UNIVERSIDAD POLITÉCNICA DE VALENCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

MASTER THESIS

Autostereoscopy vs. non-autostereoscopy on the LG Optimus 3D

Author:
Katharina Maria Kaczmarczyk

Supervisor:
Dr. M.Carmen Juan Lizandra

Kaczmarczyk, Katharina Maria
Autostereoscopy vs. non-autostereoscopy on the LG Optimus 3D
Master thesis, Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia, 14 September 2013

This thesis is a part of the Spanish project: *DEVELOPMENT AND VALIDATION OF AUGMENTED REALITY SYSTEMS FOR LEARNING-ENTERTAINMENT,* known as *APRENDRA* (TIN2009-14319-C02-01, 1-1-2010/31-12-2012).

ABSTRACT

The rapid increase of 3D capable devices has provided a series of new and advanced home entertainment systems; that indicates a higher number of demands for 3D contents, such as 3D movies, 3D TV series and 3D games. As a result, this technology has been applied already on the displays of Smartphones and handheld video gaming consoles.

In this thesis, a study between autostereoscopy and non-autostereoscopy on a Smartphone was carried out by testing a new *Android* application that provides both visualization modes with user interactions. The new app contains a number of static and dynamic objects in a 3D environment. Evaluation findings indicate that people are interested in 3D game content on Smartphones. However, perception issues of 3D virtual objects and loss of picture quality demonstrate that this technology still needs further improvements before it can become suitable for all groups of people.

RESUMEN

El rápido aumento de los dispositivos con características 3D ha permitido una serie de sistemas de entretenimiento nuevos y avanzados para casa, hecho que ha aumentado la demanda de contenidos en 3D: películas en 3D, series en 3D y videojuegos en 3D. Esta tecnología ya se ha aplicado en las pantallas de los teléfonos inteligentes y videoconsolas portátiles.

En esta tesina, se realizó un estudio sobre una aplicación para dispositivos Android con dos modos de visualización e interacción con el usuario: con autoestereoscopía y sin autoestereoscopía. Esta aplicación contiene varios objetos, tanto estáticos como dinámicos, en un entorno 3D. Tras realizar la evaluación, los resultados indican el alto grado de interés que tienen los contenidos en 3D para juegos en teléfonos inteligentes. Sin embargo, los problemas de percepción de objetos virtuales en 3D demuestran que esta tecnología todavía necesita mejoras para proporcionar una percepción de profundidad sin pérdida de nitidez en la imagen para que sea adecuada a un amplio grupo de la población.

ACKNOWLEDGEMENTS

First of all, I want to thank all staff from *Master in Artificial Intelligence, Pattern Recognition and Digital Imaging* from the *Polytechnic University of Valencia* (*Universidad Politécnica de Valencia)* for the provided training, good atmosphere and close collaboration over the past two years.

In particular I want to thank my supervisor Dr. M.Carmen Juan Lizandra for her expert guidance, her patience and inspiration during the realization of this thesis. A special thank to my beloved brothers Thomas David Kaczmarczyk and Sebastian Siegmund Kaczmarczyk for their willingness as beta-testers, encouragement through sometimes difficult moments. Another thank goes to Nuria Martínez Soriano for her permission to use her Nintendo 3DS. I am thankful to my parents, family and friends for their encouragement, understanding and support over the last few months.

"Friendship is the hardest thing in the world to explain. It's not something you learn in school. But if you haven't learned the meaning of friendship, you really haven't learned anything"

— Muhammad Ali

In order to show my thanks, I dedicate this thesis to all of them.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

I

Introduction

INTRODUCTION

1.

1.1 Motivation

The amount of devices with stereoscopic and autostereoscopic displays has increased in the past few years. Especially in the entertainment media area, this technology is gaining popularity. In stores, there are several 3D display devices, such as 3D televisions, 3D displays (LCD/LED), 3D laptops, 3D cameras and mobile devices (*Nintendo 3DS* and Smartphones). However, most of these devices are using stereoscopic displays, which require the usage of special 3D viewer (e.g. liquid crystal shutter) in order to perceive the 3D image.

In 2011, the *Nintendo 3DS*, the *LG Optimus 3D* and the *HTC Evo 3D* were released as mobile devices with an autostereoscopic display, which do not need special glasses (GSMArena, 2011; LG Electronics, 2011; Nintendo Co. America, 2011). Except for the Nintendo 3D, users interact directly through touch gestures with the autostereoscopic display.

With Smartphones now outselling feature phones and 225 million Smartphones sold worldwide in second quarter of 2013 alone (BBC, 2013a), touch gesture based controls such as *Natural User Interfaces (NUI)* have found their way into our daily schedule.

While the gaming market on modern Smartphones is dominated by 2D game-apps such as *Angry Birds*, the devices do offer sophisticated 3D graphics acceleration which allows for 3D Games such as: *Monster life*, developed by *Gameloft* (Gameloft, 2013).

Now the question is, if there is a significant difference between the perception and touch gesture in a full-3d autostereoscopic environment or not. It is known, that the accuracy difference of perception and touch gesture for 2D autostereoscopic environments are nearly-significant, especially for moving 2D objects on a Smartphone or on a 3D display with an additional remote control (*Nexus One* Smartphone). Nevertheless, some participants preferred the autostereoscopic mode over the non-autostereoscopy (Halvey et al., 2012; Van Knotsenburg, 2012).

This thesis will demonstrate that with current technology, moving virtual objects through touch gestures with full-3D autostereoscopic visualization can be done as accurately and quickly as without autostereoscopic visualization. To verify this, an *Android* app for the *LG Optimus 3D P920* was developed that contains several experiments - two individual games with a series of stages - with both visualization modes (autostereoscopic and non- autostereoscopic).

1.2 Scientific goals and research hypotheses

The main objective of this thesis is to determine the advantages of new technologies, in this case combining *Android Natural User Interfaces* with an autostereoscopic display. To achieve this, we have established several goals:

- To develop an *Android* app that attracts a user's attention and also engages them to the new technology.

- To study the interaction of users with an autostereoscopic display compared to non- autostereoscopic visualization.
- To study the interaction of users with *Android Natural User Interface* and autostereoscopic visualization.
- To test the *Android* app with a set amount of participants.
- To design two questionnaires, each with their own proposes. The first, to determine user acceptance. The second, to analyze if and how the technology aids to solve different activities given.
- To provide a statistical analysis of the results.

In order to achieve these goals, the thesis was divided in three phases:

1. **Previous study of the Android SDK in combination with LG Real3D SDK.**

At this phase, the first step was to install, configure and test all the given Android demo apps from the Android and *LG Real3D SDK*. To better understand the *LG Real3D SDK* demo examples, the two autostereoscopic demos "GLbased Earth rotation" from "Real3D samples" and "OpenGL" from "Real3dApiDemos" were combined into one demo. Apart from the visualization, the new demo provided an *Android Option Menu*, its function was to enable or disable the autostereoscopic scene visualization of the rotating earth.

For the development of an application with *Natural User Interaction*, it was important to get familiar with the *Android SDK*, in particular with the touch gestures: tap, drag and swipe. Therefore, a part of the main menu programming was done separately to the new autostereoscopic visualization demo.

2. **Development of the Android app**

The development of the app was organized in four sections:

(a) Creating virtual primitives with textures with the Android graphic library.

(b) Designing a user-friendly menu: creating images for the buttons and background, using view and its properties and finally using some XML Android scripts.

(c) Developing the visualization activity with primitives, control buttons and solution control for every stage.
(d) Writing the source code that creates a txt file in the Smartphone.

In some cases, two or more unfinished sections were executed simultaneous. Especially in the case of the app flowchart development, it was necessary to combine (b) and (c), although both sections were unfinished. The same situation occurs for the Multilanguage implementation.

After finishing the development, the app was tested by two beta-testers in order to minimize the risk of unexpected errors or behaviors during an experiment session.

3. Autostereoscopy vs. non- autostereoscopy

The objective of this study is to figure out which visualization mode is more appreciated from the subjects. Also if there exists any significant difference between both modes by realizing certain activities, such as time differences, accuracy and perception. The first of two hypotheses is that the participants will prefer the autostereoscopic visualization over the non-autostereoscopic visualization, such as in the experiment of a 2D autostereoscopic environment. And the other hypothesis is that the autostereoscopic visualization would lead to greater test result and would be easier to use.

The following facts support both hypotheses:

(a) Although both visualization modes possess the same Android Natural User Interfaces, the autostereoscopic visualization provides a 3D perception sensation that should improve the stereopsis effect.
(b) Since the release of mobile devices with autostereoscopic displays, the participant is able to adjust the eye distance factor (offset) by moving the device further away or using the implemented application’s slider.

1.3 Thesis structure

This thesis document is organized in following chapters.

Chapter 1 explains the motivation, scientific goals and research hypotheses of this study.

Chapter 2 studies the background and state of the art of autostereoscopic technology in mobile devices. Also reviews the most relevant literature related to this study.

Chapter 3 describes the development process of the used Android application for the research hypothesis of this study. Apart from the development, this chapter focuses also on the hardware and software characteristics.

Chapter 4 describes the study of this thesis, where the application was evaluated by a group of adults. They tested the application with and without autostereoscopic projection.

Chapter 5 concludes the thesis and analyses future prospects of an autostereoscopic application.

Appendix Chapter A1 presents the questionnaires, which have been used for the research hypothesis of this study

II

State of the art

STATE OF THE ART

2.

2.1 Introduction

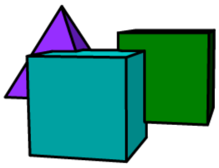
In the market, every stereoscopy or autostereoscopy display device is using the term "3D" in their brand name, e.g. *Nintendo 3DS, HTC Evo 3D* and *LG Optimus 3D*. Now, the question is, when the 3D technology was discovered, what are the differences between both 3D technologies and what are the specifications of enabled 3D devices.

2.2 Stereoscopy and Autostereoscopy

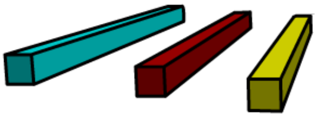
Before starting with the explanations of both 3D technologies, it is important to cover the 3D visualization types. In other words, when is an object closer or further away from the viewer?

In a 3D environment three characteristics are combined in order to achieve the monocular stereopsis vision (e.g. similar to using one eye instead of two): visible surface determination, parallax and size constancy. Or in photography and art terms, this phenomenon is known as perspective.

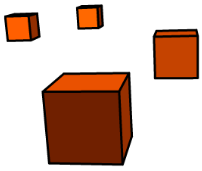
- The visible surface determination (also known as occlusion culling) points out which part of the object is visible or not, when another object is in front of it or the viewport position has changed (Figure 2.1a).
- The parallax draws the line from the front face to the end of the object in the direction of the invisible vanishing point (Figure 2.1b).
- The size constancy describes the Z-position of each object in the scene. If the object is closer to the viewer, then it is drawn bigger as the other objects, even though all objects have the same size (Figure 2.1c).



(a) Visible surface determination



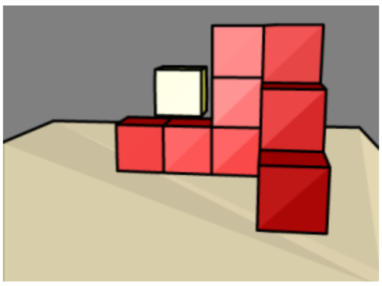
(b) Parallax



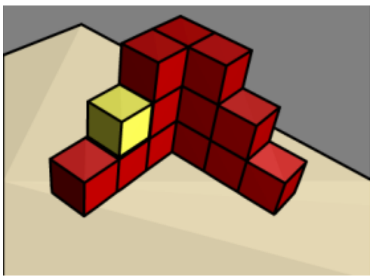
(c) Size constancy

Figure 2.1: 3D environments characteristics.

By not providing the size constancy, a change of the viewport can provoke an optical illusion. In Figure 2.2, an optical illusion from the game *Super Mario 3D Land* world 2-1 was rebuild in order to demonstrate the importance of the size constancy. The yellow cube was scaled by the factor 0.93. After the viewport was manipulated, the yellow cube had the same size as the red ones. In some cases an optical illusion is wished to create a puzzle, such as in *Super Mario 3D Land*.



(a) Yellow cube is the smallest element



(b) Yellow cube has now the same size after changing the viewport.

Figure 2.2: An optical illusion created by changing the viewport and object size.

To achieve the 3D depth-illusion of an image three characteristics are required:

- Simultaneous perception: each eye receives its image (viewport) simultaneously. In archery for example, the standard convention for bows depends upon the eye dominance. In order to know which eye is dominant, a simple trick is used. First the subject has to select a target object to observe such as a door handle, picture, etc. The next step is to do the OK hand gesture by connecting the thumb and forefinger into a circle (the O). Now the subject has to look through the O at the selected target. By closing the non dominant eye, the focused point disappears due to the viewport change. This is similar to the "jumping thumb" example.

  In Figure 2.3 the dominant eye is the left eye, therefore the final object is drawn similar to the left viewport image.
- Fusion of monocular images / binocular parallax: combining both images into one.
- Stereopsis: the impression of depth by viewing an image with binocular vision.

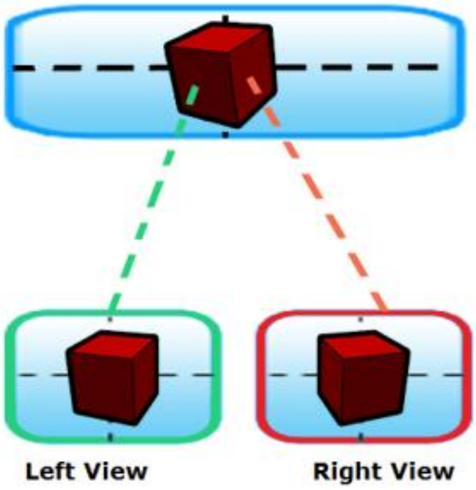
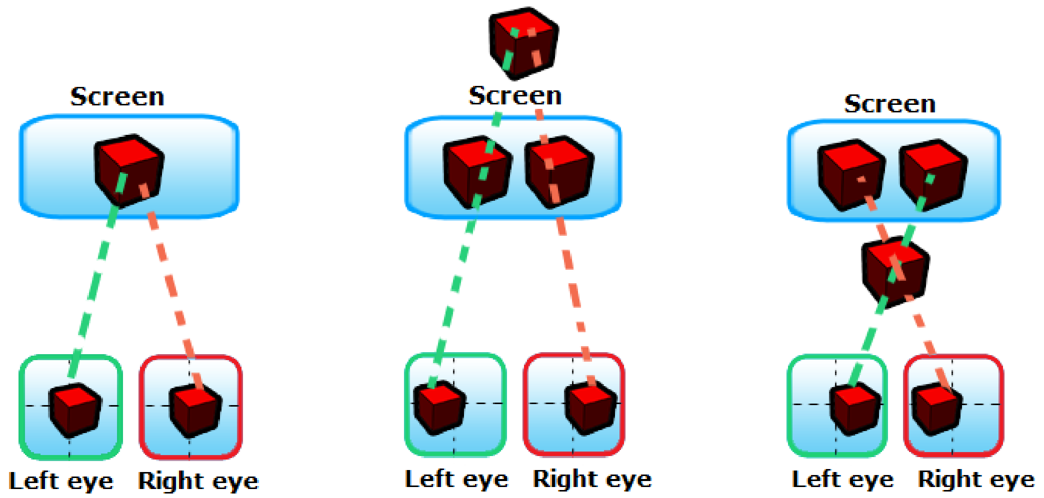


Figure 2.3: Example of both eye's viewports of an cube.

The binocular parallax is divided into 3 classes:

- Zero parallax gives impression of depth in the screen (Figure 2.4a).
- Positive parallax gives impression that the object appears further than the screen (Figure 2.4b).
- Negative parallax gives impression that the object appears outside of the screen (Figure 2.4c).



(a) zero parallax (b) positive parallax (c) negative parallax

Figure 2.4: The binocular parallax definitions.

Day (1994) has provided good examples for depth impression by using the negative parallax in a single image on a piece of paper. Each eye is focusing the image; by default the left and right images are slightly dissimilar due to the viewport (see Figure 2.3).

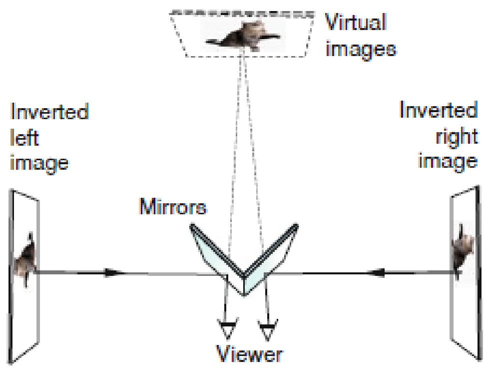
2.2.1 Background of Stereoscopy and Autostereoscopy

In the following section, an overview of the historical development of stereoscopic and autostereoscopic displays is shown. This review is a summary of the study of Surman (2013) with additional information (Funk, 2008).

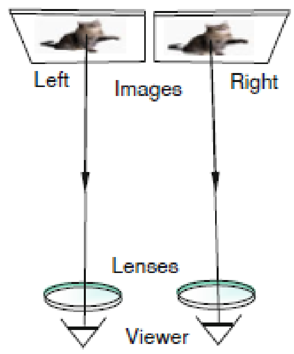
1833 - Charles Wheatstone described the characteristics of stereoscopy and has developed the mirror stereoscope. A pair of mirrors reflects two inverted images for each eye. As result, the brain will fuse the two images into one three-dimensional object, which appears in space behind the mirrors (Figure 2.5a).

1849 - David Brewster developed the Lenticular stereoscope, which was able to take stereoscopic photographs through a pair of convex lenses for the left and right eye. By watching through the lenses, the viewer's eyes were able to focus at a distance further than the actual distance of the images (Figure 2.5b).

1853 - Wilhelm Rollmann introduced the first 3D glasses, known as anaglyph 3D. The three-dimensional scene or composition was achieved by using two different filters (usually chromatically opposite e.g. red and cyan) for each eye's image.



(a) Wheatstone Stereoscope (Surman, 2013)



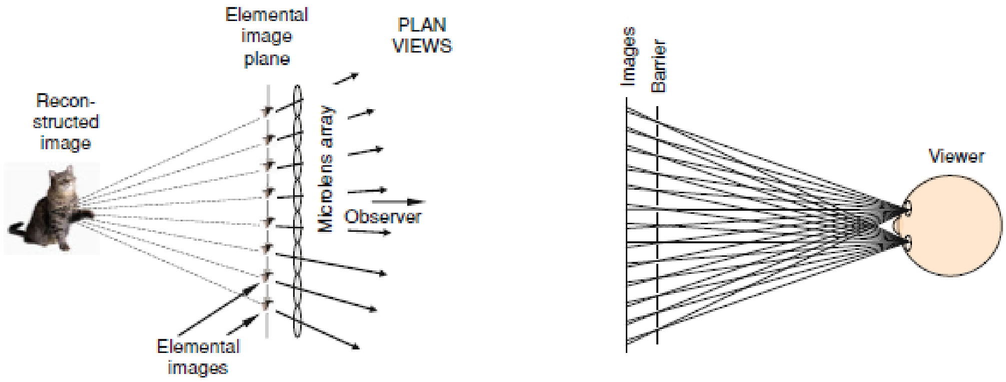
(b) Brewster Stereoscope (Surman, 2013)

Figure 2.5:Wheatstone's and Brewster's stereoscope.

1903 - Frederic Eugene Ives patented the parallax stereogram. The left and right images are directed to the appropriate left and right eyes (see Figure 2.6b). This technique is used in displays for televisions, monitors and mobile devices.

