System porting to mobile devices at the example of the SEE project

Master Thesis

Roman Gressler

Matriculation number: 3217822

June 3, 2022

Faculty 3 — Mathematics and Computer Science
Computer Science

1. Supervisor: Prof. Dr. Rainer Koschke
2. Supervisor: Prof. Dr. Zwetachter

ABSTRACT

TODO: Hier das Abstract der Arbeit. Kann deaktiviert werden.

ERKLÄRUNG

Ich versichere, diese Arbeit — sofern dies nicht explizit anders gekennzeichnet wurde — ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Bremen, den June 3, 2022

Roman Gressler

DANKSAGUNG

TODO: Danksagung hier.

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

1.2 RESEARCH QUESTION

The central research question of this thesis is: Are Android smartphones suitable of working with "*Software Engineering Experience* (SEE)"?

To answer this question this thesis will discuss a general concept to implement "SEE" on mobile devices in chapter 2. Afterwards the specific details of implementation will be introduced in chapter 3. The research question will then be answered in chapter 4 with an evaluation where the implemented mobile version of "SEE" will be compared with the desktop version. Last but not least this thesis will discuss further research and give a conclusion in chapter 5.

SEE: *Eine interaktive Visualisierung von Software, welche die* Code-City-*Metapher verwendet und einen kollaborativen Multiplayer über verschiedene Plattformen*[2] *hinweg ermöglicht.*

2 CONCEPT

In this section a concept of a mobile "see" version will be presented. Therefore, a prototype will be created to point out the features that a mobile version of "see" requires.

Prototypes are a common way to express the needs of a system. It is a low-cost way of planning an implementation, that can highlight challenges regarding constraints of a system early on.

Even though a prototype will never be able to show every aspect and need of a complex system, it should still help to answering questions like: How should the system feel? How should it be implemented and what are the key features? Houde and Hill (1997)

"see" is meant to be used by multiple platforms such as desktop devices, mobile devices and virtual reality devices. Each device has different interaction constrains. While a desktop user will control the player with mouse and keyboard a mobile user will interact with virtual joysticks on a touchscreen. Selecting nodes of a "*Code-City*" will be done by clicking it with a mouse on desktop devices, while a mobile device will require a touch input.

Code-City: *In der Code-City-Metapher werden Softwarekomponenten durch Gebäude in einer Stadt repräsentiert, wobei die Eigenschaften dieser Gebäude verschiedene Metriken der Software ausdrücken können — z. B. könnte die Höhe eines Gebäudes der Anzahl der Codezeilen entsprechen.*

2.1 INTERFACE

In the following a paper prototype will be presented that marks out a concept for the mobile interface. Since the field of mobile development is quite young there few guidelines regarding the design of mobile device interfaces. A guideline that is widely accepted is problematic to find. Renaud and Van Biljon (2017), Punchoojit and Hongwarittorrn (2017)

Major differences to desktop environments are the screen size, forms of input and input feedback. To assure as much space is used for the actual interaction of the app the menu should just take as much space as needed. As a study has found out, a size of at least 8*8 mm is needed to reduce error rates selecting the right button. Conradi et al. (2015) Parhi et al. (2006) TODO WEITER AUSFÜHREN SHORTCUTS WIE STRG Z NICHT MÖGLICH Adipat and Zhang (2005)

Moving the player will be handled with virtual joysticks as seen in figure 2.1. The left joystick will move the player through the virtual room and the right will move the camera angle or in other word the direction the player looks at. The joysticks are placed in the left and right corner and should just take as much space as needed to be handled

comfortably. This way the player is able to navigate through the virtual room with his/her thumps while still having enough space to work on the “Code-City”.

Figure 2.1: Joysticks for moving in “SEE”

The menu on the top left side seen in figure 2.2 will be called “quickbar” further on. The quickbar can be minimized to safe screen space when not needed. The quickbar is designed to offer more general functions that are needed in various situations. Because there are no shortcuts on mobile devices each function has to have a button to be activated.

The functions are redo and undo which will do an action undone again or revert an action. Then there is a camera lock that will lock the players perspective to a certain “Code-City” so that the player can only move around the selected city and move closer or further away from it. The next function is to rerotate a “Code-City”. That means the “Code-City” that was last rotated will be set back to its initial state of rotation. Last but not least there will be a button for recentering the city, which will work quite similar to the rerotate button and center the last moved “Code-City”. The button on the right can be used to collapse or expand the quickbar.

On the top right side another menu will be placed that contains different interaction modes. By clicking a button an interaction mode will be selected and moved to the top right corner. Also, the menu will be collapsed and only the buttons regarding the selected interaction mode shall be shown. By clicking the button on the top right again the menu shall expand and the other interaction modes shall be selectable. The other buttons shall be kept in the same order to reduce confusion of the user.

