Requirements and Analysis Document for Group 04

Version: 1.0.0

Date: 30/3-17

Author: Hanna Carlsson, Tobias Lindgren, Sara Kitzing, Madeleine Lexén

1 Introduction

This application's purpose is to get more young people interested in programming and learning how to code. To make the whole process less daunting, and to give an easy start to the journey. The issue at hand is the high threshold to begin coding as a complete beginner. It can be hard to get started and there is a growing demand for programmers. The point is to develop younger people's interest in programming, and give them a way to get started.

The beneficiary of the application will be youth who might be interested in learning how to code. In the long term it will also benefit companies and educational institutions within computer science. The application will primarily be used as an educational tool during leisure time, but it does not hinder it from being used in school as well.

The application will be similar to a game, as in that it will have several levels in a hierarchy where the user will have to complete a series of steps to reach the next level. These steps will be in form of questions or code assignments.

1.2 Definitions, acronyms and abbreviations

Key: The correct answer

Level: A subcategory within a main category. Contains information about the specific topic and a question on the topic.

Map: The start page, a map over the different main categories

Read more: A page which shows information about basic programming in text form that the user will learn throughout the different worlds. It can be accessed from the menu in all views.

Boss: The last level in a main category which summarizes what the previous levels has taught the user.

Fill in the blanks: A question where the user will get a number of sentences with blanks that they will have to fill in themselves.

Drag n' drop: A question where the user will be presented with a number of different code segments and will have to put them in the right order.

Write code: A question where the user will get a task where they need to write code