1908 - Gabriel Lippmann proposed the integral imaging technique. The early version of this technique used elemental images with the same size and reproduces those images by light beams that travel back in the opposite direction. As result, a pseudoscopic image was created, because the new three-dimensional object was shown from the opposite. This has been resolved by reversing the elemental images electronically (Figure 2.6a).

1918 - Clarence Kanolt patented the first method of capturing a series of images in order to "resolve" the problem of limited head movement. A series of vertical stripes were generated with different viewports of the object by moving the camera laterally with a barrier.



(a) Integral Imaging (Surman, 2013) (b) Parallax Barrier (Surman, 2013)

Figure 2.6: Integral imaging and parallax barrier operation method.

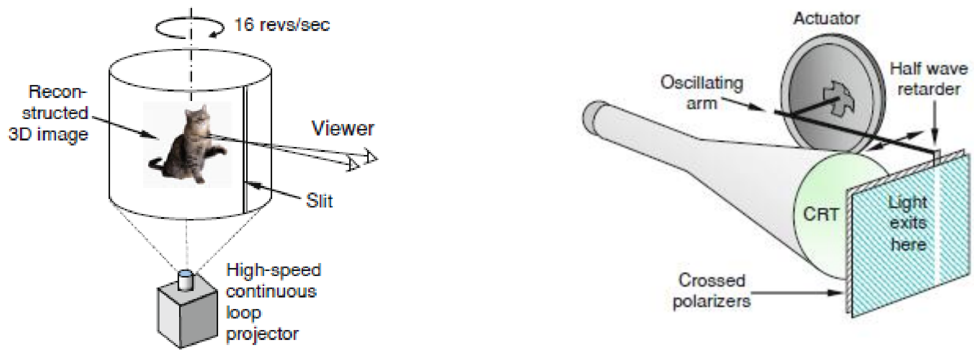
- Herbert Eugene Ives (son of Frederic Ives) developed another recording method by capturing the parallax information with a large diameter lens. The parallax barrier was located in front of the film in the camera in order to separate the direction of the rays

Before the Second World War - John Logie Baird pioneered 3D televisions by developing a standard 30-line system. The images - that have been illuminated with scanning light - were stored on a scanning disk which had two spiral apertures for the left and right eye. To reproduce those images, a neon tube illumination source was used as the output, to rapidly modulate the viewports. By observing through another double spiral scanning disk that was running in sync with the capture disk, the virtual three-dimensional object was created. Another invention of his was a "Phantoscope" that is referred to as a volumetric display. The inverse square law was used to determine the range of points on the scene surface of captured images. These images were reproduced by projecting an image on a surface that moved at the correct angles of its plane.

1936 - Edwin Land demonstrated for the first time the functionality of the linearly polarized glasses. Each eye's image is projected on a screen through orthogonal polarizing filters. By wearing linearly polarized glasses, each eye receives one of the projected images due to the filter that blocks the orthogonally polarized light. However, the head movement is limited for the viewing of the three-dimensional scene.

1949 onwards - Robert Collender invented the Stereoptiplexer with two display modes "inside-looking-out" and "outside-looking-in". The method operation for the Stereoptiplexer is similar to the Zoetrope. A virtual three-dimensional scene was created by moving the slit across the screen that contains several laterally angled viewports of an object in a natural manner. For the "outside-looking-in" displays, the virtual three-dimensional scene is generated in a free space with the same technique. Only difference, the viewer has the sensation of looking at the scene through a window (see Figure 2.7a).

1987 - Homer B. Tilton developed a variation of the Stereoptiplexer - the Parallactiscopes (parallactic oscilloscopes) - that runs without a film and produces a holoform (similar to hologram) on a cathode ray tube (CRT). To achieve a minimum mass moving aperture, a half wave is moved between crossed polarizes and the voice coil actuator moves the retarder (Figure 2.7b).



(a) Stereoptiplexer (Surman, 2013) (b) Parallactiscope (Surman, 2013)

Figure 2.7: Collender's Stereoptiplexer and Tilton's Parallactiscope

Wheatstone's and Brewster's study aids to understand the basis of human depth perception. The inventions from Lippmann, Collender and Tilton are early examples of the light field displays operation mode.

2.2.2 3D Display types

First of all, this section is an overview of actual 3D display's classifications of Surman (2013) and Mehrabi et al. (2013). Also, in this thesis the term "display" is used for all types of displays, such as television (3DTV), monitors, head-mounted-devices (HMD), mobile devices (e.g. handhelds or Smartphones), etc.

Normally in the market are two types of displays available - "3D" and "glasses-free 3D", in other words "stereoscopic" and "autostereoscopic" displays. For stereoscopic displays special 3D viewers are required. Depending on the stereoscopic display, the glasses technology is of the type active or passive. The 3D displays are divided into three classifications: "autostereoscopic", "stereoscopic" and "head-mounted-display (HMD)". Furthermore, the autostereoscopic displays are sub-divided into 3 classes, such as holographic, volumetric and multiple images, as shown in Figure 2.8.

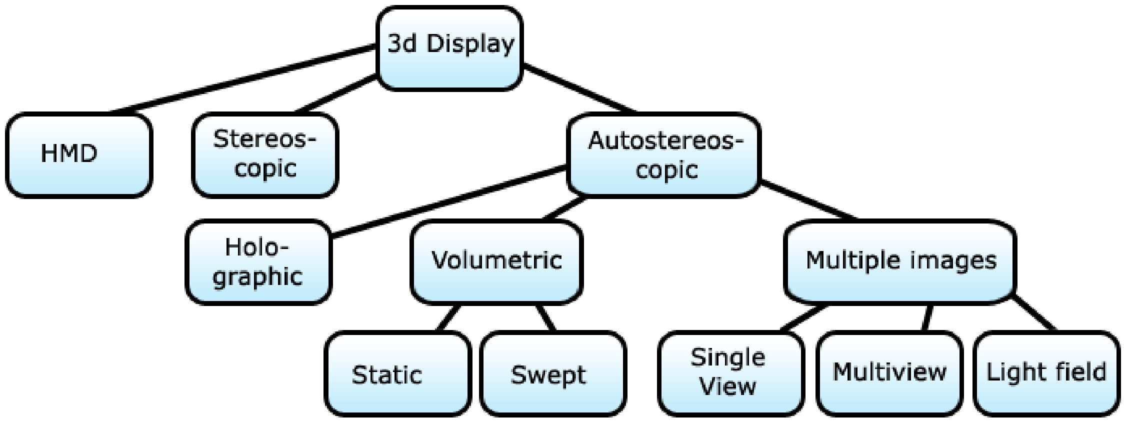


Figure 2.8: A hierarchy of 3D displays and their technologies.

As mentioned before, the technology of stereoscopic television displays depends on the used 3D viewer. A 3D display with active viewer uses the shutter system, known as liquid crystal shutter glasses (LCS). By wearing LCS, each eye receives one of the projected images by activating and deactivating the corresponded eye's image, as shown in Figure 2.9. To avoid crosstalk (one eye receives a small part of the other eye), the shutter glasses deactivated both glasses until the corresponded image is completely displayed. As result, a lack of images (flicker) becomes visible that can result in headaches after prolonged use. A small group of persons are still capable to see the images flickering effect.

The passive viewer technology uses filters (color or polarization waves) to translate the corresponded image to each eye. The color filter for each eye is used in anaglyph and chroma depth viewer. For polarized glasses, the images are projected with orthogonal polarized light on a screen and each eye's image is transmitted by blocking the orthogonal polarized light. The flickering image effect is not presented in this technology.

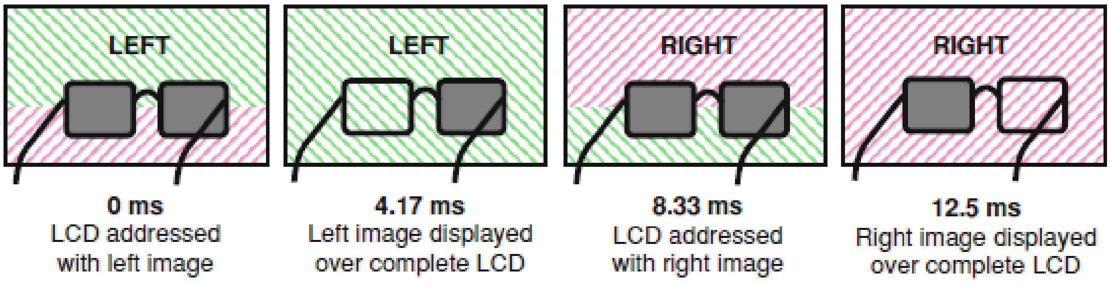


Figure 2.9:The method operation for liquid crystal shutter glasses (Surman, 2013).

Before explaining the functionality of multiple images in autostereoscopic screens, this segment begins to describe the following two displays, holographic and volumetric displays which are available for home entertainment.

A holographic display reproduces a three-dimensional scene with a series of phonograms that are generated by laser, interference, diffractions, light intensity recording and or other suitable illumination. This technique is similar to sound wave recording and reproduction, but uses light waves in place of sound waves. Apart from three- dimensional scene and object reproduction onto free space, holography is well used in ID cards and passports.

The volumetric displays draw three-dimensional objects onto a space with light points. Each light point (called voxel) corresponds to a real 3D coordinate in a volume space and is visible in a wide range. In general, the volumetric displays are subdivided into two groups of reproduction, static and swept volumes. In order to draw a three-dimensional object pattern, the addressable voxels emit light on their *on-* state, in the *off*-state the voxels are transparent. Recent research has proposed a volume space where fast infrared pulses are used to draw the voxels in nanoseconds.

In swept volumes, the voxels are drawn on several microscopic mirror displays or LCD displays. The voxels position can be manipulated by software which provides certain interaction with the displays[1]. Consequently, to avoid stretched light beams the three-dimensional volume is re-drawn with high-speed (900 rpm or 30Hz). This volume can be classified as Oscillating Planar, Varifocal and Rotating mirror.

- Oscillating Planar Mirror: Perpendicular to a CRT the microscopic mirror moves the track back- and forward.
- Varifocal Mirror: A flexible mirror that is anchored and connected to a woofer.
- Rotating Mirror: a double helix mirror o a LCD display with an RGB laser draws and rotates the three-dimensional object with a rate of 600 rpm.

Usually autostereoscopic displays are from the type multiple images (see Figure 2.8). In order to achieve the depth perception of a three-dimensional object onto the screen, two display variations are used. The most common for mobile devices is the parallax-barrier display. Two or more images of the three-dimensional scene are drawn. By separating a set of pixels on the screen with a barrier, each eye receives another directed image. The lenticular lens displays uses an array of cylindrical lenses that directs the light beams of a pixel to a defined viewing zone (see Figure 2.10).

The view option (single and multi view) defines the amount of displayed images on the screen and viewer. The single view uses two images simultaneous for each eye. In the case of multi view, five or nine images are displayed simultaneously which increase the viewing zone on one hand, but decrease the image resolution on the other hand.

Another technology is the light field that emits several light beams from each point on the screen without using holography. Integral imaging, optical modules and dynamic aperture are included in this technology. Integral imaging shows a series of pictures from different viewports.

---

[1] A demonstrative video of the Sony RayModeler, that allows user interaction, is available in YouTube: http://youtu.be/6BFKC-NKRFw (sonyelectronics, 2010, Sony RayModeler, a 360-Degree Autostereoscopic Display Prototype)

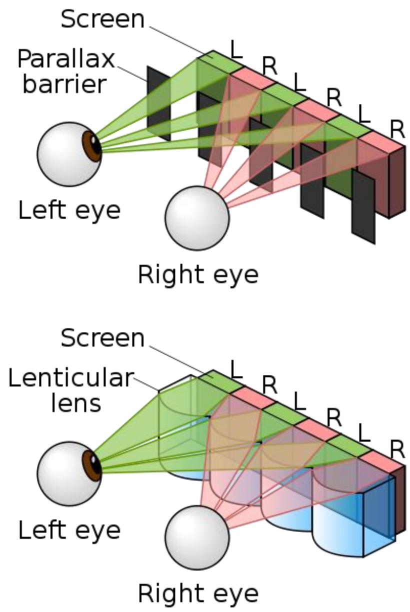


Figure 2.10: The method operation for an autostereoscopic parallax barrier and lenticular lenses display (CMG Lee, 2011).

2.2.3 Development and Research

Section 2.2.1 reviewed the background of autostereoscopy and stereoscopy until 1987. The popularity of 3D has returned since 2010, even though movie theaters showed 3D movies since 1990s.

December 2009 - Blu-ray Disc Association announced that the Blu-ray Disc is capable to store 3D contents, such as 3D movies, 3D TV series and 3D games.

2010 onwards - The first 3D capable TV was launched by Sony (Williams, 2010). Later Samsungs, Philips, Toshiba had also launched 3D TVs.

November 2010 - The virtual vocaloid Japanese idol - Mikuna Hasane - has been launched as a holographical avatar in her life concert (Fith, 2010)[2].

[2] A video from the concert is available in YouTube: http://www.youtube.com/watch?v=KxZlUq9jRbk (Ametatsu500, 2010, OneRoom(ft. Hatsune Miku)「From Y to Y」 Live HD [ENG Subs])

2011 onwards - Several autostereoscopic mobile devices were released (see section 2.3).

January 2011 - WikiPad, a glasses-free 3D tablet will be presented at CES 2012 (Keene, 2011)

August 2011 - XPAND, Panasonic, Samsung and Sony have collaborated on the development of a new technology standard for consumer 3D active glasses, under the name, "Full HD 3D Glasses Initiative." (XPAND, 2011)

October 2011 - HDMI defined with HDMI v1.4 the input/output protocols for major 3D contents (HDMI, 2011).

January 2012 - Sensics introduces Natalia™ as the first intelligent, immersive, and interactive 3D head-mounted-display (Smartgoggles, 2012).

April 2012 - Sony Computer Entertainment Inc., Tokyo (JP) patented new 3D glasses with camera based head tracking (Mao et al., 2012). YouTube enabled 3D viewing methods on some videos. (Greer, 2012)

August 2012 - The Oculus Rift virtual reality head-mounted-display was started to develop (BBC, 2012).

March 2013 - HP Research team has published a new technology for a multi-directional backlight for a wide-angle, glasses-free three-dimensional display.

August 2013 - Skype confirms 3D video calls are under development (Kelion, 2013a)

2.3 Autostereoscopic displays in mobile devices

As mentioned before, the research and development with / of autostereoscopic displays in mobile devices has increased. The following studies have been performed with autostereoscopic mobile phones:

- Ishio et al. (2011) have analyzed the readability of characters with autostereoscopic visualization on the *SHARP LYNX 3D SH-03C* Smartphone. In general, the readability of 3D characters is not depending on the age. However, a suitable size of 3D characters is required for the readability for people over 40 years of age.

- Boev et al. (2011) have performed a comparative analysis of eight 3D displays, ranging from portable autostereoscopic to large stereoscopic television sets. Two parameters have been compared, display related parameters (the 3D reproduction ability of a display) and content related parameters (suitability of stereoscopic displays, e.g. a 3D scene with particular disparity range). As a result, all portable 3D displays (except one) were capable to accommodate the disparity range of downscaled stereoscopic content. Also the comfort disparity zone of a 3D display cannot be expressed by calculations of the viewing geometry.

- Van Knotsenburg (2012) has studied the clicking accuracy in a 2D autostereoscopic environment. The study observed that the clicking accuracy difference between autostereoscopy and non-autostereoscopy is near-significant. Apart, users prefer sometimes the autostereoscopic visualization due to the true 3D perception sensation, as it was more challenging and more fun.

- Arino et al. (2013) have studied the suitability of autostereoscopic displays with augmented reality (AR) in comparison with virtual reality (VR) for children. An additional rotator peripheral device with a marker was introduced. As result, the difference between AR and VR is generally insignificant, although a group of children wanted to use the autostereoscopic screen for more games by using AR. Also the children that played the VR game version gave more importance to the negative parallax (objects is viewed out of the screen) while the other group (played the AR game) were more interested into the integration of the virtual object with the real environment.

- Bourge et al. (2013) have analyzed the characteristics of Smartphones with 3D autostereoscopic video rendering; and compared with 3D home systems (3DTV, 3D Laptops, etc.). In their opinions, Smartphones bring a new dimension of videoconferencing, gaming, augmented and virtual reality, navigation, user interfaces due to their small dimensions.

- Wei et al. (2013) have significantly improved the Peak signal-to-noise ratio of synthesized images by using a disparity-based interpolation method and two boundary refinement schemes. Also their method is effective, simple and suitable for enabled 3D Smartphones.

The characteristics and history of *Nintendo 3DS* has been studied and a brief comparison between actual available Smartphones in Europe. This section concludes the state of the arts by comparing briefly the *Nintendo 3DS* with the *LG Optimus 3D P920* Smartphone.

2.3.1 *Nintendo 3DS*

With 41.25 million sold devices, the *Nintendo 3DS* series is the most popular mobile device with a glass-free 3D display on the marked (Nintendo Co. Japan, 2013), despite ophthalmologist's warnings of potential harms to eyesight to children, if playing games in the 3D mode (Pallas et al., 2013). While Nintendo markets the 3DS with its autostereoscopic and augmented reality technology (Nintendo Co. America, 2011) a large part of its success down to the brands recognition which it got through popular franchises such as: *Super Mario Bros, Donkey Kong, Pokémon, The Legend of Zelda*, etc. (Stuart, 2013).

The *Nintendo 3DS* partially owes its existence to two experiments with 3D technology in the history of *Nintendo*. It already starts in the late 80 with the *Famicom 3D System* which was an addition to the Japanese *Familiy Computer* [3] (known as *Famicom*) gaming console. It was the first 3D video game system from Nintendo.

[3] During 1985 and 1986, *Nintendo* released *Famicom* in North America and Europe with a new design as *Nintendo Entertainment System,* abbreviated as *NES.* However, the *Famicom 3D System* was only released in Japan.

Technically, it is a liquid crystal shutter headset with a special adapter that supports two headsets simultaneously (see Figure 2.11a). However, this system was only available in Japan and proved unsuccessful, due to the side-effects, such as headaches, dizziness and nausea. According to the *Famicom*-community, the most popular games with a 3D mode were: Highway Star, Famicom Grand Prix: 3D Hot Rally, and Falsion (Famicom world, 2013c). The direct competitor for the *Famicom 3D System* was the *SegaScope 3-D* Glasses from *Sega Master Drive*.

(a) *Famicom 3D System* headset (Boffy, 2006)

(b) Falsion without goggles and with 3D goggles

Figure 2.11: ***Famicom 3D System***

Nintendo's second attempt with 3D was in 1995 with the *Virtual Boy* (abbreviated as *VB*), a table-top video game console. Released in Japan and North America, the *VB* unit contained an external controller; apart 3D display goggles that are supported by a pair of metal feet (Figure 2.12a). As a result, players were forced to hunch over in order play games. This caused physiological effects such as motion sickness and possible effects of eyestrain.

(a) the *Virtual Boy* console (Seguin, 2008)

(b) Red Alarm screenshot (Ishaan, 2009)

Figure 2.12: The *Virtual Boy* 3D system

The system for the *VB* is realized using two oscillating mirrors for each eye, which beam the light of the red LED's displays (similar to Wheatstone Stereoscope), as illustrated in Figure 2.13. For that reason, all *Virtual Boy*'s game titles were monochrome, as displayed in Figure 2.12b. By watching the games through the eyepiece, the external lights were blocked and the sensation of true "3D" was generated.