The first interaction mode, seen in figure 2.3, is for selecting nodes. Nodes can be selected by being touched and deselected by being touched again. There can be multiple nodes selected at once. The hole selection can be deselected by clicking the deselect button next to the

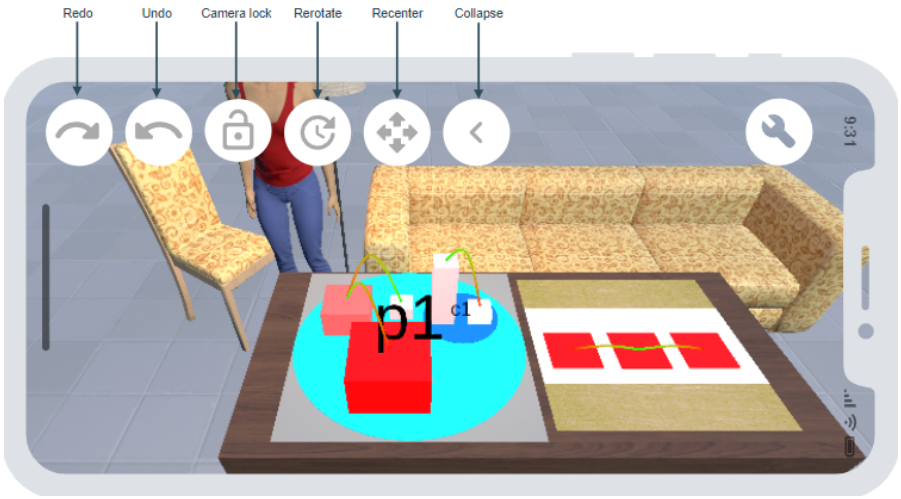

Figure 2.2: Quickbar for various interactions in "SEE"

select interaction mode button. Selected nodes shall be highlighted with a different node color and also display their name.

Figure 2.3: Selection mode in "SEE"

The second interaction mode, seen in figure 2.4, is for deleting node. It does not need additional buttons. Node will be deleted by being touched.- Unlike in the desktop version there will not be a group deletion interaction because it would require an additional menu panel. The added functionality would be minimal and selecting a group of nodes, confirming and finally deleting would require a handful more steps and would therefore most likely not be used.

The following interaction mode, seen in figure 2.5, is dedicated to the nodes and edges of a "Code-City". Starting on with the "add node" button on the right. When activated the user can create new node by clicking on a certain spot on the "Code-City" plane. The following button on the left is for adding edges. By selecting two nodes a new edge will be created between them. Then, the button one further on the

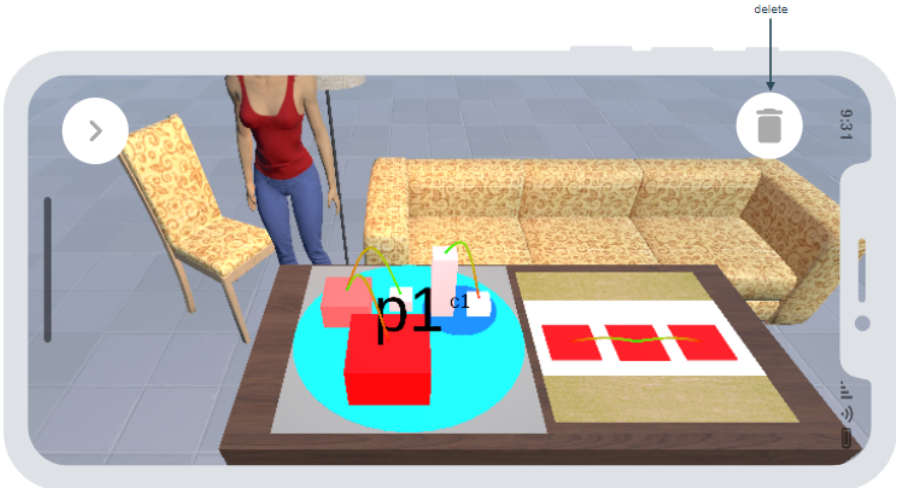

Figure 2.4: Delete mode in "SEE"

right is for editing nodes. By touching a node a window will pop up that allows the user to edit the node by changing its name and its type. Last but not least the button on the left-hand side will be used to scale nodes. That means the node height and width can be adjusted by first selecting it via touch and then hold a corner and slide it further away from the node center to increase the size or slide it towards the center to decrease the size of the node. Each button of the node interactions will be marked green after being pressed to indicate that it is active.

