Slovak University of Technology in Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

FIIT-5208-35196

Bc. Filip Hlaváček

ALTERNATIVE INTERACTION APPROACHES IN THREE-DIMENSIONAL ENVIRONMENTS ON THE SCREENS OF MOBILE DEVICES

Master's thesis

Study program:	Information Systems
Field of study:	9.2.6 Information Systems
	Institute of Applied Informatics
	Faculty of Informatics and Information Technologies
	Slovak University of Technology in Bratislava
Supervisor:	Dr. Alena Kovárová

2012, May

Annotation

Slovak University of Technology in Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: INFORMATION SYSTEMS

Author: Bc. Filip Hlaváček

Diploma project: Alternative Interaction Approaches in Three-dimensional Environments on the screens of Mobile Devices

Supervisor: Dr. Alena Kovárová

2012, May

Human-computer interaction (HCI) plays an essential role in today's technology. Large displays, various input devices, high-end mobile phones, all need to adopt specific HCI approaches in order to present their true potential to the common user. Just recently multi-touch displays became a standard for mobile devices. Common interaction techniques for 2D environment manipulation have already been adopted, but 3D interaction techniques are in development.

In this thesis we focus on multi-touch interaction techniques with the aim to design a solution that others would adopt. To reach this goal we set out by analyzing research on advanced interaction techniques and existing approaches used in applications available to basic users. We design our own techniques with the aim to provide all six degrees of freedom. The prototype is an Android OS application that should help students to acquire knowledge in the field of solid geometry, specifically cube cross sections.

Keywords:

3D geometry, Virtual Reality (VR), Virtual Reality Learning Environment (VRLE), Multi-touch interaction, Mobile devices, 6 DoF

ANOTÁCIA

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Študijný program: INFORMAČNÉ SYSTÉMY

Autor: Bc. Filip Hlaváček

Diplomový projekt: Alternatívne spôsoby práce v trojrozmernom priestore na obrazovkách mobilných zariadení

Vedenie diplomového projektu: Mgr. Alena Kovárová, PhD.

máj 2012

Interakcia človeka s počítačom (ICP) zohráva v dnešnej modernej dobe významnú úlohu. Obrovské obrazovky, rozličné vstupné zariadenia, špičkové mobilné telefóny, všetky tieto zariadenia potrebujú špecificky pristupovať k otázke ICP, aby ich potenciál mohol využiť aj obyčajný používateľ. Len nedávno sa viac dotykové obrazovky stali štandardnou výbavou mobilných zariadení. Existujú už všeobecne zaužívané techniky interakcie v dvojrozmernom prostredí, ale interakcia v trojrozmerných priestoroch je stále vo vývoji.

V tejto práci sa zameriavame na viac dotykové interakčné techniky s cieľom navrhnúť riešenie, ktoré by si ľahko osvojili aj bežní používatelia. Aby sme tento cieľ mohli dosiahnuť, začali sme analýzou existujúceho výskumu v tejto oblasti a aplikácii, ktoré sú dostupné bežným používateľom. Naše techniky definujeme s cieľom zachovania šiestich stupňov voľnosti. Plánovaný prototyp bude aplikácia pre operačný systém Android a mal by byť schopný pomôcť žiakom stredných škôl s učením sa stereometrie, konkrétne rezov kockou.

Kľúčové slová:

stereometria, virtuálna realita, viac dotyková interakcia, mobilné zariadenia

Acknowledgment

I would like to express my gratitude to my supervisor, Dr. Alena Kovárová, whose expertise, understanding, and patience are reflected in every single page of this thesis. I would especially like to thank her, for her everlasting optimism and enthusiasm which drove me forward at a steady pace, allowing me to reach so far the biggest milestone in my life, my master's degree.

I owe my deepest gratitude to my entire family, without whose support through my entire life, I would not be able to achieve any of this.

A very special thanks goes to my mom, for were it not for her sacrifices despite these grave times, I would not have finished this thesis.

In conclusion, I would like to thank everyone else, who helped me one way or another.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

2D	-	Two Dimensional
3D	-	Three Dimensional
ADT	-	Android Development Tools
AVD	-	Android Virtual Device
DoF	-	Degrees of Freedom
GPS	-	Global Positioning System
IGS	-	Interactive Geometry Software
IR	-	Infrared light
RNT	-	Rotate'N Translate
SDK	-	Software Development Kit
UI	-	User Interface
VR	-	Virtual Reality
VRLE	-	Virtual Reality Learning Environment

1 INTRODUCTION

Human-computer interaction (HCI) plays an essential role in today's technology. Large displays, various input devices, high-end mobile phones, all need to adopt specific HCI approaches in order to present their true potential to the common user. Just recently multi-touch displays became a standard for mobile devices. Common interaction techniques for 2D environment manipulation have already been adopted, but 3D interaction techniques are still in development.

This thesis reports on a research study investigating the use of virtual reality in conjunction with multi-touch mobile devices to facilitate the knowledge construction by middle school students of 3-dimenstional (3D) geometry by means of up-to-date education approaches.

New generations of young students are constantly being more difficult to educate as it is difficult to motivate them. Old educational means often do not appeal to them. I believe everyone knows what it feels like, when he does not master a lecture and later on has problems with following lectures as it is not easy to catch up. Nowadays technologies and high-end devices available to almost everyone can be used to motivate students with new educational approaches. In order to be able to stimulate students' knowledge construction, this application has to be intuitive, fast to master and efficiently provide relevant information to the students. In order to fulfill these requirements we have designed a new multi-touch technique approach with the aim to provide enhanced but easy to learn interaction possibilities.

This thesis is divided into 9 chapters. This short introduction concludes the first chapter. In the 2nd chapter we thoroughly analyze virtual reality and human-computer interaction, as the comprehension of these two areas is of essential importance in order to design a meaningful application. The 3rd chapter follows with a more detailed description of the existing approaches and techniques as well as a list of applications that support interaction in a virtual 3D world. The application requirements are described in chapter 4 and are followed with the software design in chapter 5. The 6th chapter contains information regarding the implementation of our CCS 3D application. Chapter 7 presents a thorough evaluation of the results acquired in the individual stages of our thesis. In the 8th chapter we summarize our achievements as well as future work. The thesis ends with the list of references in the 9th chapter.

2 WORLD OF 3D

Computers first emerged to serve the human race to facilitate different types of tasks. Nowadays computers run production lines with minimal human intervention or simply entertain us. One of the main aims is to provide users with simplified virtual environments where they can simulate different actions without having to face real-world consequences or create environments that eliminate various negative effects or even threats while interacting with very rare, precious, expensive or dangerous objects (be it a space shuttle, a human heart, or even an entire ecosystem). To prevent that we end up controlling these applications with simple text commands, these environments cannot simply focus on providing the simulated object's functionality but have to take into account the way people will interact with such applications. Otherwise one would have to acquire excessive knowledge just in order to be able to accomplish the most basic tasks. With today's technologies we are, however, able to enhance such applications with all kinds of different input and output devices that provide the "real feel" and therefore an intuitive interaction. And that is where both Virtual Reality (VR) environments and Human Computer Interaction (HCI) have found their way into science.

2.1 Virtual Reality

It is said that the origins of virtual reality can be traced back to the early 1950s [1], when Douglas Engelbart, a young engineer and former naval radar technician, was one of the first to think of computers as tools for digital display. At that time his futuristic concepts were not taken into account. Later on in the year 1965, after the emergence of first displays, an ARPA scientist named Ivan Sutherland published his essay "The Ultimate Display".

> *"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such room would be fatal."*[1]

This essay full of wild futuristic thoughts, as well as precise predictions, planted the seed of VR into the minds of many scientists. Three years later Sutherland created one of the first head mounted augmented reality display systems. The following years VR emerged in different areas notably in military flight simulators, movies and at last in video games.

[1] http://www.eng.utah.edu/~cs6360/Readings/UltimateDisplay.pdf

Nowadays everyone knows that there is something called “Virtual Reality”, everyone has an idea of what it theoretically is, but no one truly knows how to exactly define it.

> *“What I envision is not so much a pre-programmed virtual world that you might play as a game, but rather a virtual world that you can change from the inside that people use as a form of expression in which they’re constantly creating things together.”* Virtual Reality Pioneer – Jaron Lanier (Sun Microsystems Inc., 2003)

> *“Virtual Reality, the use of computer modeling and simulation that enables a person to interact with an artificial three-dimensional (3-D) visual or other sensory environment. VR applications immerse the user in a computer-generated environment that simulates reality through the use of interactive devices, which send and receive information and are worn as goggles, headsets, gloves, or body suits. In a typical VR format, a user wearing a helmet with a stereoscopic screen views animated images of a simulated environment.“*[2]

This perfect oxymoron “Virtual Reality” simply represents an artificial environment that has the ability to convince the user of its existence as a real world. VR resides in the thoughts of humans in different forms and images, as everyone depicts it as the means to their fabled reality. That is why VR nowadays is not the same for everyone. An application where a scientist immersed in a 2D simulation of molecules studies their lifecycles, would not be worthy of the title “Virtual Reality” in the eyes of a teenager, who spends most of his time living his life detached from the real world playing stereoscopic 3D games with stunning graphics and various input devices with haptic feedback.

The less we have to use our imagination, the more realistic the environment will be. We perceive reality with our senses. Without having to imagine or think of what those virtual objects would feel, look and sound like, were they real, we eliminate that subconscious feeling that they are artificial. By feeding various inputs to our senses, we can concentrate more on the experience than on the act of perceiving the virtual environment as reality.

Immersive VR is a term characterized as a total lack of physical distance between the immersant’s body-image and the immersive environment [14]. In other words:

> *“Total immersion is implied complete presence within the insinuated space of a virtual surrounding where everything within that sphere relates necessarily to the proposed “reality” of the world’s cyberspace and where the immersant is seemingly altogether disconnected from exterior physical space.”* [3]

[2] http://www.britannica.com/EBchecked/topic/630181/virtual-reality-VR

It takes different "doses" of VR for people to achieve this state of mind. For some it is enough to fool visual and auditory senses, others need tactile feedback as well. Some might need in addition to the previously mentioned senses, to perceive the virtual environments by olfactory and gustation senses as well. Immersive VR is not very common as the equipment required to manipulate one's mind to lose total awareness of reality is expensive and has not yet found its way into the market.

2.2 Virtual Reality Learning Environments

By being able to partially manipulate human minds, VR found its way into education. Students have trouble acquiring new knowledge as they find it often uninteresting, boring, unimportant or even useless. Motivation has been, and will forever remain, the teacher's most effective weapon. VR brings the possibility to enlighten students with new teaching methods and approaches as well as explaining things without having to explain how to interpret the explanations.

Recently VRLEs have emerged in the field of education. The goal is to provide environments for the students where knowledge can be acquired in its native form, by simply experiencing the different theoretical situations in the given environment. This eliminates the burden of processing theoretical knowledge presented in undefined environments, where apart from focusing on the subject of interest, students subconsciously process large amounts of unknown variables that define the essence of the environment in which the subject resides.

Our research will be conducted in the field of 3D geometry, also known as solid geometry. Students often have problems projecting shapes drawn on paper into a three-dimensional space in their minds. Further operations on these objects are difficult just to imagine, therefore difficult to comprehend and learn. Presenting such shapes and operations in VR enables the students to focus on the target objects and later on, when already acquainted with necessary experience based knowledge, to project these shapes and operations onto paper or semantics.

There have been many projects that have built VRLE for geometry. In the case of VRMath [2] students are opted to complete various tasks that are analyzed into details in the thesis. An advanced solution for a VRLE has been presented by Hannes Kaufmann and Dieter Schmalstieg [4] where students are allowed to interact with 3D objects in an immersive VR and therefore directly experience the knowledge.

There are many obstacles one must face in order to create an efficient VRLE. Most of the mentioned VRLEs are of scientific character and are focused more on analyzing their impact on the knowledge acquisition effectiveness of students. There are however a few applications such as Cabri 3D[3] and GeoGebra[4] that are already being used in classroom education.

2.3 HCI

With new technologies emerging daily and providing users with all kinds of different possibilities the need to evolve the interaction methods grows rapidly. The value of a VR without intuitive or at least fast learnable interaction methods degrades rapidly as the users are demotivated and unwillingly forced to concentrate on the fact that the VR environment they act in is not real, because they are not able to interact in a way they would in the real world.

Human-computer interaction (HCI) is nowadays subject of studies in many different areas. We will concentrate on HCI in the areas related to this thesis, that being everything concerning virtual reality, augmented reality, learning environments, simulations and mobile devices. In the following part of the thesis we will try to divide HCI into a few categories based on the subject of interest.

2.3.1 Target environment

When discussing different types of interaction methods, we can divide HCI into groups based on the environment the user has to interact with. Each environment is designed so that it fulfills its specific goal, therefore requires a different HCI approach.

Application specific

Simple applications (not VR) are designed to simply satisfy strictly defined requirements. Users are often informed of how to use the given application so that they can use it. The primary goal is to find interaction methods, that effectively cover all application capabilities, rather than to provide intuitive interaction methods. It does not matter, whether the user needs months of training in order to be able to use this application and years to master the whole potential. What matters is that someone who has already mastered such an application is able to work effectively and without any constraints.

---

[3] http://www.cabri.com/cabri-3d.html
[4] http://www.geogebra.org/

Augmented reality

Augmented reality (AR) enhances what we perceive with our senses with information that would normally be harder to compute in our minds. The most relevant applications of AR can be nowadays found for example in vehicles with interactive heads-up displays built-in the windshield. The driver can be provided with lot of relevant information ranging from velocity, traction control and fuel consumption, to navigation routes and destination weather. A new emerging field of application is mobile devices. Applications allow pedestrians to view relevant information by simply pointing their mobile device's camera at the buildings of interest. Users are provided with all relevant information based on accelerometer, compass and GPS data.

HCI for AR could be characterized as something in between of application specific and VR HCI. The aim is to provide the user an UI that he can control the application with, but at some point, where the application begins to partially act in the role of VR, enhanced interaction methods are required.

Virtual reality

As VR environments focus on simulating reality, human-computer interaction must correspond to the interaction possibilities in the real world. It has to be as intuitive as possible and for common tasks, it should allow the users to replicate these actions and receive the expected feedback. Replicating each and every possible interaction would however be impossible as even nowadays we do not posses such advanced technology. We evolve at a very quick pace and what was dreamed of a decade ago, is a standard nowadays. Simulating immersive VR environments has been evolving at a very fast pace as mechanisms as walking platforms like CyberWalk[5] and 3D video projections allow the user's to experience virtual environments with visual and motor sense.

2.3.2 Target audience

Human beings are unique and each and every one of us thinks in a different way. Special interaction approaches are required in order to satisfy the needs of different users.

Generic user

Focusing our VR environments on single user interaction allows us to focus on the interaction quality as there are no other attributes that need to be taken into account in

[5] http://www.cyberwalk-project.org/

order to create an intuitive, or even a realistic interaction. Creators can focus purely on the way a single user interacts with the environment.

Specific user types

Specific users have different needs. Especially handicapped people, who already interact differently with the real world, need to be treated specially when designing interaction methods.

2.3.3 Devices and technologies

Depending on the device in question, different interaction approaches have to be used. Some devices allow more straightforward techniques, minimizing the abstraction between the reality and the virtual environment.

Volumetric displays

Volumetric displays, unlike traditional flat screens that simulate the third dimension by various effects, display the object in three physical dimensions. In the work of *Tovi Grossman, Daniel Wigdor* and *Ravin Balakrishnan* [5] an autostereoscopic display is enveloped by a sphere that users can interact with. In their research they uncover several principles unique to volumetric displays:

- True 3D displays remove a layer of abstraction between input and display space, and thus tend to better afford gestural interactions.
- Because the display space is limited by the physical enclosure, all objects are within arm's reach. As a result, traditional 3D interaction techniques don't necessarily apply, necessitating the development of new techniques.
- Given that the display area is within an enclosure, gestures on and above its surface can be quite directly mapped to actions within.

Various input methods are being used nowadays to control content displayed on volumetric displays, but as stated in the work mentioned above, people have a tendency of touching the 3D imaginary floating in the display. Therefore the interaction with the outer envelope of the display seems the most natural choice. The interaction itself is based on tracking the positions of multiple fingers and can be categorized as simple motion tracking. Finger tracking could be used for any kind of volumetric display as it can be used independently on the display device.

Haptic surfaces

Haptic surfaces, known as touch surfaces are nowadays present in most mobile devices be it mobile phones, tablets or notebooks (trackpads). They provide a more direct way of input than common input devices like mice. With the emergence of multi-touch surfaces trackpads have made another great step towards substituting mice.

The evolution of touch surfaces, being more responsive, of greater size and with underlying active displays, motivates the search for the best interaction approach. Until multi-touch was introduced, these surfaces were a compact tool that simply substituted the mouse in moving the cursor on the screen. Multi-touch allowed the users to use multiple finger combinations and with the combination of drawing gestures brought trackpads one step ahead of mice. However they still lack the precision a pen tablet and velocity a mouse can offer.

Implementation of single touch interaction methods is straightforward as it does not differ from interaction capabilities of the mouse. It relies mainly on the user interface and therefore is mostly application dependent. UIs need to be optimized for specific tasks [6], where different approaches are evaluated. It is often convenient to let the user decide himself which approach he prefers.

Multi-touch surfaces have several issues that need to be addressed. Two or more fingers too close to each other can be processed as a single finger on the screen and when designing three or more finger gestures, one has to take into account the size of the screen as larger fingers may simply not fit the screen. Fingertip blobs affect the error rate [7] and their examination may help in proposing the user interface design.

Accelerometers, IR

Additional hardware components like accelerometers, infrared cameras and digital compasses can also be used to control various applications. Compasses can evoke rotation and IR cameras can determine relative location [8]. Accelerometers have found their way into mobile video gaming industry as they can simulate actions one would do in reality. Instead of steering a car by holding down buttons one can simply tilt / turn the device as a steering wheel[6,7].

Classic input devices

Input devices as hardware keyboards, mice and pen tablets do not need to be specifically described as they represent common forms of input. They are old but irreplaceable as

[6] https://market.android.com/details?id=net.osaris.turbofly

[7] http://itunes.apple.com/us/app/need-for-speed-hot-pursuit/id394732447

keyboards are the fastest text input devices, mice are of vital essence for every professional gamer and drawing without a pen is impossible.

2.3.4 Action processing

The action of creating the input signal can be categorized based on the required stimuli be it physical or mental. Different devices and techniques aim at various user skills.

Physical manipulation

Most devices nowadays require users to interact with the input devices by body movement. Without using the muscular system, one could not move the mouse, nor type on the keyboard.

Direct physical interaction techniques are those where a user changes a state of a physical object which then sends an appropriate signal to the computer. Most input hardware is based on this principle: a button has to be pressed, joystick has to be tilted, accelerometer has to be moved, etc. Each device has specific requirements that the user has to meet in order to use it efficiently. Some devices are aimed at generic usage, therefore do not require special knowledge, or skills, others however are not that easy to master especially when they are required to fulfill a specific goal.

Interaction techniques such as gesture recognition, where a user waves his hand in front of a camera have recently become the subject of many research projects. Even Grossman's et al. work [5] is based on this approach as the sphere dome itself serves only for finger guidance. The individual fingers are being tracked by a set of high-resolution infrared cameras. Another solution is digital image processing, where different objects are tracked by various algorithms that process image feeds from all kinds of cameras. In the Raffa's et al. research [9] a pipeline for efficient continuous gesture recognition is designed. Recent introduction of Xbox's Kinect controller[8] has brought indirect gesture recognition to its highlight.

Mental control

Recent studies have brought dreams and futuristic visions closer to reality. Until now interaction with simple thoughts seemed impossible. Research has once again brought down another barrier and the first mind controlling prototype devices are available[9]. By scanning the neural activity users can control objects by simply thinking of it. These

---

[8] http://www.xbox.com/kinect
[9] http://www.emotiv.com/

devices are still under heavy development and should they someday be available to generic IT consumers, environments will have to be adjusted in order to be able to efficiently collaborate with mind reading devices as completely different interaction approaches will have to be used.

3 EVALUATING EXISTING APPROACHES

3.1 Vision

From the analysis above, some of our aims are already clear. Our primary goal is to find an effective interaction method for mobile devices equipped with multi-touch screens. As most multi-touch practices focus purely on 2D environments the interaction methods in 3D environments are not yet standardized. 3D games running on mobile devices with Android and iOS operating systems use some common guidelines, but those would not satisfy our needs as they do not grant the user the six degrees of freedom (DoF). Our secondary goal is to create a simplified VRLE where students are motivated by means of game and competitiveness to improve their knowledge in geometry, specifically of cube cross sections. Therefore the main goals are:

- find an efficient multi-touch interaction method
- create a VRLE for geometry education environment that can be characterized as a video game
- target devices are mobile devices with multi-touch capacitative displays

3.2 Existing applications

Our interest falls into two application types. As we want to create a 3D geometric multi-touch application we focus on existing IGSs (Interactive Geometry Software) and iOS and Android applications that somehow process 3D data. While the IGSs are listed just to present different capabilities of geometry systems, in the mobile application section we focus more on the interaction techniques used to manipulate objects in 3D environments. From the listed applications it will become clear that no interaction approach has yet been standardized and therefore applications interpret interactions as they best suit the specific needs. For comparison we analyze the DoF the application supports.

3.2.1 IGS

We focus on the features these systems provide in order to be able to choose an application domain for our multi-touch interaction technique.

Cabri 3D[10]

From the many available 3D IGSs we have chosen Cabri 3D [16] as it is probably the best propagated one. It found its way into education and helps students to comprehend the individual problems of solid geometry.

Cabri 3D lets students experiment with geometric objects in various ways. From drawing simple points and combining them into lines and rays to unfolding 3D shapes. Students are able to explore 3D objects from any view, see cross sections of these shapes and toggle between wireframe and surface displays. Cabri 3D connects geometry (as seen in Figure 1) and algebra by measuring length, angles, area and volume and then attaching these numeric values directly to the figure to use them in calculations or in expressions using fundamental algebraic concepts, such as numbers, variables and operations. It takes time to master all the features this application offers, but once mastered, Cabri 3D can become a powerful educational tool in teacher's hands.

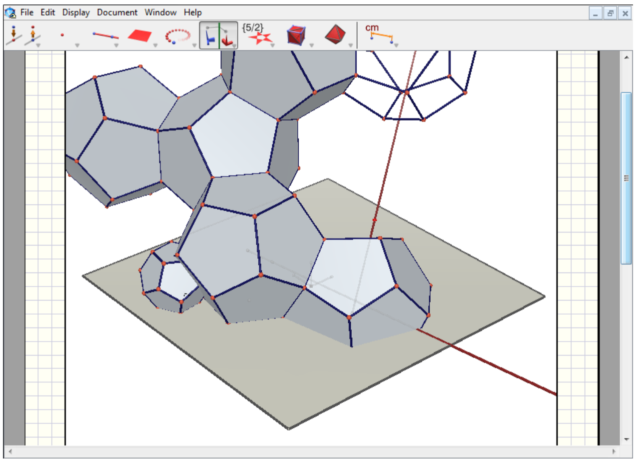


Figure 1 – Cabri 3d

What Cabri 3D lacks, is the full control over the 3D environment. The scene with objects is concentrated around the XZ plane. Controlling the height (Y axis) in which the objects are located is impossible as well as is rolling the scene (rotation about the X axis). The zoom in and out functionality is either missing or I simply did not figure it out in half an hour's time. Even thou the interaction with the 3D environment is limited, this approach partially helps the students to comprehend geometry as it eliminated the need of processing additional perspectives.