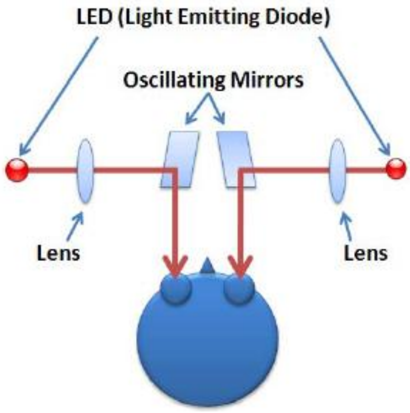


Figure 2.13: Method of operation for the *Virtual Boy* 3D goggles (Zachara et al., 2009)

In addition, the *VB* is known as *Nintendo*'s most famous market failure, due to the poor consolidated sales, especially when compared with the *Sony PlayStation* that had been sold over 7 million times worldwide. As a result, *Nintendo* never released the *VB* in Europe nor Australia; furthermore *Nintendo* discontinued production of the *VB* in 1996. Zachara et al. (2009) have studied in detail the reason for the market failure of the *Virtual Boy* in comparison with other 3D and non-3D devices. One of the most conspicuous facts was the console construction that was not portable such as the mobile devices nowadays. A further factor was the low resolution of the monochrome display.

Finally, after both 3D devices failed, *Nintendo* has released a portable handheld with an autostereoscopic display. Nevertheless, compared to current generation rival system its display resolution is still low (800 × 240) in 2D Mode which is halved in 3D Mode. The issue is more evident with the *Nintendo 3DS XL* which poses a 90% bigger screen resulting in a pixel density of 95dpi (2D Mode). In Table 2.1 the technical specification of the *Nintendo 3DS* and its successor version - the *Nintendo 3DS XL* are shown (Nintendo Co. Japan, 2011; Nintendo Co. America, 2013; Nintendo3Dbrew, 2013).

	Nintendo 3DS	Nintendo 3DS XL
Release date in Europe:	March 25, 2011	July 28, 2012
Dimensions when closed :	Width: 13.4 cm (5.28 in) Height: 7.4 cm (2.91 in) Depth: 2.1 cm (0.87 in)	Width: 15.6 cm (6.14 in) Height: 9.3 cm (3.66 in) Depth: 2.2 cm (0.87 in)
Weight (including battery pack, stylus and SD card):	235g	336g
Upper display (autostereoscopic LCD display):	3.53'' ((7.68cm × 4.61cm)) Resolution: 800 × 240 Per eye: 400 × 240 Colors: 16.77 million colors	4.88'' (10.62cm × 6.37cm) Resolution: 800 × 240 Per eye: 400 × 240 Colors: 16.77 million colors
Lower display (touch LCD display):	3.02'' (6.14cm × 4.61cm) Resolution: 320 × 240 Colors: 16.77 million colors	4.18'' (8.50cm × 6.37cm) Resolution: 320 × 240 Colors: 16.77 million colors
Cameras:	One inner camera and two outer cameras Resolution 640 × 480 (0.3 Mega pixels) Lens are single focus and uses the CMOS capture element Usage: for photos in three dimensions, for photos, for Augmented Reality games, for QR Codes	
Power:	1300 mAh lithium-ion battery 3DS games: 3 to 5 hours DS games: 5 to 8 hours	1750 mAh lithium-ion battery 3DS games: 3.5 to 6.5 hours DS games: 6 to 10 hours
Connectivity:	2.4GHz Wi-Fi (Security: WPA and WPA2) Recommended communication distance: 30cm	
Storage capacity	Included 2 GB SD card 2 GB internal flash memory (1.5 usable)	Included 4 GB SD card 2 GB internal flash memory (1.5 usable)
CPU:	Dual-core ARM11 MPCore 268MHz	
Graphics	DMP PICA2000 - 268MHz	
Memory:	2x64MB FCRAM, 6 MB VRAM	
Price:	\$169.99 ≅ 127,5234€	\$199.99 ≅ 150,0289€

Table 2.1: Detailed hardware specification of both Nintendo 3DS handhelds.

However, Jaffe et al. (2013) have demonstrated that the *Nintendo 3DS* games series do not demonstrate any significant perception advantages of the usage of the autostereoscopic visualization mode. They carried out tests with 11 subjects in the game *Mario Kart 7*. They tracked the lap times in the following stages: *Airship Fortress, Bowser's Castle* and *Rainbow Road*. They concluded that in autostereoscopic visualization mode the subjects too longer to complete the stage than during non-autostereoscopic visualization (Figure 2.14). These differences are insignificant, causes of many factors in this type of games, such as the stage, the obstacles on the road, items, etc.

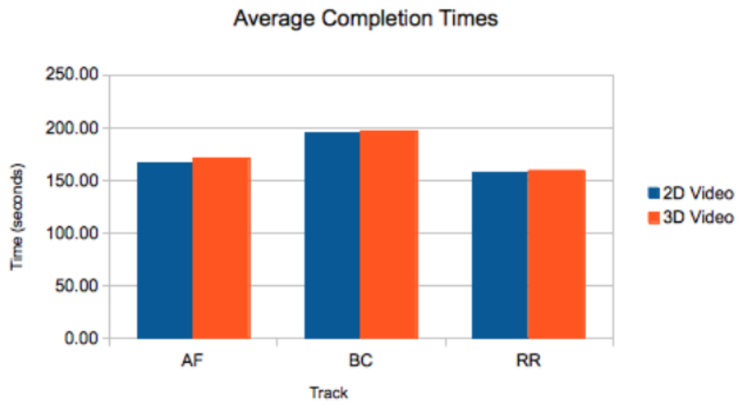


Figure 2.14: Result of the average completion times in both visualization modes.

Another example is the game *Super Mario 3D Land*, in some cases players is confronted with an optical illusion in form of a puzzle. In these stages, the game gives a hint on the right corner in order to activate the autostereoscopic visualization by moving the 3D depth slier. These optical illusions can be found in world 1-1, world 2- 1, world 4-4.

At the first moment, the player sees a normal block level. But by looking closer, the shadow of the block with the coin appears on the floor and indicates that this block is on another position as seen. In other words, it is a "destroyed" optical illusion, as illustrated in Figure 2.15.

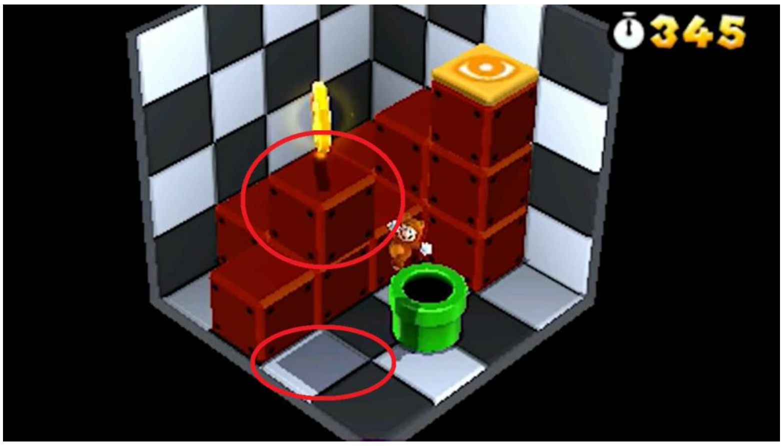


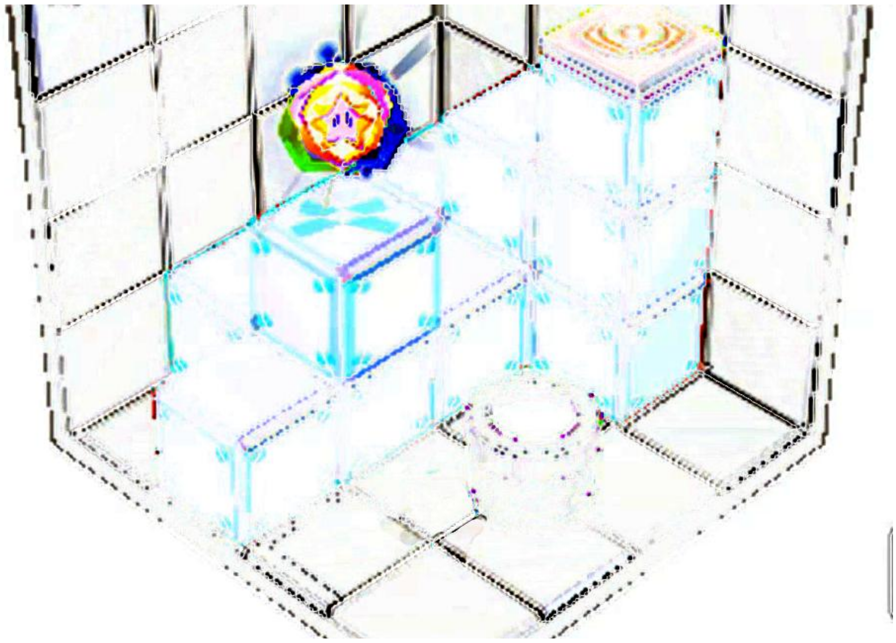
Figure 2.15: A "destroyed" optical illusion, cause of the object shadow.

By activating or deactivating the autostereoscopic visualization, the block with the coin changes its position. In Figure 2.16c, a difference image based on two frames has been created in order to illustrate the little translation. Looking closer at the lines, it is clear, that the block has moved by approximately two to three pixels forward.

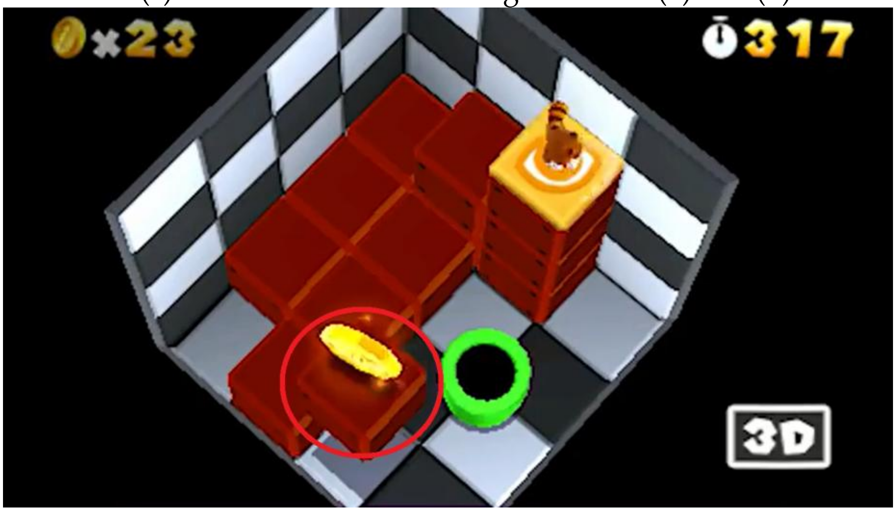
This small transformation is not visible for a small group of players, due the small display size and screen resolution. The viewport of the scene can be changed by activating the eye icon platform. As result, the player is capable to seeing the real position of the cube with the coin.

(a) Stage with non-autostereoscopic visualization

(b) Stage with autostereoscopic visualization



(c) A render differences image between (a) and (b).



(d) Real position of the block

Figure 2.16: Changing cube position by visualization mode and viewport.

The only purpose for the autostereoscopic visualization mode consists in entertainment for limited period of time, as some players experience health issues such as: headache and dizziness. Also, no *Nintendo 3DS* game title requires this mode in order to solve a puzzle or to see better obstacles.

2.3.2 Mobile phones and Smartphones

The revolution of mobile devices with 3D technology started years before with the *Sharp mova SH251iS* that was only released in Japan. The *Sharp mova SH251iS* was capable of rendering and generating 3D images on an eight inch display. A 3D demo version of the popular PC game *Quake II* was released with the mobile phone, which gave a true perception of 3D to the player (Norris, 2002).

The direct competitors of the *LG Optimus 3D P920* - the *Sharp Aquos SH80F* and the *HTC Evo 3D* Smartphone - were released in the period in Europe. Apart from the named mobile phones, the most 3D-enabled mobile phones were never released in Europe, which included the *Sharp mova SH251iS*, *Samsung SCH-B710*, *Hitachi Wooo Ketai H001*, etc.

The *LG Optimus 3D P920* has the smallest display's resolution, the biggest internal memory storage and lowest price. In comparison, the *HTC Evo 3D* is the heaviest mobile phone with the best CPU and secondary camera characteristics. The last competitor - the *Sharp Aquos Sh80F* - is the lightest and has the highest resolution and the best primary camera characteristics, also shares the best video recording quality with the *LG Optimus 3D P920*. All these Smartphones share same fate as neither them remains in production.

In Table 2.2 the *Sharp Aquos SH80F,* the *HTC Evo 3D* and *LG Optimus 3D P920* Smartphone are compared with their hardware specifications (GSMArena, 2011a; GSMArena, 2011b; GSMArena, 2011c).

	Sharp Aquos SH80F	**HTC Evo 3D**	**LG Optimus 3D P920**
Release date in Europe	October 2011	July 2011	July 2011
Dimensions	127 x 64 x 11.9 mm (5.0 x 2.52 x 0.47 in)	126 x 65 x 12.1 mm (4.96 x 2.56 x 0.48 in)	128.8 x 68 x 11.9 mm (5.07 x 2.68 x 0.47 in)
Weight	135 g	170 g	168 g
Display	3D LCD capacitive touch screen, 16M colors 540 x 960 pixels, 4.2 inches (~262 ppi pixel density) - TapFlow UI	3D LCD capacitive touch screen, 16M colors 540 x 960 pixels, 4.3 inches (~256 ppi pixel density) - HTC Sense UI	3D LCD capacitive touch screen, 16M colors 480 x 800 pixels, 4.3 inches (~217 ppi pixel density) - LG 3D UI

	Multitouch -	Multitouch Corning Gorilla Glass protection	- Corning Gorilla Glass protection
Primary camera:	Dual 8 MP, 3264x2448 pixels, autofocus, LED flash Stereoscopic photos & videos; geo-tagging Video 1080p (2D), 720p (3D)	5 MP, 2560x1920 pixels, autofocus, dual-LED flash, Stereoscopic photos (2 MP only) & videos; geo-tagging Video 720p@30fps (2D), 720p@30fps (3D)	Dual 5 MP, 2592x1944 pixels, autofocus, LED flash, Stereoscopic photos & videos; geo-tagging Video 1080p@30fps (2D), 720p@30fps (3D); video stabilization
Secondary camera:	0.3 MP (VGA)	1.3 MP	?
Power:	Li-Ion 1240 mAh battery Stand-by: Up to 450 h (2G) / Up to 410 h (3G) Talk time: Up to 5 h 20 min (2G) / Up to 5 h 20 min (3G)	Li-Ion 1730 mAh battery Stand-by: Up to 358 h (2G) / Up to 420 h (3G) Talk time: Up to 9 h 20 min (2G) / Up to 7 h 45 min (3G)	Li-Ion 1500 mAh battery Stand-by: Up to 450 h Talk time: Up to 13 h (2G) / Up to 9 h (3G)
Connectivity:	2G: GSM 900 / 1800 / 1900 3G: HSDPA 900 / 2100 WLAN: Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot Bluetooth: v3.0 with A2DP, EDR	2G: GSM 850 / 900 / 1800 / 1900 3G: HSDPA 900 / 2100 or HSDPA 850 / 2100 HSDPA 900 / 1700 / 2100 WLAN: Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot Bluetooth: v3.0 with A2DP, EDR	2G: GSM 850 / 900 / 1800 / 1900 3G: HSDPA 900 / 1900 / 2100 HSDPA 850 / 1700 / 2100 WLAN: Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot Bluetooth: v2.1 with A2DP, EDR
Storage capacity:	Internal: 2 GB storage, 512 MB RAM External: microSD, up to 32 GB	Internal: 1 GB storage, 1 GB RAM External: microSD, up to 32 GB	Internal: 8 GB storage, 512 MB RAM External: microSD, up to 32 GB
CPU:	1 GHz Scorpion	Dual-core 1.2 GHz	Dual-core 1 GHz Cortex-A9
GPU:	Adreno 205	Adreno 220	PowerVR SGX540
Sensors:	Accelerometer, proximity, compass	Accelerometer, gyro, proximity, compass	Accelerometer, gyro, proximity, compass
OS:	Android OS, v2.3 (Gingerbread)	Android OS, v2.3 (Gingerbread), upgradable to v4.0 (Ice Cream Sandwich)	Android OS, v2.2 (Froyo), v2.3, upgradable to v4.0 (Ice Cream Sandwich)
Features	- SNS integration - HDMI port	- Active noise cancellation with dedicated mic	- SNS integration - HDMI port

	- Google Search, Maps, Gmail - YouTube, Google Talk, Picasa - MP3/AAC+/WAV/WMA player - MP4/Xvid/H.263/H.264/WMV player - Organizer - 3D photo viewer - Document viewer/editor - Voice memo/dial - Predictive text input	- TV-out (via MHL A/V link) - Google Search, Maps, Gmail - YouTube, Google Talk, Picasa - MP3/AAC+/WAV/WMA player - MP4/Xvid/H.263/H.264/WMV player - Facebook, Flickr, Twitter applications - Organizer - Document viewer/editor - Voice memo/dial/commands - Predictive text input	- Google Search, Maps, Gmail - YouTube, Google Talk - MP4/DivX/XviD/H.264/H.263/WMV player - 1080p@30fps (2D), 720@30fps (3D) playback - 3D/2D video editor - MP3/WAV/WMA/eAAC+ player - Document viewer/editor - Organizer - Voice memo/dial/commands - Predictive text input
Price (from amazon.de from other sellers)	365,00 €	269,99 €	244,00 €
3D SDK[4]	Sharp Stereo Display SDK	HTC OpenSense SDK	LG Real3D SDK

Table 2.2: Technical specification of the Sharp Aquos SH80F, the HTC Evo 3D and LG Optimus 3D P920 Smartphone.

In 2012, the successor - the *LG Optimus 3D Max* - was released by *LG Electronics*. Kluczniok (2012) reported that the *LG Optimus 3D Max* is using a new 3D converter in order to provide a better 3D perception. Its weight was reduced to 148 grams and the Android OS-icons were represented in 3D. After an additional update, the *LG Optimus 3D Max* should offer a HD-converter.

[4] Sharp Stereo Display SDK: http://www.sharp.co.jp/mebius/3dmodule/sdk.html
HTC OpenSense SDK: http://www.htcdev.com/devcenter/opensense-sdk/stereoscopic-3d
LG Real3D SDK: http://developer.lge.com/resource/mobile/RetrieveDocDevLibrary.dev

2.3.3 Brief comparison between *Nintedo3DS* and *LG Optimus 3D P920*

It was not possible to realize a detailed comparison between the autostereoscopic visualization of the *Nintendo 3DS* and *LG Optimus 3D P920* due to not having the *Nintendo 3DS SDK*. Therefore, this comparison was realized by small tests, such as moving the screen of device in different angles and personal perception of the autostereoscopic scene. For the comparison, the developed *Android* app - *3D Block Puzzle*, the *Puzzle Swap*[5] 3D scenes and the optical illusion of *Super Mario 3D Land* world 2-1 from the *Nintendo3D* have been used.

- The *LG Optimus 3D P920* supports a higher angle range than the *Nintendo 3DS*. Which means, the *Nintendo 3DS*-user is forced to look directly to the screen, by moving the console a little bit, the autostereoscopic visualization effect is less pronounced.
- The *Nintendo 3DS* requires complex objects for good autostereoscopic visualization results, such as shown in Figure 2.17a. Also important, the object has to have sharp edges pointing out in Z-direction (see Figure 2.17b).
- The translation of the objects in optical illusion stages from *Super Mario 3D Land* by activating or deactivating the autostereoscopy, during our experiments a small group of persons could not see any differences in the scene, although they had no problem to use stereoscopic or autostereoscopic devices.
- Both mobile devices can provoke known side-effects such as headaches and dizziness, after an excessive usage.