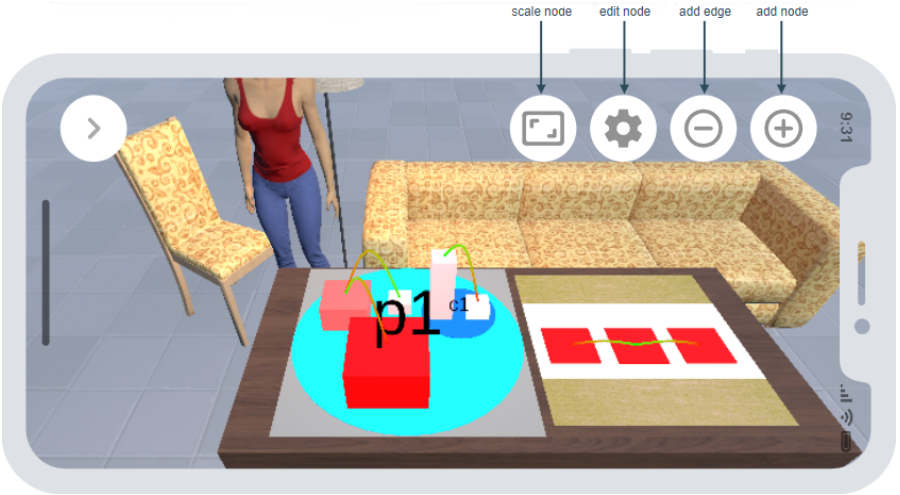

Figure 2.5: Node interactions in "SEE"

Then there will be a button for rotation interactions that can be seen in figure 2.6. Starting with the first activatable button that lets the user rotate the hole "Code-City" by touching any point on it and then sliding away from that point. Similar to that there will be a button that lets the user rotate just a single node on the "Code-City". In addition to that there will be a button that activates the so-called "locked-rotation" mode. While in "locked-rotation" mode the rotation of a node or

"Code-City" will be done in eight predefined steps to a full rotation. Each step will have the same 45° range. The last button of this group will be for changing the center of the rotations. There are to options: the first option is a center of rotation in the middle of the "Code-City" and the second is in the middle of a node selection made with the interactions seen in figure 2.3. The second option can be activated by pressing the last button.

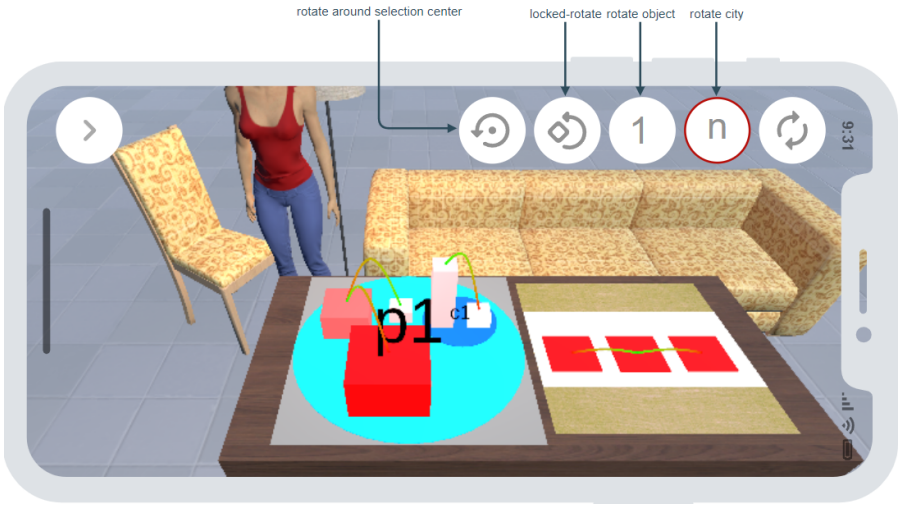

Figure 2.6: Rotation mode in "SEE"

The last interaction group, seen in figure 2.7, is for moving the "Code-City" or a single node. The move interactions are quite similar to the rotation interactions. There will be a button to move a hole "Code-City" as well as a button to move only single nodes. In addition to that there will be a button that restricts the movement of the "Code-City" or node to a predefined direction. The directions will be again in 45° angles and objects can be moved on a straight line on that angle. Moving a node or a "Code-City" can be achieved by touching and holding it and then moving it to the desired position.

2.2 INTERACTION

Smartphones are quite limited in space and there are few input possibilities. Unlike a desktop computer there is no mouse and there is no physical keyboard. Smartphones use virtual keyboards but due to the restriction of screen space the keyboard is hidden most of the time. Which would make keyboard shortcuts uncomfortable because the user has to open the keyboard first. Therefore, smartphones need different ways of interaction such as touch gestures.