Apart from Cabri 3D there are many other, very similar IGSs, e.g.:

- Yenka 3D shapes[11]

[10] http://www.cabri.com/cabri-3d.html

- Archimedes Geo3D[12]
- GeoGebra[13]
- Geomspace[14]
- Geometria[15]

Aplollonius

Apollonius is probably one of the first IGS applications available for mobile devices. This application however works only in the field of planar geometry where the interaction techniques are relatively straightforward.

Single finger actions are used for object translation and selection while with two fingers on the screen the canvas can be panned, zoomed and rotated.

3.2.2 Mobile applications

As we focus on multi-touch displays we analyze a few applications that are available at the Apple store and Android market. Both Android and iOS operating systems support multi-touch mobile devices. Chosen applications are free and manipulate 3D environments and objects.

The devices the applications were tested on were an apple iPod 2nd generation and a HTC Desire HD.

iSculptor[16]

iSculptor is a 3D modeling application for the iPhone, iPod Touch and iPad. By moving and editing polygons and vertices one can create 3d models that can then be imported into most popular 3D packages that accept wavefront OBJ files.

This application provides a huge quantity of various actions one can execute on the model. The UI consists of a toolbar with buttons changing modes that determine what the various action and gestures mean. A dominant part of the UI is a thick border around the canvas (Figure 2) which serves as a scroll panel and enables users to rotate the canvas around fixed axes that do not rotate with the object.

---

[11] http://www.yenka.com/
[12] http://www.raumgeometrie.de/drupal/en
[13] http://www.geogebra.org/
[14] http://sourceforge.net/projects/geomspace/
[15] http://geocentral.net/geometria/
[16] http://itunes.apple.com/us/app/isculptor/id370525280?mt=8#

One finger interaction is used for object, surface and vertices selection as well as object transformation in the different modes where selected objects are to be transformed.

Two finger interactions are used for panning the canvas as well as for zooming by using the pinch gesture.

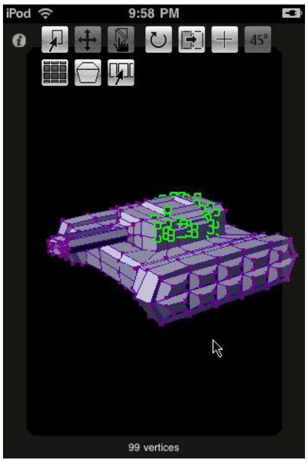


Figure 2 - iSculptor[17]

iDough[18]

Another application aimed at Apple mobile devices. This application was one of the first I encountered that used multiple fingers for something else than the pinching gesture. Aimed as much at professionals as at beginners, this application is a powerful portable sculpting tool.

Control over the 3D environment is rather limited as the application provides only one object to interact with. This object is always positioned in the center of the screen and no translations are required. Rotations are however limited as well. Rotation about the Y axis is not limited, but rotation about the X axis is limited to an angle of 180°. Rotating about the Z axis is done indirectly through a series of rotations of the other two planes. Rotations are applied to the coordinate system itself.

One finger interaction executes the selected command on the object independent on whether the finger just taps the object, or is dragged along (start point must be on object).

Two fingers dragged from side to side rotate the object about the Y axis, while dragging two fingers from top to bottom, or the other way around, makes the object rotate about the X axis. The standard pinch gesture is used to zoom-in / zoom-out.

[17] http://a2.mzstatic.com/us/r1000/004/Purple/5c/c6/c2/mzl.muwqgnrz.320x480-75.jpg
[18] http://itunes.apple.com/us/app/idough/id386752314?mt=8#

Even thou this application's DoF are limited, most users are more than satisfied and are able to create astonishing creations.

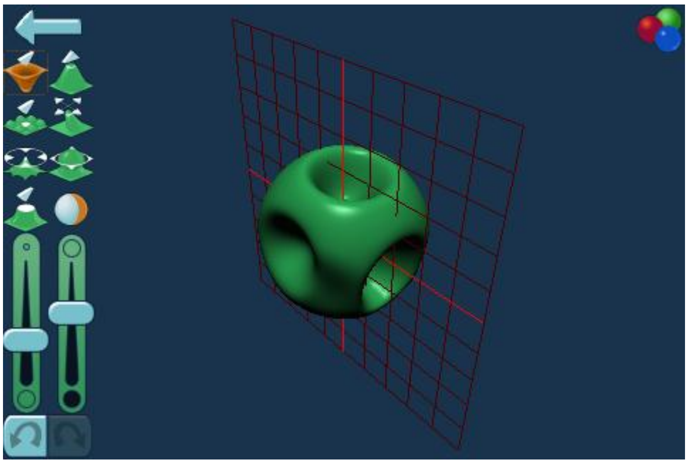


Figure 3 - iDough

LookAtCAD

This iOS application is a simple viewer of 3D models. It allows users to connect to FTP locations and download 3D object stored in different formats. Later on users can load downloaded objects and observe them from different angles while the object is always centered on the screen (no translations)(Figure 4).

Rotations about the X and Y axes are allowed by simple dragging of on finger.

Two fingers are used to zoom in and out using the pinch gesture.

This application is an exhibit that demonstrates how a wrong interaction approach can demotivate the user. If one wants to view an object from a specific angle, he need to be very patient as rotating the object is the opposite of intuitive.

Figure 4 - LookAtCAD

i3dViewer[19]

This is another simple model viewing application, but this time, with a more friendly interaction approach. When viewing models the user controls the camera. By one finger dragging he can change the viewing angle, while when dragging with two fingers he can move the camera sideways.

Figure 5 - i3dViewer

ModelView[20]

One of the few free Android applications available in the market that allow the user to view .OFF and .OBJ files. As the author himself states, this project is "just a learning exercise and serves no real purposes". The project started a simple OpenGL learning project. I find this application worth of mentioning as there are not many like it (Figure 5).

One finger dragging rotates the object. Dragging from side to side, the object can be rotated about the Y axis, while dragging vertically rotates the object about the X axis. Along with the object the whole coordinate system is being rotated, therefore roll rotations (rotation about the Z axis) are impossible to achieve.

Two fingers are used only for the pinch gesture that allows zooming in and out.

What I like about this application is its spin feature. When releasing a dragged finger without stopping it first, the object keeps spinning at a decelerating rate until it comes to a halt.

[19] http://itunes.apple.com/us/app/i3dviewer/id371652694
[20] http://zerocredibility.wordpress.com/2010/12/07/3d-model-viewer-for-android/

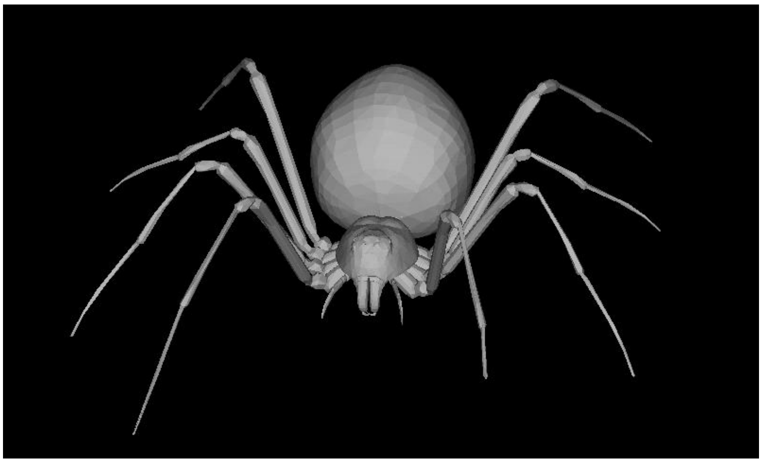


Figure 6 - ModelView

Nao3d Viewer Free

Nao3d Viewer is another Android application for viewing 3D models. This application however lacks user friendly manipulation. Objects can be rotated about an invisible centre placed at their bottom by dragging one finger across the screen. The relation between distance the finger travels on the screen and the angle to object rotates is set incorrectly as one finger stroke hardly rotates the object. This application is an example of user-unfriendly interaction.

AutoCAD WS[21]

This android application lets users explore complex 2D and 3D models. The application itself comes with a few samples, to demonstrate the application capabilities. The interaction technique is very intuitive and one of the best we came across. It uses common approaches and lets the user rotate the object with one finger and translate with two. The roll action is mapped to a two finger circle gesture. What we found unpleasant was that the rotation by one finger dragging depends on where you start the gesture. Therefore it behaves inconsistently and some users might find it confusing.

Apart from this minor flaw the application has the best interaction technique we have noticed so far.

[21] https://play.google.com/store/apps/details?id=com.autodesk.autocadws

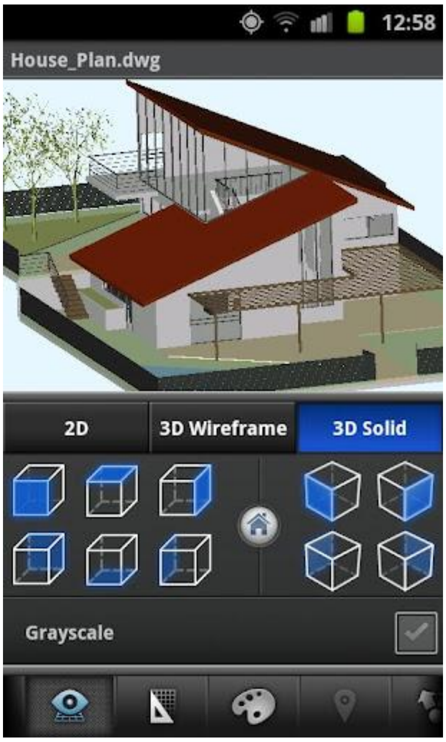


Figure 7 – AutoCAD WS

Other applications

There are a few more probably relevant applications on the mobile market. But as they are paid applications, were not accessible for us and we found no relevant information, we could not categorize them as the ones mentioned above. Among these applications are:

- iCrosss (iOS)
- 3d Geometry (iOS)

3.3 Common multi-touch interaction approach

This section examines multi-touch interaction techniques that are backed up by scientific research. We start off by presenting solutions of existing contributions in this area of research. The various techniques are analyzed in detail in tables containing the following columns:

- DoF – which DoF this action covers
- Action
    - as <fingers_on_screen_count>F action
    - “3 pinch” would mean that 3 fingers need to be in contact with the screen and the pinch gesture has to be performed
- Result

3.3.1 Research based techniques

Fiorella et al. [10] conducted an experiment comparing classic button UIs with multi-touch UIs. Their multi-touch interaction technique supports only 4 DoF. I suspect this to be the reasons, which lead them to the conclusion, that "further work is needed in order to achieve a completely satisfactory gesture mapping implementation".

Table 1 - Fiorella et al. multi-touch interaction technique

DoF	Action	Result
forward / back	N/A	
roll	N/A	
yaw	1F horizontal drag	rotate about the *Y* axis
pitch	1F vertical drag	rotate about the *X* axis / pitch
left / right	2F horizontal drag	translate along the *X* axis
up / down	2F vertical drag	translate along the *Y* axis

Hancock, Carpendale and Cockburn have designed three interaction techniques to manipulate 3D objects on tabletop displays [11]. However, only their multi-touch techniques allow the six degrees of freedom. Their aim was to develop shallow-depth interaction techniques for tabletop displays, therefore their main aim was to provide only 5 DoF:

- *x* & *y* – the position on the surface of the table
- object rotation about the *z*-axis / yaw
- object rotation about the *y*-axis / roll
- object rotation about the *x*-axis / pitch

The first proposed is a two finger technique based on the Rotate'N Translate (RNT) algorithm [12]. This technique lacks the yaw DoF as smaller displays (such as mobile phone displays) cannot take full advantage of the RNT algorithm.

Table 2 - Hancock et al. two finger multi-touch interaction technique

DoF	Action	Result
yaw	1F drag	rotate about the *Z* axis by moving the point of contact closer to the finger
forward / back	1F vertical drag	translate along the *Y* axis
left / right	1F horizontal drag	translate along the *X* axis
roll	2F vertical drag	rotate about the *Y* axis / roll
pitch	2F horizontal drag	rotate about the *X* axis / pitch
up / down	2F pinch	translate along the *Z* axis

The second proposed technique is a three finger multi-touch technique. The mappings are listed in the table below.

Table 3 - Hancock et al. three finger multi-touch interaction technique

DoF	Action	Result
forward / back	1F vertical drag	translate along the *Y* axis
left / right	1F horizontal drag	translate along the *X* axis
yaw	2F drag	rotate about the *Z* axis
up / down	2F pinch	translate along the *Z* axis
roll	3F vertical drag	rotate about the *Z* axis / roll
pitch	3F horizontal drag	rotate about the *X* axis / pitch

In their work they state that "there has been a general consensus about the separability of rotation and translation. It is widely believed that input is superior if these are kept separate", but on the other hand at the end of their work they state that "People are not only capable of separable simultaneous control of rotation and translation, but prefer it". We believe that whether it is an advantage or a disadvantage depends from the target application. Tabletop displays require fast and imprecise manipulation with objects and therefore do not suffer from minor undesired transformations. Geometry applications on the other hand, could become frustrating to use especially on smaller screens, where finger precision is not as accurate as on larger screens.

Martinet et al. [13] embrace Hancock et al.'s three finger technique as the Z-technique and compare it to the standard viewport technique enhanced with multi-touch capabilities. Their controlled experiment shows that both techniques are equivalent in performance, but the Z-technique was preferred by most participants.

4 SPECIFICATION

In this chapter we deduce requirements based on set goals with the aim to create an application that will help us to find an effective and intuitive mean of interaction with abstract objects in a 3D world.

4.1 Goals

So far we have analyzed various applications that have something in common with interaction in 3D environments. Based on our research we were able to specify the following goals, which will be discussed in more detail in this chapter:

- geometry education,
- mobile devices,
- multi-touch interaction,

and a special goal

- motivation (see section 4.1.4)

4.1.1 Geometry education

Designing a robust system that would be capable of educating students and would cover whole middle school knowledge, would require a lot more time than one can dedicate to this diploma project. This goal is just a secondary goal. We will focus on applying our solution in the field of solid geometry because of the following reasons:

1. *three dimensions*: as our solution focuses on the interaction in 3D environments, solid geometry is the logical choice
2. *user feedback*: teacher and student feedback on how well the application eliminates the need of a good perspective understanding in order to understand the curriculum is of great importance as a fully immersive VR would require none at all
3. *education*: ongoing research has proven that alternative educational approaches can lead to better results with students and we would like our application to be a contribution in this area as well

4. *innovation*: at the moment we found no 3D IGS mobile applications, with our application we will be the first to offer the combination of a 3D environment educating geometry

Our application (as a side result of this thesis) will only cover a specific subject of solid geometry. We have chosen cube cross sections as our definite application domain. Covering all possible actions that students are required to execute in order to successfully solve cube cross section tasks should lead us to building the core of the application in a manner that will allow further implementation of other solid geometry curriculum.

The application itself should allow students to solve constructive problems using methods they learn at school. Basically the functionality our application has to allow is almost the same as the application Rezy kocky[22]. That would be:

1. *vertices*: creation, selection, positioning and removal of vertices on existing lines, rays, line segments and even planes
2. *lines, rays, line segments*: creation, selection, transformation and removal
   - defined by two vertices
   - parallel lines defined by a source line and a vertex that the parallel line will pass through
3. *planes*: creation, selection, transformation and removal
   - defined by 3 vertices that are not part of a single line
   - defined by a line and a vertex that is not part of the line
   - the slicing plane can be directly manipulated
4. *cross section wizard*: the "teacher"
   - animated hints
   - animated step by step tutorial
   - instant solution
5. *the cube*: the cube itself can be manipulated as well
   - un/folding the net
   - splitting the cube into two separate pieces
6. *measurements and calculations*: in later development phases support for measuring distances, angles and calculation of volumes could be implemented

[22] http://www.infovek.sk/predmety/matem/index.php?k=313

4.1.2 Mobile devices

Present fast paced evolution of mobile devices has caused that even elementary school students own smartphones with huge multi-touch displays. Mobile carriers often offer high-end devices much cheaper with two-year wireless contracts and make them available to the generic customer. On the assumption that 90%[23] of students own mobile phones, from which 20% (and growing steadily) can be categorized as smartphones, we conclude that developing an application for the iOS or/and Android platform is the most reasonable option. Application ("app") distribution can be as simple as downloading the application directly from Apple App Store or Google Android Market.

We decided to develop for Android for the following reasons:

1. *constantly growing user base*: even thou Apple still holds the majority of market shares, Android is catching up fast
2. *extended developer support*: for the open source nature of Android there is a large community of developers eager to help one another
3. *app distribution*: apart from Google Android Market, apps can be distributed via third-party sites and by the devices themselves
4. *affordable devices*: Android phones as well as tablets are considerably cheaper than the Apple iPhone or iPad (Android 4.0 Ice Cream Sandwich tablet NOVO7[24] costs just $100)

As this thesis focuses on a specific interaction method, to be able to experience the full potential of our app one will need a device that meets certain specifications. The screen size is the most relevant because we will be using three finger gestures and comfortable use can be hindered by fingers covering up the whole environment visualization area of small screens.

4.1.3 Multi-touch interaction

Our primary goal is to design an effective interaction interface that will give the users maximum freedom of interaction within a 3D environment. As mentioned in the analysis, most existing applications lack the 6 DoF. Our aim is to experiment with various interaction approaches and evaluate them.

[23] http://www.textually.org/textually/archives/2005/02/007109.htm
[24] http://www.ainovo.com/

Based on assumptions that we developed through the examination of techniques mentioned above, we designed our own technique (Table 4) that will be implemented in our prototype application. By conducting a series of experiments and user testing we will then be able to modify and improve this solution.

Table 4 - Custom multi-touch interaction technique

DoF	Action	Result
left / right	1F horizontal drag	translate along the *X* axis
up / down	1F vertical drag	translate along the *Y* axis
yaw	2F horizontal drag	rotate about the *Y* axis
pitch	2F vertical drag	rotate about the *X* axis
	2F pinch	change size / zoom
roll	2F circle	rotate about the *Z* axis
forward / back	3F vertical drag	translate along the *Z* axis

The forward / backward movement could be mapped to a different gesture, e.g. 3F pinch or 3F horizontal drag. As you can see, the 2F pinch action has no DoF mapped to it. We are sure, that everyone uses the 2 finger pinch gesture to change zoom levels and that is not exactly the same as translation along the *Z* axis.

We believe that this technique could be used in any 3D environment that requires precise manipulation as the task of translation is separated from rotation. A proposed list of detailed actions is available in the table below.

The current technique mapping is aimed at manipulating either the camera or an object in the scene. The suitable camera types are either the look at camera, where the camera is fixed on a point in the scene and rotations result in the camera being rotated around this point, or a normal camera where the rotations are done by the camera itself (around its center). First person cameras would not benefit from these mappings. However with minor changes to the various DoF gesture mappings, means of controlling a first person camera could be achieved.

Table 5 - Detailed multi-touch interaction technique description

Action	Condition	Result
1F tap	target = unselected vertex selection = null	create new selection add vertex to selection
1F tap	target = unselected vertex selection = active	add vertex to selection
1F tap	target = selected vertex	remove vertex from selection
1F single tap	target = void	deactivate selection
1F single tap	target = object	activate selection
1F double tap	target = object	add each vertex from group to selection
1F double tap	target = void	clear selection
1F drag	origin = unselected vertex selection = active	add each vertex on drawn path to selection

1F drag	origin = selected vertex selection = active	remove each vertex on drawn path from selection
1F drag	target origin = object / void selection = active	move selection along the X and Y axis
1F drag	selection = inactive	pan camera left, right, up and down
2F horizontal drag	selection = active	rotate selection about the Y axis / yaw
2F vertical drag	selection = active	rotate selection about the X axis / pitch
2F horizontal drag	selection = inactive	rotate camera about the Y axis / yaw
2F vertical drag	selection = inactive	rotate camera about the X axis / pitch
2F pinch	selection = active	change selection size
2F pinch	selection = inactive	camera zoom in and zoom out
2F circle	selection = active	rotate selection about the Z axis / roll
2F circle	selection = inactive	rotate camera about the Z axis / roll
3F vertical drag	selection = active	translate selection along the Z axis
3F vertical drag	selection = inactive	move camera along the Z axis

Apart from multi-touch interaction we would like to implement the possibility to control the environment using the accelerometer available in most smartphones nowadays, as the hardware input can be mapped straight to the 6 DoF.

4.1.4 Motivation

One might wonder why we focus on motivating potential users as early as in design stages. By motivating users we aim to:

1. receive better feedback (better in quality as well as quantity)
2. increase knowledge acquisition effectiveness

Motivation can be achieved in various ways. We decided to go with the "school by play" principle. Apart from simply being an educational tool, we would like to turn Cube Cross Sections 3D into a game where students can compete against each other. The game highlights would be:

- player rankings on the mobile social gaming network openfeint[25], gained points are effected by:
  - speed (time it takes to solve the puzzle)
  - accuracy (number of actions taken)
- stunning animations (e.g. particle effects)

[25] http://openfeint.com/

4.2 Requirements

After reviewing our goals, we were able to transform them into requirements and plans that helped us to develop our prototype application. We decided to name this application "Cube Cross Sections 3D" or "CCS 3D" in short.

4.2.1 Functional requirements

Based on the analyzed goals we set the following functional requirements. Users, or rather players, can choose between two game modes. The first, simple mode lets the users manipulate a plane. By moving and rotating the plane various cross sections can be achieved and the full potential of the 6 DoF can be unleashed. In the advanced game mode, the goal is to construct the cross section plane starting with just 3 points on the cube net. In this advanced mode, the user has to interact with points, segments and lines. However translations will not be used that often as most of the time the cube will be rotated. Translations will be used in cases an intersection of two lines originates "outside" of the screen. Interaction with objects in this mode mainly consists of object state / existence manipulation (create, remove, hide) and rotations (3 DoF). In order to be able to extensively evaluate our interaction method, the application would greatly benefit if user usage statistics were collected. The most important requirements for our application are summarized in the following figure (Figure 7), where the core requirements are the bright ones and the gray ones represent requirement that would provide a robust application.