[5] Puzzle Swap is a preinstalled mini collection game on Nintendo 3DS. Players collect puzzle pieces by passing with other 3DS-players or by using the Play Coins in the StreetPass Mii Plaza. However, there is no guaranty of a new Puzzle Swap piece. The Play Coins are obtained by walking. One Play Coin corresponds to 100 steps and the maximal amount of Play Coins a day is 10. After completing a Puzzle Swap, the image is available as a 3D scene that supports autostereoscopic visualization.

(a) A complex polygon object.

(b) An object with sharp edges.

Figure 2.17: ***Nintendo 3DS's Puzzle Swap*** **scenes.**

III

Development

DEVELOPMENT

3.

3.1 Introduction

This chapter presents the development of an interactive Android app, which runs on a Smartphone with an autostereoscopic parallax barrier display - and is divided in two sections. Section one corresponds to the detailed information about the autostereoscopic projection system, hardware and software. The other section explains in detail the characteristics of the developed app, which supports two visualization modes - autostereoscopy and non-autostereoscopy.

3.2 Autostereoscopic projection system

The idea of this app is to provide a series of activities which need to be solved by using touch gestures. These activities can be fulfilled using autostereoscopic and non-autostereoscopic visualization. For a comprehensive comparison between, each task is timed to determine accurately the subject's performance in either mode.

Each element in the app is represented as a virtual object for both visualization modes, except buttons and images, which are non-autostereoscopic elements. For the touch detection in the app, the Android SDK touch gesture event manager method (*onTouchEvent(MotionEvent event)*) was used. As a result, the user was capable to translate a movable primitive to a new position by moving a finger on the touch-screen or rotate the scene by activating the "camera mode".

3.2.1 Hardware

A *LG Optimus 3D P920* with Android 4.0.4[6] was used for the comparison study between autostereoscopic and non-autostereoscopic user perception. *Android* provides a certain guaranty to the user, which means, no access permission to important operating system data. In order to obtain complete access to the data partition on *Android*, "root" access is required. This is equivalent a Linux Super User. However, for security purposes Super User access has been disabled in *Android*. Re-Enabling root access requires hacks which are not supported by the manufacturers voiding the warranty and are not suitable for end users. In this case, the easiest way to store the test results of each task was to use the extra external storage facility (a 4GB microSD card).

3.2.2 Software

The Integrated development environment *Eclipse for Mobile Developers* version *Juno Service Release 1* (Build id: 20120920-0800) with the *Android Development Tool* (*ADT*) Plug-in *version 21.0.1.v201212060256-543035*, that includes a set of development tools such as debugger, libraries, an emulator based on QEMU (*Android Virtual Device Manager)*, documentation, sample code, and tutorials were used to develop the app.

[6] The Android Ice Cream Sandwich actualization was not available till 9th April 2013. Before, the Smartphone possess Android 2.3.5. The app was developed starting by Android 2.3.5 until 4.2 with the Real3DSDK version 10, before updating to Android 4.0.4.

For the 3D models rendering, the *Open Graphics Libraries for Embedded Systems version 2.0* (*OpenGL ES 2.0*) application programming interface (*API*) was used (http://developer.android.com/guide/topics/graphics/opengl.html). *OpenGL* is a multi-platform *API* for rendering 2D and 3D computer graphics developed by *Khronos Group*. Original author is *Silicon Graphics Inc. (SGI)*, which developed *OpenGL* from 1991 and released in January 1992. *OpenGL ES* is a subset of *OpenGL* also developed by *Khronos Group*; and is designed for embedded systems such as mobile phones, PDAs, and video game consoles. *OpenGL ES 2.0* is based roughly on *OpenGL 2.0* and using similar transition from *OpenGL 3.0* to *3.1*. Due to this modification *OpenGL ES 2.0* is not backward compatible with older versions, but is supported by *Android* platforms, *Apple* devices (*iPad, iPad Mini, iPhone 3GS* and *iPod Touch* 3rd generation or later), several *Nokia* phones, BlackBerry devices (with *BlackBerry OS 7.0*, *Blackberry 10* and *BlackBerry PlayBook*), etc.

In the case of *Android*, the *OpenGL ES 2.0* framework *API* is nearly equivalent to the *J2ME JSR239 OpenGL ES API*, but not identical. *Android* supports *OpenGL* through the *Native Development Kit* and framework *API* (*Android* 2.2 or higher versions). Due to the reduction, this *API* is only compatible with primitives, which are drawn with points, lines or triangles. Therefore, it is not compatible with quads or n-sided polygons such as used in *OpenGL*. However, it allows developer to write shaders by providing a fully programmable pipeline.

OpenGL ES 3.0 is available starting with *Android* 4.3, is backwards-compatible with the *OpenGL ES 2.0 API* and provides additional texture compression support, in order to increase the performance by reducing memory requirements and making more efficient use of memory bandwidth.

The autostereoscopic visualization was performed by using the *LG Real3D SDK* (http://developer.lge.com/resource/mobile/RetrieveDocDevLibrary.dev) for *Android 4.0* (version 14)[1], which can be installed easily with the *Android SDK Manager* in *Eclipse*. Also, provides technical *Android* application demos to demonstrate the functionality and power of the *LG Real3D APIs*, runs on the device and on the *Android Virtual Emulator* (with the use of with red-cyan anaglyph glasses), provides to developers certain controls over the display's hardware block interleaving in order to display 3D applications on the screen. The *LG Real3D SDK* allows developer to program with 2D and 3D objects. In the case of a 2D autostereoscopic environment, the *LG Real3D SDK* provides specific variables such as *R3DSpaceView, depthImage, R3DImgTestView*, etc.

In the other case, the *LG Real3D SDK* uses *OpenGL ES* with a modification on the glSurfaceView with the *Real3D setReal3DInfo* method, such as shown in Listing 3.1. In the app, the value for "type" is "*REAL3D_TYPE_SS*", which means the usage of "Side-by-Side stereoscopic surface"; and for "order" it is "*REAL3D_ORDER_LR*", which implies, that the left image will be drawn before the right one.

```
    glSurfaceView = new GLSurfaceView(this);
    mReal3D = new Real3D(glSurfaceView.getHolder());
//...  mReal3D.setReal3DInfo(new RealInfo(Boolean disableAutoDetection, int type, int order))
    mReal3D.setReal3DInfo(new Real3DInfo(true, Real3D.REAL3D_TYPE_SS,
Real3D.REAL3D_ORDER_LR));
```

Listing 3.1: Example of setReal3DInfo method usage.

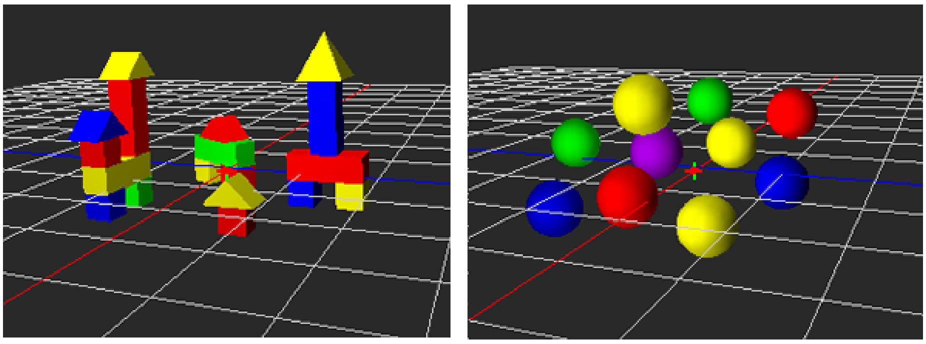
In general, the app was launched and debugged directly on the device by setting up the *Android Debug Bridge (adb)* in *Eclipse*; instead of using the given *Virtual Android Emulator.* The Emulator is able to run the app, however its execution is slower as on the device; also it is not able to represent the autostereoscopic vision without red-cyan anaglyph glasses.

3.3 Android app

To study the user perception of the autostereoscopic display, the app was forced to combine autostereoscopic / non- autostereoscopic visualization with User Interactions in a simple game. Also, it had to demonstrate two different types of game playing with autostereoscopic visualization; which has been achieved by developing two individual full-3D games.

3.3.1 App design

The app is divided into two sub-games. The first is similar to a memory brain training game with stereopsis characteristics, named *3D Ball game*. Mission of the game is to count all balls in the scene, counting all elements with a specific color or knowing which object is closer to the subject. Figure 3.1b shows a 3D concept art for this sub-game. The second sub-game represents a mental and stereopsis game, called *3D Block Puzzle*. During the game, users have to build one or various constructions such as towers made of simple primitives (cubes, quads, pyramids and triangular prisms) with different colors (see Figure 3.1a).



(a) *3D Block Puzzle* concept (b) *3D Ball game* concept

Figure 3.1: 3D idea concept created in Swift3D for the both games.

3.3.2 App development

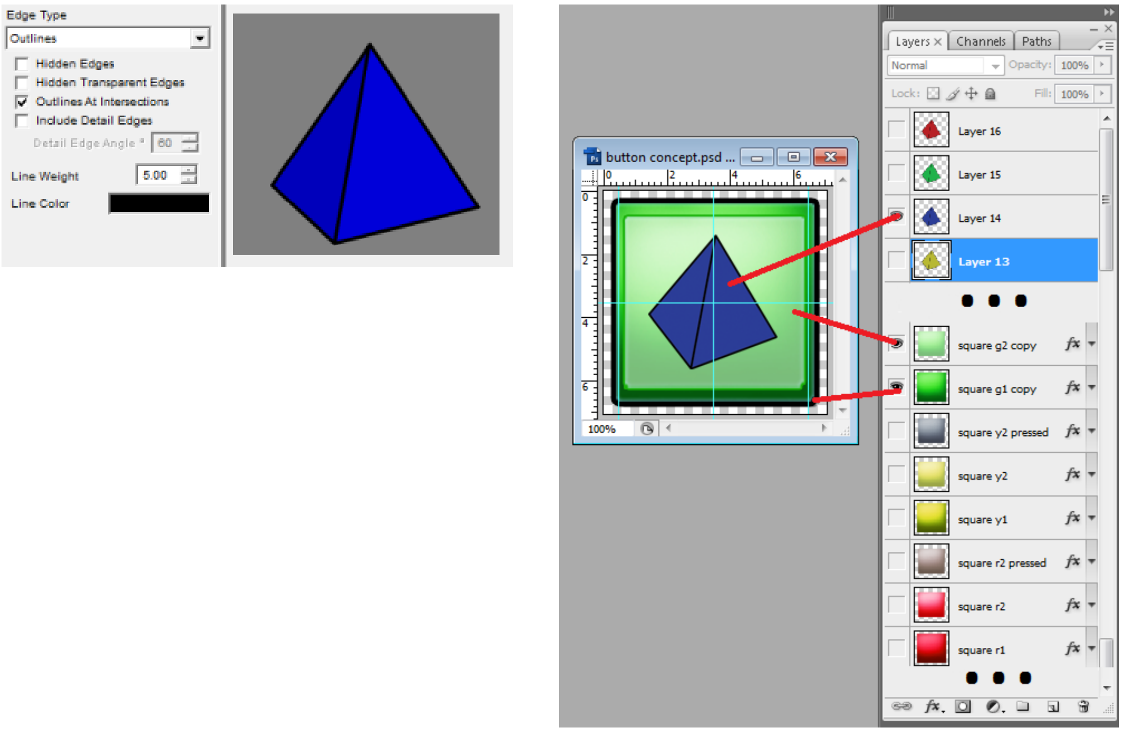
First of all, all in-game elements such as buttons icons, images, logo types and background images were designed with a trial version of *Adobe Photoshop CS3*, which provides a series of image filters.

In this app, three different buttons were designed, such as menu, stage and icon image button. The buttons and View Layouts were designed similar to the *Android* game “Where’s my Water?” from *Disney* (Google play, 2013), in order to provide a user-friendly app. In Figure 3.2, a screenshot of the second stage of “Where’s my Water?” is shown, which was used as reference for the design.

(a) Buttons and icon reference. (b) Clicked button

Figure 3.2: Design reference for buttons, icons and the layout view.

For the 3D object icon buttons that are used in *3D Block Puzzle* (see Chapter 3.3.3 App characteristics), a combination of *Swift3D* together with *Adobe Photoshop* was used, as shown in Figure 3.3. This combination was also used to create the background and logotype image.



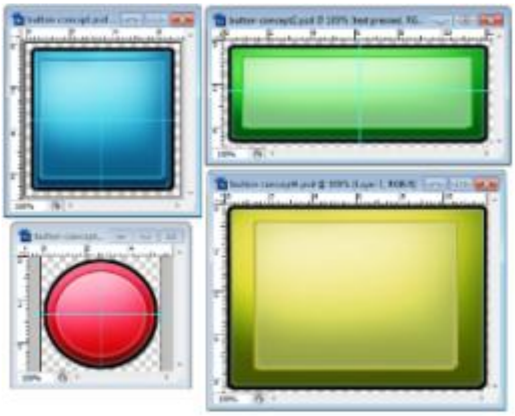
(a) Create image with outlines in *Swift3D*.

(b) Import Image from *Swift3D* and combine it with proper layout elements plus filters in *Adobe Photoshop*.

Figure 3.3: Creating process for the 3D object icon images

The menu, stage and icon buttons were designed in *Adobe Photoshop* using proper layout elements (square, circle and rectangle) and three fronts:

- *Comic Sans* for the menu buttons (Figure 3.4b).
- *AR Christy* for the stage numbers and for the 3D icon button (Figure 3.4c).
- *Windings 1-3* for the icons: return, correct and wrong (Figure 3.4d).



(a) A set of proper layout elements.

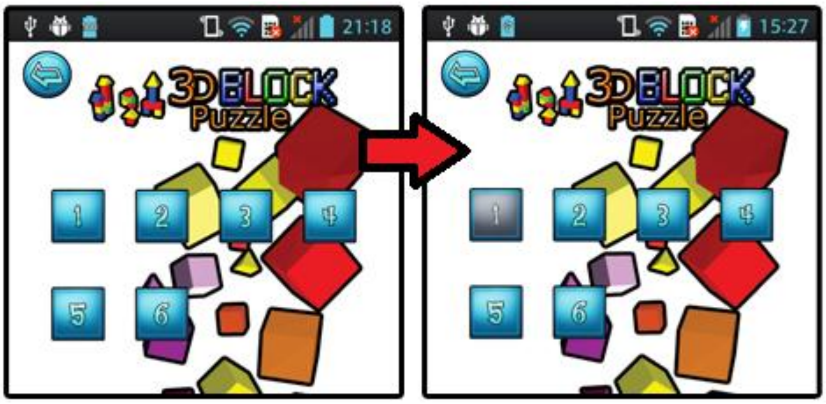
(b) A set of menu buttons.

(c) A set of in-game buttons (stage or mode selection).

(d) A set of icon buttons.

Figure 3.4: Proper button designs.

In order to highlight a pressed button, a second image was developed for each button and managed through a XML script (DiMarzio, 2011). Figure 3.5 and Listing 3.2 illustrate the XML script code for the button with the blue number image. Apart from the button design two icons were created, one to show how many attempts are left (see Chapter 3.3.3 App characteristics) and a help image for *3D Block Puzzle*, in order to show the final construction.



(a) Button is not clicked (b) Button is pressed

Figure 3.5: Button image change according the state.

```xml
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item android:state_pressed="true"
                android:drawable="@drawable/pglevel101btndown" />
<item android:drawable="@drawable/pglevel101btn" />
</selector>
```

Listing 3.2: XML script for buttons.

All primitives (cube, pyramid and triangular prism) were created in OpenGL ES without using 3D graphic-tools e.g. Blender. In Listing 3.3, part of the source-code is presented in order to draw the triangular prism.

```java
float vertices[] = {
        //frontface
        -1.0f, -1.0f, 1.0f, //Vertex 0
        1.0f, -1.0f, 1.0f,  //v1
        0.0f, 1.0f, 1.0f,  //v2

        //rightface
        1.0f, -1.0f, 1.0f, //v3...
        1.0f, -1.0f, -1.0f, //v4
        0.0f, 1.0f, 1.0f,  //v5
        0.0f, 1.0f, -1.0f,  //v6

        //backface
        1.0f, -1.0f, -1.0f,  //v7
        -1.0f, -1.0f, -1.0f,  //v8
```

```
        0.0f, 1.0f, -1.0f,  //v9

        left
        -1.0f, -1.0f, -1.0f, //v10
        -1.0f, -1.0f, 1.0f,   //v11
        0.0f, 1.0f, -1.0f,  //v12
        0.0f, 1.0f, 1.0f,  //v13

        //bottom
        -1.0f, -1.0f, -1.0f, //v14
        1.0f, -1.0f, -1.0f,  //v15
        -1.0f, -1.0f, 1.0f,  //v16
        1.0f, -1.0f, 1.0f,  //v17
};

float textures[] = {
        0.0f, 0.0f,
        0.5f, 0.0f,
        0.25f, 1.0f,

        0.5f, 0.0f,
        0.5f, 1.0f,
        1.0f, 0.0f,
        1.0f, 1.0f,

        0.0f, 0.0f,
        0.5f, 0.0f,
        0.25f, 1.0f,

        0.5f, 0.0f,
        0.5f, 1.0f,
        1.0f, 0.0f,
        1.0f, 1.0f,

        0.5f, 0.0f,
        0.5f, 1.0f,
        1.0f, 0.0f,
        1.0f, 1.0f,
};

byte indices[] = {
        //Faces definition
        0,1,2, //front
        3,4,6, 3,6,5, //right
        7,8,9, //back
        10,11,13, 10,13,12, //left
        14,15,17, 14,17,16, //bottom
};
```

Listing 3.3: Triangular Prism drawing source code.

In Figure 3.6a is the corresponding texture for the triangular prisms shown. Half of this texture was also used for the pyramid. For each texture it was necessary to create a border, therewith the participant can better distinguish whether the moveable object is in the correct position or not. The help cube - which is not movable - has a star on its texture (see Figure 3.6b).

(a) Triangular texture

(b) Texture for the help cube.

Figure 3.6: Texture samples from *3D Block Puzzle*.

Before starting to explain how the app was developed, the following steps have to been covered: general structure of an *Android* app, an app with autostereoscopy and finally the structure of an interactive *Android* app with both visualization modes (autostereoscopic and non- autostereoscopic). Generally an *Android* app contains three elements, the *Activity Manager*, the *Activities* and the *View Layout*, as illustrated in Figure 3.7:

- The *Activity Manager* can contain several activities and can switch between them during runtime in order to provide a user-friendly application. Also it is managing the libraries and the permissions, which are required from the app e.g. *WRITE_EXTERNAL_STORAGE*.
- The *Activity* can be compared with a full screen window on a desktop operating system that uses *View Layouts*. *Activities* can share *View Layouts*. Therefore, it is not required to create an equal number of *Activities* and *View Layouts*. Apart from the appearance, Activities can provide certain controls through touch-gesture events e.g. *setOnClickListener* and key-down-events (pressing the menu button or another button).
- The *View Layout* is a basic unit of the *Android Natural User Interface* and represents a widget on the *Activity Screen*, which can appear as visible or invisible. Basic visible Views are for example: *TextView, EditText, Button, ImageButton, CheckBox, RadioButton, ImageView* and *glSurfaceView*. Invisible Views help to enforce certain rules for the layout design in form of: *GridLayout, LinearLayout (Vertical), LinearLayout (Horizontal), RelativeLayout, FrameLayout*, etc. A *View Layout* can be created with the *layout-XML* in *Eclipse* or in runtime with *Java*. During runtime a *View Layout* can be changed through *Java* source code e.g. changing an image after a special event. This image modification can be done with the *setImageDrawable*-method.