Zooming in to a "Code-City" happens by scrolling on a desktop environment. The is no option to scroll on mobile devices, but there are at least two popular alternatives. The first option would be to double tap on the "Code-City" to zoom in. The double tap would zoom in,

Figure 2.7: Movement mode in "SEE"

in predefined steps and after reaching a certain level of closeness it would trigger to zoom out again. In "SEE" zooming in, in predefined steps might not be precise enough because there could be a quite large "Code-City" or a rather small one. Finding predefined steps that would fit every situation is rather hard. Therefore, a second option by zooming in with a two finger gesture might be better. In this option the user uses two fingers and slides them towards each other to zoom in or slides the two fingers away from each other to zoom out. This way there are no predefined steps necessary and zooming interactions can be done precisely.

2.3 REQUIREMENTS

In the following a list of requirements will be given, which will specify in detail what the implementation of a mobile version has to take care of. The list will be referred to multiple times in the upcoming realization part in chapter 3. Requirements are essential for the planning phase as they give a good fundamental structure for the developer to rely on. Robertson and Robertson (2012); Stevens and Pooley (2005)

[R1] The application shall run on Android devices

[R2] The application shall be controlled via touchscreen

- [R2.1] The player and camera shall be moved with virtual joysticks
- [R2.2] Needed shortcuts of the desktop version shall be handled with buttons
- [R2.3] Zooming shall be handled with a two finger gesture

[R3] The user shall be able to select a node of a "Code-City"

- [R3.1] After selecting the name of the node shall be shown

- [R3.2] The user shall be able to deselect single nodes or a group of nodes

- [R4] The user shall be able to delete nodes

- [R5] The user shall be able to interact with nodes
  - [R5.1] The user shall be able to add nodes
  - [R5.2] The user shall be able to add edges
  - [R5.3] The user shall be able to edit nodes
  - [R5.4] The user shall be able to scale nodes

- [R6] The user shall be able to rotate a “Code-City”
  - [R6.1] The user shall be able to rotate a “Code-City” in 45° steps
  - [R6.2] The user shall be able to rotate single objects
  - [R6.3] The user shall be able to rotate around a center of selected nodes
  - [R6.4] The user shall be able to undo the rotation

- [R7] The user shall be able to move a “Code-City”
  - [R7.1] The user shall be able to move single object of a “Code-City”
  - [R7.2] The user shall be able to restore the “Code-City” initial position
  - [R7.3] The user shall be able to move a “Code-City” or single node in predefined directions

- [R8] The user shall be able to undo and redo actions

- [R9] The user shall be able to lock the camera to a selected “Code-City”

3 IMPLEMENTATION

...

4 EVALUATION

In the following chapter the mobile implication of "SEE" will be evaluated in a user study. Therefore, the mobile application will be compared with the desktop version.

This chapter will start with a description of the desktop version and its main differences in section 4.1. Continuing with a defined aim and the precise hypotheses for the user study in section 4.2. After sketching the first experiment set up in section 4.3 the actual experiment set up will be discussed in detail in section 4.4 including the used survey tool, questionnaires and the pilot study.

4.1 "SEE" DESKTOP

In this section the desktop version of "SEE" will be explained. In this evaluation the mobile version of "SEE" will be compared with the desktop version. Therefore, it is necessary to take a deeper look at the differences between those two versions. Especially at how the interactions differ and what impact it could have on the user experience.

One outstanding difference from the desktop version to the mobile version is the selection of the interaction modes. While in the mobile version the menu for the interaction modes is always visible in the desktop version by pressing space a menu screen opens as seen in figure 4.1. Alternatively interaction modes can be changed by pressing one of the "1-9" keys, which, however requires the user to memorize which number belongs to which mode.

Another difference is the type of user input. The desktop version uses mouse hovering to display the name of a hovered "*Node*" or "*Plane*". This is a faster method then touching the object first in the mobile version. In addition to that in the mobile version the object also has to be deselected otherwise there will be a lot of "Node" and "Plane" names displayed, and it will soon get quite messy. Also, the precision of object selection differs because touch input can never be as precise as selecting with a mouse cursor. This could force the mobile user to zoom further in because with a touch input it will not be possible to select small objects like it might be with a cursor. Which, of course, would require more time.

One more key difference is the available keyboard for desktop users. It allows using "*Shortcuts*", which makes some menu items unnecessary but also requires the user to memorize those "Shortcuts". The desktop

Node: *A point in a diagram where lines intersect. In "SEE" it usually displays a software class*

Plane: *An area that bundeles "nodes". Could for example represent a namespace.*

Shortcut: *A combination of key that will call an action like for example "ctrl" + "c" for coping a text.*