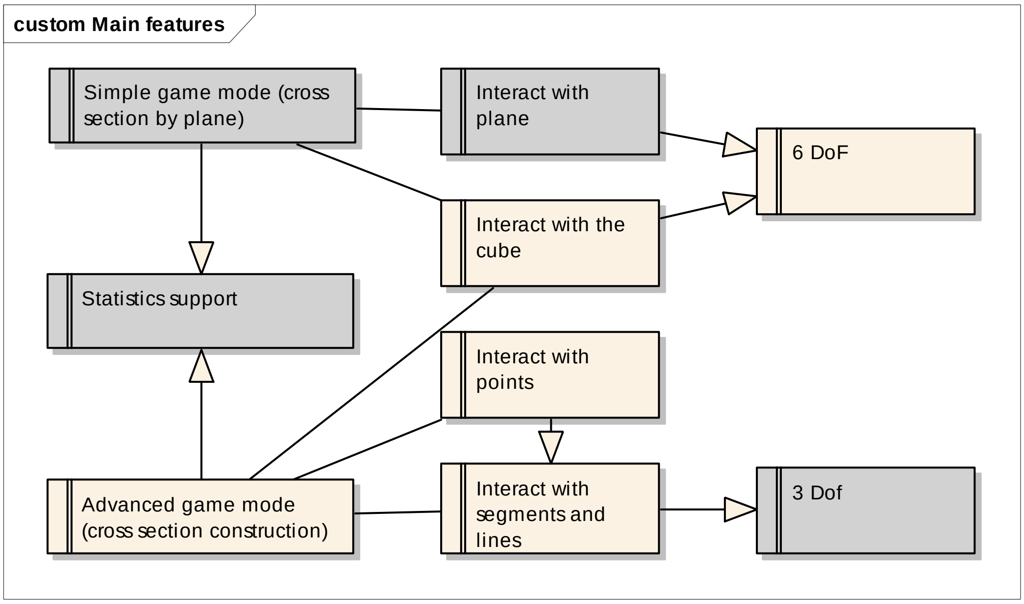


Figure 8 - Functional requirements

Apart from these requirements we defined a few complementary ones, which are not of low importance. However if the application met those requirements, it would greatly benefit from them.

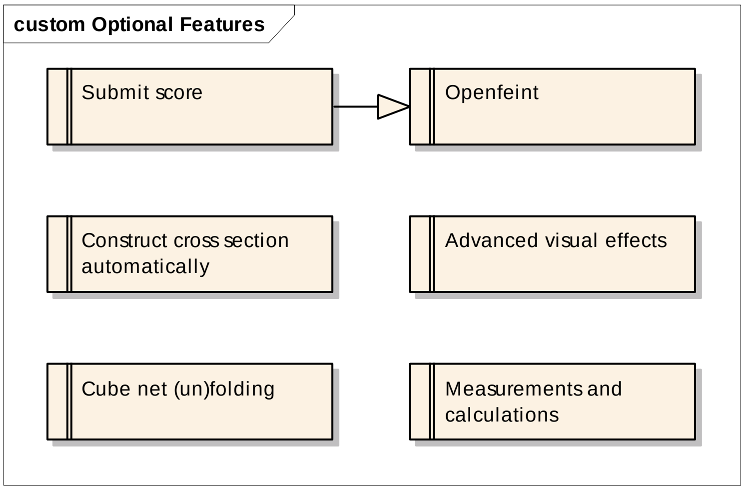


Figure 9 - Optional functional requirements

4.2.2 Non-Functional requirements

As this thesis focuses on multi-touch interaction, the device that will run our application has to have multi-touch support. Most devices running nowadays have multi-touch capable screens. To support older devices as well however, apart from having an adequate screen, the device has to run on Android 2.0 or higher, as lower versions do not support multi-touch. The following are the non-functional requirements:

- Android 2.0 Eclair
- Multi-Touch screen
- Screen size > 3”

Optional:

- Accelerometer
- Internet access

5 CCS 3D SOFTWARE DESIGN

In the first part of this chapter, the general architecture of our application is discussed and presented in UML notation. After a brief overview the individual components are discussed in more detail. The next part describes the application features with a set of use cases. And at the end of this chapter a simple user interface has been sketched.

5.1 Architecture

The designed application will consist of 3 basic modules as displayed on the following diagram (Figure 9):

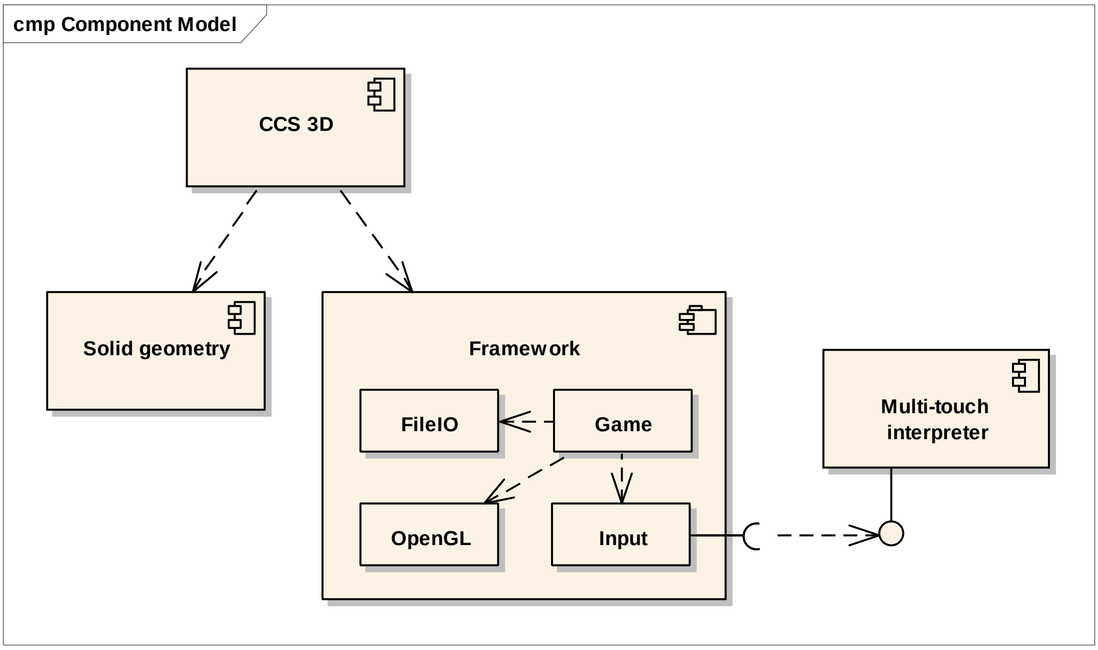


Figure 10 - Component model

5.1.1 CCS 3D component

This core component is the game itself. It contains the logic that ties the other components together. The role of this component is to allocate assets, communicate with the device, make sure it meets the specified requirements, call the individual activities when user navigates through the various screens, etc.

5.1.2 Solid geometry component

This component take care of all the logic associated with geometry. Classes as Point, Line, Plane and Cube will represent the various objects on our scene. This component represents the imaginary world of a cube where all our objects that users can interact with reside. This world has to be able to adapt and respond to various impulses, mostly user invoked.

5.1.3 Framework component

Our application will be built on top of a framework. The most basic subcomponents that our framework should have are Game, OpenGL and Input (as displayed on Figure 9).

The Game subcomponent is the "mastermind" of the framework. It delegates tasks to the various subcomponents and makes sure everything works as it is supposed to.

The OpenGL component, as the title states, takes care of OpenGL ES. We decided to go with OpenGL 1.0 and 1.1 as OpenGL ES 2.0 is not supported on older devices. Tasks as rendering the final frame, double buffering, drawing sprites, applying projection and model matrices are all being executed within this component.

The Input subcomponent is the most relevant for us. It processes all input methods and events and allows us to use the input devices in a simple manner. The minimal requirements for an Input component are multi-touch and accelerometer support.

5.1.4 Multi-touch interpreter component

Probably most of the frameworks will need to be adjusted or provide some extension capability in order to support our various multi-touch interaction methods. And that is where our Multi-touch interpreter component will be used. Connected with the framework Input component it will be able to process the various events into predefined gestures and output easy to interpret values that can be used to alter the objects states.

Apart from providing the user with a means of interaction, this component would be responsible for detailed input data collection. It would collect information on how the users interact with the application. This data will contain information such as when a finger touched the screen, how long was this finger down, what distance did it travel, were any other fingers down at that time, etc. This data output combined with information regarding the state of the game world (the cube) will allow us to later examine what actions did a user take to accomplish various tasks.

5.2 Use cases

Based on the defined requirements we defined a number of use cases that describe how users can interact with our application. The most important use cases which are crucial for our application are the ones that extend the “Start new game” use case. The others, such as “Start application”, “Change settings”, etc are of minor importance and can be applied to almost any game app. However the “Cross section by plane intersection”, “Cross section construction”, and use cases that extend them are application specific and define the behavior of our game app. All the use cases are available in the following figure (Figure 10). Their titles should be self-explanatory.

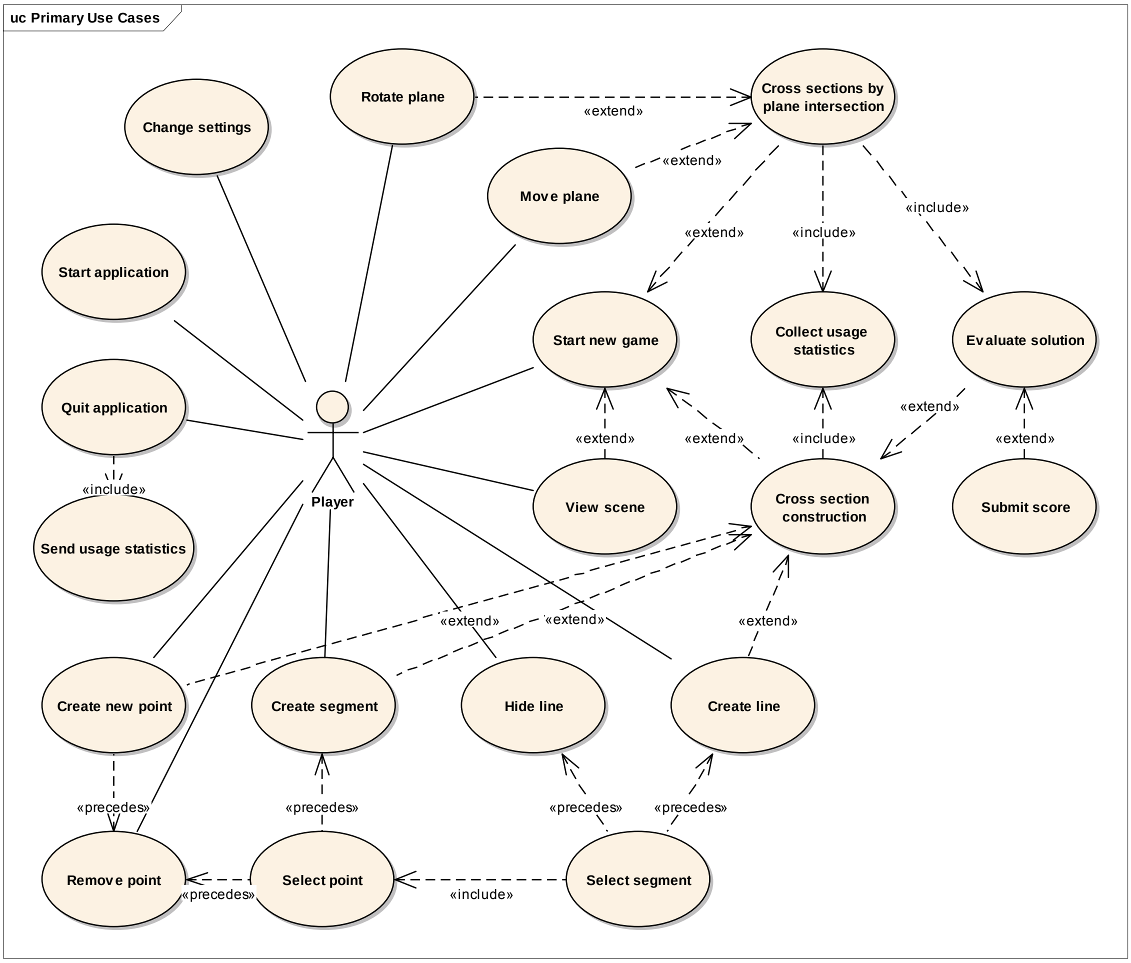


Figure 11 - Use cases diagram

5.3 User interface

The user interface will allow only touch interaction, so in order to navigate to the next screen the user has to touch the appropriate button, or use the back button available on all Android devices. The game consists of five activities (see Figure 11), where an activity in Android represents a screen:

- Main menu: when the application starts the main menu screen activity is launched. From here the user can navigate to the settings, highscore, help and game screens. In order to correctly terminate the application the user has to exit from this activity.
- Settings: here the user will be able to change the various interaction methods. We will propose a simple interaction method based on a button interface as well as multiple multi-touch interaction methods. Users will be able to control snapping and hints as well.
- Highscores: here users can find a simple list of user nicknames and the points scored in the two game modes.
- Help: this activity will contain a series of simple screens each holding a picture and text that will explain to the users, how to interact with our application. The following help screens will be available:
    - a help screen describing the possible interactions with the cube
    - at least one screen for each interaction method
    - a screen describing each of the two game modes
    - a series of screens explaining the construction of cube cross sections
- Game screen: this screen will change according the selected game mode and interaction method. On this screen the cube will be displayed and users will be able to interact with it.

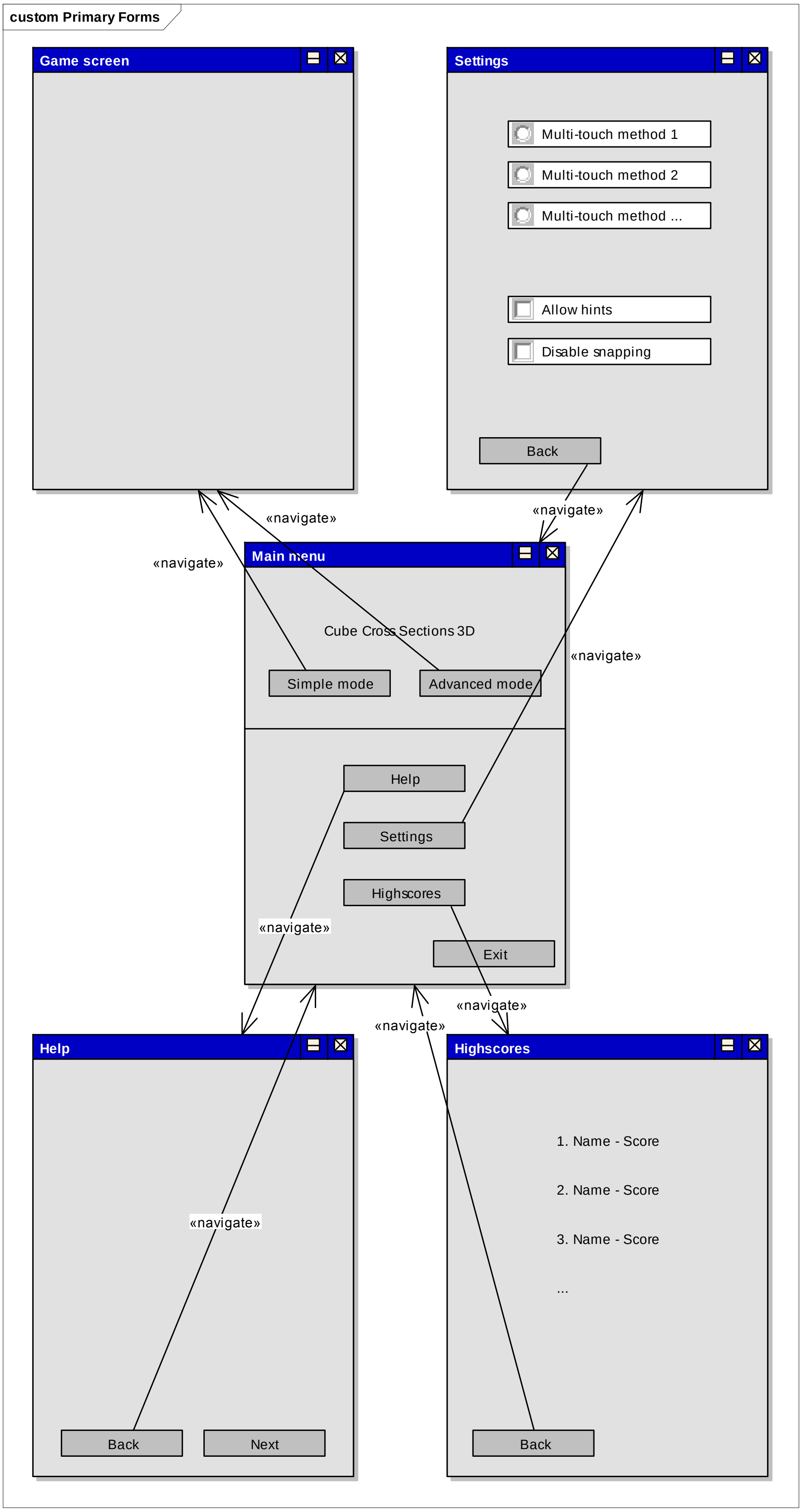


Figure 12 - User interface design

6 IMPLEMENTATION

This chapter contains details regarding the process of implementing the Cube Cross Sections 3D application. There were various obstacles and challenges that we had to overcome during this process and some had impact on the application. This chapter is divided into two parts: the prototype and the final application.

6.1 The prototype

We built a custom framework while learning Android basics. We believe that this was the best choice, as it allowed us to keep up with the deadlines and deliver an acceptable prototype on time. The prototype itself consists of a few activities that prove that on our framework and geometry component is able to construct the final application.

As of our development environment we work with the following:

- Windows 7 Professional operating system
- Java SDK, Android SDK
- Eclipse Indigo integrated development environment with the ADT plugin
- HTC Desire HD as our testing device

The Android SDK comes with an AVD (Android Virtual Device) manager, which provides us with the option to test our applications on the Android emulator. However when it comes to testing OpenGL and multi-touch interaction, the emulator is simply not enough. It has only limited OpenGL ES support and OpenGL heavy applications simply do not run as smooth as on the target device. Multi-touch interaction is crucial for our application and having to simulate simultaneous touches with a single mouse pointer would be frustrating and slow us down. Therefore all testing was done exclusively on the HTC Desire HD.

6.1.1 The framework

The framework was developed while getting to know Android programming. We closely followed the book Beginning Android Games [15], that allowed us to understand all the details and complexity of the framework in question. The framework is aimed at OpenGL games. We omitted parts of it that were of no interest to us, parts like audio streaming, .OBJ file importers, etc. Our framework contains only classes that are needed for our application (for more details consult Appendix B).

6.1.2 The prototype application

The prototype application itself is made of 4 test activities. Each of them tests a certain area. Apart from the visual output on the device we use LogCat in Eclipse to access debug messages sent from the device.

WireframeTest

This simple activity was the first one created. It served us to learn how to rendering the cube. The cube is being drawn with the GL10.GL_LINES OpenGL primitives. The lines are a bit jagged, so we tried applying antialiasing. However this resulted in very thin – hardly visible lines, which ignored line width parameters. In the end we decided not to use antialiasing as older devices might not support it.

LettersTest

This activity was designed to test the description of the points of the cube with letters. The SpriteBatcher class did not provide the needed functionality. It lacked the ability of placing the sprites into different depths so we had to extend it into the FontBatcher class which allows us to place texture sprites containing the letters anywhere in the scene.

Multitouchtest

Most attention was dedicated to this activity. The result proves that, with our framework we will be able to create a multi-touch interpreter component that will handle these interaction methods. We took the liberty of experimenting with a new interaction technique aimed at 3 finger gestures focused on rotation (Table 6).

Table 6 - Prototype multi-touch interaction technique

DoF	Action	Result
pitch	3F - rotate 2F around fixed 1F	rotate about the X axis
yaw	3F pinch	rotate about the Y axis
roll	3F - rotate 1F around fixed 2F	rotate about the Z axis

Testing this interaction method brought us to several conclusions, which are discussed in the prototype evaluation chapter.

LinesTest

This activity presents in a simplified form the various display options for our main objects. In this activity the user can:

- toggle point visibility
- toggle segment visibility
- toggle line visibility

- create a point at a random position on the wireframe / remove the random point

6.2 CCS 3D

In the last phase of our implementation the prototype has been turned into our CCS 3D application. Due to limited time we had, only the core functions were implemented. The application has not become a game, but it sufficed to be able to evaluate the interaction technique. The application consists of three activities.

SurfaceCutTest

This is the prototype version of the second application mode, where the user manipulates a plane cutting the cube. The plane can be rotated and moved around in the 3D space.

F12CCS

This is the Cube Cross Section construction mode with the two finger technique used. Apart from 6 DoF camera manipulation, users are presented with 6 icons. They provide the following functionality:

- Toggle point names
- Construct parallel line – when a user select two points on an existing segment or line, he can click on this icon to activate the construction of a parallel line; after that he just needs to select a point through which the new parallel line should pass
- Toggle segment – toggles visibility and in case no such segment exist, a new one is created
- Toggle line – toggles visibility and in case no such line exist, a new one is created
- Toggle selection – when this icon is clicked it deselects all selected points and temporarily stores their state; in case the user wants to return to the previous selection, he simply clicks the button again
- Evaluate cut – when all points of the cut are selected and this button is pressed, the cut is highlighted

F23CCS

This is the Cube Cross Section construction mode with the three finger technique used. Apart from exactly the same functionality as for the F12CCS activity, this activity allows the user to do consecutive point selection by simply dragging the finger across the point (without having to lift the finger off the screen to select another point).

7 TECHNIQUE USABILITY

After having designed a multi-touch technique, its usability had to be tested. A series of tests were performed in various stages of the implementation. Based on the evaluation of each phase the techniques were adjusted and in the end, the final designs were submitted to thorough evaluation. The main emphasis was on the usability and user acceptance.

7.1 Prototype evaluation

Based on the empirical evaluation of the first interaction technique implemented in the prototype, we designed a new technique (Table 7) that was later compared to the one designed in the first phase (Table 4).

The use of three fingers for the task of translation on the Z axis was not intuitive and limited the user to a simultaneous translation on a maximum of two axes. In order to introduce three finger touch interaction it requires it to be a great improvement over various current approaches in order for the users to accept it. Simply adding three fingers as a new gesture, especially when translating on the Z axis is almost identical to the zooming action (based on the specific application) would not convince the user to adapt to using three fingers.

In the prototype an experimental technique (Table 6) was applied as well. Its mappings were not designed to be intuitive, but mapped all the three rotational DoF to three fingers. Based on this experiment we further came to the conclusion that complex three finger gestures are difficult to use because:

- three fingers obscure the objects displayed on the screen and therefore lack detailed visual feedback
- most phones screen sizes limit the user “workspace” as they are not big enough and finger movements are limited to only short strokes
- gestures that in combination allow rotations around more than one axis simultaneously are difficult to adopt, unless simulate real world experience
- unintuitive gestures will not be adapted by the general user

With regard to these revelations, a new technique has been proposed after our alpha-testing. This technique described in the Table 8 focuses on the following:

- gesture grouping for separate interaction categories

    - 1 finger for selection
    - 2 fingers for rotation
    - 3 fingers for translation
- intuitive use
    - XY plane common gesture approaches (simple dragging results in translation on the XY plane – touchpad like)
    - pinch gesture (translation on the Z axis)

Translation has been chosen to have the three finger gestures mapped because it is the easier task. Rotation being more difficult should be mapped to two fingers so that the users can concentrate on rotating instead of laying three fingers on the screen. Using three fingers for the pinch gesture leaves two fingers for the pinch gesture that could be used for zooming, however if not of vital importance, the zooming feature should rather be left out. Otherwise the user could end up confused, not being able to see the difference between zooming and translation on the Z axis.