Figure 3.7: General *Android Activity* architecture.

An app that uses the Real3D SDK from LG with *OpenGL* has following components: the *Activity Manager*, the *Activity*, the *Renderer*, the *OpenGL Scenes* (by default two *OpenGL Scenes*), the *Real3D* converter and finally the *View Layout*, as shown in Figure 3.8.

- For this type of apps the *Activity Manager* has to provide the libraries and features for OpenGL and Real3D.
- The *Activity* is managing all required elements for the interactive full screen window. In this case, calling the View Layout, the Renderer and the Real3D-objects in combination with an event manager.
- The *Renderer* provides all requirements for the 3D Scene visualization, which means initialization (*onSurfaceCreated*), manipulation (*onSurfaceChanged*) and drawing (*onDrawFrame*) of the Scene.
- The *OpenGL Scene* can be a single object or a set of objects in the *Renderer Scene*.
- The *Real3D* generates the autostereoscopic visualization with two *OpenGL Scene's images*.
- The *View Layout* can be with the *layout-XML* or in *Java-runtime*, as mentioned before. Mostly the *View Layout* is written with *Java-runtime* code in order to avoid creating further *Layout*-classes.

Figure 3.8: Android app architecture that runs with Real3D and OpenGL.

For the app to work with the *Real3D SDK*, certain modifications had to be done to the *Android Mainifest.xml* before executing the source-code. Import for the modification is the <uses-feature> and <uses library> xml tag. Listing 3.4 shows the *Mainifest.xml* file of the developed app.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.puzzlegame3d"
   android:versionCode="1"
   android:versionName="1.0" >
   <uses-sdk
      android:minSdkVersion="8"
      android:targetSdkVersion="17" />
   <uses-feature
      android:name="lge.hardware.real3d.barrier.landscape"
      android:required="false" />
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" >
   </uses-permission>
   <application
      android:allowBackup="true"
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@android:style/Theme.NoTitleBar" >
      <uses-library
         android:name="com.lge.real3d"
         android:required="true" />
      <activity
         android:name="com.example.puzzlegame3d.PG_UPVsplashscreen"
         android:screenOrientation="landscape" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
         <!-- ... activities... -->
      </activity>
   </application>
</manifest>
```

Listing 3.4: Mainifest.XML with modification for the Real3D SDK.

As mentioned before, the Android app was required two visualization modes, hence the *Activity* structure is a combination of a general app; an app with *Real3D* plus *OpenGL* and an app with only *OpenGL* (see Figure 3.9).

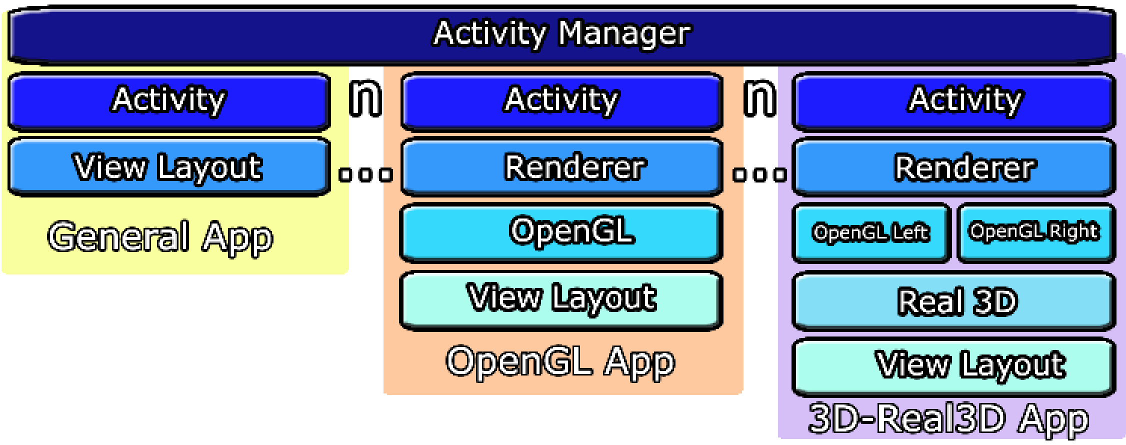


Figure 3.9: Structure of the developed *Android* app.

Before an *Activity* starts, the *Android* app executes the following steps:

(1) The *Activity Manager* consults if the *Activity* is of the type autostereoscopic or non-autostereoscopic, in the case that a *Renderer* is implemented.

(2) The *Renderer* also checks which visualization mode is active.

(3) According to the mode, each component executes another part from the visualization-source-code. Listing 3.5 shows in pseudo-code what will be done in the case of autostereoscopy.

```
//Activity
if(autostereoscopy){
    initialize Real3D
    use the technique: Side-by-Side stereoscopic surface
    draw Left Image first
}

//Renderer
if(autostereoscopy){
    draw left image
    draw right image
} else { draw image }
```

Listing 3.5: Pseudo-code in the case of autostereoscopy.

In the final app architecture is divided into two blocks main menu and games architecture. The main menu architecture represents a simple menu with four elements: start, credits, settings and exit. The games architecture is simplified into two elements, the games and their stages. Figure 3.9 shows a simplification of the final architecture in which the exit element is not presented and the start element is represented as the arrow to Games. In addition, the games element represents both games, according to the selection; the corresponding stage will be load. All levels contain two sub-elements, which are the *3D Scene* (*Primitives* and *Renderer*) plus the *Game Logic* (*Solution* and *Game Engine*).

- *Primitives* represent all available 3D objects in this app e.g. spheres, cubes, quads, pyramids and triangular prisms.
- *Renderer* represents the combination of the *Renderer* with *OpenGL Scene.*
- *Solution* / *Solution Engine,* is the control engine, which has been used. For the *3D Ball Game*, the *Solution Engine,* checked only if the given answers were correct. In *3D Block Puzzle*, the *Solution Engine,* checked if the object is collocated correctly.
- *Game Engine* realizes relevant modifications / operations in the app such as, changing between autostereoscopy and non-autostereoscopy, moving an object or changing the camera and saving the result into a TXT-File.

Another import element - that is not given in Figure 3.10 - is the *Engine*, which is used in this app in all *Activities*, in order to be able to change between languages and register a user for the test with participants.

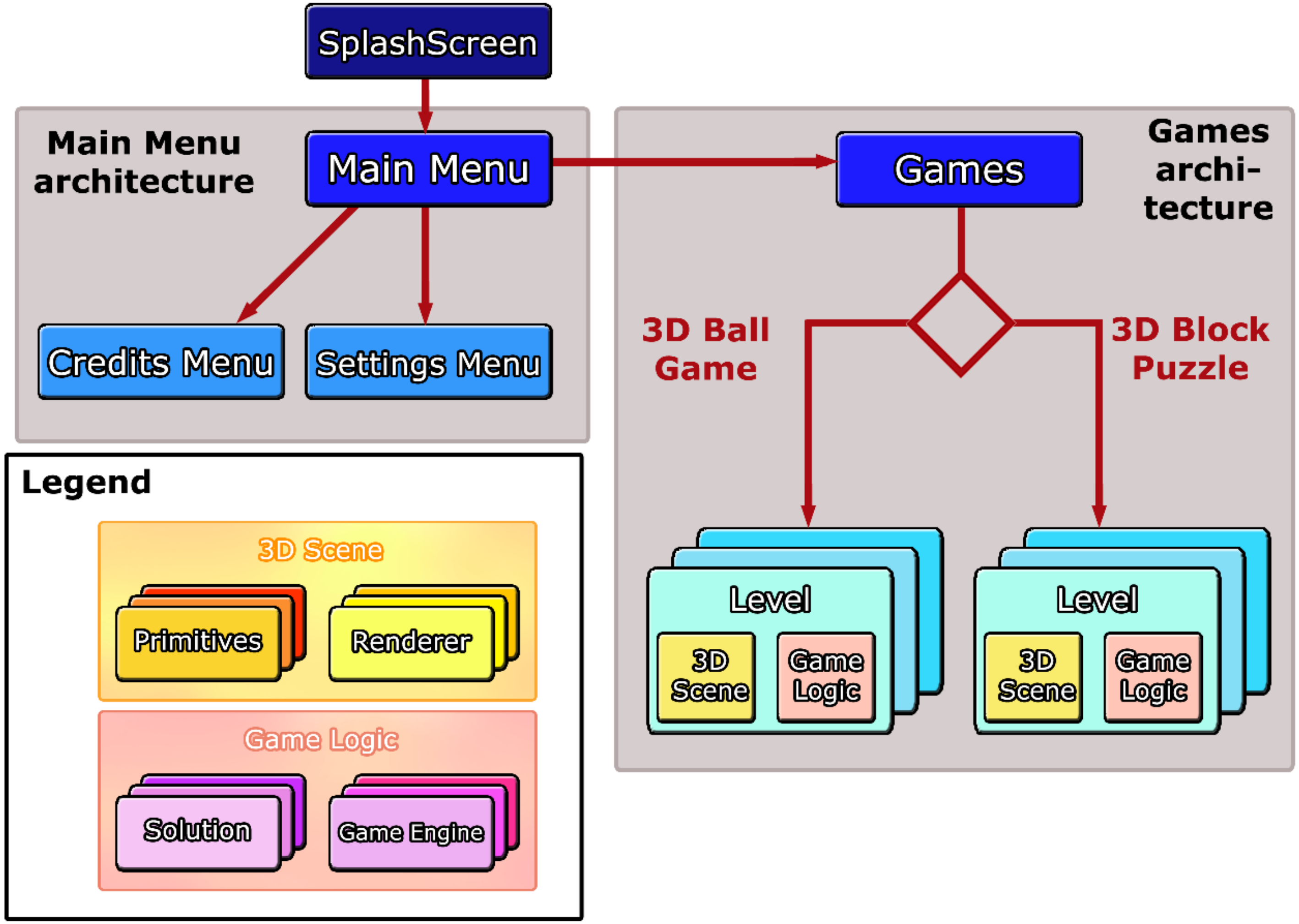


Figure 3.10: Developed *Android* app architecture.

For *3D Block Puzzle*, an additional epsilon value was used to provide a certain object movement range in *Solution Engine*, which means that the object has to be between the solution position ± epsilon. Each stage uses the same epsilon values.

- Epsilon X (*epx*): 0,13
- Epsilon Y (*epy*): 0,1
- Epsilon Z (*epz*): 02

The solution for each stage is following (Table 3.1):

Level	Star-Cube1	Star-Cube1	Object1	Object2	Object3
			Red		-
2	(0, -1, -2.5)	-	(0, 0, -2.5)	-	-
			Blue	Red	
3	(-1.5, 0, 0)	-	(-1.5, -1, 0)	(-1.5, 1, 0)	-
			Blue	Green	Red
4	(0, -1, 1)	(0, 0, 0)	(0, 0, 1)	(0, 1, 0.5)	(0, -1, 0)
			Blue	Yellow	Green
5	(0, -1, -1)	(2, -1, -3)	(1, -0.5, -1)	(1, 1, -2)	(1, -0.5, -3)

Table 3.1: ***3D Block Puzzle*** **coordinates object solution for all stages.**

In addition, each stage uses a point system in order to know if the user has passed the stage or not. If the object is in the solution range (X±epx, Y±epy, Z±epz), the system gives one point for each correct coordinate. To pass the stage, all points have been collected. In the other case, the stage is incomplete and the participant has to repeat that stage. The following Table 3.2 shows the point system for all levels:

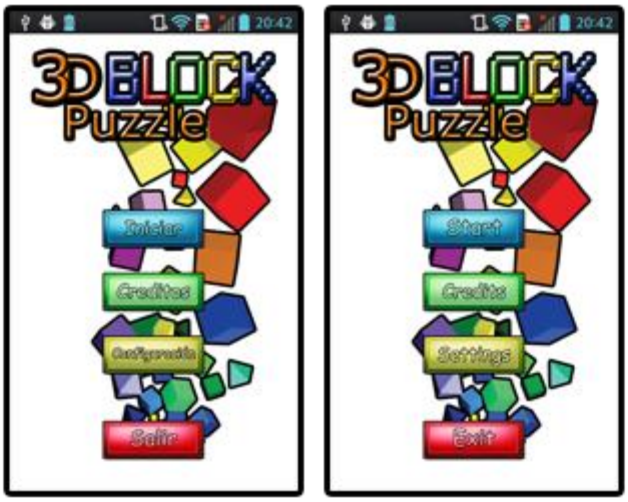
Level	Object1	Object2	Object3	Total
2	3	-	-	3
3	3	3	-	6
4	3	3	3	9
5	3	3	3	9

Table 3.2: ***3D Block Puzzle*** **points system for all stages.**

This app is multilingual; for the case that one participant does not understand Spanish. The Android platform supports Multilanguage files by using different "`values`"-folder in "`res/`". Every "`values`"-folder has to contain a "`string.xml`" file with all strings used in the TextView in the activity (see Figure 3.11). The native language for each Smartphone is the default selected language in Android mobile operating system. If an app supports different languages, the first language will be the default Android Smartphone operating language. It is not recommended to change Android Smartphone operating language during app runtime, although it is possible (Android Developers, 2013).

Figure 3.11: Android developer application file management.

Apart from the "`values-es`"-folder, it was necessary to use two different images and a XML scripts for the ImageButton. Due to the small amount of available languages, it was no problem to generate all required files. However, it is not recommended to use *ImageButton* with text in the image, in case the app has to be multilingual. The result can be seen in Figure 3.12.



(a) Spanish App home screen (b) English App home screen

Figure 3.12: Language change in the home app screen.

In order to draw the correct image, the Activity had to check which language was active and take the corresponding image for the *ImageButton*, as shown in Listing 3.6.

```
//if English is selected
if(engine.getGAME_LANGUAGES_SELECTED() ==
PG_Engine.GAME_LANGUAGES_AVAILIBLE[0]){
        gamelisttitle.setImageDrawable(getResources().getDrawable(R.drawable.pglang1title));
        game2title.setImageDrawable(getResources().getDrawable(R.drawable.pglang1game1title));
        game1title.setImageDrawable(getResources().getDrawable(R.drawable.pglang1game2title));
}
else {
        gamelisttitle.setImageDrawable(getResources().getDrawable(R.drawable.pglang2title));
        game2title.setImageDrawable(getResources().getDrawable(R.drawable.pglang2game1title));
        game1title.setImageDrawable(getResources().getDrawable(R.drawable.pglang2game2title));
}
```

Listing 3.6: Source-code to draw correct image for selected languages.

3.3.3 App characteristics

First of all, the aim of the games is to analyze differences in perception between both visualization modes, which may support further autostereoscopy application studies. In this study, the two games share the same view layout designs for both modes in order to accurately analyze the behavior, the reaction and perception of a subject during game-play. The only difference between both layout designs consists in the manipulation of the autostereoscopic eye distance factor, allowing each participant to adjust the eye distance in autostereoscopic mode by using the seek bar in the upper right corner for each task (see Figure 3.13). During non-autostereoscopic mode this seek bar has no effect on the rendered image.

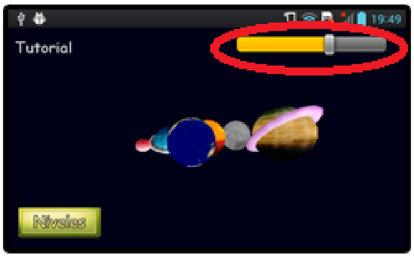


Figure 3.13: Position of the seek bar in the screen.

Each game contains several stages. The first stage in the two games is a demo level in order to make the participant familiar with the touch gesture control. The *3D Ball game* does not require any advanced manipulation of the scene. In the *3D Block Puzzle*, however it is important to learn, how to combine the "normal mode", "camera mode" and "z-mode" in order to be able to build the shown construction correctly. The amount of movable primitives increases gradually with each level, starting with one until three objects.

To progress to the next stage the subject has to successfully pass the current stage; or the user has to redo the stage. The stage clear screen is for both games and for all stages similar, as shown in Figure 3.14. There are certain differences due the amount of information which has to be displayed such as: passed time to clear the stage, left retries, passed / failed icon, object coordinates, etc.

(a) Stage clear screen of *3D Block Puzzle* (b) Stage clear screen of *3D Ball game*

Figure 3.14: Stage clear screen

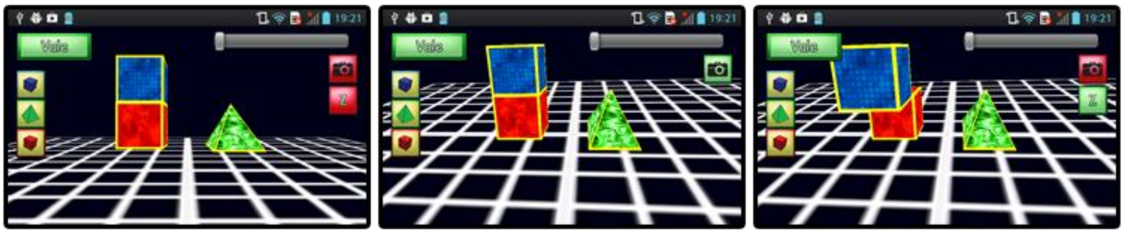
For both games a maximum of three retries for each stage is given, which are represented by stars in the stage-clear-screen. After losing all stars, the user is forced to start the next stage. The retry cycle of the first level from *3D Block Puzzle* is presented at Figure 3.15.

(a) Failing the first time (b) Failing the second time (c) Failing the third time

Figure 3.15: Participant retry-cycle, except passed state.

The user interaction is given by the Android platform supported touch gestures, including: tap, double tap, tap and hold, swipe, drag, pinch open and finally pinch close (Lehtimaki, 2013). Both games make use of the following touch gestures: tap and drag. Only the game selection activity supports the swipe gesture, which has been realized by combining HorizontalScrollView, TableLayout and ImageButtons. The Tap touch gesture is used to realize a click action on buttons.

In some cases a button triggers a state change of an implemented mode (visualization, camera and z-mode) or change of object id. Therefore, it is necessary to inform the user about which mode is active, also which object is selected. In *3D Block Puzzle*, the user can switch between "camera mode" and "z-mode"; however, there is a restriction that both modes cannot be selected at the same time. In order to select one mode, the other one has to be deactivated. An example of activating and deactivating the "camera mode" and "z-mode", also changing the object is given in Figure 3.16.



(a) None mode is active (b) Camera is active (c) Z is active

(d) Selection object information

Figure 3.16: Button triggers "Toad" action

The drag gesture is used to rotate the scene or to translate an object through the scene in two directions, by default X and Y. For the Z translation of an object, it was required to provide another mode, called "z-mode". In "z-mode" an object can be translated in X and Z direction. For all stages in *3D Block Puzzle*, a translation restriction stops objects from moving out the screen. However, the rotation operation has no restriction in order to keep the ability of seeing every face of the objects.

Apart from the user interactions, the app grants an administration area with a login scenario, to change between the visualization modes. Another important aspect of the administration area is also the language setting (change between English and Spanish) and user id registration, as shown in Figure 3.17. After completing a stage, the app saves all results in a txt file with the user id that has been modified before in the administration area. Listing 3.7 shows the txt file content with the results from *3D Block Puzzle* Level 05 of user A01.

(a) Login scenario (b) User registration

Figure 3.17: Administration area

```
User: A01   3D Mode: true   Time: 168039 ms   Life: 3
   Starcube coordinates:  0.0 -1.0 -1.0
   Starcube2 coordinates:  2.0 -1.0 -3.0
   Movable Quad coordinates: 0.8196352 -0.4721779 -0.19911389
   Movable Triangle coordinates: 0.0 0.0 0.0
   Movable Quad coordinates: 1.8062363 -0.28634065 -0.84574777
   Points: 2   Valuation: failed