2 Requirements

2.1 User interface

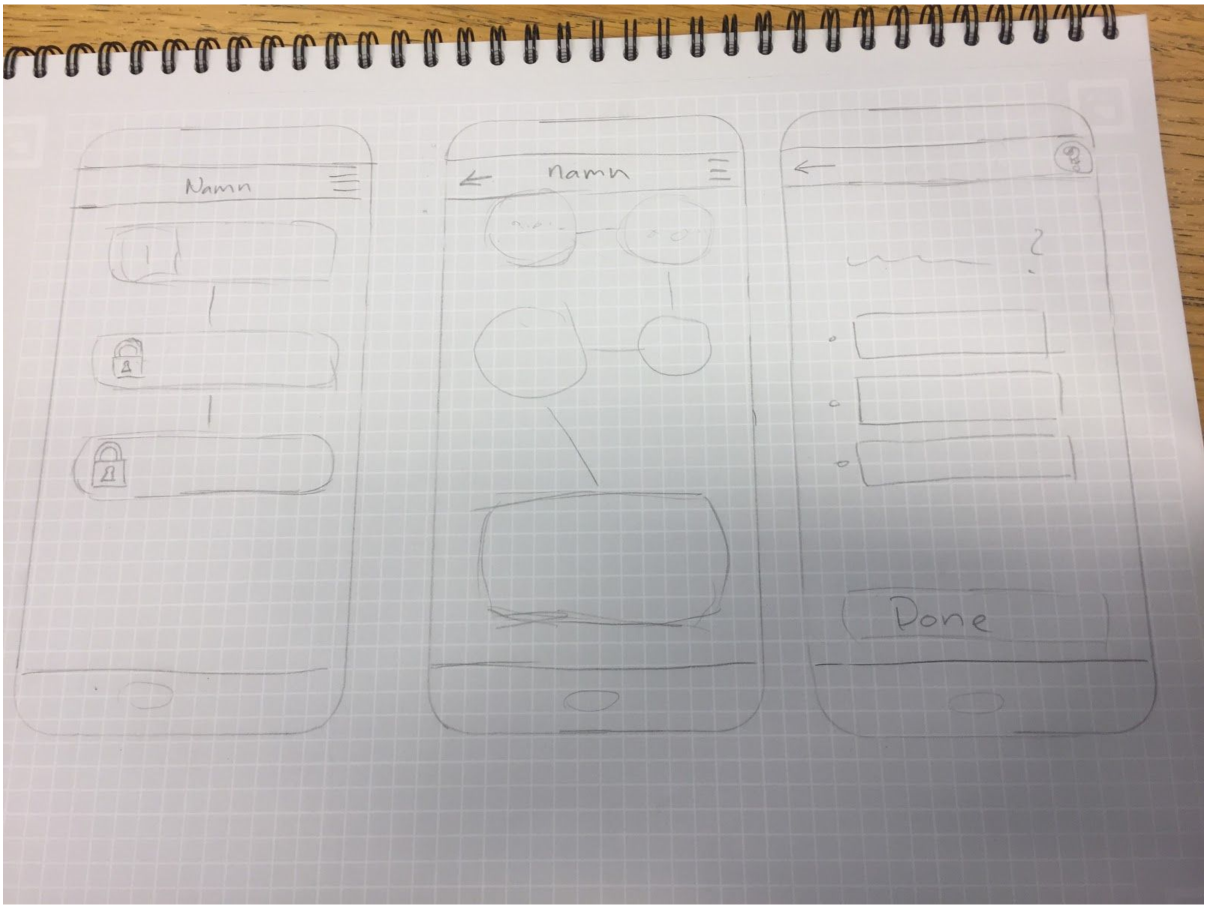


Figure 1. The left view is the start page of the app. Shows the main map of the different worlds. The middle view is the levels view, shown when clicked on a world. This shows the different steps you need to complete to get to the next world. The right view is shown when clicked on a level. This displays a question you need to answer correctly in order to pass the level.

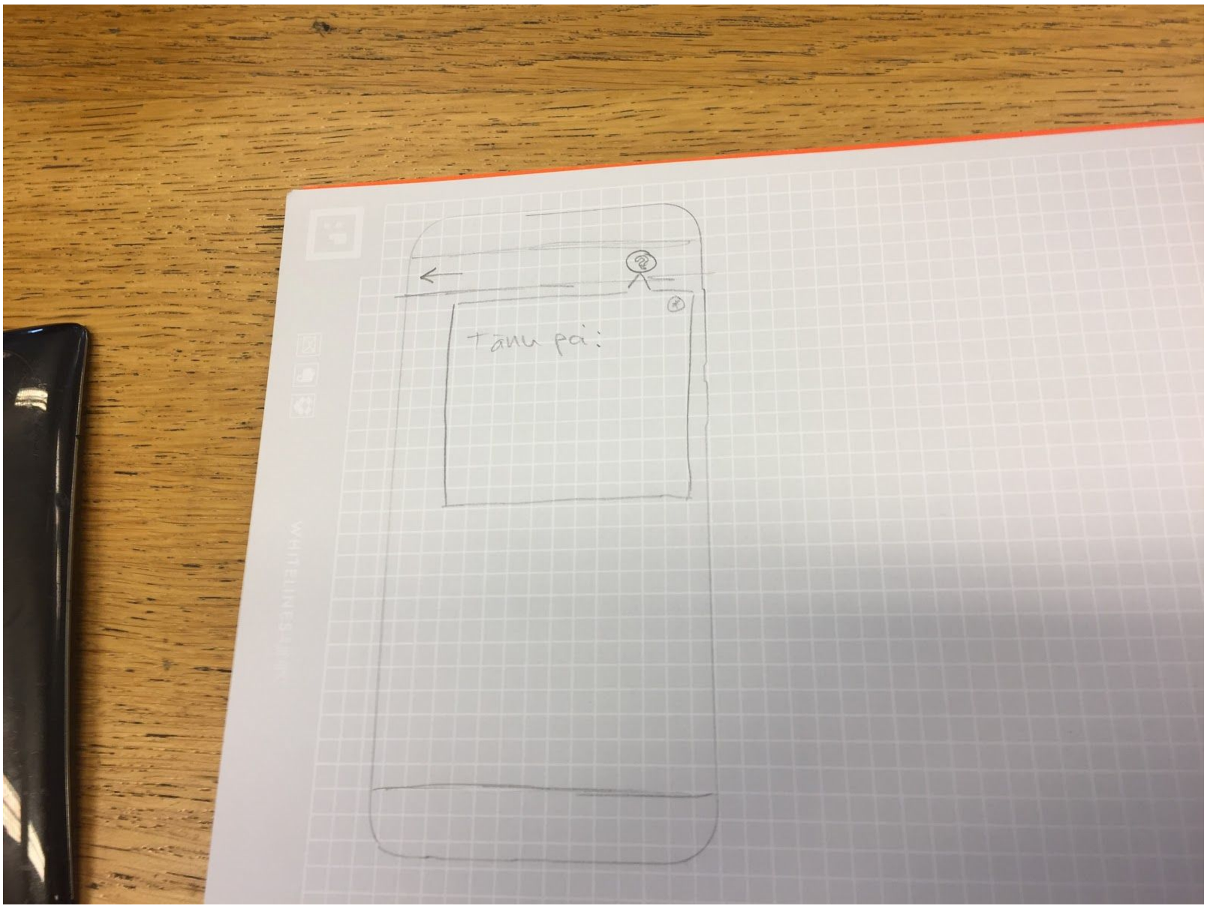


Figure 2: The hint view. Shown when you click on the question mark in the right corner.