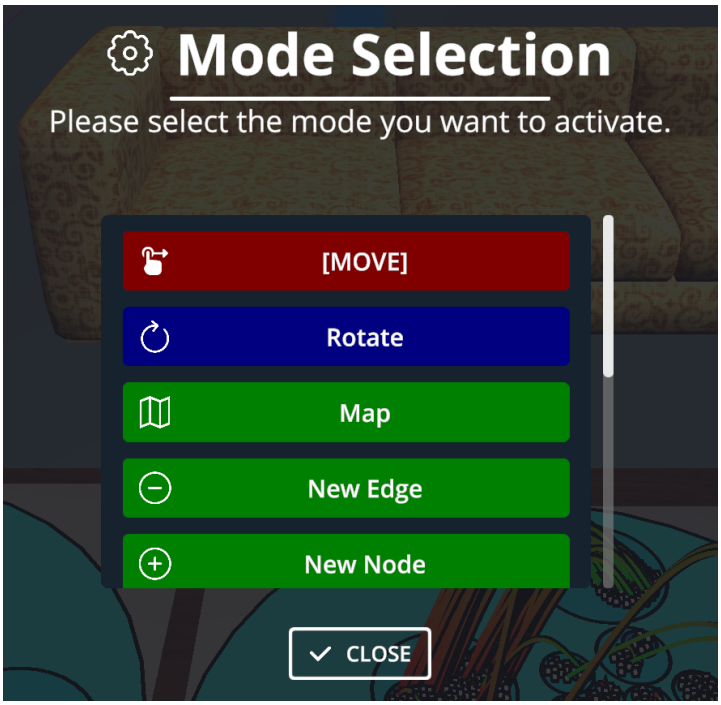

Figure 4.1: The desktop menu for selecting interaction modes.

version for example uses the "R" key in the move and rotation mode to recenter or rerotate a "Code-City". In the mobile version on the other side the user will find a button for both actions. With the right amount of training both actions should probably equal in the amount of time they need but the mobile version sacrifices screen space for those buttons. If however the user has to type more text like in renaming objects, the common desktop keyboard should come in handy as a study from Kim et al. (2014) shows that even at a same keyboard size, a virtual one will lack in productivity.

4.2 AIM AND HYPOTHESIS

The finished prototype of the mobile extension shall be evaluated. Therefore, the system shall be compared on smartphones as well as desktop computers. Comparing these two use cases shall give insight on how much impact the constraints of mobile devices have on the usability and overall user experience.

The hypotheses of this thesis is that the mobile version of "SEE" lacks slightly in usability compared to the desktop version. This due to the constraints of the mobile device. A smartphone has far less screen space and also does not have a physical keyboard, which would allow many more shortcuts.

4.3 EXPERIMENT SET UP

The system shall be tested in two groups each starting with a different device. Each group does the test on both devices, but one group will start with the mobile application and the other one with the desktop application. The participants will be assigned random to the groups. The testers will have various tasks to test the usability of the two applications. Afterwards the users will get a survey in English to document their impressions.