User: A01   3D Mode: true   Time: 129791 ms   Life: 2
   Starcube coordinates:  0.0 -1.0 -1.0
   Starcube2 coordinates:  2.0 -1.0 -3.0
```

```
Movable Quad coordinates: 2.0092483 -0.4664889 0.0
Movable Triangle coordinates: 2.0149405 1.035377 -0.93866694
Movable Quad coordinates: 2.0718622 -0.4513176 -1.9076788
Points: 9   Valuation: passed
```

Listing 3.7: ***3D Block Puzzle*** **Level 05-file structure.**

The complete app is divided in four sections. Section one represents the app adjustment flowchart. Section two shows the flowchart of *3D Ball Game* with four stages (one demo stage and three game stages). The third section shows a flowchart of *3D Block Puzzle* with six stages (one demo stage, four game stages and an extra stage for measuring the participant's autostereoscopic depth visualization).

Before starting the experiment with the participants, the splash screen pop ups and after two seconds appears the main menu. Once everything is adjusted –including user registration, visualization mode and language selection (see Figure 3.18), the participant has to select and run the first game (*3D Ball Game)*, as presented in Figure 3.19. After finishing *3D Ball Game*, the participant has to run *3D Block Puzzle*, as shown in Figure 3.20. Now the administrator has to re-adjust the application in another visualization mode as before for a second run with the same user in the same order. Figure 3.21 represents the completed flowchart of the app for the case that the participant is running the app for the first time with non-autostereoscopic visualization. After finishing the first run, the "visualization mode" for autostereoscopy will be activated by the administrator.

At any moment it is possible to access any section of the game by using the menu button of the Smartphone. There is no restriction for using the menu options, however, only the administrators know about this option in order to avoid any unexpected behavior of the participant during the experiment.

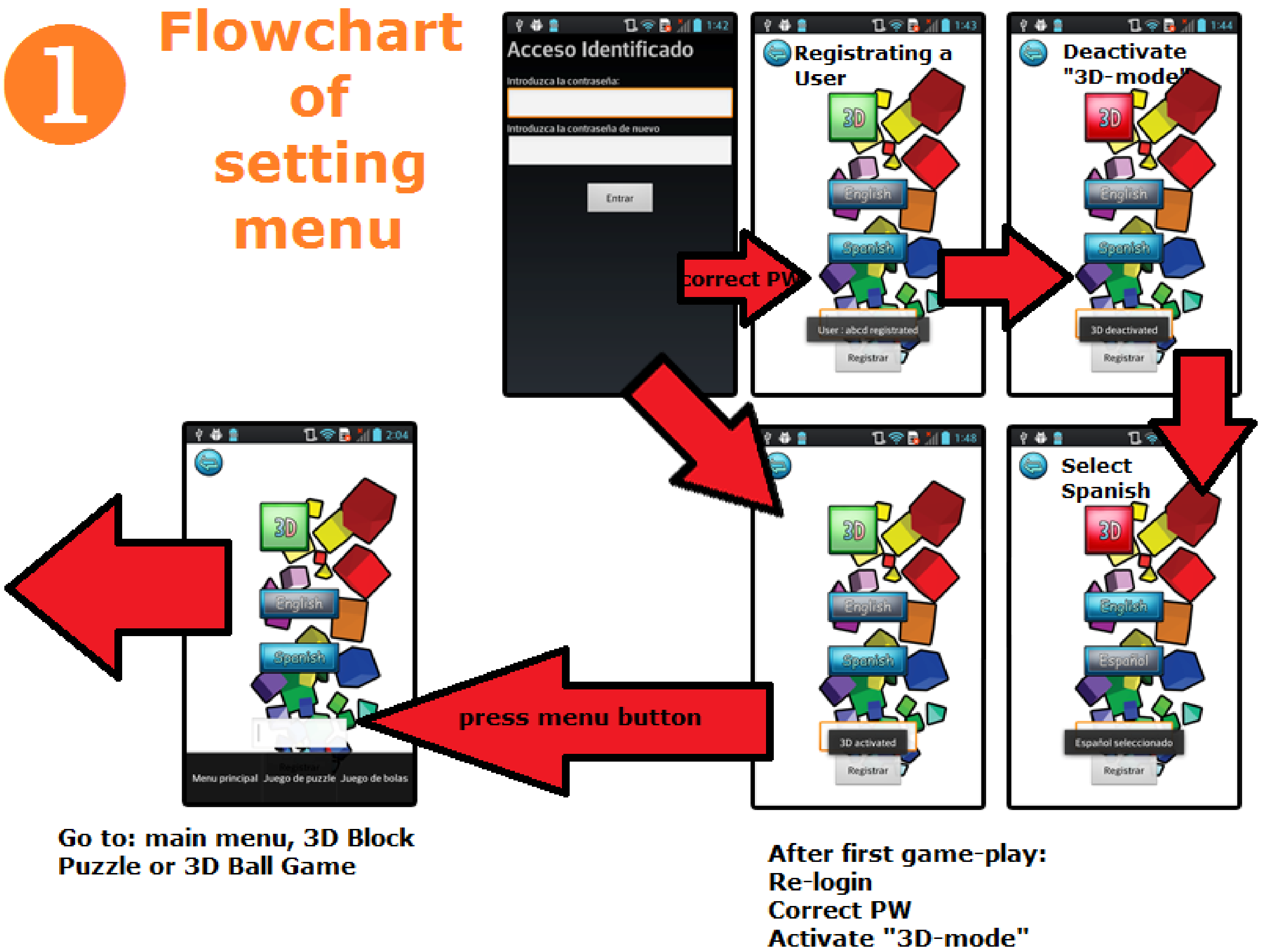


Figure 3.18: A possible game adjustment flowchart scenario.

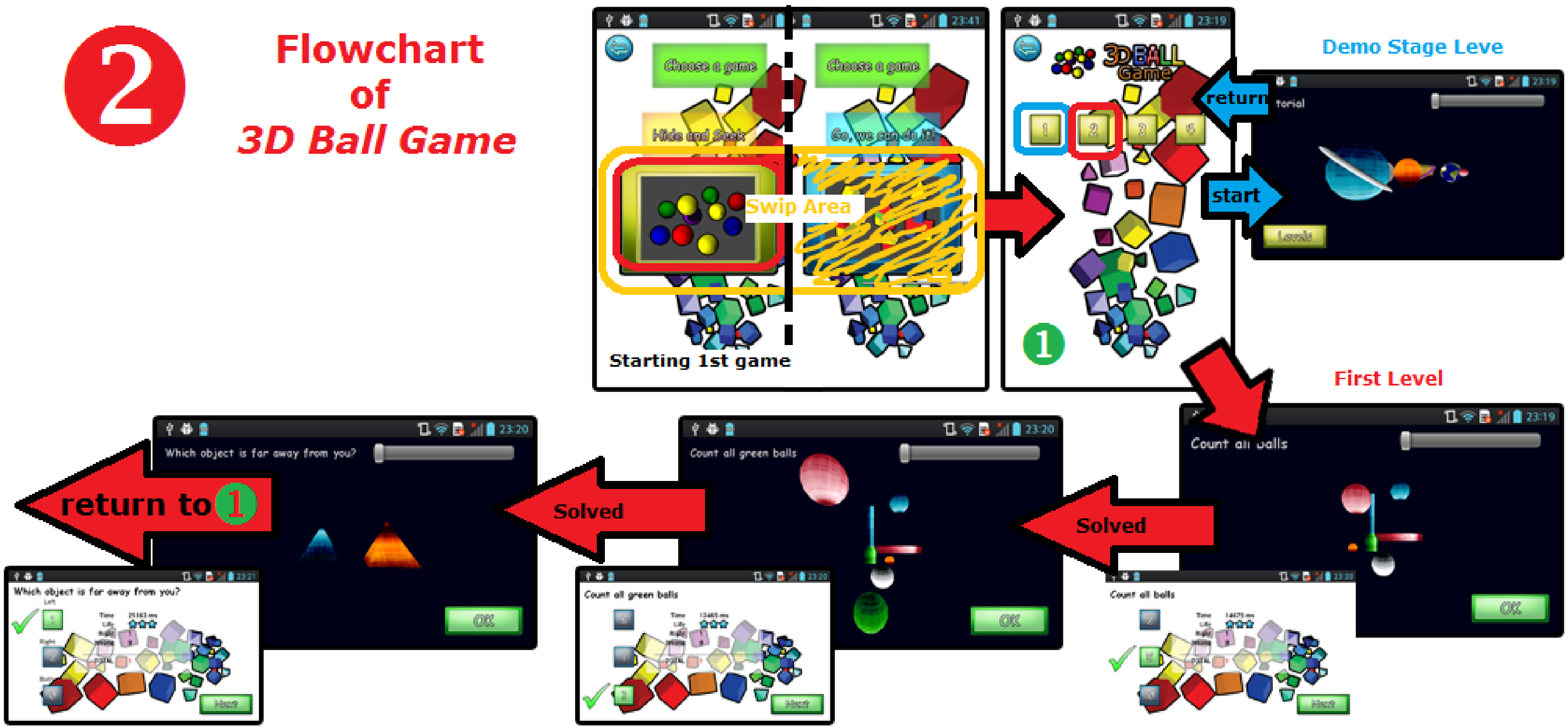


Figure 3.19: Flowchart of running first game.

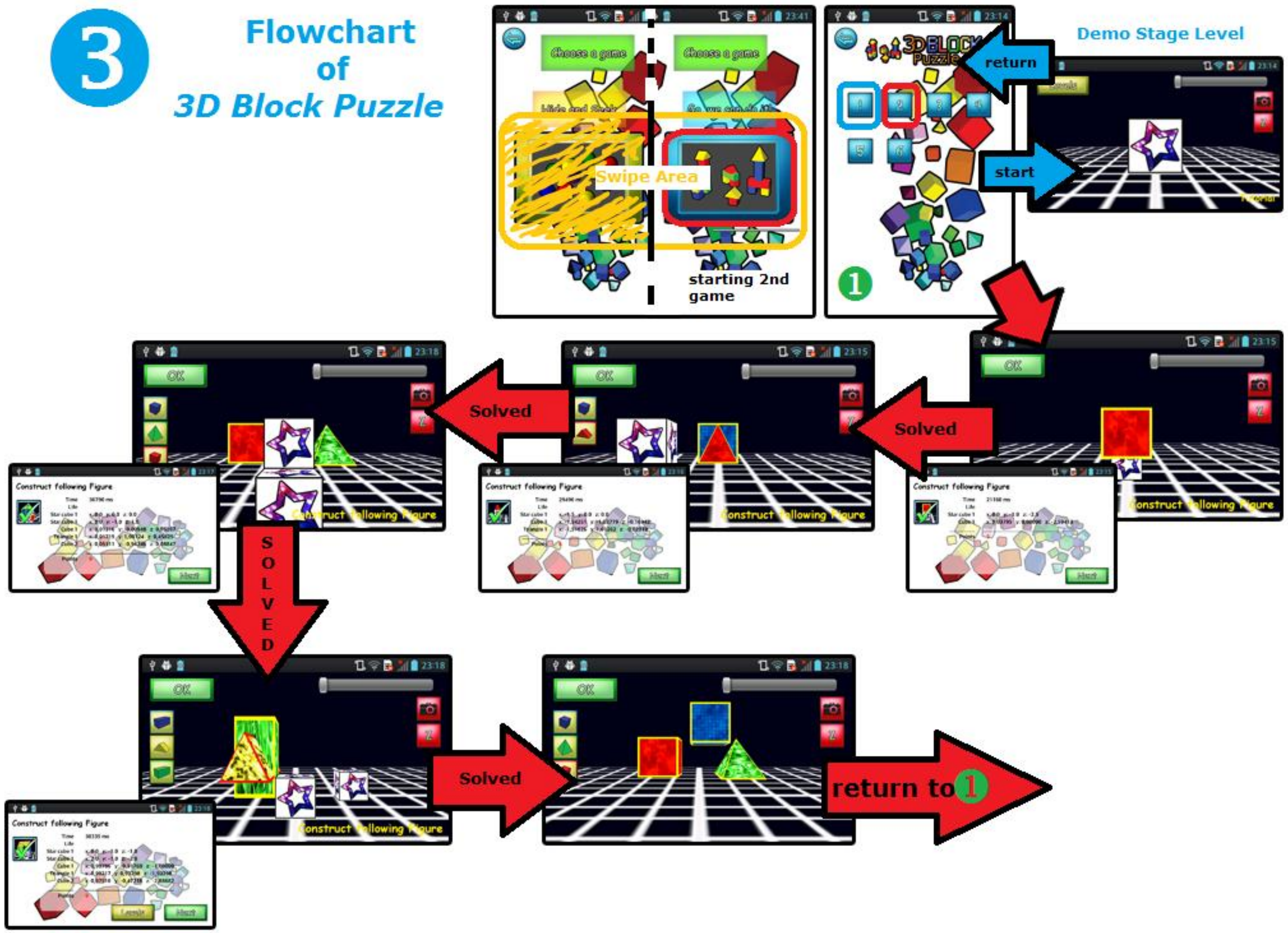


Figure 3.20: Flowchart of running second game.

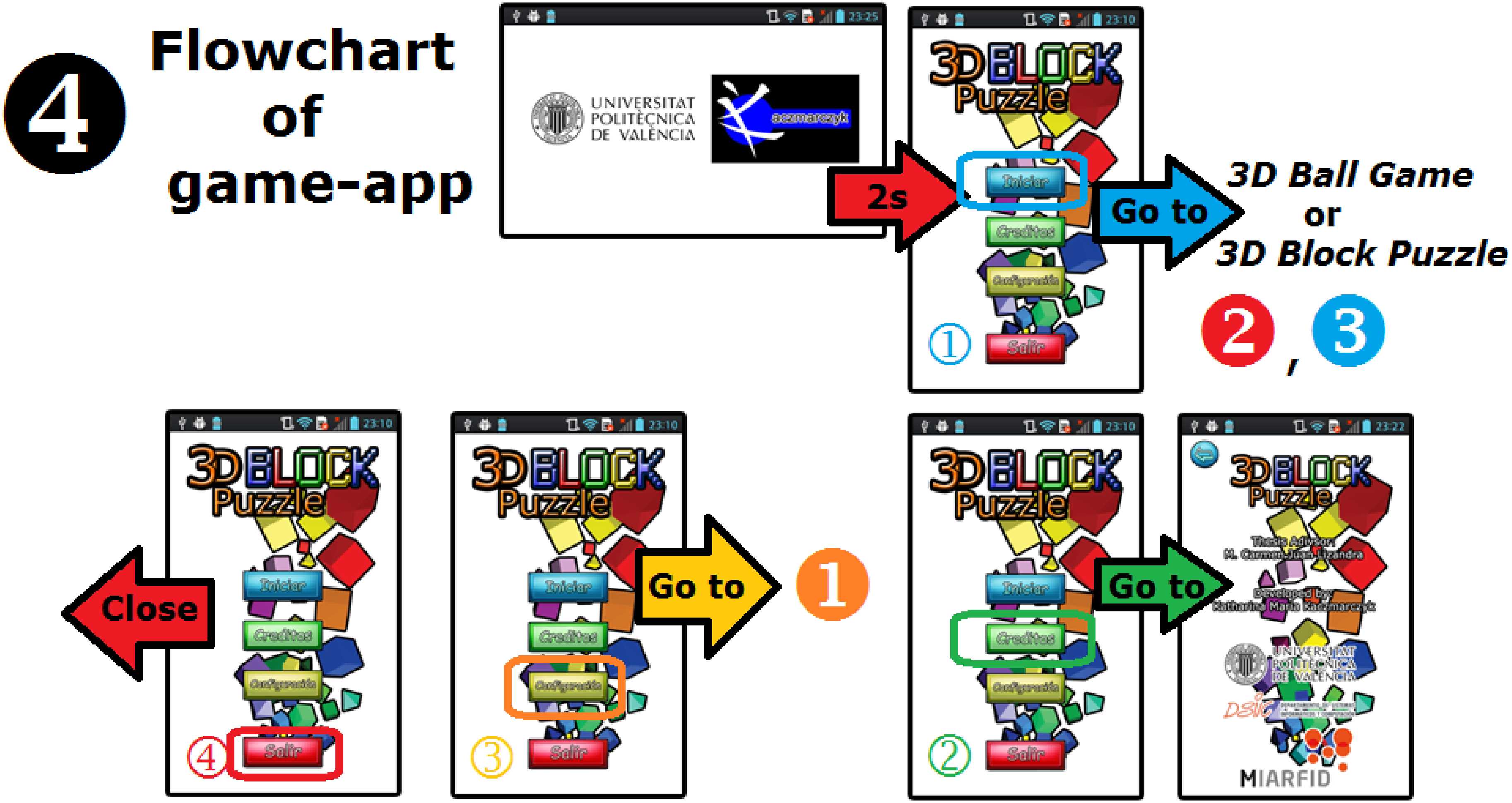


Figure 3.21: Complete Flowchart

IV

Study

STUDY

4.

4.1 Introduction

First of all, the aim of this study is to determine if there is a significant difference between the autostereoscopic and non-autostereoscopic visualization. In addition, to analyze the depth perception level of the autostereoscopic screen of the *LG Optimus 3D P920*. Hence, the developed Android app (see section 3.3) was played and evaluated by two groups of adults: firstly played the app with autostereoscopy and then without; the other group played the app without autostereoscopy and later with autostereoscopy.

This chapter explains in detail who participated, which measurements have been done during the evaluation, the procedure of this study and the analysis of obtained results (in game values and questionnaires).

4.2 Participants

A total number of twenty adults between twenty-two and thirty-eight years old participated in this study. There were five women (25%) and fifteen men (75%). The total mean age was 26.65 $\pm$ 4.512 years old. In particular, the mean age for women was 24 $\pm$ 1.871 and for men was 27.53 $\pm$ 4.824. The participants were divided into two groups of ten persons.

4.3 Measurements

For this study two different post-questionnaires were used. The first questionnaire (see Table A1.1) was filled out after playing one of the games for the first time in order to measure the functionality and user perception of this app.

Once the participant had played the games with the two visualization modes (autostereoscopy and non-autostereoscopy), the second questionnaire (Table A1.2) was filled out. This questionnaire was used to determine the participant's preference for either mode.

In addition to the questionnaires, the app also saved the results of each task in a TXT-file, as shown in chapter 3, Listing 3.3. These have been analyzed for both visualization modes. While the position difference was stored, the variation of the values was too small to be of use for analysis. Hence only the time required to solve the "puzzle" was analyzed. An interesting aspect which is also visible in the time variable was that in some cases the subjects used different strategies to solve the "puzzle" after getting familiar with the control.

4.4 Procedure

To avoid that participant's opinions could affect the others; the participant entered the activity room one by one. Each participant was assigned to one of the following two groups:

- Group A plays first in autostereoscopy and afterwards with non-autostereoscopic visualization.
- Group B plays first with non-autostereoscopic visualization and afterwards in autostereoscopy.

Figure 4.1 illustrates graphically the procedure for both groups. The following protocol was used:

(1) The participant came into the room. Before the activity started the app was configured (visualization mode and participant-ID registration). After the configuration, the participant started to play the Demo stage of *3D Ball Game* in order to get familiar with the touch gestures.
(2) The second stage of *3D Ball Game* was launched and the subject was informed about the game rules e.g.: "find all green balls by moving through the scene with the index finger".
(3) If the subject gave a wrong answer in the *Solution Screen*. He or she had restarted the stage by using the continue button. Although the answer was wrong, the app stored the result (participant-ID, visualization mode, time and answer). In the other case, the participant started with the next stage (similar to point 2).
(4) If the number of repetition was three, the app started automatically the next stage.
(5) After finishing all stages of *3D Ball Game*, the Demo stage of 3D Puzzle Game was started. The participant first had to use the camera mode in order to find the red cube in the scene. Then, he or she had to move the cube in direction X or Z.
(6) The second stage of *3D Block Puzzle* was launched and the subject was informed about the game rules.

(7) If the subject failed to build the construction exact enough, the stage was reloaded by using the continue button and the primitives were reset. Although the construction was wrong, the app stored the result (participant-ID, visualization mode, time and coordinates of the moved objects). In the other case, the game started the next stage (similar to point 6).

(8) If the number of repetition was three, the app started automatically the next stage.

(9) After finishing both games, the participant was asked to fill out a questionnaire.

(10) The test was repeated by changing the visualization mode from autostereoscopy to non-autostereoscopy or from non-autostereoscopy to autostereoscopy.

(11) To conclude the test, the participant filled out the second questionnaire.

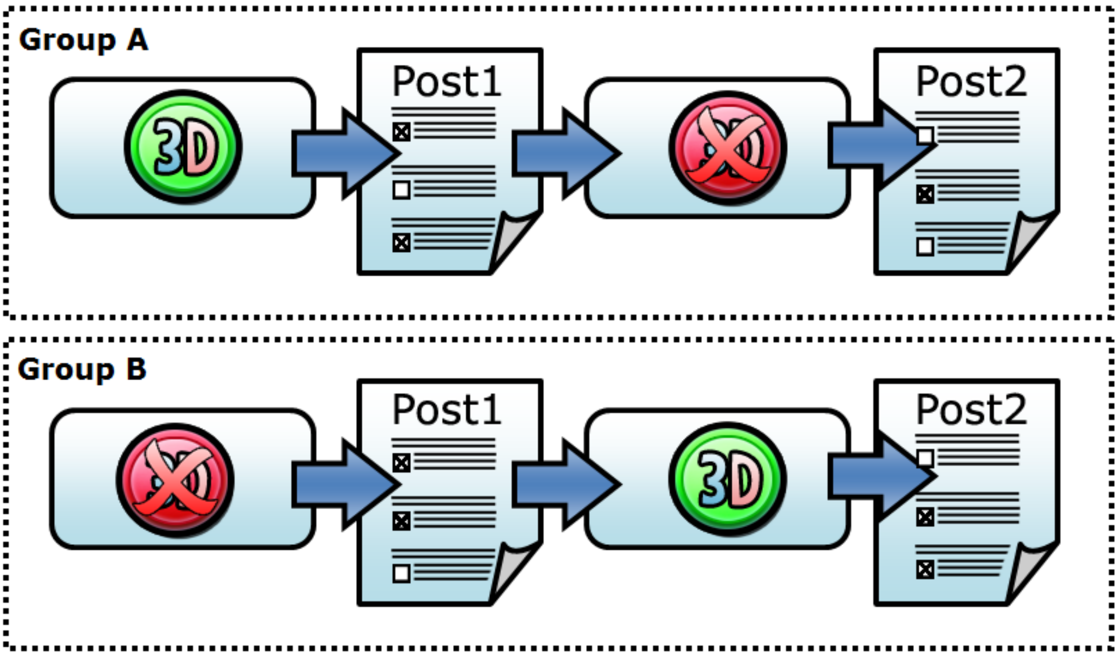


Figure 4.1: Study procedure for each group with both visualization modes.

4.5 Results of the app

In this section, a one-way-ANOVA with $\alpha = 0.05$ was used to determine if there was a significant difference between both visualization modes by using the recorded total time (in some cases a participant had several tries) to pass a stage. Apart from the ANOVA, the Box plot diagram was drawn for the two games and all of its stages in order to graphically illustrate the data variances between autostereoscopy and non-autostereoscopy.

4.5.1 *3D Ball Game*

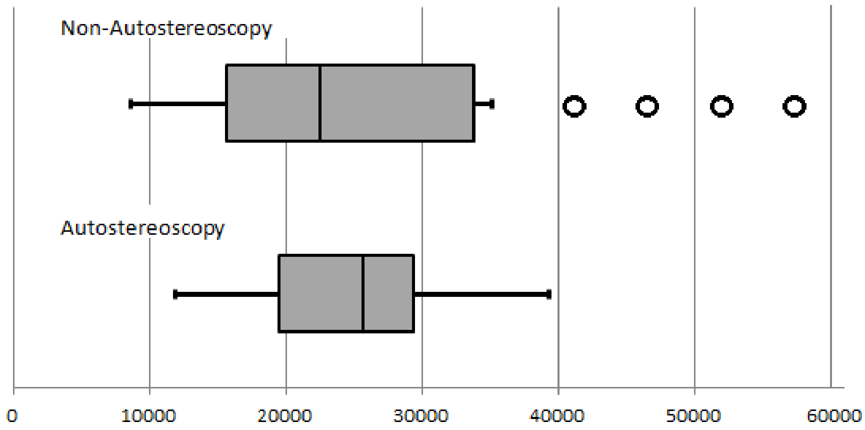
At the end the *3D Ball Game* turned out to be too simple, as subjects were not forced to move through the scene to count all balls. In the second run, the most of them remembered the solution from the first attempt.

However, the group A observed the 3D environment unhurriedly. Some of them started to move through the scene in order to play with the dept-perception, which caused extreme values (see Figure 4.2). Furthermore, the group B that played this game with autostereoscopy was more fascinated and confused by autostereoscopy as group A during their first exposure to this mode.

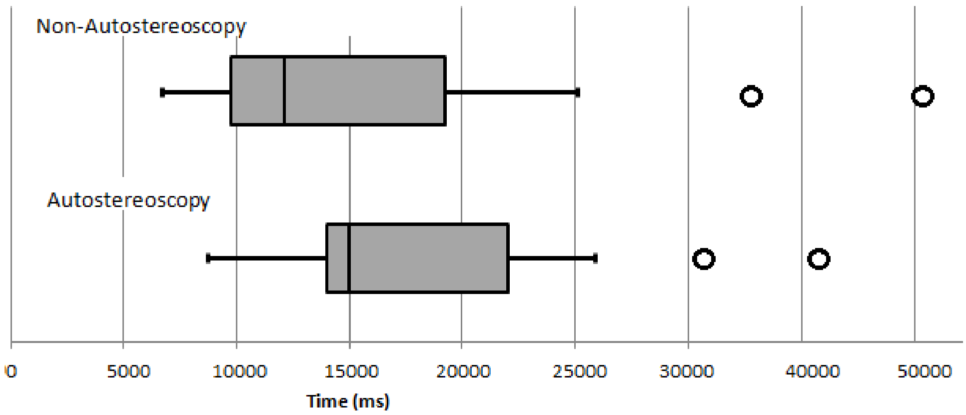
In Table 4.1 no significant difference was found in all stages, which could a result of the stages low difficulty and the fact that some participants started to behave different between the two visualization modes.

	Auto	*Non-Auto*	*P-value*	*F*
L2	*25087.200 ± 8165.295*	*26121.950 ± 14372.471*	*0.781*	*0.0784*
L3	*18436.400 ± 7693.095*	*16228.85 ± 10905.548*	*0.464*	*0.547*
L4	*12150.250 ± 5245.457*	*12862.200 ± 8973.724*	*0.762*	*0.092*

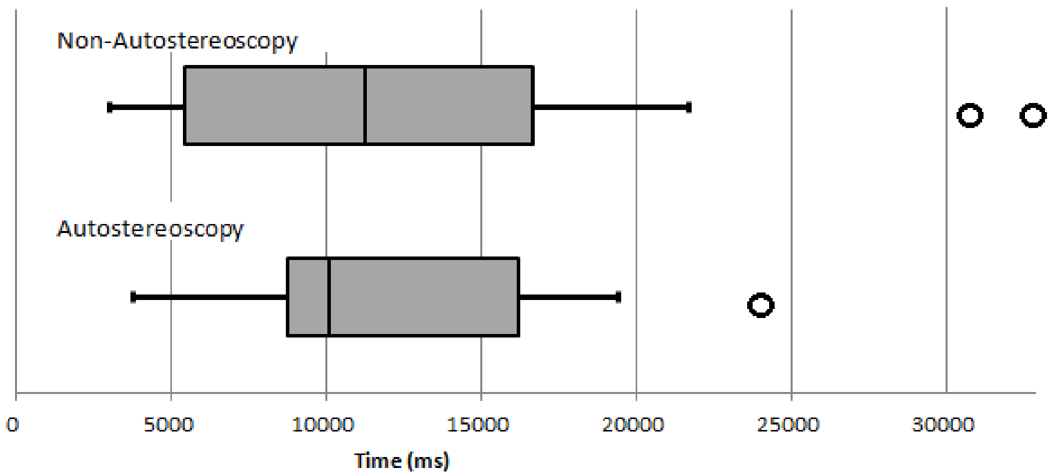
Table 4.1: ANOVA results for the stage comparison outcomes of 3D Ball Game with d.f. 38.



(a) Stage 2 time results.



(b) Stage 3 time results.



(c) Stage 4 time results.

Figure 4.2: Graphically test results demonstration of *3D Ball Game*.

4.5.2 *3D Block Puzzle*

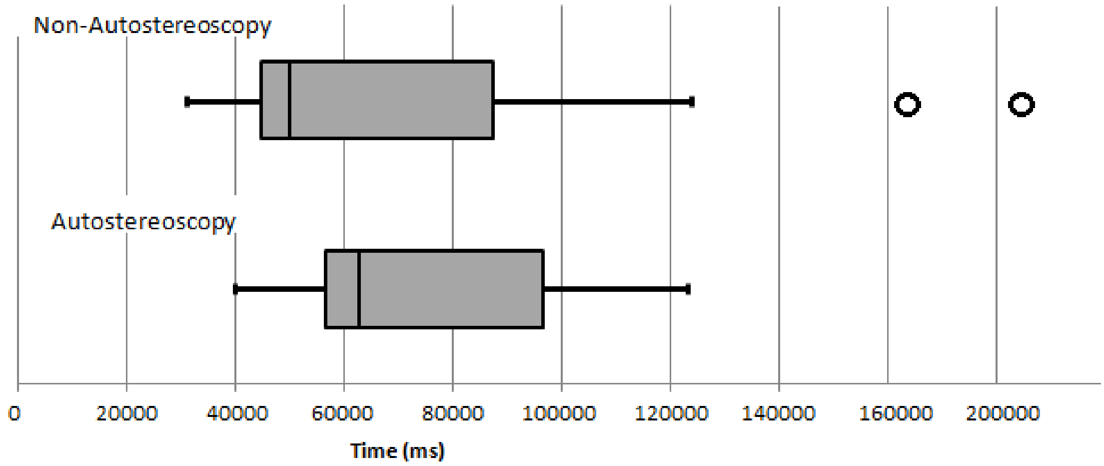
During the test, most of the participants started play excessively with the "camera-mode" in all stages, which caused extreme values (Figure 4.3) of two types: losing time by watching all faces of the construction and / or getting an unlucky projection of the scene that resulted in a failed to observe a misalignment in their construction, thus not obtaining all points (see appendix).

No significant difference was found in all stages, due to of the "low" 3D dept-perception. Although in the final stage (L5) the P-value is conspicuously low compared to the other stages, nevertheless no statistically significant, which is demonstrated in Table 4.2.

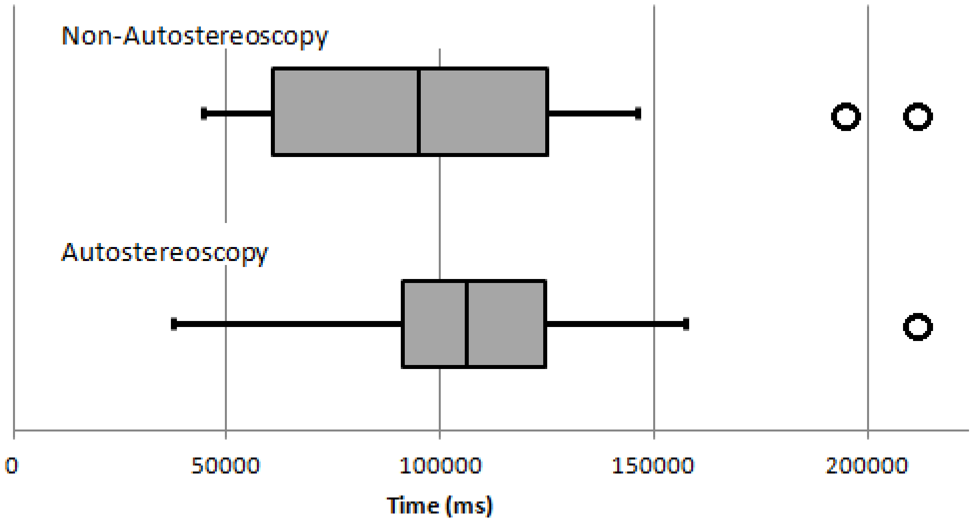
Another important aspect, almost all participants had explained that the difficulty was increasing extraordinarily fast in the first run. In addition, the last stage was scored as the hardest stage of the 3D Block Puzzle, the "help-image" displayed the result in a different camera angle as default, thus forcing to think in 3 Dimensions. This was made worse by the relatively small size of the "help-image" and resulted in participants loosing valuable time. This stage was named by some of the subject as "Brainfuck".

