actor rotates the ship (around a local z-axis) with the rotation keys.

	Actor	**System**
1	Presses one of the rotation keys (LEFT or RIGHT).	
2		Ship object is rotated in the direction corresponding to the users key press. It continues to rotate as long as the user keeps the key pressed. While rotating a swoosh sound (different to that playing while moving as described above) plays.

Normal flow of events 4

Actor stops rotation of the ship by releasing the currently pressed rotation key.

	Actor	**System**
1	Releases the rotation key.	

2		The ship object's rotation decelerates quickly until it comes to a complete stop.

Alternate flow of events

The ship's movement results in it hitting an asteroid.

	Actor	**System**
1	Moves the ship so that it hits an asteroid.	
2		**Krocka**

Use Case: Krocka

Summary: The user's ship hits an asteroid in phase 2.
Priority: low
Extends: Navigera 2
Includes: Förlora spel
Participators: System

Normal flow of events

Ship collides with the asteroid head-on.

	Actor	**System**
1		Actor loses control of the ship.
2		Ship object is removed and explosion animation plays.
3		**Förlora spel**

Alternate flow of events

Ship collides with the asteroid from the side, takes damage but doesn't run out of health.

	Actor	**System**
1		Health bar goes down slightly and becomes red. Fades back to normal color over 1 second.
2		Cow counter goes down by a small number,

		but doesn't go below 0. Number becomes white and fades back to normal color over 1 second.

Alternate flow of events 2
Ship collides with the asteroid from the side, takes damage and runs out of health.

	Actor	**System**
1		Health bar goes down to 0.
2		Actor loses control of ship.
3		Ship object is removed and explosion animation plays.
4		**Förlora spel**

Use case: Lose game

Summary: The player runs out of life and loses the game
Priority: low
Extends: Start game, Crash?
Includes: -
Participators: System

Normal flow of events:

	Actor	**System**
1		Lifebar turns red and start blinking to signal that the player is about to run out of life
2	The player touches/ crashes into an asteroid	
3		Life bar gets empty
4		Dialogue pops up saying: "Game over"
		The exit-screen appears with the high score and the alternatives: "Restart" and "Exit" game"

Use Case: Klara spel 2

Summary: The user's ship successfylly gets to the end of phase 2 and releases the cows on its home planet.
Priority: low
Extends: -
Includes: Highscore list
Participators: System

Normal flow of events

Ship reaches the end of the asteroid belt with at least one cow and releases it's goods onto its home planet.

	Actor	**System**
1		Ship reaches the end of the asteroid belt.
2		Winning animation is played (something like: new view where the ship hovers over its home planet, opens its hatch and releases all of its cows onto an already huge pile of cows on the surface).
3		**Highscore list**

Alternate flow of events

Ship reaches the end of the asteroid belt with no cows left, gets destroyed and loses the game.

	Actor	**System**
1		Ship reaches the end of the asteroid belt.
2		Losing animation is played (something like: new view where the ship hovers over its home planet, opens its hatch to release some cows but doesn't have anything to release and then gets destroyed by a destroyer beam from the very angry control tower).
3		Losing screen is shown telling the user that she lost and asking her if she wants to try again or go back to the main menu.

Use case: Change the settings of the game

Summary: The player chooses to change the settings in the game menu
Priority: low
Extends: Start game, Pause game
Includes: All alternative use cases for game settings
Participators: System, Player

Normal flow of events:

	Actor	**System**
1	Player presses “Change settings”	
2		A menu with the alternative: “Change resolution”, “Change language”, “Turn off sound” etc. appears
3	The player presses “Return”	
4		Start menu screen re-appear

Use Case: Skjuta

Summary: The user’s ship shoots some of it’s goods out in phase 2.
Priority: low
Extends: Navigera 2
Includes: -
Participators: Actor, System

Normal flow of events

Ship shoots out it’s most recently gathered item and it hits an asteroid..

	Actor	**System**
1	Presses shoot button.	
2		The item on top of the UI item stack gets bigger and then shrinks to nothing to signal that it no longer is on the stack. All the other items get moved up one step.
3		The item gets shot out from the front of the

		ship, moving forward with great speed. An appropriate sound (e.g. a “MOOO” if the item is a cow) is played.
4		The item hits an asteroid. Both the item and the asteroid get destroyed. (Visualised as the object disappearing in a small explosion while the asteroid gets divided into many small stones scattering in all directions.)

Alternate flow of events

Ship shoots out it’s most recently gathered item and it doesn’t hit anything..

	Actor	**System**
1	Presses shoot button.	
2		The item on top of the UI item stack gets bigger and then shrinks to nothing to signal that it no longer is on the stack. All the other items get moved up one step.
3		The item gets shot out from the front of the ship, moving forward with great speed. An appropriate sound (e.g. a “MOOO” if the item is a cow) is played.
4		No asteroid is in the path of the item, so it just gradually fades away in the distance.

Exceptional flow of events

Ship tries to shoot out it’s most recently gathered item but doesn’t have any items stored.

	Actor	**System**
1	Presses shoot button.	
2		A sound indicating an empty storage is played.

Use Case: Maskhål

Summary: The user’s ship travels through a wormhole during phase 2.
Priority: low

Extends: Navigera 2
Includes: -
Participators: Actor, System

Normal flow of events

The ship enters a wormhole and gets teleported to the corresponding “out” wormhole.

	Actor	**System**
1	The actor navigates the ship so that it enters a wormhole.	
2		The ship disappears into the wormhole while a teleporting sound is played.
3		The whole view very quickly moves to the corresponding “out” wormhole and the ship reappears there (coming out of this wormhole).
4		**Navigation**

4 Domain model

An UML class diagram.

4.1 Class responsibilities

Explanation of responsibilities of classes in diagram

5 References