Table 7 - New interaction technique based on prototype evaluation

DoF	Action	Result
left / right	3F horizontal drag	translate along the *X* axis
up / down	3F vertical drag	translate along the *Y* axis
forward / back	3F pinch	translate along the *Z* axis
pitch	2F vertical drag	rotate about the *X* axis
yaw	2F horizontal drag	rotate about the *Y* axis
roll	2F circle	rotate about the *Z* axis

For a more detailed explanation of the described multi-touch technique, please refer to Appendix A.1.

7.2 CCS 3D beta-testing

While developing the CCS 3D application, we communicated our ideas to various users. We paid attention to the reactions, when presenting the idea of interacting with three fingers. Around a dozen in count, these users were people closely related to IT, be it colleagues or fellow students.

The vast majority initially disliked the idea of using three fingers but when confronted and presented with the advantages, the initial negative attitude slowly faded away. Based on the fact, that using three fingers simply does not inspire the users at all, we designed one more technique. It uses two fingers at most and focuses on being intuitive, simple, easy to use and based on gestures already known to the users. The main advantages of this technique (described in Table 8) are that most users will find already acquainted with it and that it is suitable for use with mobile phones, where screens are simply too small to provide enough space for efficiently using three finger gestures.

Table 8 – Two finger interaction technique

DoF	Action	Result
left / right	3F horizontal drag	translate along the *X* axis
up / down	3F vertical drag	translate along the *Y* axis
forward / back	3F pinch	translate along the *Z* axis
pitch	2F vertical drag	rotate about the *X* axis
yaw	2F horizontal drag	rotate about the *Y* axis
roll	2F circle	rotate about the *Z* axis

This technique is based on approaches that are already being used. The two finger translation is nowadays being used in most laptops with multi-touch trackpads. Even thou the application domains are mostly in a two dimensional environment, we believe users will not require a different approach for three dimensional environments. This technique has been implemented in CCS 3D as well and in the final evaluation compared with the three finger technique.

7.3 CCS 3D user testing

The two designed techniques (2F/3F and 1F/2F) were presented to the users while being asked to fulfill given instructions. The observations and results were noted into a questionnaire form. The full questionnaire can be found in Appendix B.

This test should be regarded as an introductory test to verify that the designed techniques can be regarded prospective and promising. There are things that can be done better in order to provide more detailed results (discussed in chapter 7.4), however for our needs this test is more than sufficient.

7.3.1 Test setup

Our group of testers consisted of high school students from the Hubeného 23 high school. Most of them (79%) already passed curriculum containing cube cross sections. Students that had no knowledge of cube cross section construction were tasked with simple instructions aimed at the use of the 6 DoF. Such instructions included:

- turn the cube so that the CDHG plane is in the front

- turn the cube so that the CDHG plane is in the front, but the GH segment is at the bottom
- zoom in
- move the cube to the side so that only half of it can be seen

Students were questioned one by one, each sitting at a desk where the mobile device was located. Other students were not present, so their answers would not be influenced. Approximate test duration was 10 minutes per student with cube cross section knowledge (5 minutes for the other students as they were not required to solve the cross section construction).

Apart from the questionnaire, the following was used:

- stopwatch
- mobile device: HTC EVO 3D
- applications: MultiTouch Tester (by the511plus), CCS 3D

7.3.2 Test execution

The following scenario has been executed with each student. First the student was seated and asked to answer the questions in section (in this chapter - 7.3.2 Test execution - the term “section” refers purely to the sections of the questionnaire) *1 General information* of our questionnaire.

Section *2 Interaction – user experience / intuition evaluation* was answered based on what the user did in the MultiTouch Tester application. Because the first three test subjects staggered on the request to rotate the “coin” displayed on the screen, the other respondents were first asked to do the translations, which made the upcoming request to rotate the “coin” clearer.

Next section *7 Task evaluation – observer point of view* was filled out based on the time it took the user to execute the given tasks using the designed three finger technique (2F/3F). In section 7.1 the numbers were filled, only if the user managed to fulfill the task under 30 seconds. If he did not manage to guess the gesture in time, he was given a hint “try to use three fingers”. Accomplishing a task after being given a hint was noted in section 7.2.

After being acquainted with the interaction methods and the user interface (UI), the student was asked to construct the cross section (section 7.2.8). Then the sections *6 CCS*

3D application evaluation and *3 Interaction – custom design 2F/3F user evaluation* were filled out.

In the end the student was presented with the latter, two finger technique (1F/2F). After having tried it out in the application, the remaining sections *3 Interaction – custom design 1F/2F user evaluation* and *5 Interaction – custom design in general* were filled out.

7.3.3 Test results

In most of the test questions the students were asked to evaluate something and rate it on a scale from 1 to 10 (noted as 0-9), where 1 is the worst and 10 is the best possible rating. The rest were Yes/No questions as well as questions where the user was asked to give his opinion (blank fields). Detailed result statistics can be found in Appendix B. The rated values have been transformed into percentage values for a clearer representation.

The fact that 93% of the tested students rated our application with a rating above 75% leads us to the conclusion that the application was accepted positively. With a usability rating of 79% and user interface rating of 89% we can conclude that the user interaction evaluation was not negatively affected by the application itself.

The vast majority, 84% of the students asked, preferred the two finger technique (1F/2F). The three main causes which led the students to this conclusion were the following:

- habit – as 58% of the students tested own a phone with iOS or Android, where 2 finger rotation and the pinch gesture are widely spread, they found it a habit to use those gestures and therefore rated the interaction technique higher
- simplicity – using three fingers cramped on a small screen requires practice, it took time for the students to find a way of placing the fingers so that they can be conveniently moved around the screen
- small screen – by laying three fingers on the screen of a mobile phone and dragging them around, most of the screen is covered by the users hand and therefore the lack of visual feedback decreases usability (68% rating for 2F/3F against 90% rating of 1F/2F)

While in the two fingers technique students did not differentiate dramatically between the various DoF gestures, the rating for the three finger technique (2F/3F) has a wider spread. The highest rated in the 1F/2F technique was the use of one finger for rotation (score of 95%) and the second best, the well-known pinch gesture (score of 92%). The remaining two finger gestures were all rated around 90%. In the 2F/3F technique the two finger gestures were rated an average of 84%, which compared to the two finger gestures for translation in the 1F/2F technique is 6% lower. However the same gesture mapping for the roll rotation has a 6% lower rating as well, therefore we believe that the individual DoF gesture rating is affected by the general technique perception. As already mentioned earlier, the three finger gestures have a noticeably lower rating of 68%. What is interesting is that the three finger pinch gesture has 10% higher rating than the three finger gesture for translation on the XY plane. Even thou that simply dragging three fingers on one hand simultaneously without changing their position relative to each other is simpler than moving two fingers away or closer to the third finger, the pinch gesture is preferred due to its two finger variant which is very popular.

The final overall rating is 93% against 76% in favor of the two finger technique. When asked to rate the suitability of the two techniques for mobile phones the two finger technique scored 89% against 70% for the three finger technique. However the students rated an average of 87% for tablet suitability for both techniques.

When asked if the interaction technique provided them all possibilities to accomplish any kind of translation or rotation, they scored 100% for both techniques.

By observing students' way of interaction with the cube, we were able to estimate the frequency of using the individual degrees of freedom individually as well as simultaneously. When asked if the student realizes that the three finger technique allows simultaneous translation for all 3 DoF 65% answered yes, however only 35% took advantage of it. Simultaneous rotation for all 3 DoF was noticed only by 24% of the students (or so they answered) but only 6% used it. On the other hand, one students requested simultaneous translation with the roll rotation (using the 2F/3F technique).

Based on evaluation of section 2 of the questionnaire we can clearly see that no user even considered the use of three fingers. Two fingers were suggested for the pinch gesture (all students) and the roll gesture (56%). The assigning of two fingers for rotation or translation depended on which question the student was asked first. If first asked to assign a gesture to translation, he chose one finger (1F) drag. If first asked about rotation, the most common choice was one finger drag as well. If the 2 finger drag gesture has been assigned, it was almost always chosen as the second choice (when one finger drag has already been assigned).

7.4 Evaluation

As mentioned above, the fact that users prefer the two finger interaction technique can be credited to screen size and habits. The test results clearly show that the possibilities that the three fingers technique offers are equal to (and greater than) the two finger technique. The main disadvantage is the use on small screens, where scene obscuring occurs. On the other hand the three fingers technique gives room for one finger interaction. It can be used for various interaction enhancements, like shape recognition, path tracing, etc. In our CCS 3D application we let users do selecting by simply dragging one finger across the screen. This cannot be done with the two fingers technique, as dragging one finger is mapped to cube rotation.

Even thou the results indicate that both techniques are equally suitable for tablets, we fear that applying a different test scenario (let the user try out both and then rate both of them at once) and even testing on a tablet, would result in a better tablet suitability score for the two finger technique. However that would be just because the two finger technique consists of common gestures.

When taking into account all revelations, the following facts can be stated:

- users want to keep things simple, the fewer fingers, the better
- three fingers on the screens of mobile phones are one too many
- users prefer gestures they are already acquainted with, (like the pinch gesture)
- simultaneous translation in all 3 DoF is appreciated and can increase efficiency
- simultaneous rotation is preferred for pitch and yaw only, simultaneous use of all three rotation DoF is very hard to understand and even harder to use

Based on these facts we suggest the two fingers technique be used on mobile phones (devices with smaller screens) and our three fingers technique on tablets. The advantage of the three finger technique is the mapping transparency. It is important to understand the difference between gestures triggering certain action and gestures being directly translated into actions. When simply rotating a picture by 45 degree steps, the gesture just triggers the rotation. On the other hand, in an application where the user wants to rotate with less than a degree steps, such gesture requires to be directly translated. In the latter case if the roll rotation is allowed simultaneously with translation, unwanted roll or translation may occur. Separating the roll rotation from the translations gives us a smooth stable Z axis translation without unwanted roll of the camera / object.

In the year 2016 there shall be more tablets PCs than notebook PCs and they shall be four times as robust as the tablets available nowadays[26]. This computing power will allow users to use tablets for all kinds of activities, such as work, entertainment, gaming, etc… Many application that present 3D environments available only on computers nowadays, will become a part of mobile applications, be it architect software, 3D modeling or simple 3D object browsing. Especially applications where work efficiency plays a key role would greatly benefit from our three finger technique for camera / object manipulation.

As for mobiles phones and other devices with smaller screens, the two finger technique is a variant among other existing similar techniques, but fulfills all user expectations and requirements.

Further research involves thorough tests, where the application should not be domain specific. While testing our application, not everyone had sufficient knowledge of cube cross sections and therefore the results might have been negatively affected. Simple tasks as “move the ball into the box” etc. would not limit the test subject to a specific group of people. Apart from the domain, the application should be enhanced to collect usage statistics and record user actions, so they can be played back later.

[26] http://www.technewsworld.com/story/75039.html

8 Conclusion

We analyzed the most relevant areas related to virtual reality, human-computer interaction, multi-touch interaction techniques and virtual reality learning environments. Based on the evaluation of existing approaches, we designed an interaction technique that allows 6 DoF. This technique has been improved and redesigned after an internal prototype testing. Based on the feedback received from users that were presented with the three finger technique, one more, a simpler technique, has been designed before the final evaluation.

In our research we compare these two techniques to techniques available in mobile applications nowadays, as well as to techniques designed in other research studies on multi-touch gestures. Our paper regarding this research has been published in the IIT.SRC proceedings and received an award from the Czechoslovakia Section of IEEE.

User testing was executed on our CCS 3D application, which allows students to interact with a cube and construct cube cross sections. Tests are designed so that students have to execute various tasks from simple cube rotations, to tasks where all 6 DoF have to be put to use. By evaluating the collected results we were able to conclude that the two fingers technique is the right choice for mobile phones and other devices with smaller screens. For tablets the two fingers technique might be preferred as well, however this can be attributed to habit and user unwillingness to embrace something new. As this technology is young, there is still room for setting a new standard. The three fingers technique has clearly more potential than the two fingers technique. The technique is also more suitable for 3D environments and when embraced, it can deliver higher efficiency and usability results.

Through extensive research and a thorough design of an efficiently usable, intuitive, and easy to master technique we bring VR one step closer to the mobile device users.

9 REFERENCES

1. Horne, M. and Thompson, E.M. (2008) 'The role of virtual reality in built environment education', Journal for Education in the Built Environment, 3 (1), pp. 5-24. ISSN: 1747-4205 http://hdl.handle.net/10145/108455
2. Andy Yeh. 2004. VRMath: knowledge construction of 3D geometry in virtual reality microworlds. In CHI '04 extended abstracts on Human factors in computing systems (CHI EA '04). ACM, New York, NY, USA, 1061-1062. DOI=10.1145/985921.985979 http://doi.acm.org/10.1145/985921.985979
3. Nechvatal, Joseph. 1999. Immersive Ideals / Critical Distances. Inquiry, no. 14. http://www.eyewithwings.net/nechvatal/iicd.pdf
4. Hannes Kaufmann and Dieter Schmalstieg. 2006. Designing Immersive Virtual Reality for Geometry Education. In Proceedings of the IEEE conference on Virtual Reality (VR '06). IEEE Computer Society, Washington, DC, USA, 51-58. DOI=10.1109/VR.2006.48 http://dx.doi.org/10.1109/VR
5. Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. 2004. Multi-finger gestural interaction with 3d volumetric displays. In Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04). ACM, New York, NY, USA, 61-70. DOI=10.1145/1029632.1029644 http://doi.acm.org/10.1145/1029632.1029644
6. HoSiyong, A.; Kenny, C.; , "Evaluation of on screen navigational methods for a touch screen device," Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on , vol., no., pp.83-84, 2-5 March 2010
doi: 10.1109/HRI.2010.5453258
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5453258&isnumber=5453161
7. Ahsanullah; Mahmood, A.K.B.; Sulaiman, S.; , "Investigation of fingertip blobs on optical multi-touch screen," Information Technology (ITSim), 2010 International Symposium in , vol.1, no., pp.1-6, 15-17 June 2010
doi: 10.1109/ITSIM.2010.5561307
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5561307&isnumber=5561289
8. Abu Saleh Md Mahfujur Rahman, M Anwar Hossain, and Abdulmotaleb El Saddik. 2010. Spatial-geometric approach to physical mobile interaction based on accelerometer and IR sensory data fusion. ACM Trans. Multimedia Comput. Commun. Appl. 6, 4, Article 28 (November 2010), 23 pages. DOI=10.1145/1865106.1865112 http://doi.acm.org/10.1145/1865106.1865112
9. Raffa, G.; Jinwon Lee; Nachman, L.; Junehwa Song; , "Don't slow me down: Bringing energy efficiency to continuous gesture recognition," Wearable Computers (ISWC), 2010 International Symposium on , vol., no., pp.1-8, 10-13 Oct. 2010
doi: 10.1109/ISWC.2010.5665872
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5665872&isnumber=5665849

10. Donato Fiorella, Andrea Sanna, Fabrizio Lamberti, 2010. Multi-touch user interface evaluation for 3D object manipulation on mobile devices, Journal on Multimodal User Interfaces, Springer Berlin / Heidelberg, Issn: 1783-7677
Doi: 10.1007/s12193-009-0034-4
Url: http://dx.doi.org/10.1007/s12193-009-0034-4
11. Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07). ACM, New York, NY, USA, 1147-1156. DOI=10.1145/1240624.1240798 http://doi.acm.org/10.1145/1240624.1240798
12. Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. 2005. Fluid integration of rotation and translation. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '05). ACM, New York, NY, USA, 601-610. DOI=10.1145/1054972.1055055 http://doi.acm.org/10.1145/1054972.1055055
13. Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2009. 3D positioning techniques for multi-touch displays. In Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST '09), Steven N. Spencer (Ed.). ACM, New York, NY, USA, 227-228. DOI=10.1145/1643928.1643978 http://doi.acm.org/10.1145/1643928.1643978
14. Michael Heim. 2000. *Virtual Realism* (1st ed.). Oxford University Press, Inc., New York, NY, USA
15. Mario Zechner. 2011. Beginning Android Games. Apress, USA. ISBN-13: 978-1-4302-3042-7
16. de Cotret, S., & de Cotret, P. R. (2005). *Cabri 3D User Manual.* Cabrilog.

APPENDICES

APPENDIX A

We have designed two interaction techniques. In order to describe these interactions, we devised a simple table annotation that maps finger count and action to a single DoF. Here we would like to explain this annotation as well as the designed techniques on a series of illustrations. For each DoF we will show an explanatory illustration and explain the notation.

A.1 Multi-touch technique for medium-sized screens

The technique has been defined as follows:

DoF	Action	Result
left / right	3F horizontal drag	translate along the *X* axis
up / down	3F vertical drag	translate along the *Y* axis
forward / back	3F pinch	translate along the *Z* axis
pitch	2F vertical drag	rotate about the *X* axis
yaw	2F horizontal drag	rotate about the *Y* axis
roll	2F circle	rotate about the *Z* axis

The first is the left / right DoF. The notation 3F means that three fingers are required to be touching the screen. When dragged horizontally, the cube moves to the sides.

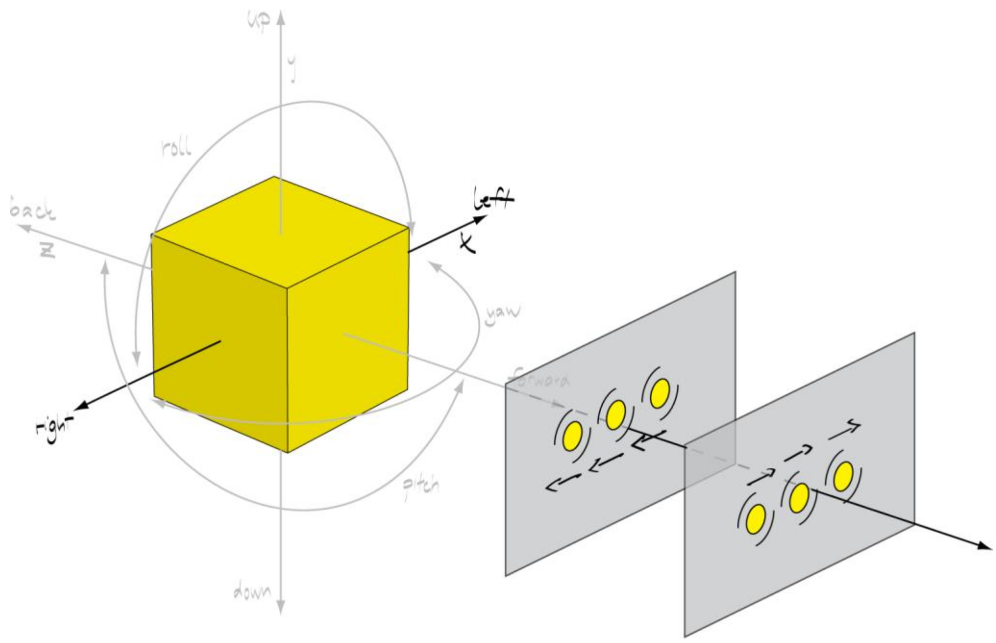


Illustration 1 – 3F horizontal drag resulting in left / right object movement

The up /down DoF has the notation 3F vertical drag assigned. It behaves the same as the previous DoF action.

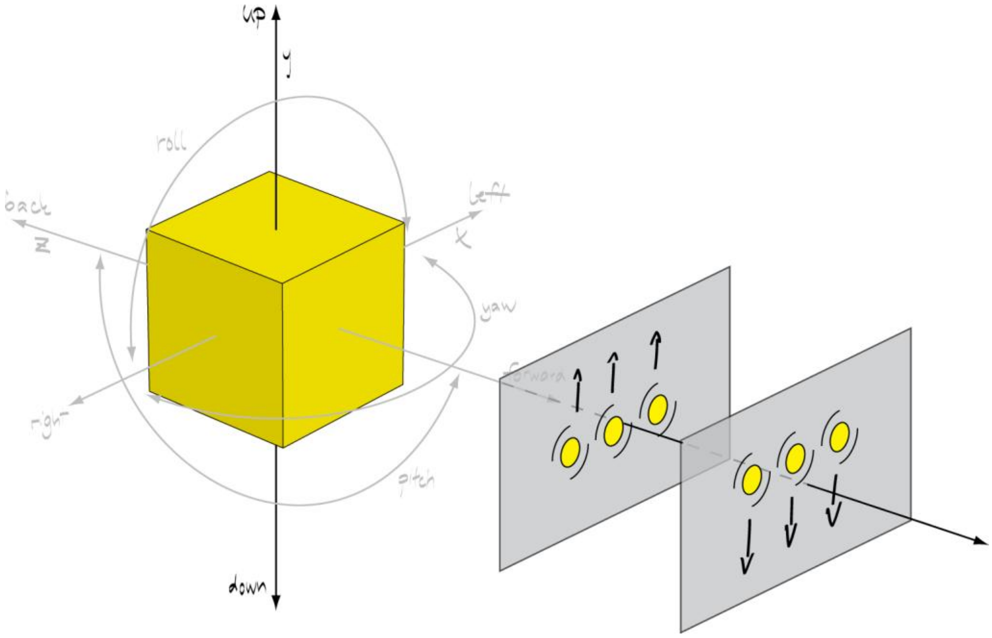


Illustration 2 - 3F vertical drag resulting in up / down object movement

The forward / back movement is incorporated via the pinch gesture. The notation 3F pinch means that the finger move away from each other or closer to each other depending on whether we want to move the cube towards us, or away from us.