2.2 Functional requirements

The user can advance through the main categories by unlocking them. This is done by completing every level in previous main categories, which unlocks the next main category. The levels themselves consist of one or more assignments and at the end of the level there is a test that the user has to pass. If the user struggles with the answer to an assignment there is a hint, that guides the user to the answer. If the user still isn't able to find the right answer, there will be an option to show the key to the assignment.

The assignments are constructed accordingly; first there is an explanation of the issue at hand, then follows a question testing if the user understood the information. These test can take on one of four forms, which are fill-in-the-blanks, multiple choice, write code and drag and drop.

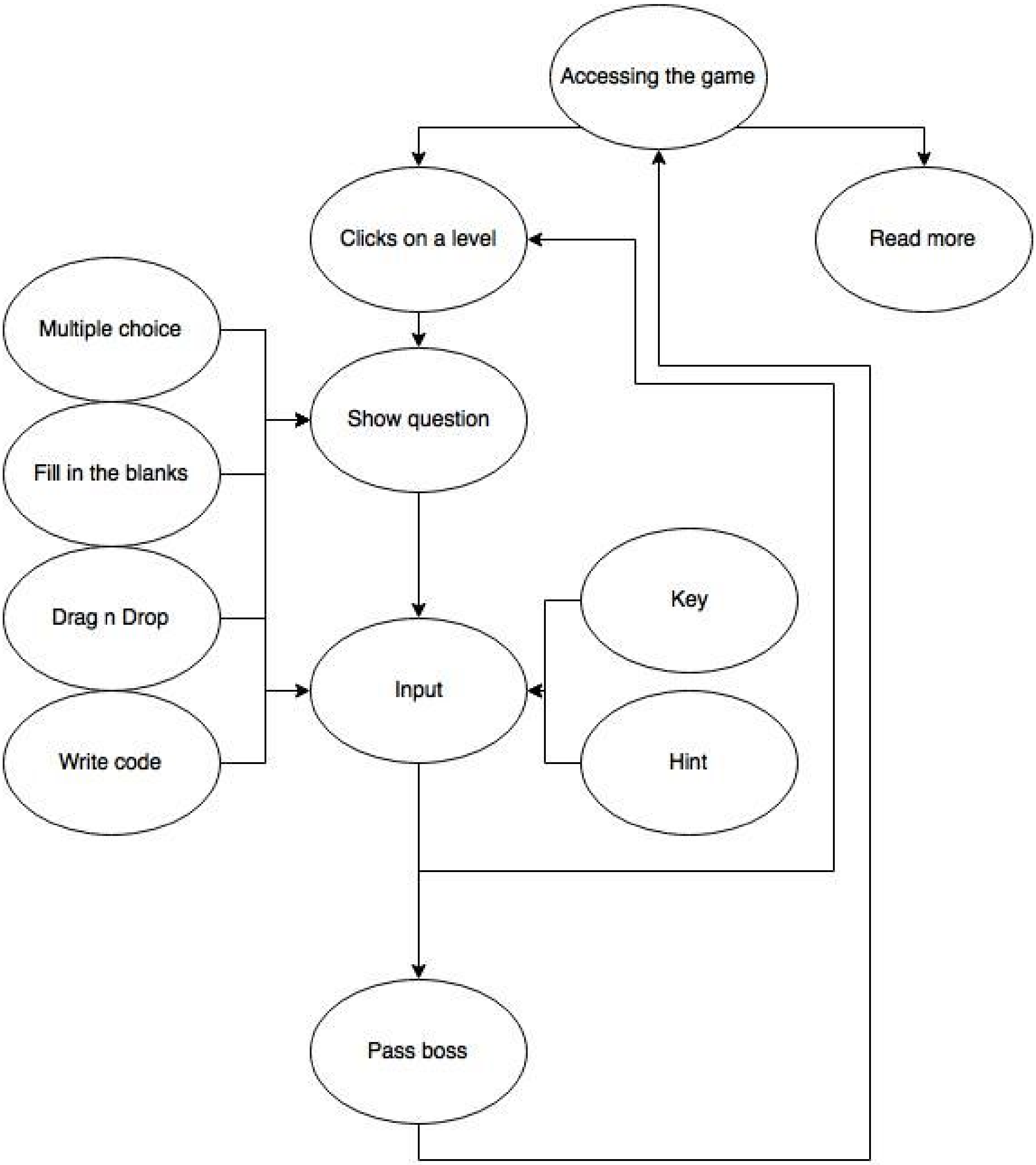
There will also be a section where the user can read more about what it is learning about, with a vocabulary and texts explaining the subjects further.