	Auto	*Non-Auto*	*P-value*	*F*
L2	*73326.700 ± 27015.939*	*73064.900 ± 47663.981*	*0,983*	*0.0005*
L3	*109552.450 ± 44715.769*	*104924.350 ± 49323.088*	*0.756*	*0.097*
L4	*186185 ± 70854.837*	*189003.300 ± 77306.678*	*0.905*	*0.014*
L5	*204089.700 ± 139332.561*	*164249.550 ± 55695.479*	*0.244*	*1.410*

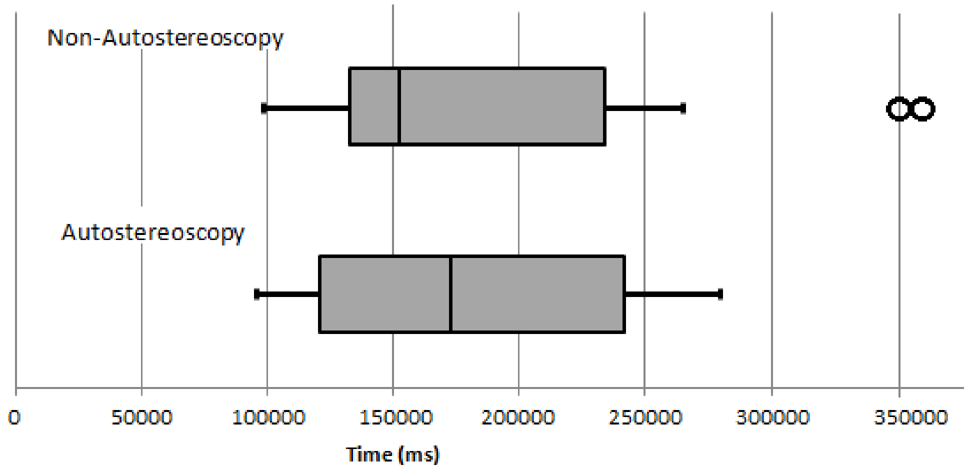
Table 4.2: ANOVA results for the stage comparison outcomes of 3D Block Puzzle with d.f. 38.



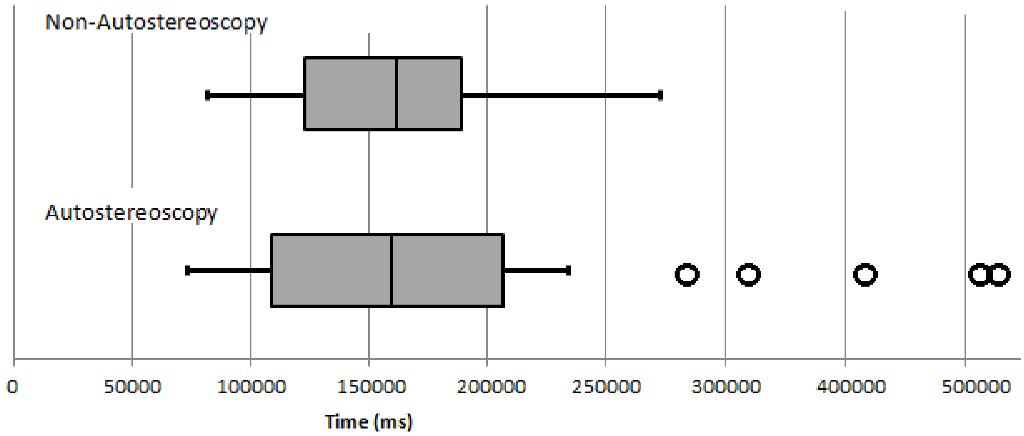
(a) Stage 2 time results.



(b) Stage 3 time results.



(c) Stage 4 time results.



(d) Stage 5 time results.

Figure 4.3: Graphically test results demonstration of *3D Block Puzzle*.

4.6 Results of the questionnaires

A one-way-ANOVA with $\alpha = 0.05$ was used to test if there was a significant difference between both groups in the first questionnaire (Table A1.1). For the second questionnaire (Table A1.2) a simple percent calculation was done.

4.6.1 System comparison outcomes

No significant difference was found in either question (see Table 4.3). Nonetheless, in *Q4* the *P-value* demonstrates that the groups had different opinions about the in-game objects.

In Q6 the subjects were asked if the shown objects / constructions were coming out of the screen. A statistical significant was expected as at this stage group B had only played this game in non-autostereoscopy. However, the ANOVA demonstrates the opposite. It could be that the 3D dept-perception was not good enough and/or the sample size was too small. Another interesting aspect was found in *Q8* where participants were asked if they would play the game again. In this test group A had a greater variance; however the ANOVA suggests that both groups had a similar general opinion.

	Auto	*Non-Auto*	*P-value*	*F*
Q1	*3,9 ± 0,876*	*3,8 ± 0,632*	*0.773*	*0.0857*
Q2	*3,2 ± 1,033*	*3,1 ± 0,568*	*0.7915*	*0.007*
Q3	*4,6 ± 0,516*	*4,5 ± 0,707*	*0.722*	*0.130*
Q4	4,1 ± 1,101	*3,6 ± 0,843*	*0.269*	*1.301*
Q5	3,8 ± 1,229	*3,4 ± 1,174*	*0.466*	*0.554*
Q6	2,9 ± 1,595	*2,0 ± 0,667*	*0.117*	*2.710*
Q7	2,6 ± 1,430	*2,7 ± 1,059*	*0.860*	*0.0316*
Q8	3,4 ± *1,430*	*3,4 ± 0,699*	*0.999*	*<0.001*
Q9	3,6 ± 1,075	*3,7 ± 0,675*	*0.806*	*0.062*

Table 4.3: ANOVA results for the system comparison outcomes with d.f. 18.

4.6.2 Satisfaction outcomes

In general, 70% of the participants found it easier to control the game in non-autostereoscopic visualization. The in-game objects - especially in *3D Puzzle Game* - were easier to see in non-autostereoscopy for 75%. Also 65% selected that for this type of activity it was better to use non-autostereoscopy. In Q4, slight deviations were found between both groups (see Figure 4.4). However, 65% preferred autostereoscopy mode, even though it was harder to control. The reason for this selection was: the objects were more realistic in a 3D environment, more intuitive, never played a mobile game with autostereoscopy making it an extraordinary event, an interesting dept-perception impression and funnier. Nevertheless, in autostereoscopic mode some participants suffered from side-effects such as, headaches, dizziness and seeing the objects / scene double.

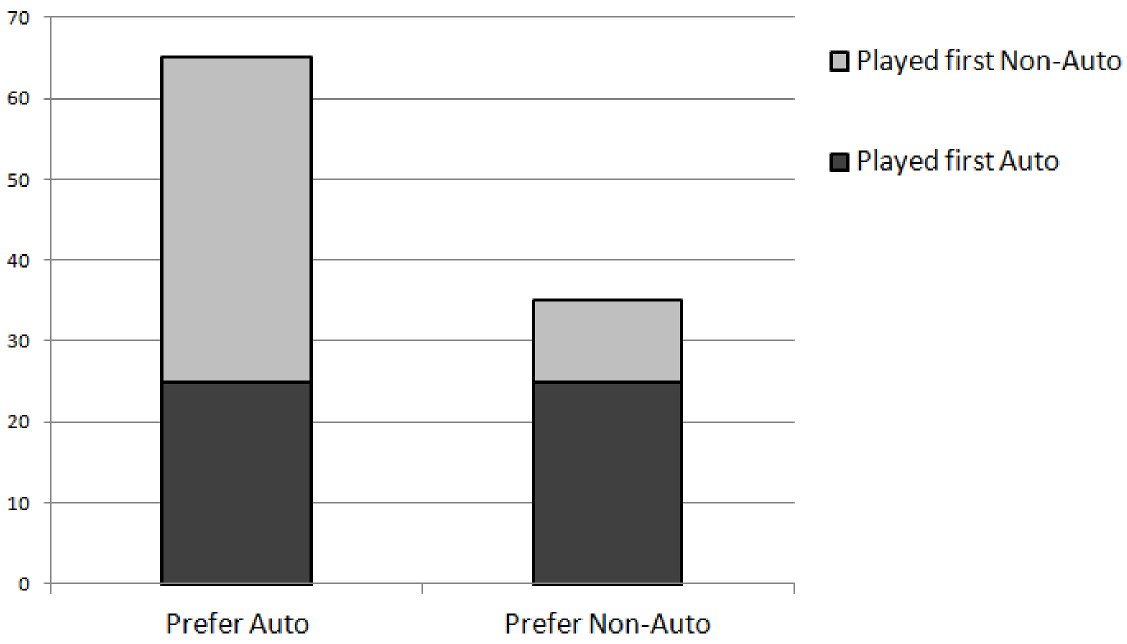


Figure 4.4: User's visualization preference.

V

Discussion

CONCLUDING REMARKS

5.

5.1 Conclusions

In this thesis, we carried out a research over the 3D depth-perception with the LG Optimus 3D Smartphone by developing an Android app with two types of games. To our knowledge, this is the first full autostereoscopic research app that supports two visualization modes (autostereoscopy and non-autostereoscopy).

The study showed that differences between autostereoscopy and non-autostereoscopy are no statistically significant. Moreover, the participants spent more time to solve the puzzle, due to two factors. Firstly, the impression of the depth-perception caused excessive and unnecessary use of the "camera-mode". Secondly some subjects suffered from adverse effects such as health issues (headaches, etc.), and / or crosstalk (receiving both images) which hindered the participants. The satisfaction outcomes demonstrated that the participant were interested and fascinated in this technology.

After a two year trial, the British Broadcasting Corporation (BBC) dropped their 3D TV broadcasting plans for an indefinite time, due to lower 3D viewing figures than expected, even though "half of the estimated 1.5 million households in the UK with a 3D-enabled television watched last summer's Olympics opening ceremony in 3D"(BBC, 2013c). Also Nintendo changed their focus on stereoscopy with the announcement of the new Nintendo 2DS. The Nintendo 2DS is a budget, flat design version of the Nintendo 3DS and will be released on 12 October 2013 in Europe. It is capable of running Nintendo 3DS games; however to save costs Nintendo dropped autostereoscopy support for this handheld (Stuart, 2013).

These developments suggest that after three years, it appears the 3D hype to be over and the beginning of a new towards high resolution display devices e.g. 4K TV and 2K mobile devices (Lane, 2013; Kelion, 2013a) which is further confirmed by the abundance of Full HD Smartphones such as the Samsung Galaxy S4. However in a recent announcement by Microsoft which confirmed their intentions to develop a protocol to support 3D video calls in Skype; may provide a new way for stereoscopy and autostereoscopy.

Stereoscopy and Autostereoscopy still need further improvements in order to provide a better experience for all users by improving the depth-impression and through the reduction of side-effects. Especially for mobile devices, this technology will depend on a solution to the view zone issue, as current devices rely on the eye's characteristics and still lack the support of several view zones (multi views) e.g. restricting the experience to only one active viewer.

5.2 Future work

During the development of the app, the given Tech-Demos from LG were reviewed. The best 3D dept-perception was given by 2D autostereoscopic examples and CameraPreview+OpenGL. Therefore, another aspect for the study could be the development of a 2D autostereoscopic app or of a full-3D environment with augmented reality.

For the both apps (2D autostereoscopic and augmented reality app), it would be interesting to provide other touch gestures, such as pinch open, pinch close, tap and hold, double tap and finally swipe. The pinch open and pinch close gestures could be used to realize a zoom in or zoom out into the environment. For design reasons this touch gesture was also missed in the final app version in order to not clutter up the screen. Furthermore potential use can be found in the utilization of the accelerometer and gyroscope sensors in Android. These sensors could be used to translate an object in X or Y or to reset the scene if the device is shaken.

As some subjects intuitively tried to touch the other primitives and moved accidentally the actual selected one user friendliness could be improved by using the ray picking method[7] to an object rather than a button as is used currently. Another proposal is to re-program the app as a computer video-game that works with on an autostereoscopic monitor in combination with the rotator peripheral device from Arino et al. (2013). This device could be used to rotate the scene and in some cases also the object.

The other two apps and the computer video-game should be tested by a group of children with a larger sample size. During the test with adults, a small group of them were lacked interest in the app and autostereoscopy.

A paper of this thesis will be submitted to the *Displays* Journal (Journal indexed in JCR).

[7] A good example is given by Beauchamp G in his blog. Available from: http://android-raypick.blogspot.com.es/2012/04/first-i-want-to-state-this-is-my-first.html

Bibliography

Android Developers, Supporting Different Languages [online].
Available at: http://developer.android.com/training/basics/supporting-devices/languages.html [Accessed 10 April 2013].

Arino J. J., Juan M. C., Gil-Gomez J. A. and Molla R., 2013. A comparative study using an autostereoscopic display with augmented and virtual reality, *Behaviour & Information Technology*.
In press, http://dx.doi.org/10.1080/0144929X.2013.815277 [Accessed 5 April 2013].

BBC, 2012, Oculus Rift virtual reality headset gets Kickstarter cash [online].
Available at: http://www.bbc.co.uk/news/technology-19085967 [Accessed 21 July 2013]

BBC, 2013a, Smartphones outsell basic feature handsets [online].
Available at: http://www.bbc.co.uk/news/business-23707341 [Accessed 5 August 2013]

BBC, 2013b, BBC 3D programming 'on hold' indefinitely [online].
Available at: http://www.bbc.co.uk/news/entertainment-arts-23195479 [Accessed 5 August 2013]

Boev A. and Gotchev A., 2011, Comparative study of autostereoscopic displays for mobile devices. In *Proc. SPIE 7881, Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V*, pages 112-124 [online].
Available at: http://dx.doi.org/10.1117/12.879225 [Accessed 20 July 2013]

Boffy B, 2006, Famicom 3D System, a liquid crystal shutter headset for the Nintendo Famicom which gives a 3D effect on specially-designed games [online].
Available at: http://en.wikipedia.org/wiki/File:Famicom_3d_system.jpg [Accessed 15 July 2013]

Bourge A. and Bellon A., 2013, 3D Video on Mobile Devices. In *Emerging Technologies for 3D video, creation, coding, transmission and rendering*, pages 432-449

CMG Lee, 2011, Comparison of parallax-barrier and lenticular autostereoscopic displays [online].
Available at:
http://en.wikipedia.org/wiki/File:Parallax_barrier_vs_lenticular_screen.svg [Accessed 18 July 2013].

DiMarzio J. F., 2011, Practical Android 4 Games Development, New York: Apress

Day D., 1994, Power vision, Day Dream D Publsihing, Inc.

Famicom world, 2013a, Famicom Disk System (FDS) [online].
Available from: http://famicomworld.com/system/other/3d-system/ [Accessed 15 July 2013]

Famicom world, 2013b, Famicom other Systems [online].
Available at: http://famicomworld.com/system/other/ [Accessed 15 July 2013]