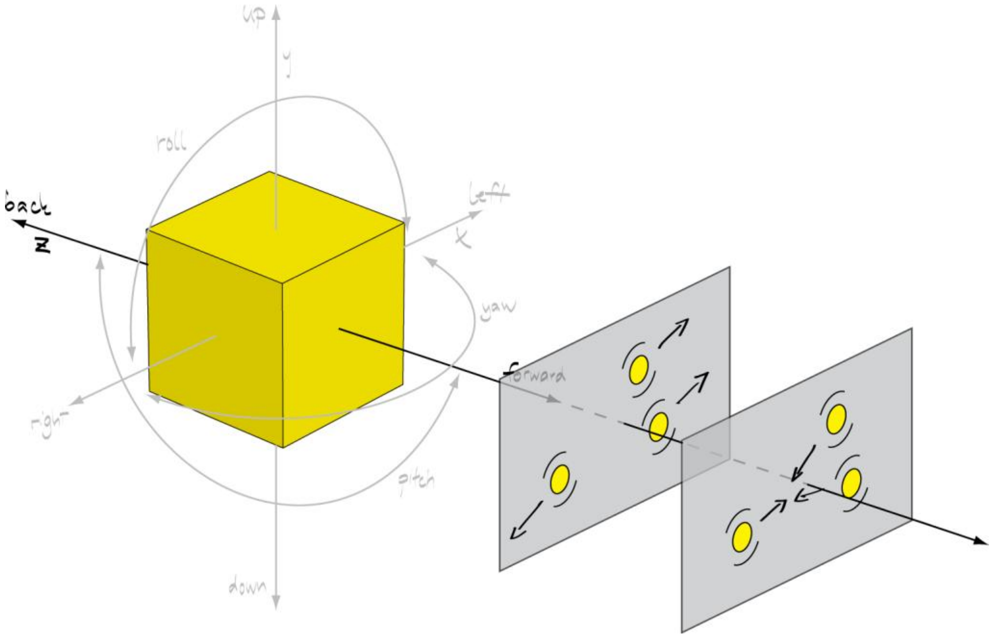


Illustration 3 - 3F pinch gesture resulting in forward / backward movement

In this technique all translation DsoF have been mapped to 3 finger gestures and rotation is mapped to 2 finger gestures. In order to pitch the cube the user has to do the 2F vertical drag gesture. Two fingers simply dragged up or down.

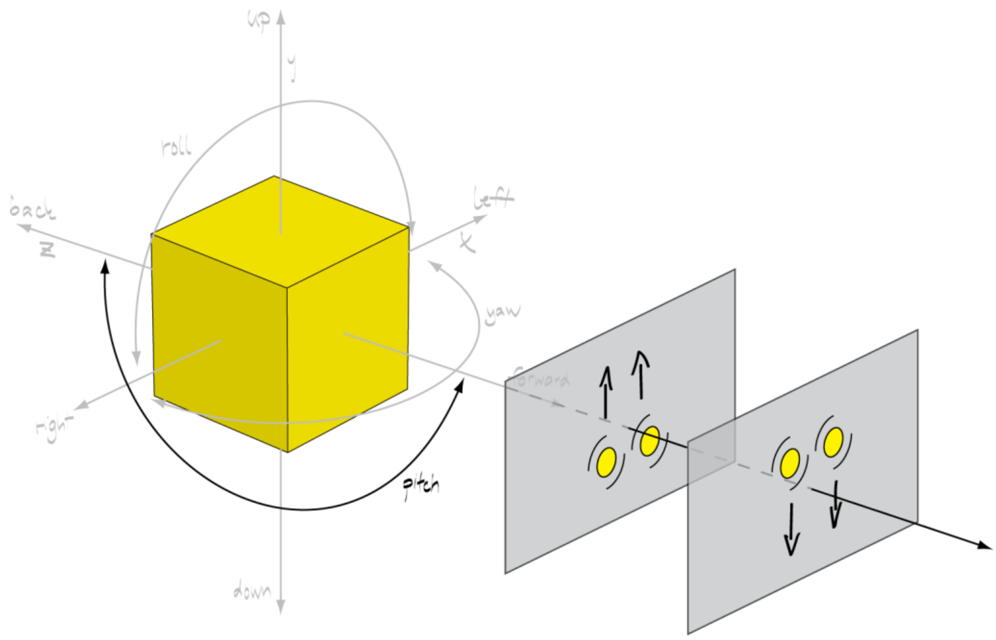


Illustration 4 - 2F vertical drag is used to rotate the cube around the X axis

Yaw is controlled by simply dragging 2 fingers horizontally as displayed on the next illustration.

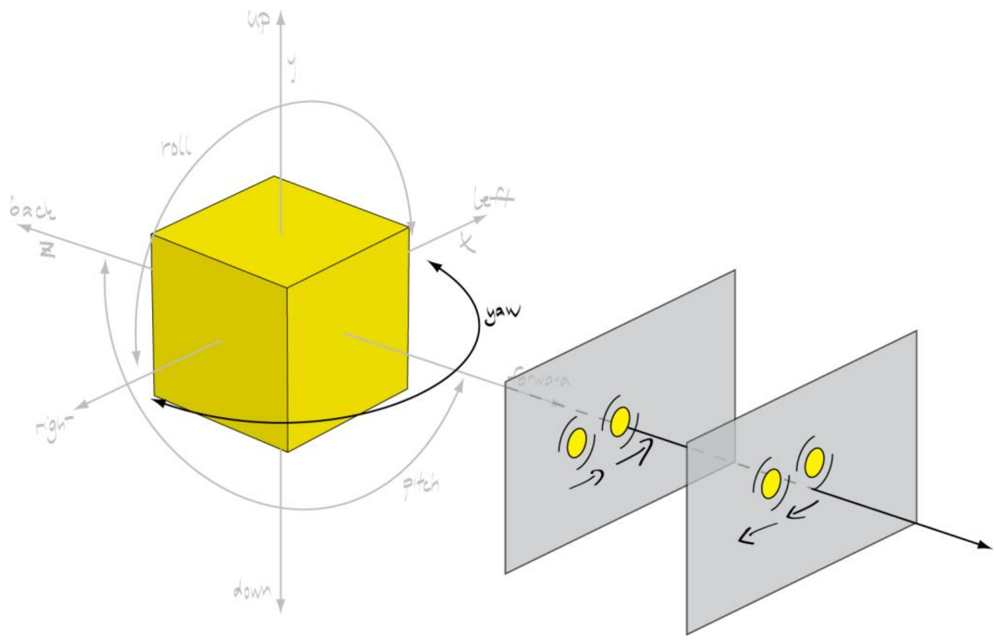


Illustration 5 - 2F horizontal drag results in rotations about the Y axis

The most complex to implement is the 2F circle gesture. The circle descriptor means that the fingers are moved as if on a ring of a circle (see next illustration).

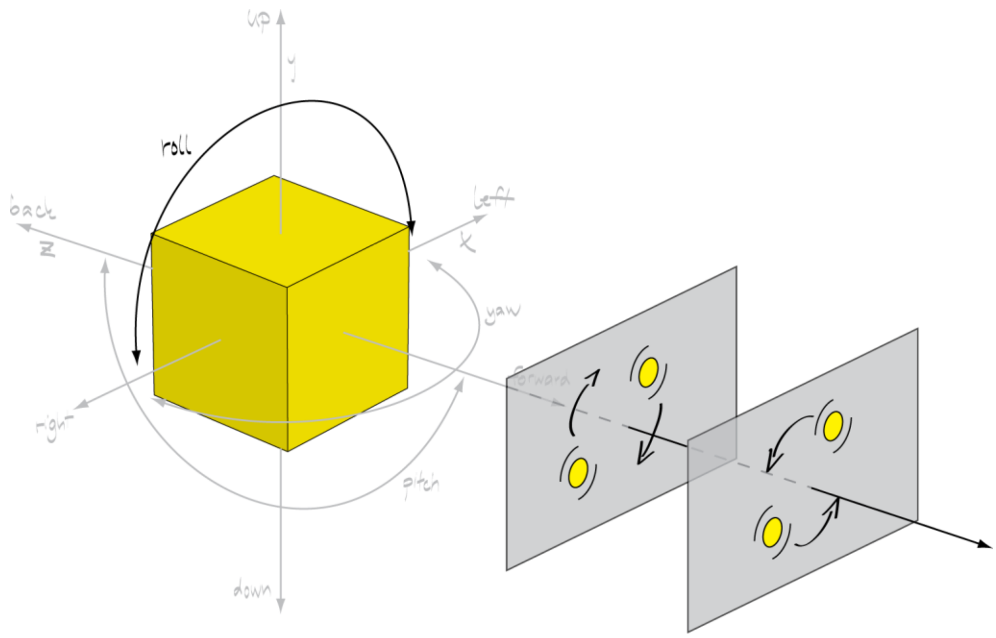


Illustration 6 - 3F circle action results in rotation about the Z axis

A.2 Multi-touch technique for small-sized screens

The second technique is designed for small screens of mobile devices, where the use of three fingers is cumbersome and inconvenient. This technique has been defined as follows:

DoF	Action	Result
left / right	2F horizontal drag	translate along the *X* axis
up / down	2F vertical drag	translate along the *Y* axis
forward / back	2F pinch	translate along the *Z* axis
pitch	1F vertical drag	rotate about the *X* axis
yaw	1F horizontal drag	rotate about the *Y* axis
roll	2F circle	rotate about the *Z* axis

The first is the left / right DoF. Two fingers are required in this case to move the cube to the sides.

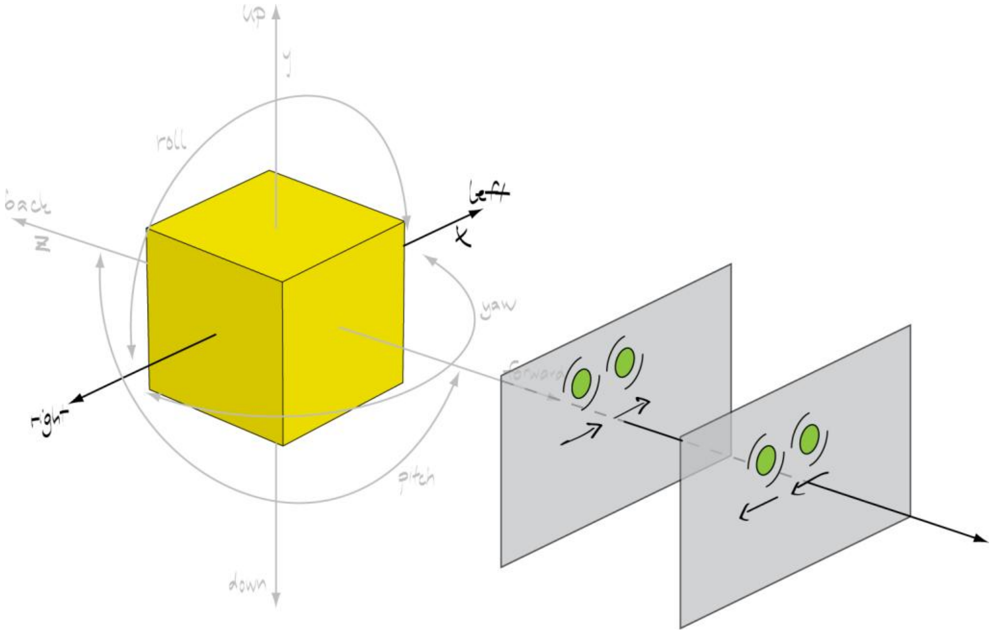


Illustration 7 – 2F horizontal drag resulting in left / right object movement

The up /down DoF behaves the same as the previous DoF action.

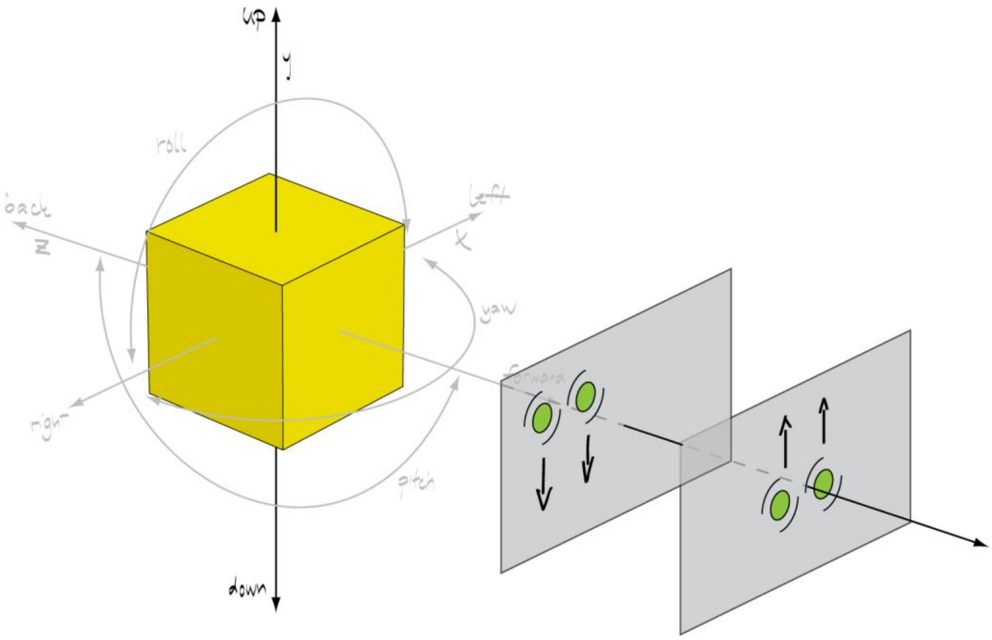


Illustration 8 - 2F vertical drag resulting in up / down object movement

The forward / back movement is incorporated via the pinch gesture. The notation 2F pinch means, that moving fingers away from each other, or closer to each other results in the translation on the Z axis. Even thou Z axis translation is not the same as zooming, this gesture can be used for both, just needs to be treated specially by the application.

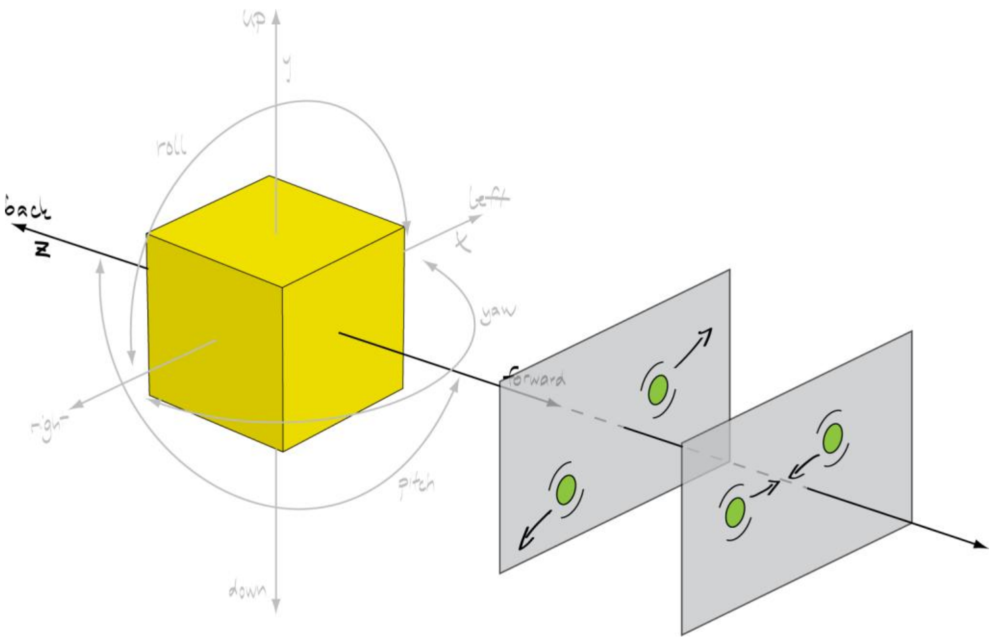


Illustration 9 - 2F pinch gesture resulting in forward / backward movement

In this technique all translation DoF have been mapped to 2 finger gestures and rotations are mapped to 1 and 2 finger gestures. In order to pitch the cube the user has to do drag one finger vertically.

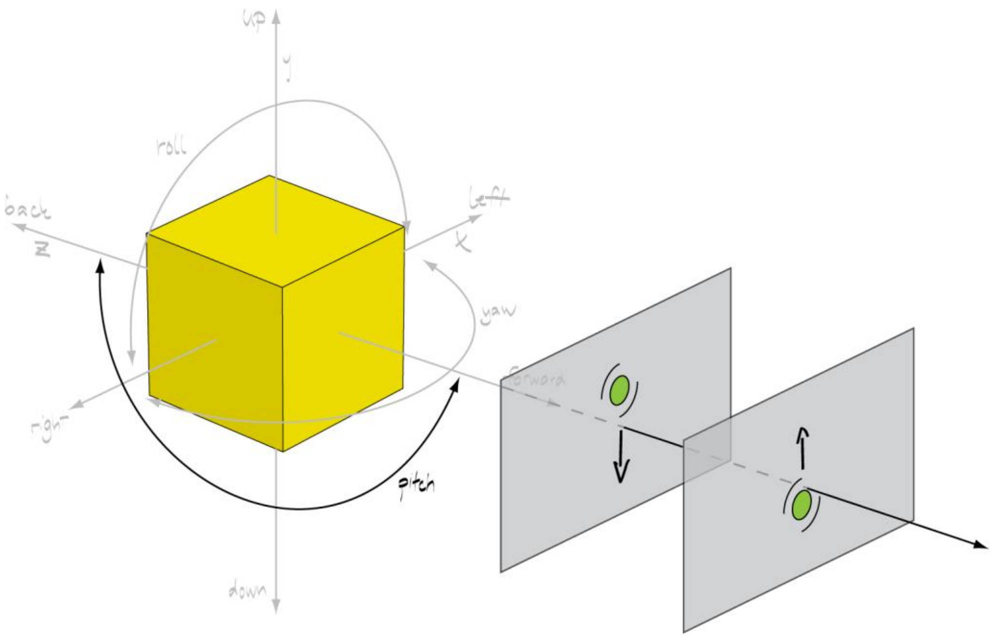


Illustration 10 - 1F vertical drag is used to rotate the cube around the X axis

Yaw is controlled by simply one finger horizontally as displayed on the next illustration.

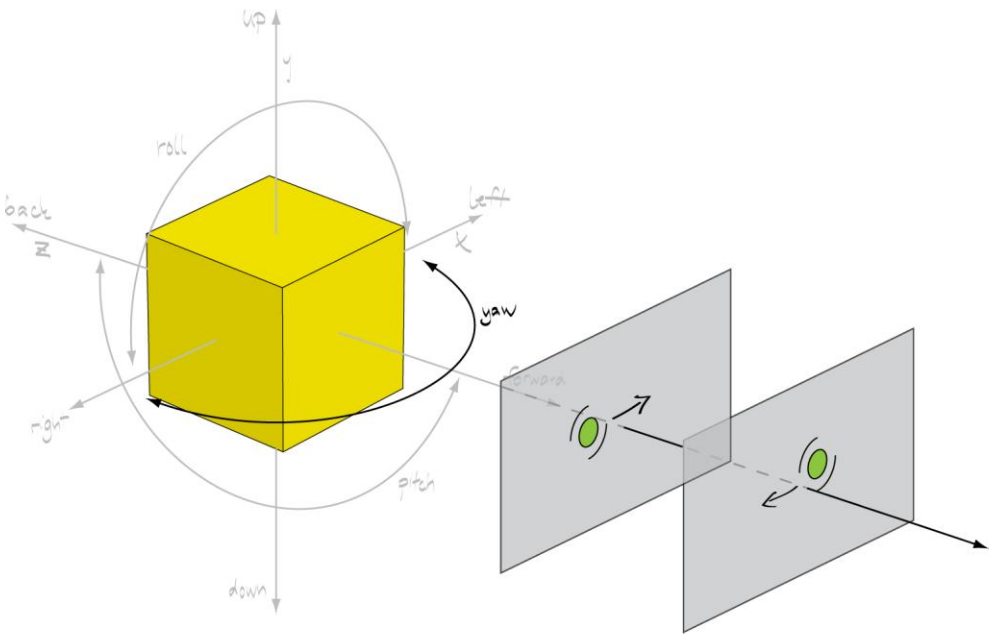


Illustration 11 - 1F horizontal drag results in rotations about the Y axis

The most complex to implement is the 2F circle gesture. The circle descriptor means that the fingers are moved as if on a ring of a circle (see next illustration).

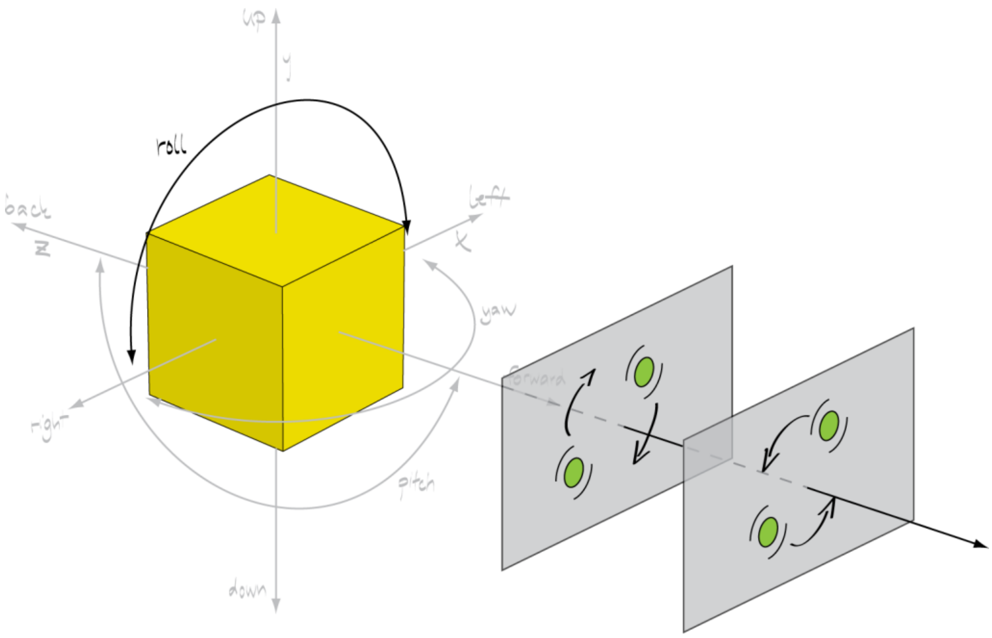


Illustration 12 - 2F circle action results in rotation about the Z axis

Appendix B

B.1 Student questionnaire

1. General information	
1.1. Person	
1.1.1. Age	
1.1.2. Sex	Male Female
1.2. Mobile devices	
1.2.1. Do you own a mobile phone?	Yes No
1.2.2. Do you own an Android or iOS phone?	Yes No
1.2.3. Does your phone have a multi-touch screen?	Yes No
1.2.4. Do you own other mobile devices (not phones)?	Yes No
1.2.5. What do you use your mobile devices for?	
1.3. Applications	
1.3.1. Do you know any 3D applications?	
1.3.2. What 3D geometry or 3D modeling applications do you know?	
1.3.3. Did you ever use them?	Yes No
1.3.4. What is your knowledge of cube cross sections?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
2. Interaction – user experience / intuition evaluation	
2.1. How would you rotate the cube to the left or right?	
2.2. How would you rotate the cube up or down?	
2.3. How would you roll the cube to the side?	
2.4. How would you move the cube to the left or right?	
2.5. How would you move the cube up or down?	
2.6. How would you move the cube closer to you or away from you?	