1. Accessing the game
2. Input
3. Hint
4. Show the correct answer
5. Passes the boss
6. Read more

2.3 Non-functional requirements

The application should be able to run on all android smartphone devices which run on an OS Lollipop 5.0 or higher.

3 Use cases



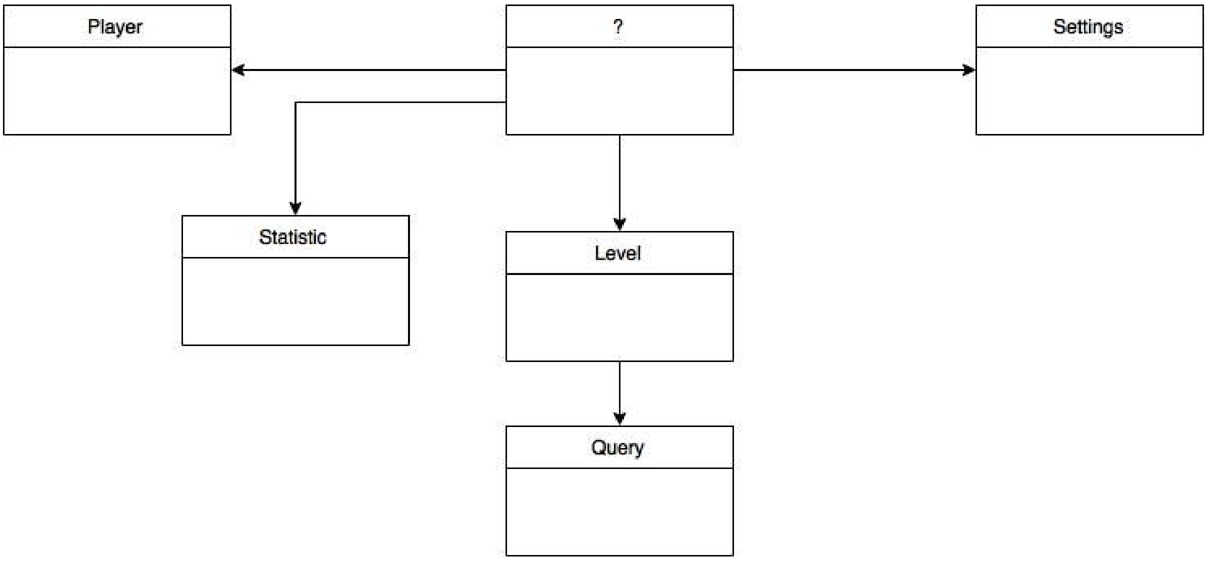
3.1 Use case listing

- Accessing the game

- Read more
- Clicks on level
- Show question
    - Multiple choice
    - Fill in the blanks
    - Drag n’ drop
    - Write code
- Input
- Passes the boss
- Key
- Hint

The full user cases can be found in the appendix

4 Domain model An UML class diagram .



4.1 Class responsibilities

Main: Is the overall representation of the game.
World map: The map containing the different worlds, the first view, handles the different worlds.
World: The map containing the different levels.
Level: The object containing the assignment.
Assignment: Contains a type of question, a hint and the key. It gives the question, receives an answer and checks if it is correct.
Info: Information about how the assignment is supposed to work.
Hint: The object containing tips of how to solve the assignment.
Key: Contains the answer
Menu: Contains the settings and the Read more.
Read more: Contains a longer file with explanation of keywords of programming.

Drag n Drop: Contains the structure for the drag n drop questions.
Fill in the blanks: Contains the structure for the fill in the blanks questions
Write code: Contains the structure for the write code questions.
Multiple Choices: Contains the structure for the Multiple choices questions.
Settings: Contains settings of the app such as volume.

5 References

6 Appendix

6.1.1

Use case: Accessing the game

Summary: The user opens the app and wants to play the "game"
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The player enters the right answer.

	Actor	System
1	Opens the app	
2		Shows the main page (the game's main map)
3.1	Clicks on a unlocked "world"	
3.2		Alternative flow 3.2
4.1	Clicks on the menu	

4.2		Shows the menu in the same window
5.1	Clicks on a locked “world”	
5.2		Nothing happens, you can’t click on a locked “world”

Alternate flow

Flow 3.2 (Clicks on unlocked “world”)

The player clicks on a unlocked “world”.

