Requirements and Analysis Document for group 16

Erik Sjöström, Filip Labe, Jonatan Källman, Sarosh J. Nasir

March 23, 2017
v1.0

1 Introduction

We have indentified a huge gap in the world of clicker games, namely Lord of the Rings based clicker games. Our project aims to fill that gap.
This game will simulate the adventure of Frodo and friends on their quest to destroy the one ring, and will benifit anyone who likes both LOTR and clickers. This game might be useful on the bus, to pass the time when moving between locataions, or when sitting on the toilet.

1.1 Definitions, acronyms, abbriviations

LOTR: Lord of the Rings.

2 Requirements

2.1 User interface

2.2 Functional requirements

2.3 Non-functional requirements

3 Use cases

3.1 Use case listings

3.1.1 Use case: Attack

Summary: The user clicks the screen and the game calculates new health for the monster.
Priority: High
Extends: undefined
Includes: undefined

Participators: Player
Normal flow of event: The player attacks the monster and the monster survives.

	Actor	System
1	Player clicks screen	
2		Attack animation plays
3		Calculate new health for monster

Alternate flow: The player attacks the monster and the monster dies. The level is not cleared.

	Actor	System
1	Player clicks screen	
2		Attack animation plays
3		Calculate new health for monster
4		Health ≤ 0
5		Death animation plays
6		Replace dead monster with new monster

Alternate flow: The player attacks the monster and the monster dies. The level is cleared.

	Actor	System
1	Player clicks screen	
2		Attack animation plays
3		Calculate new health for monster
4		Health ≤ 0
5		Death animation plays
6		Replace dead monster with new monster
7		The option to change level is made avliable

3.1.2 Use case: Buy upgrade

Summary: The user tries to buy an upgrade.
Priority: Medium
Extends: undefined
Includes: undefined
Participators: Player
Normal flow of event: The player clicks on the upgrade and has enough money.

	Actor	System
1	Player clicks on the upgrade	
2		Checks that the player has enough money to buy the upgrade
3		The upgrade is applied to the player

Alternate flow: The player clicks on the upgrade but does not have enough money.

	Actor	System
1	Player clicks on the upgrade	
2		Checks that the player has enough money to buy the upgrade

3.1.3 Use case: Open map

Summary: The player clicks the button labeled ”map” Priority: Medium
Extends: undefined
Includes: undefined
Participators: Player
Normal flow of event: The player clicks the button labeled ”map”

	Actor	System
1	Player clicks on the button	
2		Displays map page

3.1.4 Use case: Use map

Summary: The user tries to move to a different level.
Priority: High
Extends: undefined
Includes: undefined
Participators: Player
Normal flow of event: The player clicks on the level they want to move to, the level is unlocked.

	Actor	System
1	Player clicks on the level	
2		Checks that the level is unlocked
3		Loads new level

Alternate flow of event: The player clicks on the level they want to move to, the level is locked.

	Actor	System
1	Player clicks on the level	
2		Checks that the level is unlocked

3.1.5 Use case: Show stats

Summary: The use clicks the button labeled ”Stats”
Priority: low
Extends: undefined
Includes: undefined
Participators: Player
Normal flow of event: The player clicks the button labled ”Stats”

	Actor	System
1	Player clicks on the button	
2		Displays stats page

3.1.6 Use case: Go home

Summary: The user wants to go home
Priority: Low
Extends: undefined

Includes: undefined
Participators: Player
Normal flow of event: The player clicks the button labeled ”Home”

	Actor	System
1	Player clicks on the button	
2		Displays the player home screen

3.1.7 Use case: Home improvements

Summary: The user wants to improve their home
Priority: low
Extends: undefined
Includes: undefined
Participators: Player
Normal flow of event: The player wants to buy an animal, has enough money and enough space.

	Actor	System
1		Checks that player has enough money
2		Checks that player has enough space
3		Enables button.
4	Player clicks on a buy animal button	
5		Deduct money from player
6		Add animal to home

Alternate flow: The player wants to buy an animal, does not have enough money

	Actor	System
1		Checks that player has enough money
2		Checks that the player has enough space
3		Does not enable button

Alternate flow: The player wants to buy an animal, does not have enough space

	Actor	System
1		Checks that player has enough money
2		Checks that the player has enough space
3		Does not enable button

Alternate flow: The player wants to buy more space, has enough money.

	Actor	System
1		Check that the player has enough money
2		Enables button.
3	Player clicks on button	
4		Deduct money from player
5		Add more space to player

Alternate flow: The player wants to buy more space, does not have enough money.

	Actor	System
1		Check that the player has enough money
2		Does not enable button.

Alternate flow: The player wants to sell an animal, has animals.

	Actor	System
1		Checks that player has animals
2		Enable button
3	Player clicks button	
4		Remove one animal from player
5		Add money to player

Alternate flow: The player wants to sell an animal, has no animals.

	Actor	System
1		Checks that player has animals
2		Does not enable button

4 Domain model

5 References