3. Interaction – custom design 2F/3F user evaluation	
3.1. Was the interaction intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.2. Was the interaction comfortable?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.3. Left or right rotation	
3.3.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.3.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.3.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.3.4. Would you map a different interaction to it? (if yes, why and what)	
3.4. Up or down rotation	
3.4.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.4.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.4.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.4.4. Would you map a different interaction to it? (if yes, why and what)	
3.5. Rolling clockwise or counter-clockwise	
3.5.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.5.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.5.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.5.4. Would you map a different interaction to it? (if yes, why and what)	
3.6. Moving left or right	
3.6.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.6.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.6.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.6.4. Would you map a different interaction to it? (if yes, why and what)	
3.7. Moving up or down	
3.7.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.7.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.7.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.7.4. Would you map a different interaction to it? (if yes, why and what)	
3.8. Moving away or closer to the user	
3.8.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.8.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.8.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.8.4. Would you map a different interaction to it? (if yes, why and what)	

3.9. Were you able to accomplish every desired action? (if not, what did it lack?)	
3.10. How suitable do you find this technique for mobile phones?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.11. How suitable do you find this technique for tablets?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
3.12. Did you notice that you can simultaneously move the cube in all directions?	Yes No
3.13. At most in how many directions simultaneously did you move the cube?	1 .. 2 .. 3
3.14. Did you notice that you can simultaneously rotate the cube in any direction?	Yes No
3.15. At most in how many directions simultaneously did you rotate the cube?	1 .. 2 .. 3
3.16. Comments…	

4. Interaction – custom design 1F/2F user evaluation	
4.1. Was the interaction intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.2. Was the interaction comfortable?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.3. Left or right rotation	
4.3.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.3.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.3.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.3.4. Would you map a different interaction to it? (if yes, why and what)	
4.4. Up or down rotation	
4.4.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.4.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.4.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.4.4. Would you map a different interaction to it? (if yes, why and what)	
4.5. Rolling clockwise or counter-clockwise	
4.5.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.5.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.5.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9

4.5.4. Would you map a different interaction to it? (if yes, why and what)	
4.6. Moving left or right	
4.6.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.6.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.6.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.6.4. Would you map a different interaction to it? (if yes, why and what)	
4.7. Moving up or down	
4.7.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.7.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.7.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.7.4. Would you map a different interaction to it? (if yes, why and what)	
4.8. Moving away or closer to the user	
4.8.1. How often did you use it?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.8.2. Did you find it intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.8.3. Did you find it comfortable and easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.8.4. Would you map a different interaction to it? (if yes, why and what)	
4.9. Were you able to accomplish every desired action? (if not, what did it lack?)	
4.10. How suitable do you find this technique for mobile phones?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.11. How suitable do you find this technique for tablets?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9
4.12. Did you notice that you can simultaneously move the cube in all directions?	Yes No
4.13. At most in how many directions simultaneously did you move the cube?	1 .. 2 .. 3
4.14. Did you notice that you cannot simultaneously rotate the cube in all directions?	Yes No
4.15. Did you miss the possibility to roll the cube while rotating in other directions?	1 .. 2 .. 3
4.16. Comments…	

5. Interaction – custom design in general		
5.1. Which interaction technique did you like better?	2 & 3 fingers	1 & 2 fingers
5.2. Why?		
6. CCS 3D application evaluation		
6.1. How would you rate the application?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9	
6.2. Was it easy to use?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9	
6.3. Was the graphical user interface intuitive?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9	
6.4. What did you miss in the application?		
6.5. What did you dislike in the application?		
6.6. Comments…		
6.7. Bugs…		
7. Task evaluation – observer point of view		
7.1. Without knowing (guessing)		
7.1.1. Try rotating the cube left or right.		
7.1.2. Try rotating the cube up or down.		
7.1.3. Try rolling the cube to the sides.		
7.1.4. Try moving the cube left or right.		
7.1.5. Try moving the cube up or down.		
7.1.6. Try moving the cube to the back or to the front.		
7.2. With given tutorial		
7.2.1. Try rotating the cube left or right.		
7.2.2. Try rotating the cube up or down.		
7.2.3. Try rolling the cube to the sides.		
7.2.4. Try moving the cube left or right.		
7.2.5. Try moving the cube up or down.		
7.2.6. Try moving the cube to the back or to the front.		
7.2.7. Try the CCS 3D functions		
7.2.8. Solve the cross section		
7.2.8.1. Was it easy?	0 .. 1 .. 2 .. 3 .. 4 .. 5 .. 6 .. 7 .. 8 .. 9	
7.2.8.2. How long did it take you to solve it?		

B.2 Questionnaire evaluation

The first part contains the interaction evaluation. The first table presents the average rating for the specific action. The table below has the values transformed into percentage rating for easier interpretation. Complete results can be found on the attached CD.

		Interaction rating		Rotate			Translate		
		Overall	Calculated	Y	X	Z	X	Y	Z
2F/3F	Intuitiveness	76%	76%	84%	84%	83%	65%	65%	76%
	Comfort	75%	76%	84%	84%	85%	65%	65%	73%
	Frequency			38%	38%	23%	10%	10%	23%
1F/2F	Intuitiveness	92%	91%	94%	94%	90%	89%	89%	92%
	Comfort	94%	92%	96%	96%	90%	89%	89%	92%

Suitability

	Phone	Tablet
2F/3F	70%	87%
1F/2F	89%	87%

Sufficiency

	Rating	Samples
2F/3F	100%	18
1F/2F	100%	3

Preference

2F/3F	1F/2F
16%	84%

User defined actions

	Yaw	Pitch	Roll	Move X	Move Y	Move Z
1f	75%	90%	44%	89%	95%	0%
2f	25%	10%	56%	11%	5%	100%

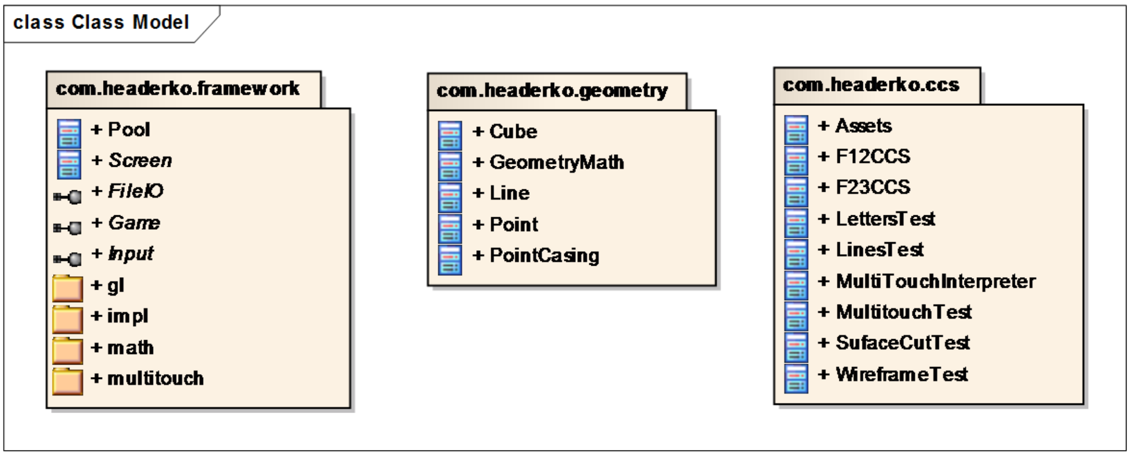
3 DoF simultaneous use

Translation				Rotation			
Noticed	1 DoF	2 DoF	3 DoF	Noticed	1 DoF	2 DoF	3 DoF
65%	12%	53%	35%	24%	6%	88%	6%

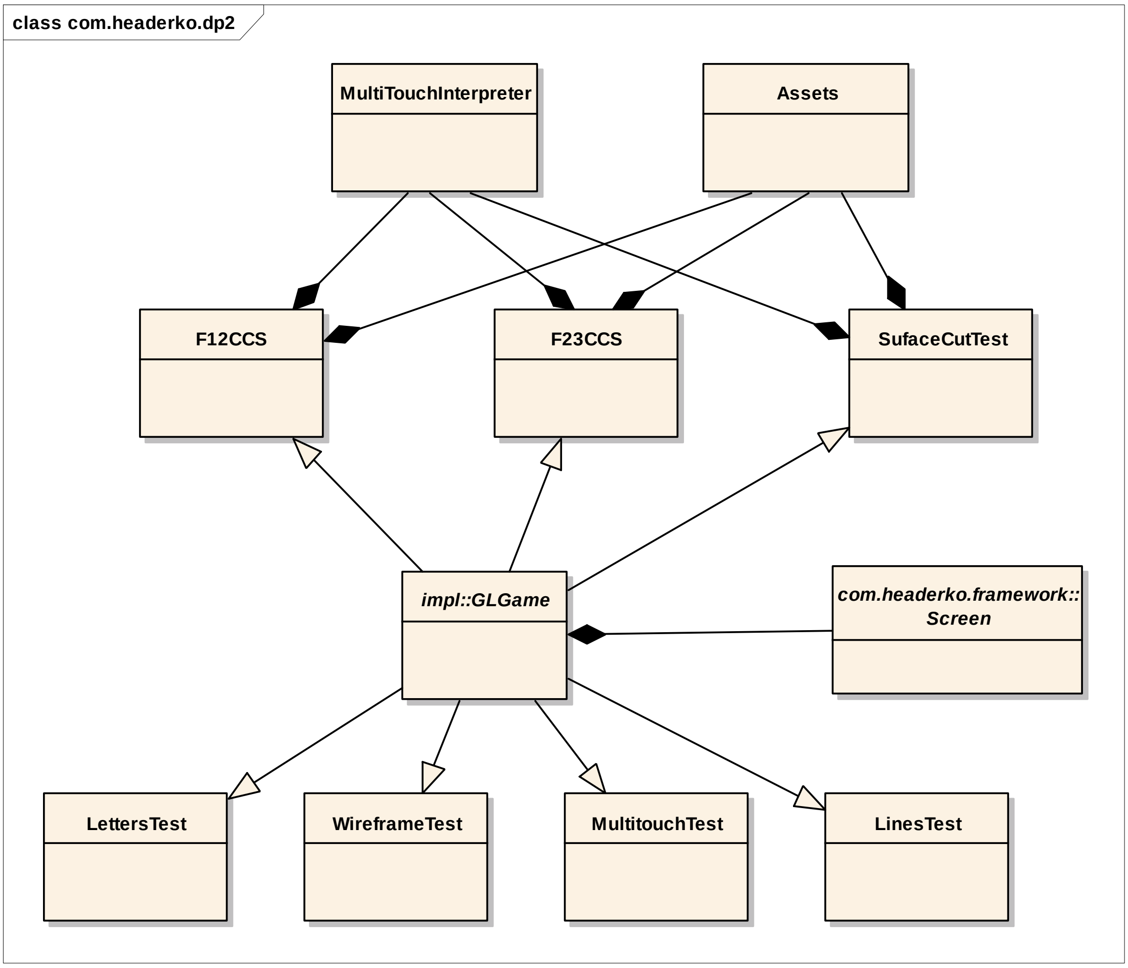
APPENDIX C

C.1 Class model

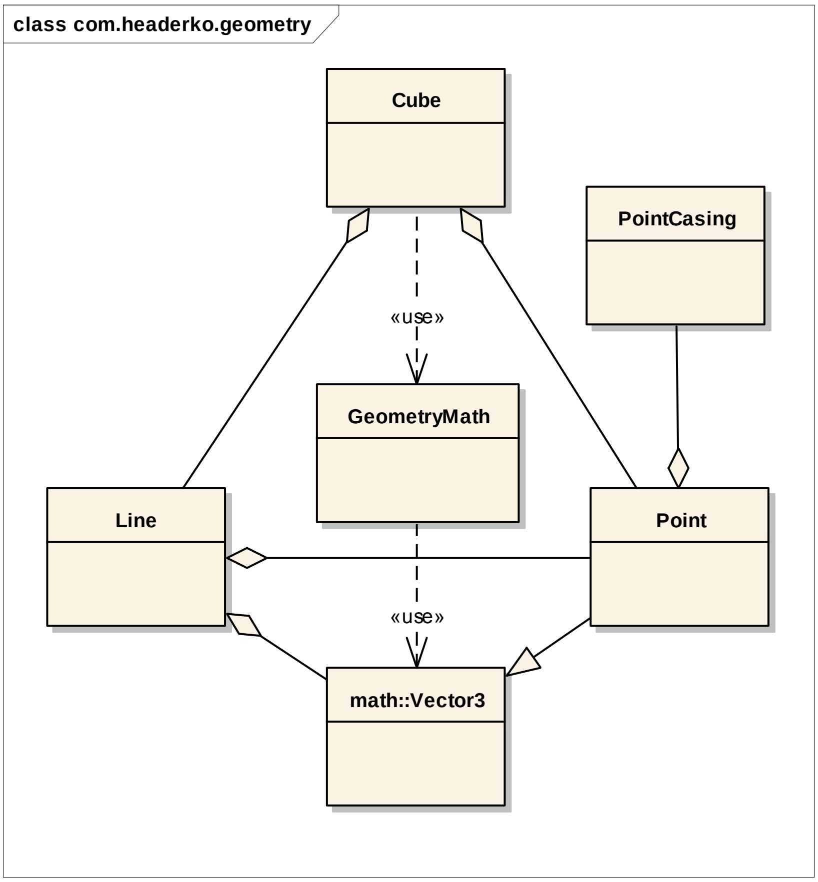
This appendix contains the all class diagrams of our application.



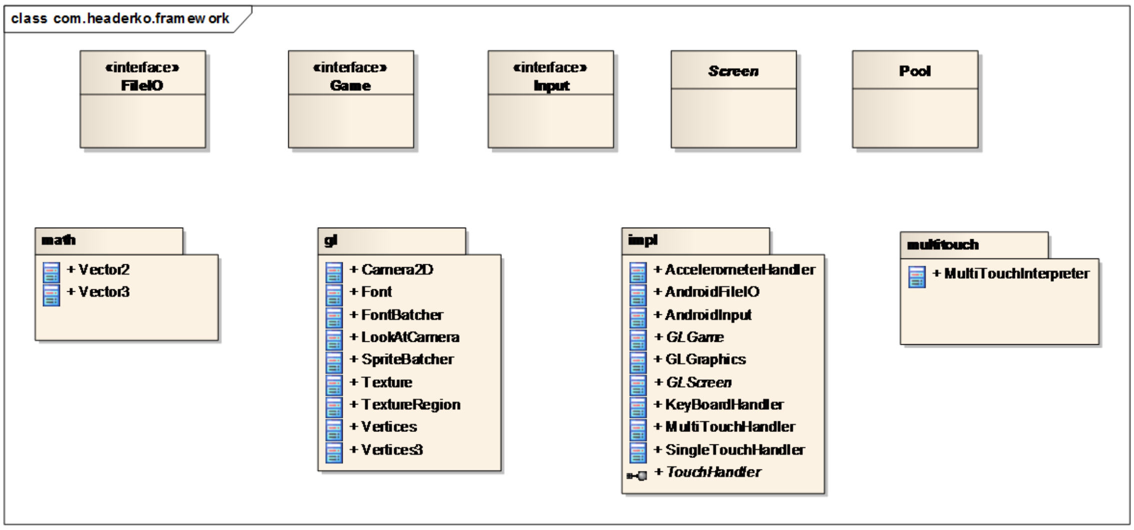
Class model 1 - Top level packages



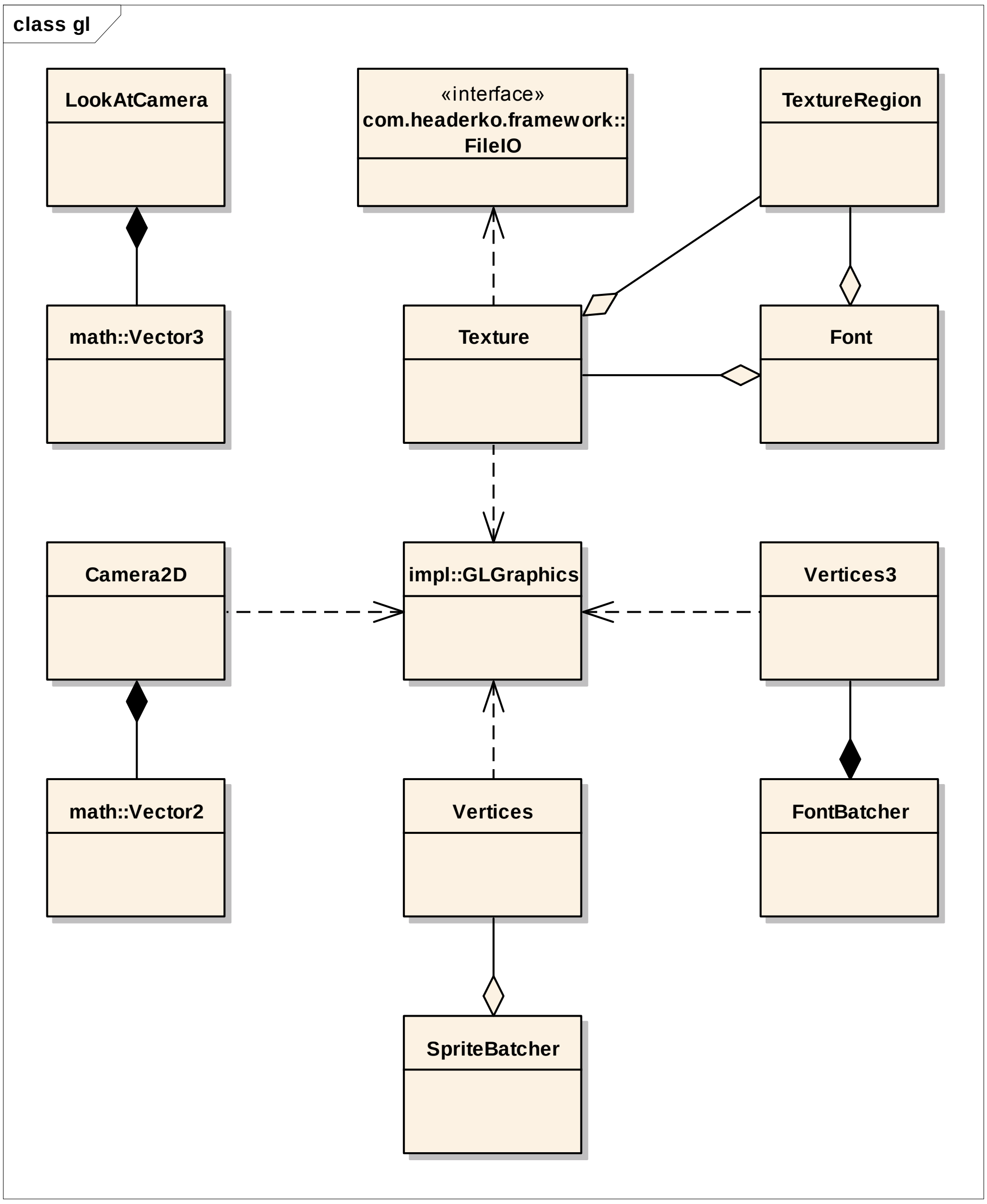
Class model 2 - Package com.headerko.ccs



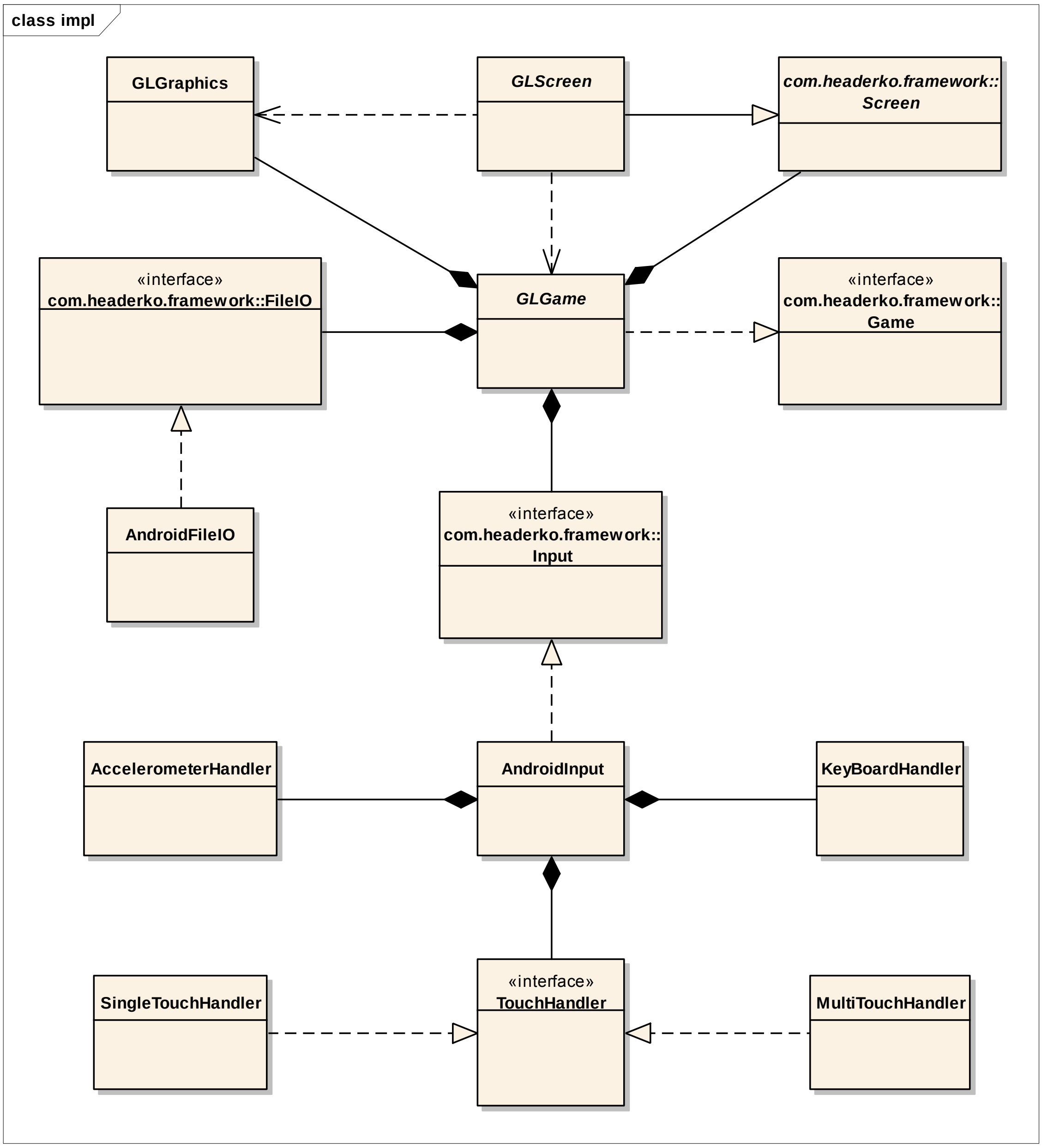
Class model 3 - Package com.headerko.geometry



Class model 4 - Package com.headerko.framework



Class model 5 - Package com.headerko.framework.gl



Class model 6 - Package com.headerko.framework.impl

Appendix D

D.1 IIT.SRC 2012 Paper

This paper was published in the IIT.SRC 2012 student conference proceedings. It was also awarded the Czechoslovakia Section of IEEE price.