	Actor	System
1		Shows next screen with the different levels within the “world”
2.1	Clicks on a level	
2.2		Display the levels questions in a new view
2.3		See user case: Input

6.1.2 Use case: Read more

Summary: The user wishes to read more about a subject than is disclosed in the excersizes/challenges.
Priority: low
Extends: (other UC)

Includes: (other UCs)
Participators: The user

Normal flow of events

	Actor	System
1	Klicks “Read more”-button	
2		Shows read more page

6.1.3

Use case: Clicks on level

Summary: The user clicks on an unlocked level
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participants: The user

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1	Clicks on “level”-button	
2		Shows relevant info for the assignment
3	Clicks on “next”-button	
4		See user case show question

Alternate flow

Flow 2 (add heading)

User clicks on “back”-button

	Actor	System
1	Clicks on “back”-button	
2		Shows level map

6.1.4

Use case: Show question

Summary: The program shows the question depending on which type it is
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participants: The program

Normal flow of events

The program shows relevant question

	Actor	System
1		Shows relevant question (see use case: multiple choice, fill in the blanks, drag n’ drop, write code)

6.1.4.1

Use case: Multiple choice

Summary: The type of question where you answer a question with a pre-written answer
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1		Shows relevant question with multiple choices
2	See user case input	

6.1.4.2

Use case: Fill in the blanks

Summary: The type of question where you get a pre-written answer with blanks to fill in
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1		Shows relevant sentence with blanks
2	See user case input	

6.1.4.3

Use case: Drag n’ drop

Summary: The type of question where you get a number of pre-written code segments and have to put them in the correct order to execute the assignment.
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1		Shows relevant question with multiple choices
2	See user case input	

6.1.4.4

Use case: Write code

Summary: The type of question where you get an assignment which you answer by writing code by yourself
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The user clicks on an unlocked level

	Actor	System
1		Shows relevant assignment
2	See user case input	

6.1.5

Use case: Input

Summary: The user enters the an answer and the app acts upon it.
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The player enters the right answer.

	Actor	System
1	Enters an answer	

2	Clicks “check answer”-button	
3		Checks if the answer is correct
4		The answer was right. Plays animation which tells the user it was correct.
5		Gives opportunity to go to the next level or quit.

Alternate flow

Flow 2 (add heading)

The player enters the wrong answer, at least once.

	Actor	System
1	Enters an answer	
2	Clicks “check answer”-button	
3		Checks if the answer is correct
4		The answer was wrong. Plays animation which tells the user to try again.
5	Enters answer	
6	Clicks “check answer”-button	
7		Checks if the answer was correct
8.1		The answer was right. Normal flow of events

8.2		The answer was wrong. Alternative flow 8.2
9	Enters answer	
10	Clicks “check answer”-button	
11		Checks if the answer was correct
12.1		The answer was right. Normal flow of events
12.2		The answer was wrong. Alternative flow 12.2
13	Enters new answer	
14	Clicks “check answer”-button	
15		Checks if the answer was correct.
16		The answer was right. Normal flow of events

Flow 8.2 (add heading)

The user enters the wrong answer twice.

	Actor	System
8.2.1		Gives opportunity to get a hint.
8.2.2		User case get hint

Flow 12.2 (add heading)

The user enters the wrong answer more than twice.

	Actor	System
12.2.1		Gives opportunity to get

		the key.
12.2.2		User case show answer

6.1.6

Use case: Passes the Boss

Summary: The user passes the Boss
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The player enters the right answer.

	Actor	System
1	Enters an answer	
2	Clicks “check answer”-button	
3		Checks if the answer is correct
4		The answer was right. Plays animation which tells the user it was correct.
5		Shows animation which moves from current level to next level
6		Moves back to the map of levels
7		Animation which unlocks new “world”

8.1	User can clicks on the new world	
8.2		Go to alternative flow 8.2

Alternate flow

Flow 8.2 (add heading)

The player clicks on next world.

	Actor	System
1	Clicks on next world	
2		Shows next view with a map of the level and grafic indicator of where the user is

6.1.7

Use case: Key

Summary: The user wants to see the correct answer.
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The player enters the right answer.

	Actor	System
1	Enters the wrong answer	
2	Checks the hints	
3		Shows hints (See hint user case)
4	Enters wrong answer again	
5		(Repeats til there’s no more hints)
6		Changes the hint button to see correct answer
7	Clicks show answer	
8		Screen changes to show the correct answer
9		The show correct answer button changes to a back button

6.1.8

Use case: Hint

Summary: The user is unsure of how to solve an objective.
Priority: high
Extends: (other UC)
Includes: (other UCs)
Participators: The user

Normal flow of events

The user checks the hint

	Actor	System
1	Clicks “hint”-button	
2		Plays animation. A hint shows on the screen

Alternate flow

Flow 2 (add heading)

The player enters the wrong answer, at least once.

Flow 8.2 (add heading)

The user enters the wrong answer twice.