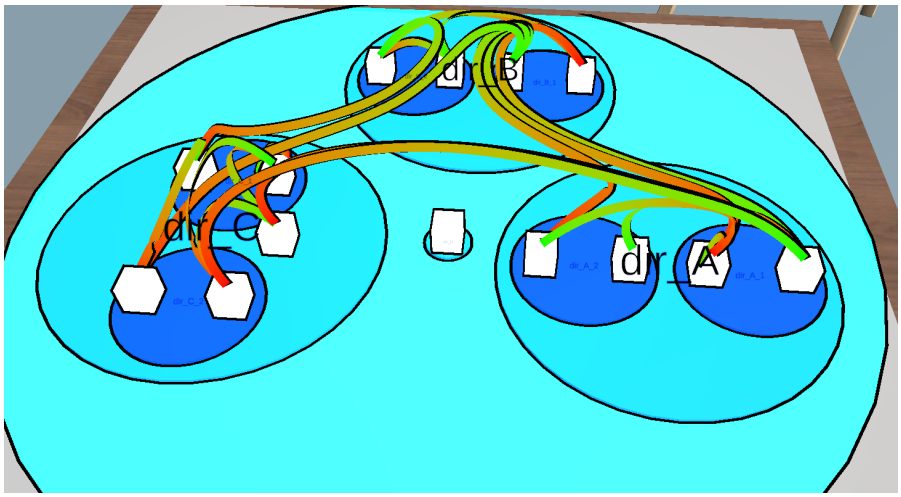

Figure 4.2: The first "Code-City" for the user study

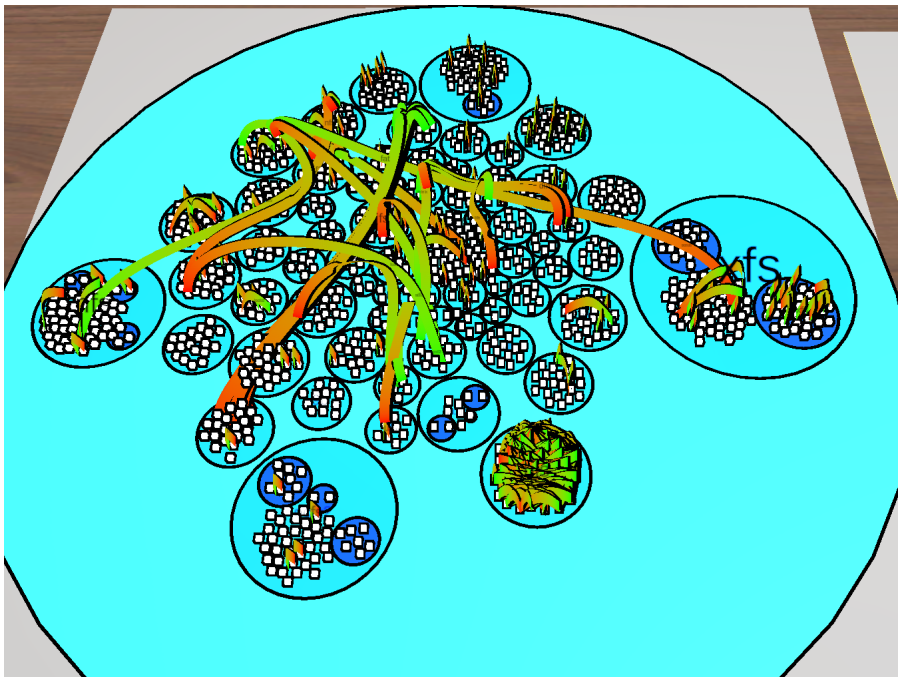

Figure 4.3: The second "Code-City" for the user study

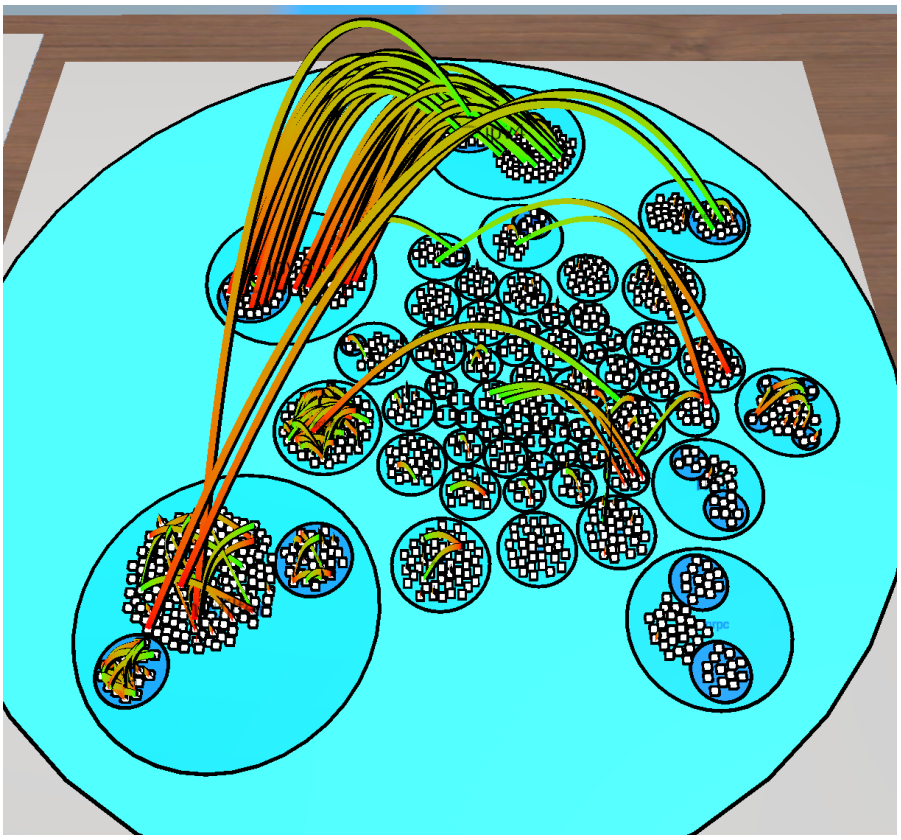

Figure 4.4: The third "Code-City" for the user study

In this survey the subjects will be asked various demographic questions as well as what Android device and version they will be using. In addition to that the subjects will be asked if they are experienced with "SEE" and if they are experienced with software development. Before the subjects will be asked to solve various tasks they will be asked to watch a short tutorial video on each application. After the video they will get a training task where every subject can get used to the system and ask questions if they have trouble solving the training task. The overseer will also make sure that every essential action will be practiced such as zooming and moving the "Code-City". Figure 4.2 shows a small arranged "Code-City" that shall be used for the training tasks. The structure of the training "Code-City" is generic and follows a simple pattern. That shall ensure that the user can focus on the training and that the user does not get overwhelmed.