Multi-touch Interaction Technique Designed for Three-dimensional Environments on the Screens of Mobile Devices

Filip HLAVÁČEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
filip.hlavacek@gmail.com

Abstract. In this paper we focus on multi-touch interaction techniques with the aim to design an intuitive, easy-to-learn and efficient solution that users would embrace. To reach this goal, we set out by analyzing research on advanced interaction techniques and existing approaches used in applications available on mobile devices. We design our own technique with regard to the six degrees of freedom. The efficiency of our solution shall be verified by applying this technique in an Android based application used by students to support cube cross section education.

1 Introduction

Human-computer interaction (HCI) plays an essential role in today's technology. Large displays, various input devices, high-end mobile phones, all need to adopt specific HCI approaches in order to present their true potential to the common user. Just recently multi-touch displays became a standard for mobile devices. Common interaction techniques for 2D environment manipulation have already been adopted, but 3D interaction techniques are still in development.

This paper reports on a research study investigating the use of virtual reality (VR) in conjunction with multi-touch mobile devices to facilitate the knowledge construction by middle school students of 3-dimenstional (3D) geometry by means of up-to-date education approaches.

Nowadays, technologies and high-end devices available to almost everyone can be used to motivate students with new educational approaches. In order to be able to stimulate students' knowledge construction, such educational application has to be intuitive, fast to master and efficiently provide relevant information to the students. In order to fulfill these requirements we have designed a new multi-touch technique approach with the aim to provide enhanced but easy to learn interaction possibilities.

---

* Master degree study programme in field: Information Systems
Supervisor: Dr. Alena Kovárová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2012, Bratislava, April 25, 2012, pp. 1–6.

2 World of 3D

With new technologies emerging daily and providing users with all kinds of different possibilities the need to evolve the interaction methods grows rapidly. The value of a VR without intuitive or at least fast learnable interaction methods degrades rapidly as the users are demotivated and unwillingly forced to concentrate on the fact that the VR environment they act in is not real, because they are not able to interact in a way they would in the real world.

Human-computer interaction (HCI) is nowadays subject of studies in many different areas. In our research we concentrate on HCI in the areas related to multi-touch interaction, virtual reality, learning environments and mobile devices. The research will be conducted in the field of 3D geometry, also known as solid geometry. Students often have problems projecting shapes drawn on paper into a three-dimensional space in their minds [4]. Further operations on these objects are difficult just to imagine, therefore difficult to comprehend and learn. Presenting such shapes and operations in VR enables the students to focus on the target objects and later on, when already acquainted with necessary experience based knowledge, to project these shapes and operations onto paper or semantics.

There have been many projects that have built virtual reality learning environments (VRLE) for geometry. In the case of VRMath [2] students are opted to complete various tasks. An advanced solution for a VRLE has been presented by Hannes Kaufmann and Dieter Schmalstieg [6] where students are allowed to interact with 3D objects in an immersive VR and therefore directly experience the knowledge.

2.1 Multi-touch interaction

In order to be able to design a usable multi-touch technique, we thoroughly analyzed existing approaches as well as related HCI. When researching interaction techniques we analyzed HCI from the following points of view:

2.1.1 Environment with intuitive control

Each environment is designed so that it fulfills its specific goal, therefore requires a different HCI approach. As these environments focus on simulating reality (VR), HCI must correspond to the interaction possibilities in the real world. It has to be as intuitive as possible and for common tasks, it should allow the users to easily replicate these actions and receive the expected feedback. On the opposite, the primary goal of application specific HCI is to find interaction methods, that effectively cover all application capabilities, rather than to provide intuitive interaction methods.

2.1.2 Haptic surfaces and finger gestures

Depending on the device in question, different interaction approaches have to be used. Some devices allow more straightforward techniques, minimizing the abstraction between the reality and the virtual environment. Haptic surfaces, known as touch surfaces are nowadays present in most mobile devices be it mobile phones, tablets or notebooks (trackpads). Multi-touch surfaces have several issues that need to be addressed. When designing three or more finger gestures, one has to take into account the size of the screen as larger fingers may simply not fit the screen. Fingertip blobs affect the error rate [1] and their examination may help in proposing the user interface design. Apart from straightforward touch interaction techniques, accelerometers and g-sensors have found their way of interaction.

The action of creating the input signal is categorized (based on the required stimuli) into physical or mental. Different devices and techniques aim at various user skills. Multi-touch interaction can be categorized as physical manipulation as users use the muscular system.

3 Evaluating existing approaches

Our primary goal is to find an effective interaction method for mobile devices equipped with multi-touch screens. As most multi-touch practices focus purely on 2D environments the interaction methods in 3D environments are not yet standardized.

Our interest falls onto two application types. As we want to create a 3D geometric multi-touch application, we focus on existing interactive geometry software (IGS) and Android and iOS applications that enable interaction with 3D objects. While the IGS applications in our research are analyzed to present different capabilities of geometry systems, the mobile applications section focuses mainly on the interaction techniques used to manipulate objects in 3D environments. From the analyzed applications and research (e.g. VRMath [2], Construct3D [6] etc.) it is clear that no interaction approach has yet been standardized and therefore applications interpret interactions as they best suit the specific needs. For comparison we analyze the degrees of freedom (DoF) the applications allow.

Tested applications are available at the Apple store or the Android market for free. The devices the applications were tested on are an Apple iPod 2nd generation and a HTC Desire HD. We name only a few of the tested applications: *iSculptor*[1], *iDough*[2], *LookAtCAD*, *i3dViewer*, *ModelView*[3], *Nao3d Viewer Free*.

Apart from analyzing the existing applications, we examined multi-touch interaction techniques backed up by scientific research. The various techniques are analyzed in detail in tables that follow (interactions are noted as [*number of fingers touching the screen*]F [*action*], Table 1 explains the degrees of freedom mapping used in the rest of the tables).

Table 1 - Degrees of freedom mapping

DoF	Result
left / right	translate along the *X* axis
up / down	translate along the *Y* axis
forward / back	translate along the *Z* axis
pitch	rotate about the *X* axis
yaw	rotate about the *Y* axis
roll	rotate about the *Z* axis

Fiorella et al. [5] conducted an experiment comparing classic button user interfaces (UIs) with multi-touch UIs. Their multi-touch interaction technique supports only 4 DoF (see Table 2). This is probably the reasons, which lead them to the conclusion, that “further work is needed in order to achieve a completely satisfactory gesture mapping implementation“.

Table 2 - Fiorella et al. multi-touch interaction technique

DoF	Action
left / right	2F horizontal drag
up / down	2F vertical drag
forward / back	N/A
pitch	1F vertical drag
yaw	1F horizontal drag
roll	N/A

[1] http://itunes.apple.com/us/app/isculptor/id370525280?mt=8#
[2] http://itunes.apple.com/us/app/idough/id386752314?mt=8#
[3] http://zerocredibility.wordpress.com/2010/12/07/3d-model-viewer-for-android/

Hancock, Carpendale and Cockburn have designed three interaction techniques to manipulate 3D objects on tabletop displays [7]. However, only their multi-touch techniques support six degrees of freedom (see Table 3). Their aim was to develop shallow-depth interaction techniques for tabletop displays (the up/down DoF is swapped with the forward/back DoF as we look straight at the top of a table).

The first proposed is a two finger technique based on the Rotate'N Translate (RNT) algorithm [8]. This technique has problems with the yaw DoF as smaller displays (such as mobile phone displays) cannot take full advantage of the RNT algorithm.

Table 3 - Hancock et al. two and three finger multi-touch interaction technique

DoF	Two finger technique	Three finger technique
left / right	1F horizontal drag	1F horizontal drag
up / down	2F pinch	2F pinch
forward / back	1F vertical drag	1F vertical drag
pitch	2F horizontal drag	3F horizontal drag
yaw	1F moving the point of contact	2F drag
roll	2F vertical drag	3F vertical drag

In their work they state that "there has been a general consensus about the separability of rotation and translation. It is widely believed that input is superior if these are kept separate", but on the other hand at the end of their work they state that "People are not only capable of separable simultaneous control of rotation and translation, but prefer it". We believe that whether it is an advantage or a disadvantage depends from the target application. Tabletop displays require fast and imprecise manipulation with objects and therefore do not suffer from minor undesired transformations. Geometry applications on the other hand, could become frustrating to use especially on smaller screens, where finger precision is not as accurate as on larger screens.

Martinet et al. [3] embrace Hancock et al.'s three finger technique as the Z-technique and compare it to the standard viewport technique enhanced with multi-touch capabilities. Their controlled experiment shows that both techniques are equivalent in performance, but the Z-technique was preferred by most participants.

4 Implementation

Designed techniques have been evaluated on a prototype application. The application is aimed for the Android platform as related devices are of different sizes and therefore enable us to evaluate our research more extensively. A custom game framework and OpenGL ES rendering of a 3D environment ensure that students are motivated to test our application.

Based on the empirical evaluation of our first interaction technique, we were able to improve it and design a better one. Both will be compared in extensive user testing and upon evaluation adjustments should be made to provide the best interaction technique.

Thanks to our prototype, we were able to empirically come to the conclusion that complex three finger gestures are difficult to use because:

- three fingers obscure the objects displayed on the screen and therefore lack visual feedback
- most phones screen sizes limit the user "workspace" as they are not big enough and finger movements are limited to only short strokes
- gestures that in combination allow rotations around more than one axis simultaneously are difficult to adopt, unless simulate real world experience

5 Designed interaction technique

In our work we have analyzed various applications related to interaction in 3D environments. Our primary goal is to design an effective interaction interface that will give the users maximum freedom of interaction within a 3D environment. As mentioned earlier, most existing applications lack the 6 DoF. Our aim was to experiment with various interaction approaches and evaluate them.

Based on assumptions that we developed through the examination of techniques mentioned above, we designed our own technique (Table 4) that we implemented in our prototype application.

Table 4 - Custom multi-touch interaction technique

DoF	Action
left / right	1F horizontal drag
up / down	1F vertical drag
yaw	2F horizontal drag
pitch	2F vertical drag
roll	2F circle
forward / back	3F vertical drag

The forward / backward movement could be mapped to a different gesture, e.g. 3F pinch or 3F horizontal drag. As you can see, the 2F pinch action has no DoF mapped to it. In this first technique design we purposely separated the 2 finger pinch gesture to change zoom levels from the 3F drag gesture that translates along the *Z* axis.

Based on the evaluation of the designed interaction technique on our prototype application, we were able to improve this technique. Our redesigned technique described in the table 5 focuses on the following:

- similar gestures for separate interaction categories: 1 finger for selection, 2 fingers for translation, 3 fingers for rotation
- intuitive use: XY plane, X axis and Y axis common gesture approaches (simple dragging results in translation on the XY plane, or rotation about the Y axis for a horizontal stroke)

Table 5 – Improved interaction technique based on prototype evaluation

DoF	Action
left / right	2F horizontal drag
up / down	2F vertical drag
forward / back	2F pinch
pitch	3F vertical drag
yaw	3F horizontal drag
roll	3F circle

Based on the application domain, users might prefer 2 finger gestures for rotation if it will be the more frequent task.

6 Conclusion

We analyzed the most relevant areas related to virtual reality, human-computer interaction, multi-touch interaction techniques and virtual reality learning environments. Based on the evaluation of existing approaches, we designed an interaction technique that allows 6 DoF. This technique has been improved and redesigned after an internal prototype testing. Both techniques are suitable for the use on mobile devices with touch screens. In our research we compare these two techniques to

techniques available in mobile applications nowadays, as well as to techniques designed in other research studies on multi-touch gestures.
After extensive user testing that is planned, we will issue a final verdict, whether we managed to design a technique that can be easily embraced by both users and developers. The testing itself will be executed on our CCS 3D application, which allows students to interact with a cube and construct cube cross sections. Tests are designed so that students have to execute various tasks from simple cube rotations, to tasks where all 6 DoF have to be put to use. At the end, the collected results will be summarized and evaluated.

Through extensive research and a thorough design of an efficiently usable, intuitive, and easy to master technique we bring VR one step closer to the mobile device users.

References

[1] Ahsanullah; Mahmood, A.K.B.; Sulaiman, S.; , "Investigation of fingertip blobs on optical multi-touch screen," Information Technology (ITSim), 2010 International Symposium in, vol.1, no., pp.1-6, 15-17 June 2010 doi: 10.1109/ITSIM.2010.5561307 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5561307&isnumber=5561289

[2] Andy Yeh. 2004. VRMath: knowledge construction of 3D geometry in virtual reality microworlds. In CHI '04 extended abstracts on Human factors in computing systems (CHI EA '04). ACM, New York, NY, USA, 1061-1062. DOI=10.1145/985921.985979 http://doi.acm.org/10.1145/985921.985979

[3] Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2009. 3D positioning techniques for multi-touch displays. In Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST '09), Steven N. Spencer (Ed.). ACM, New York, NY, USA, 227-228. DOI=10.1145/1643928.1643978 http://doi.acm.org/10.1145/1643928.1643978

[4] Cohen, C. A., & Hegarty, M. (2007). Sources of Difficulty in Imagining Cross Sections of 3D Objects. *The 29th Annual Conference of the Cognitive Science Society.* Nashville, Tennessee, USA.

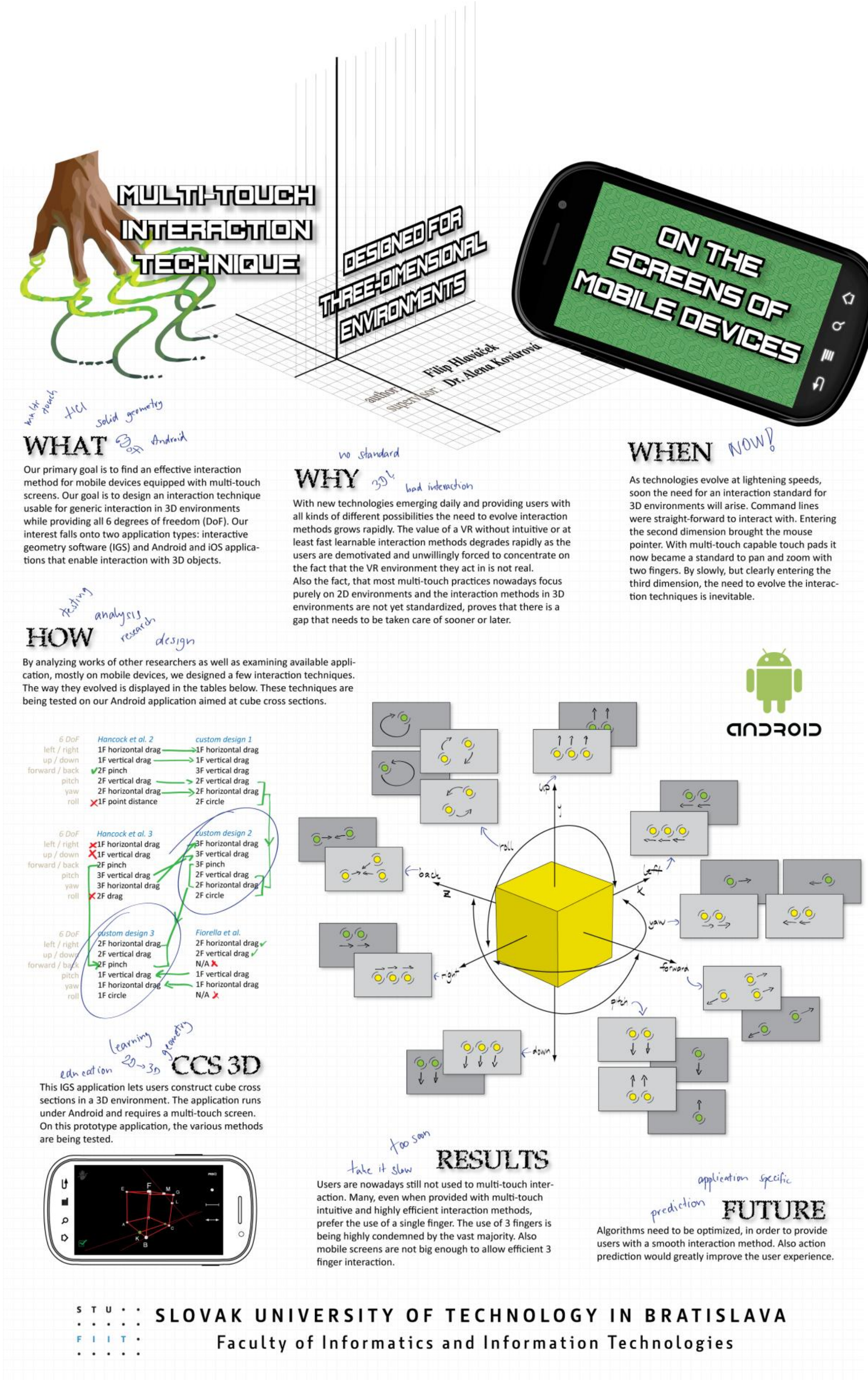
[5] Donato Fiorella, Andrea Sanna, Fabrizio Lamberti, 2010. Multi-touch user interface evaluation for 3D object manipulation on mobile devices, Journal on Multimodal User Interfaces, Springer Berlin / Heidelberg, Issn: 1783-7677 Doi: 10.1007/s12193-009-0034-4 Url: http://dx.doi.org/10.1007/s12193-009-0034-4

[6] Hannes Kaufmann and Dieter Schmalstieg. 2006. Designing Immersive Virtual Reality for Geometry Education. In Proceedings of the IEEE conference on Virtual Reality (VR '06). IEEE Computer Society, Washington, DC, USA, 51-58. DOI=10.1109/VR.2006.48 http://dx.doi.org/10.1109/VR

[7] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07). ACM, New York, NY, USA, 1147-1156. DOI=10.1145/1240624.1240798 http://doi.acm.org/10.1145/1240624.1240798

[8] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. 2005. Fluid integration of rotation and translation. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '05). ACM, New York, NY, USA, 601-610. DOI=10.1145/1054972.1055055 http://doi.acm.org/10.1145/1054972.1055055

D.2 IIT.SRC 2012 Poster



Appendix E

E.1 NordiChi 2012 Short paper

Multi-touch Interaction Technique Designed for Three-dimensional Environments on the Screens of Mobile Devices

ABSTRACT

Human-computer interaction (HCI) plays an essential role in today's technology. Large displays, various input devices, high-end mobile phones, all need to adopt specific HCI approaches in order to present their true potential to the common user. Just recently multi-touch displays became a standard for mobile devices. Common interaction techniques for 2D environment manipulation have already been adopted, but 3D interaction techniques are still in development.

In this paper we focus on multi-touch interaction techniques with the aim to design an intuitive, easy-to-learn and efficient solution that users would embrace. To reach this goal, we set out by analyzing research on advanced interaction techniques and existing approaches used in applications available on mobile devices. We design our own technique with regard to the six degrees of freedom. The efficiency of our solution shall be verified by applying this technique in an Android based application used by students to support cube cross section education.

Author Keywords

Mobile devices; multi-touch; 3D; 6 degrees of freedom; touch interaction; interaction style; multi-touch interfaces.

ACM Classification Keywords

H.5.1 Information interfaces and presentation – Multimedia Information Systems; Artificial, augmented, and virtual realities

H.5.2 Information interfaces and presentation – User Interfaces (D.2.2, H.1.2, I.3.6): Graphical user interfaces (GUI); Input devices and strategies; Interaction styles

General Terms

Design; Experimentation; Standardization.

INTRODUCTION

This paper reports on a research study investigating the use of a multi-touch interaction technique designed for mobile devices in conjunction with interactive geometry software to facilitate the knowledge construction of middle school students of 3-dimenstional (3D) geometry.

With new technologies emerging daily and providing users with all kinds of different possibilities the need to evolve the interaction methods grows rapidly. For example, the value of a VR without intuitive or at least fast learnable interaction methods degrades rapidly as the users are demotivated and unwillingly forced to concentrate on the fact that the VR environment they act in is not real, because they are not able to interact in a way they would in the real world.

Human-computer interaction (HCI) is nowadays subject of studies in many different areas. In our research we concentrate on HCI in the areas related to multi-touch interaction, virtual reality, learning environments and mobile devices. The research will be conducted in the field of 3D geometry, also known as solid geometry. Students often have problems projecting shapes drawn on paper into a three-dimensional space in their minds [4]. Further operations on these objects are difficult just to imagine, therefore difficult to comprehend and learn. Presenting such shapes and operations in VR enables the students to focus on the target objects and later on, when already acquainted with necessary experience based knowledge, to project these shapes and operations onto paper or semantics.

There have been many projects that have built virtual reality learning environments (VRLE) for geometry. In the case of VRMath [2] students are opted to complete various tasks. An advanced solution for a VRLE has been presented by Hannes Kaufmann and Dieter Schmalstieg [6] where students are allowed to interact with 3D objects in an immersive VR and therefore directly experience the knowledge.

Environment with intuitive control

Each environment is designed so that it fulfills its specific goal, therefore requires a different HCI approach. As these environments focus on simulating reality (VR), HCI must correspond to the interaction possibilities in the real world. It has to be as intuitive as possible and for common tasks, it should allow the users to easily replicate these actions and receive the expected feedback. On the opposite, the primary goal of application specific HCI is to find interaction methods, that effectively cover all application capabilities, rather than to provide intuitive interaction methods.

Haptic surfaces and finger gestures

Depending on the device in question, different interaction approaches have to be used. Some devices allow more straightforward techniques, minimizing the abstraction between the reality and the virtual environment. Haptic surfaces, known as touch surfaces are nowadays present in most mobile devices be it mobile phones, tablets or notebooks (trackpads). Multi-touch surfaces have several issues that need to be addressed. When designing three or more finger gestures, one has to take into account the size of the screen as larger fingers may simply not fit the screen. Fingertip blobs affect the error rate [1] and their examination may help in proposing the user interface design.

The action of creating the input signal is categorized (based on the required stimuli) into physical or mental. Different devices and techniques aim at various user skills. Multi-touch interaction can be categorized as physical manipulation as users use the muscular system.

EVALUATING EXISTING APPROACHES

Our primary goal is to find an effective interaction method for mobile devices equipped with multi-touch screens. As most multi-touch practices focus purely on 2D environments the interaction methods in 3D environments are not yet standardized.

Our interest falls onto two application types. As we want to create a 3D geometric multi-touch application, we focus on existing interactive geometry software (IGS) and Android and iOS applications that enable interaction with 3D objects. While the IGS applications in our research are analyzed to present different capabilities of geometry systems, the mobile applications section focuses mainly on the interaction techniques used to manipulate objects in 3D environments. From the analyzed applications and research (e.g. VRMath [2], Construct3D [6] etc.) it is clear that no interaction approach has yet been standardized and therefore applications interpret interactions as they best suit the specific needs.

DoF	Result
left / right	translate along the *X* axis
up / down	translate along the *Y* axis
forward / back	translate along the *Z* axis
pitch	rotate about the *X* axis
yaw	rotate about the *Y* axis
roll	rotate about the *Z* axis

Table 1 - Degrees of freedom mapping.

The devices the applications were tested on are an Apple iPod 2nd generation, on a HTC Desire HD and on an HTC EVO 3D. We name only a few of the tested applications: iSculptor1, iDough2, LookAtCAD, i3dViewer, ModelView3, Nao3d Viewer Free, AutoCAD WS.