Famicom world, 2013c, Famicom 3D Systems [online].
Available at: http://famicomworld.com/system/other/3d-system/ [Accessed 15 July 2013]

Famicom word, 2013d, Famicom converter [online].
Availabe at: http://famicomworld.com/system/converters/ [Accessed 15 July 2013]

Firth N., 2010, Japanese 3D singing hologram Hatsune Miku becomes nation's strangest pop star [online].
Available at: http://www.dailymail.co.uk/sciencetech/article-1329040/Japanese-3D-singing-hologram-Hatsune-Miku-nations-biggest-pop-star.html [Accessed 15 August 2013]

Funk W., 2008, Stereoptiplexer cinema system – "outside-looking-in" [online].
Available from: http://www.123seminarsonly.com/Seminar-Reports/018/64642674-Stereoptiplexer.pdf [Accessed 15 August 2013]

Gameloft, 2013, Monster life [online].
Available at: http://www.gameloft.com/minisites/monster-life/ [Accessed 10 July 2013]

Google play, 2013, Where's My Water? Free [online].
Available at:
https://play.google.com/store/apps/details?id=com.disney.WMWLite&hl=en [Accessed 05 September 2013]

Greer D., 2012, YouTube Adds 3D Video Capability and Nobody Cares [online].
Available at: http://www.premiumbeat.com/blog/youtube-adds-3d/ [Accessed 10 August 2013]

GSMArena, 2011a, LG Optimus 3D P920 [online].
Available at: http://www.gsmarena.com/lg_optimus_3d_p920-3759.php [Accessed 30 July 2013]

GSMArena, 2011b, HTC EVO 3D [online].
Available at: http://www.gsmarena.com/htc_evo_3d-3901.php [Accessed 30 July 2013]

GSMArena, 2011c, Sharp Aquos SH80F [online].
Available at: http://www.gsmarena.com/sharp_aquos_sh80f-4207.php [Accessed 30 July 2013]

Halvey M., Hannah D., Wilson G., and Brewster S. A., 2012, Investigating Gesture and Pressure Interaction with a 3D Display. *In Advances in Multimedia Modeling; Lecture Notes in Computer Science Volume 7131,* pages 395-405

HDMI, 2011, HDMI 1.4 Specification [online].
Available at: http://www.hdmi.org/manufacturer/specification.aspx [Accessed 10 August 2013]

Heise, 2009, 3D-Spezifikationen für Blu-ray Discs sind fertig [online].
Available at: http://www.heise.de/newsticker/meldung/3D-Spezifikationen-fuer-Blu-ray-Discs-sind-fertig-888117.html [Accessed 10 August 2013]

Ishaan, 2009, The Remarkable History Of The 3DS [online].
Available at: http://www.siliconera.com/2010/06/19/the-remarkable-history-of-the-3ds/ [Accessed 15 July 2013]

Ishio H., Sano S., Shiomi T., Kanda T., Hori H., Uemoto K., Sugiyama A., Niwa M., Hasegawa A., Matsunuma S. and Miyao Ma., 2011, Visibility Experiment and Evaluation of 3D Character Representation on Mobile Displays. In *HCI International Communications in Computer and Information Science, Volume 174*, pages 46-51

Jaffe S., Walker C. and Oswald E., 2013, Effects of 3D Audio and Video in Video Games, Bachelor of Science Thesis, Worcester Polytechnic Institute [online]. Available at: http://www.wpi.edu/Pubs/E-project/Available/E-project-030713-133549/unrestricted/Effects_of_3D_IQP.pdf [Accessed 25 July 2013]

Keene J., 2011, WikiPad pitches glasses-free 3D and Ice Cream Sandwich for less than $250: hands on (update) [online].
Available at: http://www.theverge.com/2012/1/11/2699768/wikipad-glasses-free-3d-tablet-android-4 [Accessed 7 September 2013]

Kelion L., 2013a, LG Display phone screen offers 'highest resolution' [online].
Available at: http://www.bbc.co.uk/news/technology-23777780 [Accessed 29 August 2013]

Kelion L., 2013b, Skype confirms 3D video calls are under development [online].
Available at: http://www.bbc.co.uk/news/technology-23866593 [Accessed 29 August 2013]

Kluczniok J., 2012, LG Optimus 3D Max: Marktstart der zweiten 3D-Handy-Generation [online].
Available at: http://www.netzwelt.de/news/91956-lg-optimus-3d-max-marktstart-zweiten-3d-handy-generation.html [Accessed 30 July 2013]

Lane A., 2013, CES 2013: BBC man says 3D TV is dead but 4K and Google TV are great [online].
Available at: http://recombu.com/digital/news/ces-2013-bbc-man-says-3d-tv-is-dead-but-4k-and-google-tv-are-great_M11163.html [Accessed 29 August 2013]

LG Electronics, 2011, The mobile revolution begins… [online].
Available at: http://www.lg.com/uk/mobile-phones/lg-P920-optimus-3d [Accessed 30 July 2013]

LG Developers, 2011, Basic Principle of 3D Stereoscopy [online].
Availaboe at:
http://developer.lge.com/resource/mobile/RetrieveDocDevLibrary.dev;jsessionid=A250F7ED6D1DE5F73DA8B5DA987A895B.node_sdpidl10 [Accessed 10 July 2013]

Lehtimaki, J., 2013, Smashing Android UI, United Kingdom: Wiley

Mao X., Marks R. and Mallinson D., 2012, 3-D glasses with camera based head tracking, patent [online].
Available at:

https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US20120088581.pdf [Accessed 15 August 2013]

Mehrabi M., Peek E.M., Wuensche B.C. and Lutteroth C., 2013, Making 3D Work: A Classification of Visual Depth Cues, 3D Display Technologies and Their Applications. In *Australian Computer Science Communication, Volume 35 Number* 5, pages 91-100

Nintendo3DBrew, 2013, Nintendo 3DS Hardware information [online]. Available at: http://3dbrew.org/wiki/Hardware#XL_model [Accessed 15 July 2013]

Nintendo Co. America, 2011, Your Nintendo 3DS System [online]. Available at: http://www.nintendo.co.uk/Nintendo-3DS/Features/Features-114646.html [Accessed 15 July 2013]

Nintendo Co. America, 2013, Nintendo DS Hardware Specs comparison [online]. Available at: http://www.nintendo.com/3ds/features/compare [Accessed 15 July 2013]

Nintendo Co. Japan, 2011, Nintendo 3DS Hardware Specs [online]. Available at: http://www.nintendo.co.jp/3ds/hardware/spec/index.html [Accessed 15 July 2013]

Nintendo Co. Japan, 2013, Consolidated Sales Transition by Region [online]. Available at: http://www.nintendo.co.jp/ir/library/historical_data/pdf/consolidated_sales_e1306.pdf [Accessed 15 August 2013]

Norris A., 2002, The return of 3D [online]. Available at: http://www.theguardian.com/technology/2002/dec/06/mobilephones.ashleynorris [Accessed 20 August 2013]

Pallas A., Meyer C.H. and Mojon D., 2013, Nintendo 3DS. In *Der Ophthalmologe Volume 110, Issue 3*, pages 263-266

Seguin G., 2008, Nintendo Virtual Boy. Festival du jeu vidéo 2008 (video game festival) (Paris, France) [online]. Available at: http://commons.wikimedia.org/wiki/File:Festival_du_jeu_video_20080926_017.jpg [Accessed 20 August 2013]

Science AAAS, 2013, 3D TV, Without the Glasses [online]. Available at: http://news.sciencemag.org/2013/03/3d-tv-without-glasses [Accessed 20 August 2013]

Smartgoggles, 2012, Sensics Introduces the Natalia 3D Goggles, Incorporating SmartGoggles™ Technology [online]. Available at: http://smartgoggles.net/sensics-introduces-the-natalia-3d-goggles-incorporating-smartgoggles-technology/ [Accessed 20 August 2013]

Stuart K, 2013, Nintendo 2DS: is this the end of 3D gaming? [online]. Available at: http://www.theguardian.com/technology/2013/aug/29/nintendo-2ds-future-of-gaming [Accessed 29 August 2013]

Surman P., 2013, Stereoscopic and Autostereoscopic Displays. In *3D-TV System with Depth-Image-Based Rendering*, pages 375-411

Van Knotsenburg J., 2012, Analysis of User Interactions on 2D and Autostereoscopic 3D Mobile Touch Devices, Master Thesis, Utrecht University [online]. Available at: http://igitur-archive.library.uu.nl/student-theses/2012-0925-200726/Analysis%20of%20User%20Interactions%20on%202D%20and%20Autostereoscopic%203D%20Mobile%20Touch%20Devices.pdf [Accessed 10 July 2013]

Wei C.H., Chiang C.K., Sun Y.W., Lin M.H. and Lai S.H., 2013, Novel Multi-view Synthesis from a Stereo Image Pair for 3D Display on Mobile Phone. In *Computer Vision - ACCV 2012 Workshops Lecture Notes in Computer Science Volume 7729*, pages 568-579

Williams M., 2010, Sony to Begin Worldwide 3D TV Launch in June [online]. Available at: http://www.pcworld.com/article/191056/article.html [Accessed 10 August 2013]

XPAND, 2011, Four Companies Will Seek Development of Joint Licensing on 'BLUETOOTH® ENABLED RF' and 'IR' Consumer 3D Active Glasses [online]. Available at: http://www.xpand.me/news/92/ [Accessed 10 August 2013]

Zachara M. and Zagal J.P., 2009, Challenges for success in stereo gaming: a Virtual Boy case study. In *ACE '09 Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, pages 99-106

A

Appendices

QUESTIONNAIRES

A1.

This appendix presents the two used questionnaires in form of tables. For each question in both questionnaires, only one answer can be selected. All possible answers will be shown below the question.

A1.1 Application questions (Post1)

In order to measure the functionality and user perception of this app, all participant were asked the following questions, after executing the app for the first time.

Q1	**Was it fun to play?**				
	Not at all	A little bit	Average	Good	Great
Q2	**Do you find it easy to play?**				
	Very difficult	Difficult	Average	Easy	Very Easy
Q3	**Did you always understand the rules of the game?**				
	Never	Rarely	Sometimes	Mostly	Always
Q4	**How did you like the objects in the game?**				
	Not at all	A little bit	Average	Good	Great
Q5	**Would you recommend this game to your friends?**				
	None	Almost none	I don't know	To some	To all
Q6	**Did it feel as if the objects were coming out of the screen?**				
	Not at all	A little bit	Average	Moderately	A lot
Q7	**Did it feel as if you were able to touch any object?**				
	Not at all	A little bit	Average	Moderately	A lot
Q8	**Would you play it again?**				
	Never	Rarely	Sometimes	Mostly	Always
Q9	**Please rate the game from 1 to 5**				
	Very Bad	Bad	Average	Good	Great

Table A1.1: Application questionnaire

A1.2 System comparison questions (Post2)

In order to determine the perception differences between the two applied visualization modes (autostereoscopy or non- autostereoscopy), each participant were asked the following questions.

Q1	**Which mode was easier to control?**
	Autostereoscopic (3D) Non- Autostereoscopic
Q2	**In which mode, have you seen the objects better?**
	Autostereoscopic (3D) Non- Autostereoscopic
Q3	**What do you think is better for the activity, you have done?**
	Autostereoscopic (3D) Non- Autostereoscopic
Q4	**Which mode did you enjoy the most?**
	Autostereoscopic (3D) Non- Autostereoscopic
	Why?
Q5	**Add comments / suggestions**

Table A1.2: System comparison questionnaire.