Following the first questions and the training, the subjects can start with the main tasks. For each application there will be two tasks and after each task the subjects will be handed a "*Post-Task*" questionnaire. Last but not least there will another questionnaire that aims to scale the "*Usability*" of the two applications. For the "Post-Task" questions the "*After-Scenario Questionnaire* (ASQ)" will be used and for the "Usability" questions "*System Usability Scale* (SUS)" will be used. Both questionnaires will be discussed later on in section 4.4.2. For each

Post-Task: *A questionnaire that is taken after every task of an experiment.*

Usability: *A term that describes how well a (software) system can be used.*

ASQ: *A post-task questionnaire consisting of three questions that is used to access how difficult a user perceived a task (See "Post-Task").*

SUS: *The Sytem Usability Scale consists of ten questions that measure "Usability"*

main task the overseer will also take the completion time of every main task. The first and second task on the first device will be performed on the "Code-City" that can be seen in figure 4.3 and the third and forth task on device two will be performed on the "Code-City" that can be seen on figure 4.4. These examples are much larger than the training "Code-City" and represent real life code. The second "Code-City" shows the file system of Linux and the third one shows the network component of Linux. That way the tasks might reflect better on real world uses for "SEE".

To not exhaust the testers too much the experiment shall not take longer than one hour. This also ensures that there is no to little variance due to exhaustion. Each participant might have a different concentration span, but this shall not be the focus of this experiment.

4.4 REALIZATION

4.4.1 *Survey tool*

4.4.2 *Questionnaires*

4.4.3 *Pilot study*

In a first test the pilot study was executed with one tester. Afterwards the study was discussed and checked for errors. It stood out that the example "Code-City" of task one was too different to the one in the second task. Therefore, the "Code-City" of the first task was exchanged with a larger and better comparable one. Further on a "Code-City" with 1288 nodes (see figure 4.3) as well as one with 1464 nodes (see figure 4.4) will be used.

Also, the tasks were not comparable because they differed in the types of interactions they used. In one task the user was asked to rename a node and in the other one the user shall add four nodes. For renaming a node the user has to use a keyboard which does not make it comparable to just click and add nodes in the second task.

4.4.4 *Final experiment set up*

Demographic questions:

- Age
  - 0-15 years old
  - 16-30 years old
  - 31-45 years old
  - 46+ years old

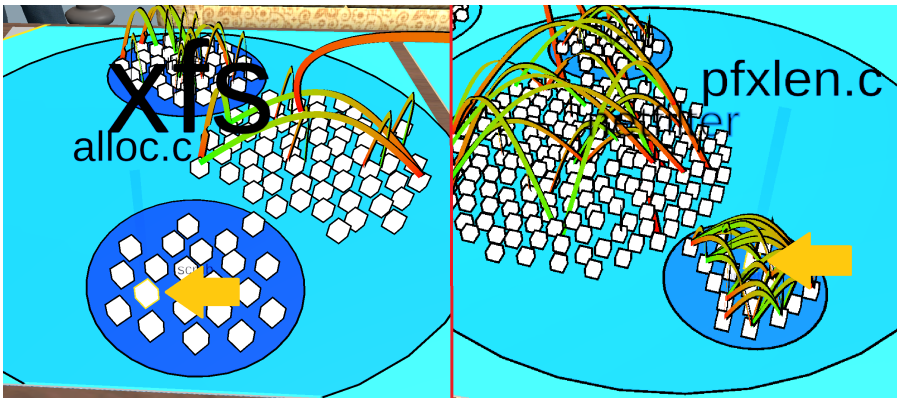

Figure 4.5: The two key nodes are marked with a yellow arrow

- What gender do you identify as?
  - Male
  - Female
  - Other ...
  - Prefer not to say
- What is the highest degree or level of education you have completed?
  - Some High School (Hauptschule/Realschule...)
  - High School (Abitur)
  - Bachelor's Degree
  - Master's Degree
  - Ph.D. or higher
  - Prefer not to say
  - Other ...

Questions regarding used hardware and experience

- Are you already experienced with See?
- Do or did you play first person video games?
- Do or did you develop software?
- On which Android device will you attend?
- Which Android version are you using?*

4.4.5 *Execution*

Nr.	Task	Expected time
Training	Navigate through the planes “dir_root” >“dir_B” >“dir_B_2”. On that plane select “b2_b.cpp” and rename it “b42”.	1 - 5 mins
1	Detect the largest plane “xfs”. On that plane find plane “scrub”. Then find and delete node “alloc.c”.	0.5 - 5 mins
2	Find the plane with one blue child plane (“btrfs”). On the blue child plane “tests” add four new nodes.	1 - 5 mins
Training	Navigate through the planes “dir_root” >“dir_C” >“dir_C_2”. On that plane select “c2_b.cpp” and rename it “c42”.	1 - 5 mins
3	Detect the largest plane “netfilter”. On that plane find plane “ipset”. Then find and delete node “pfxlen.c”.	0.5 - 5 mins
4	On the plane with the most edges (“ipv6”) find the smallest plane “ila” and connect all four nodes on it.	1 - 5 mins

Table 4.1: The tasks used for the experiment. The device will be switched after task 2.