There is no consistency in the interaction techniques designed for these applications however some of them can be regarded as sufficient.

Apart from analyzing the existing applications, we examined multi-touch interaction techniques backed up by scientific research. The various techniques are analyzed in detail in tables that follow (interactions are noted as [number of fingers touching the screen]F [action], Table 1 explains the degrees of freedom mapping used in the rest of the tables).

Fiorella et al. [5] conducted an experiment comparing classic button user interfaces (UIs) with multi-touch UIs. Their multi-touch interaction technique supports only 4 DoF (see Table 2). This is probably the reasons, which lead them to the conclusion, that "further work is needed in order to achieve a completely satisfactory gesture mapping implementation".

Hancock, Carpendale and Cockburn have designed three interaction techniques to manipulate 3D objects on tabletop displays [7]. However, only their multi-touch techniques support six degrees of freedom (see Table 3). Their aim was to develop shallow-depth interaction techniques for tabletop displays (the up/down DoF is swapped with the forward/back DoF as we look straight at the top of a table).

DoF	Action
left / right	2F horizontal drag
up / down	2F vertical drag
forward / back	N/A
pitch	1F vertical drag
yaw	1F horizontal drag
roll	N/A

Table 1 - Fiorella et al. multi-touch interaction technique.

Two finger technique	DoF	Three finger technique
1F horizontal drag	left / right	1F horizontal drag
2F pinch	up / down	2F pinch
1F vertical drag	forward / back	1F vertical drag
2F horizontal drag	pitch	3F horizontal drag
1F moving the point of contact	yaw	2F drag
2F vertical drag	roll	3F vertical drag

Table 2 - Hancock et al. two and three finger multi-touch interaction technique

The first proposed is a two finger technique based on the Rotate'N Translate (RNT) algorithm [8]. This technique has problems with the yaw DoF as smaller displays (such as mobile phone displays) cannot take full advantage of the RNT algorithm.

In their work they state that "there has been a general consensus about the separability of rotation and translation. It is widely believed that input is superior if these are kept separate", but on the other hand at the end of their work they state that "People are not only capable of separable simultaneous control of rotation and translation, but prefer it". We believe that whether it is an advantage or a disadvantage depends from the target application. Tabletop displays require fast and imprecise manipulation with objects and therefore do not suffer from minor undesired transformations. Geometry applications on the other hand, could become frustrating to use especially on smaller screens, where finger precision is not as accurate as on larger screens.

Martinet et al. [3] embrace Hancock et al.'s three finger technique as the Z-technique and compare it to the standard viewport technique enhanced with multi-touch capabilities. Their controlled experiment shows that both techniques are equivalent in performance, but the Z-technique was preferred by most participants.

IMPLEMENTATION

Designed techniques have been evaluated on an application (see Figure 1). Attractive OpenGL ES rendering of a 3D environment motivates others to play with the application.

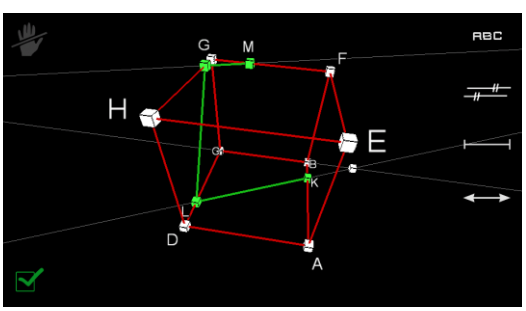


Figure 1 – Cube Cross Sections 3D application

Based on the empirical evaluation of our draft interaction technique, we were able to improve it and redesign it to a better one. Thanks to our application, we were able to empirically come to the conclusion that complex three finger gestures are difficult to use because:

- three fingers obscure the objects displayed on the screen and therefore lack visual feedback
- most phones screen sizes limit the user "workspace" as they are not big enough and finger movements are limited to only short strokes
- gestures that in combination allow rotations around more than one axis simultaneously are difficult to adopt, unless simulate real world experience

DESIGNED INTERACTION TECHNIQUE

As mentioned earlier, most existing applications lack the 6 DoF. Our aim was to experiment with various interaction approaches and evaluate them.

Based on assumptions which we developed through the examination of techniques mentioned above, we designed our first technique (Table 4) that we implemented in our prototype application.

The forward / backward movement could be mapped to a different gesture, e.g. 3F pinch or 3F horizontal drag. As you can see, the 2F pinch action has no DoF mapped to it. In this first technique design we purposely separated the 2 finger pinch gesture to change zoom levels from the 3F drag gesture that translates along the *Z* axis.

Based on the evaluation of the designed interaction technique on our prototype application, we were able to improve this technique.

DoF	Action
left / right	1F horizontal drag
up / down	1F vertical drag
yaw	2F horizontal drag
pitch	2F vertical drag
roll	2F circle
forward / back	3F vertical drag

Table 4 - Custom multi-touch interaction technique

DoF	Action
left / right	3F horizontal drag
up / down	3F vertical drag
forward / back	3F pinch
pitch	2F vertical drag
yaw	2F horizontal drag
roll	2F circle

Table 5 – Redesigned interaction technique

Our redesigned technique described in the table 5 is improved by the following:

- similar gestures for separate interaction categories: 1 finger for selection, 2 fingers for translation, 3 fingers for rotation
- two fingers are used for rotation, as rotations are harder to master for most users. Translations, being simple, easy to understand, actions, are mapped to three fingers, because most users are not used to three finger interaction
- unspecified single finger interaction: apart from selection, it can be used with gesture drawing

In our test of this technique on a mobile phone, most participants were surprised with the concept of using three fingers. Many participants disliked the concept, because three fingers are simply one too many on small screens. Based on further evaluation we came to the conclusion that on devices with small screens, where three fingers touching the screen simultaneously obscure the visualization, this technique is not suitable. On the account on rotation separation, and one finger interaction being limited to rotation only, we defined a small screen interaction technique, presented in table 6.

DoF	Action
left / right	2F horizontal drag
up / down	2F vertical drag
forward / back	2F pinch
pitch	1F vertical drag
yaw	1F horizontal drag
roll	2F circle

Table 6 – Small screen limited interaction technique

CONCLUSION

We analyzed the most relevant areas related human-computer interaction in conjunction with multi-touch interaction. Based on the evaluation of existing approaches, we designed an interaction technique that allows 6 DoF. This technique has been redesigned and another technique for small screens has been defined. In our research we compare these two techniques to techniques available in mobile applications nowadays, as well as to techniques designed in other research studies on multi-touch gestures.

Most users mastered the small screen technique easily, as its mutations are nowadays used across various devices (trackpads). When presented with the three finger technique the users were puzzled at first, but were able to interact effortlessly and use the technique efficiently after a short while. Based on promising test results, we believe that after thorough testing and evaluation this interaction technique could lay the foundations for the path to the standardization of 3D environment interaction on multi-touch mobile screens.

Through extensive research and a thorough design of an efficiently usable, intuitive, and easy to master technique we bring VR one step closer to the mobile device users.

REFERENCES

1. Ahsanullah; Mahmood, A.K.B.; Sulaiman, S.; , "Investigation of fingertip blobs on optical multi-touch screen," Information Technology (ITSim), *2010 International Symposium* in, vol.1, no., pp.1-6, 15-17.
2. Andy Yeh. 2004. VRMath: knowledge construction of 3D geometry in virtual reality microworlds. *CHI EA '04*. ACM, New York, NY, USA, 1061-1062.
3. Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2009. 3D positioning techniques for multi-touch displays. *Proc. VRST '09*, Steven N. Spencer (Ed.). ACM, New York, NY, USA, 227-228.
4. Cohen, C. A., & Hegarty, M. (2007). Sources of Difficulty in Imagining Cross Sections of 3D Objects. *The 29th Annual Conference of the Cognitive Science Society.* Nashville, Tennessee, USA.
5. Donato Fiorella, Andrea Sanna, Fabrizio Lamberti, 2010. Multi-touch user interface evaluation for 3D object manipulation on mobile devices, Journal on Multimodal User Interfaces, Springer Berlin / Heidelberg
6. Hannes Kaufmann and Dieter Schmalstieg. 2006. Designing Immersive Virtual Reality for Geometry Education. *Proc. IEEE VR '06*. IEEE Computer Society, Washington, DC, USA, 51-58.
7. Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. *Proc. CHI '07*. ACM, New York, NY, USA, 1147-1156.
8. Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. 2005. Fluid integration of rotation and translation. *Proc. CHI '05*. ACM, New York, NY, USA, 601-61

APPENDIX F

F.1 Compact disk contents

\application

\apk

\source code

\documents

\thesis

\IIT.SRC2012

\NordiCHI2012

\questionnaire

\template

\collected data

RESUMÉ

V dnešnej dobe postupuje vývoj technológii míľovými krokmi. Len nedávno sa rozšírili inteligentné mobilné telefóny medzi bežných používateľov, hlavne vďaka cenovej dostupnosti a atraktívnej ponuke. V posledných rokoch sa stali štandardom veľké dotykové obrazovky, na ktorých používateľ priamo interaguje s daným prostredím. Dnes je už štandardom ponúkať na týchto zariadeniach viac dotykové obrazovky. Našim cieľom bolo zamerať sa na ovládanie trojrozmerného priestoru virtuálnej reality interagovaním viacerými prstami súčasne.

Oblasti záujmu

Našu prácu sme začali výskumom. Zamerali sme sa na existujúce riešenia ako aj analýzu oblastí týkajúcich sa našej problematiky. Keďže ako doménu na testovanie navrhnutej interakcie sme si zvolili stredoškolské rezy kockou, zaujímali nás aj existujúce výučbové prostredia vo virtuálnej realite. Tieto sa zameriavajú na poskytovanie takých prostriedkov pre študentov, ktoré im umožňujú jednoduchšie získať dané vedomosti. Zamerali sme sa na výučbu geometrie a medzi hlavných reprezentantov tejto kategórie patria Cabri 3d, Archimedes Geo3D, GeoGebra a iné.

Oblasť interakcie človeka s počítačom sme rozoberali z viacerých hľadísk. V prípade, že sa snažíme kategorizovať interakciu z pohľadu cieľového prostredia, vieme typy interakcie rozdeliť do troch skupín, a to: závislá na aplikácii, obohatená realita, virtuálna realita. Každá skupina by sa dala charakterizovať úrovňou pohltenia používateľa. Pričom pri interakcii aplikačne závislej je takmer nulová, vo virtuálnej realite sa snaží dosahovať maxima. Spôsoby interakcie však môžeme ďalej kategorizovať podľa typu používateľa pre ktorého sú určené, podľa zariadení a technológii, ktoré využíva, prípadne podľa vstupného stimulu. Interagovať totiž môžeme fyzicky alebo mentálne. Pri fyzickej interakcii využívame naše svalstvo, či už rukami, nohami alebo aj očami či ústami, pričom pri mentálnom ovládaní, stačí „myslieť".

Existujúce prístupy

V tejto časti práce sme sa zamerali na hľadanie existujúcich riešení. Zamerali sme sa opäť na oblasti spomínane vyššie. Okrem podrobnejšej analýzy softvérových riešení výučby geometrie v trojrozmernom priestore sme sa venovali prevažne aplikáciám na mobilných zaradeniach. Operačné systémy na ktoré sme sa zamerali sú Android a iOS, keďže v dnešnej dobe majú najväčšie zastúpenie. Medzi preberané aplikácie patria napríklad: iDough, LookAtCAD, ModelView a AutoCAD WS. Okrem samotných aplikácii, prebieha v danej oblasti aj vedecký výskum. Medzi významné publikácie v tejto oblasti patria určite výskumy pánov Fiorella

a kolektívu a pánov Hancock, Carpendale a Cockburn. Fiorella a kol. navrhli techniku, ktorá však výrazne obmedzovala používateľa, keďže ponúka len 4 zo 6 stupňov voľnosti (DoF).

DoF	Akcia
Posun vľavo / vpravo	2 prstami horizontálny posun
Posun dole / hore	2 prstami vertikálny posun
Posun vpred / vzad	-
Natáčanie hore/dole	1 prstom vertikálny posun
Otáčanie do strán (vpravo a vľavo)	1 prstom horizontálny posun
Nakláňanie (otáčanie v smere a proti smeru hodinových ručičiek)	-

Hancock a kol. navrhli dve techniky, avšak iba ich trojprstová technika efektívne ponúka všetkých 6 DoF.

DoF	Two finger technique	Three finger technique
Posun vľavo / vpravo	1 prstom horizontálny posun	1 prstom horizontálny posun
Posun dole / hore	2 prsty k sebe / od seba	2 prsty k sebe / od seba
Posun vpred / vzad	1 prstom vertikálny posun	1 prstom vertikálny posun
Natáčanie hore/dole	2 prstami horizontálny posun	3 prstami horizontálny posun
Otáčanie do strán (vpravo a vľavo)	1 prstom posun bodu kontaktu	2 prstami posun
Nakláňanie (otáčanie v smere a proti smeru hodinových ručičiek)	2 prstami vertikálny posun	3 prstami vertikálny posun

Ich riešenie bolo zameraná na „plytké“ interaktívne stoly. Ich dvojprstová technika narazila na problém, že nie je efektívne použiteľná na malých obrazovkách, lebo sa nedá využiť plný potenciál Rotate’N Translate algoritmu. V ich práci vyhlásili, že „nie len že sú ľudia schopní separovaného simultánneho ovládania rotácie a translácie, ale preferujú ho“. Myslíme si, že kombinovať simultánnu transláciu s rotáciou je vhodné len v niektorých špecifických prípadov, prípadne záleží na aplikačnej doméne, kde podľa potreby definujeme, súbor niektorých zo stupňov voľnosti, ktorými budeme najčastejšie manipulovať. To však opäť koncového používateľa núti oboznamovať sa v každej aplikácii samostatne s novým spôsobom interakcie. Preto sme sa rozhodli navrhnúť naše techniky tak, že by boli univerzálne použiteľné.

Špecifikácia

Na základe spomínanej analýzy sme si stanovili nasledovné ciele:

- výučba stereometrie
- mobilné zariadenia
- multi-dotykové ovládanie
- a doplnkový cieľ – motivácia

Pre prvé dva ciele sme si stanovilo rôzne požiadavky a nároky na konečnú aplikáciu. Medzi mobilné zariadenie, ktoré nás zaujímajú patria inteligentné mobilne telefóny a tablety. Pre tieto zariadenia sme navrhli na základe analýzy prvotnú techniku interakcie. Cieľom nášho návrhu ovládania, je ponúknuť používateľom všetkých 6 stupňov voľnosti. V nasledujúcej tabuľke je mapovanie jednotlivých vybraných gest na stupne voľnosti (DoF).

DoF	**Akcia**
Posun vľavo / vpravo	1 prstom horizontálny posun
Posun dole / hore	1 prstom vertikálny posun
Posun vpred / vzad	3 prstami vertikálny posun
Natáčanie hore/dole	2 prstami vertikálny posun
Otáčanie do strán (vpravo a vľavo)	2 prstami horizontálny posun
Nakláňanie (otáčanie v smere a proti smeru hodinových ručičiek)	2 prstami krúženie

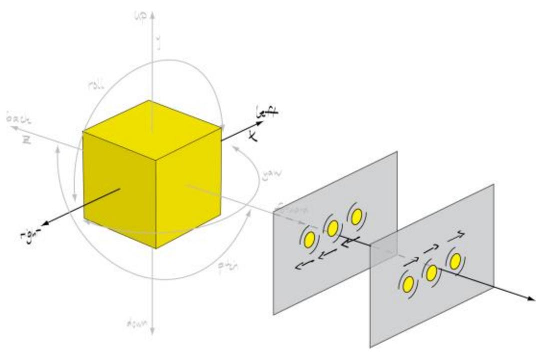
Implementácia

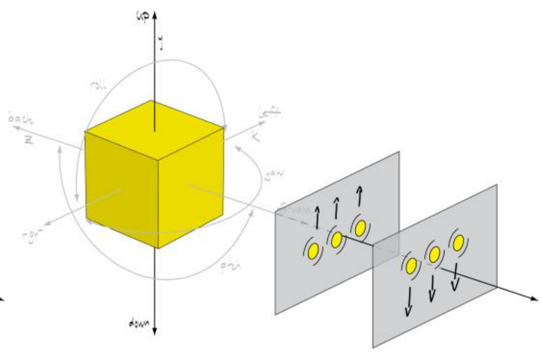
Nasledovala implementácia prototypu. Na prototype sme interne experimentovali s ďalšími komplexnými návrhmi, zameranými na získanie poznatkov z rôznych uhlov pohľadu. Po prototypovaní sme dospeli k záveru, že navrhovanú techniku treba vylepšiť a to z nasledujúcich dôvodov:

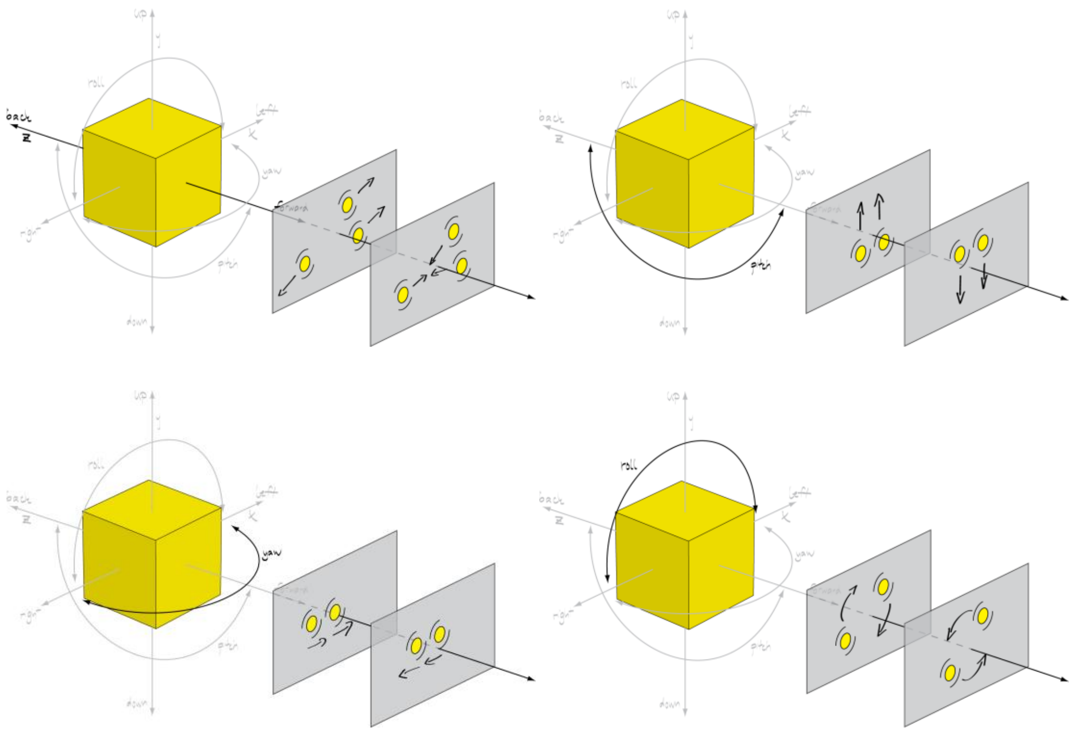
- posun po osi Z nebol dostatočne intuitívny
- translácia po osi Z bola separovaná od simultánnej translácii na osiach X a Y

V nasledujúcej tabuľke uvádzame upravenú techniku. Jej výhodou je separovanie translácie od rotácie a zároveň poskytuje priestor na ľubovoľné mapovanie interakcie na jeden prst. Translácia je namapovaná na tri prsty z dôvodu, že je jednoduchším úkonom ako rotácia. Vzhľadom na nezvyk a obtiažnejšie používanie troch prstov je korektné namapovať na ne jednoduchšiu operáciu. Za tabuľkou nasledujú ilustrácie reprezentujúce danú techniku.

DoF	**Akcia**
Posun vľavo / vpravo	3 prstami horizontálny posun
Posun dole / hore	3 prstami vertikálny posun
Posun vpred / vzad	3 prsty k sebe / od seba
Natáčanie hore/dole	2 prstami vertikálny posun
Otáčanie do strán (vpravo a vľavo)	2 prstami horizontálny posun
Nakláňanie (otáčanie v smere a proti smeru hodinových ručičiek)	2 prstami kružnica







Pri prezentovaní našich návrhov okoliu, sme dostavali prvotnú odozvu bazírujúcu na ich reakcii. Väčšine oslovených sa nápad s tromi prstami nepáčil (už dva prsty im prišli zbytočné). Na základe týchto impulzov sme navrhli zjednodušenú metódu interakcie, ktorá využíva maximálne dva prsty. Uvedená je v nasledujúcej tabuľke ako naša dvojprstová technika.

DoF	Akcia
Posun vľavo / vpravo	2 prstami horizontálny posun
Posun dole / hore	2 prstami vertikálny posun
Posun vpred / vzad	2 prsty k sebe / od seba
Natáčanie hore/dole	1 prstom vertikálny posun
Otáčanie do strán (vpravo a vľavo)	1 prstom horizontálny posun
Nakláňanie (otáčanie v smere a proti smeru hodinových ručičiek)	1 prstom krúženie

Testovanie

Jednou z najrelevantnejších častí našej práce je testovanie. Spomínané techniky boli testované na študentoch gymnázia na Hubeného ulici č.23. Dokopy sa nám podarilo nazbierať 19 vzoriek. Test pozostával z vyplňovania dotazníka a interakcie s mobilným zariadením. Testovanie prebehlo na mobilnom telefóne HTC EVO 3D. Aplikácia na ktorej boli spomínané techniky testované bola naša CCS 3D aplikácia, ktorá ponúka používateľom možnosť riešiť konštrukciu rezov kocky v trojrozmernom priestore. Cieľom študenta bolo zostrojiť rez s využitím našej trojprstovej techniky.

Na základe zozbieraných výsledkov sme dospeli k nasledovným tvrdeniam:

- používatelia preferujú jednoduchosť, čím menej prstov, tým lepšie
- tri prsty na obrazovke mobilného telefónu je priveľa

- používatelia preferujú gestá, s ktorými sa už stretli
- simultánne ovládanie translácie vo všetkých troch stupňoch voľnosti je hodnotené pozitívne a dokáže zvýšiť efektívnosť
- simultánna rotácia do strán a hore a dole je preferovaná, ale v kombinácii s nakláňaním je ťažko pochopiteľná, nie to ešte použiteľná

Záver

Výsledkom našej práce je návrh dvoch techník interakcie. Dvojprstová technika je vhodná pre mobilné telefóny, ktorých obrazovka neposkytuje dostatok priestoru na pohodlnú interakciu troma prstami. Trojprstovú techniku odporúčame pre tablety, pri ktorých sa veľkosti obrazovky pohybujú od 7 palcov vyššie. Keďže tablety sa práve dostávajú do popredia, je teraz ten správny čas na zavedenie nového štandardu. Interkacia v trojrozmernom priestore by výrazne ťažila z navrhnutej trojprstovej interakcie.