Phase		Description	
Pre-Experiment		Demographic questionnaire	
	City	Group 1	Group 2
Training	Figure 4.2	Desktop	Mobile
Task 1	Figure 4.3		
ASQ			
Task 2	Figure 4.3		
ASQ			
SUS			
Training	Figure 4.2	Mobile	Desktop
Task 3	Figure 4.4		
ASQ			
Task 4	Figure 4.4		
ASQ			
SUS			

Table 4.2: Experimental procedure per subject. The procedure is swapped per group.

5 CONCLUSION

...

5.1 OUTLOOK

AR - Santos et al. (2016)

A

GLOSSARY

Code-City In der Code-City-Metapher werden Softwarekomponenten durch Gebäude in einer Stadt repräsentiert, wobei die Eigenschaften dieser Gebäude verschiedene Metriken der Software ausdrücken können — z. B. könnte die Höhe eines Gebäudes der Anzahl der Codezeilen entsprechen. 1–7, 12–15, 25

Node A point in a diagram where lines intersect. In "SEE" it usually displays a software class 11, 21

Plane An area that bundeles "nodes". Could for example represent a namespace. 11

Post-Task A questionnaire that is taken after every task of an experiment. 14, 23

Shortcut A combination of key that will call an action like for example "ctrl" + "c" for coping a text. 11

Usability A term that describes how well a (software) system can be used. 14, 23

B

ACRONYMS

ASQ A post-task questionnaire consisting of three questions that is used to access how difficult a user perceived a task (See "Post-Task"). 14

SEE Eine interaktive Visualisierung von Software, welche die *Code-City*-Metapher verwendet und einen kollaborativen Multiplayer über verschiedene Plattformen[1] hinweg ermöglicht. 1–6, 11, 12, 14, 15, 21, 25

SUS The Sytem Usability Scale consists of ten questions that measure "Usability" 14

1 Neben Desktop- und Touch-Umgebungen noch Virtual Reality (z. B. *Valve Index*) und Augmented Reality (z. B. *Microsoft HoloLens*)

C

LIST OF FIGURES

D

LIST OF TABLES

Regie: Kontrolliere am Ende, ob alle bibliographischen Angaben vollständig sind. Wird also die Zeitschrift oder Konferenz aufgeführt, in der ein Artikel veröffentlicht wurde? Sind überall die Seitenangabe aufgeführt? Bei Verweisen auf Web-Seiten, ist überall angegeben, wann der letzte Zugriff darauf erfolgte? Sind Umlaute und andere Sonderzeichen korrekt in LaTeX beschrieben worden?

BIBLIOGRAPHY

Stephanie Houde and Charles Hill. What do prototypes prototype? In *Handbook of human-computer interaction*, pages 367–381. Elsevier, 1997.

Karen Renaud and Judy Van Biljon. Demarcating mobile phone interface design guidelines to expedite selection. *South African Computer Journal*, 29(3):127–144, 2017.

Lumpapun Punchoojit and Nuttanont Hongwarittorrn. Usability studies on mobile user interface design patterns: a systematic literature review. *Advances in Human-Computer Interaction*, 2017, 2017.

Jessica Conradi, Olivia Busch, and Thomas Alexander. Optimal touch button size for the use of mobile devices while walking. *Procedia Manufacturing*, 3:387–394, 2015.

Pekka Parhi, Amy K Karlson, and Benjamin B Bederson. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 203–210, 2006.

Boonlit Adipat and Dongsong Zhang. Interface design for mobile applications. *AMCIS 2005 proceedings*, page 494, 2005.

Suzanne Robertson and James Robertson. *Mastering the Requirements Process: Getting Requirements Righ*. Addison-Wesley Professional, 2012. ISBN 978-0-13-294285-0.

P. Stevens and R. Pooley. *Software Engineering with Objects and Components*. Springer, 2005.

Jeong Ho Kim, Lovenoor Aulck, Michael C Bartha, Christy A Harper, and Peter W Johnson. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied ergonomics*, 45(6):1406–1413, 2014.

Carlos Santos, Brunelli Miranda, Tiago Araujo, Nikolas Carneiro, Anderson Marques, Marcelle Mota, Jefferson Morais, and Bianchi Meiguins. Guidelines for graphical user interface design in mobile augmented reality applications. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 71–80. Springer, 2016.