An Augmented Reality LEGO Set

To Lan Nguyet Druckmiller
H00117235

Heriot-Watt University
Computer Science
School of Mathematical and Computer Sciences
Dissertation submitted as part of the requirements for the award of the degree of
MSc in Creative Software Systems
Acknowledgement

In alphapetical order, I would like to thank Nicolas Eisele, Michael Ibrahim Heins, and Eglė Petrauskaitė, who gave me a shelter and supported me in every single phase of my project. Thanks to Giulia Linusvanpelt Agnisola, Pier Daniel Cornacchia, Mauricio Alejandro Escamilla, and Giacomo Frega for giving me a lot of strength. Many thanks to my parents who gave me the opportunity to be here.
Next to that, I am very grateful to have Judy Robertson as my supervisor who cared a lot about the project, listened to my ideas, and helped me out whenever she could.

Thank you!

Originality Declaration

I, To Lan Druckmiller, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signature:

Date:
Abstract

Almost everyone can remember a time in which computers were not present in our daily life. Seeing children playing video games reminds us of how we grew up, and the games we liked to play when technology was not as advanced and widespread as it is today.

This project will combine a traditional style of play using building blocks with new technology. This approach replaces the mere use of mouse and keyboard as input devices by placing the interaction back into physical objects in the three dimensional world.

Imagine a child playing with building blocks on the floor. As s/he moves the blocks in real life, 3D blocks will move correspondingly in a virtual world. The task of this project is to make use of this virtuality, so that it can act as added value to an already existing toy, a LEGO set. For this, a prototype that uses augmented reality as core technology will be developed and evaluated.
# Contents

1 **Introduction** 3  
   1.1 Development Methodology 3  
   1.2 Aims and Objectives 4  

2 **Literature Review** 5  
   2.1 Taxonomy 5  
   2.2 Related Works 12  
   2.3 Technical Aspects 18  
   2.4 Summary 23  

3 **Concept** 25  
   3.1 Professional, Legal, and Ethical Issues 25  
   3.2 Concept Creation 26  

4 **Evaluation: First Iteration** 28  
   4.1 Requirements Analysis 28  
   4.2 First User Study 32  

5 **Evaluation: Second Iteration** 41  
   5.1 Extended Requirements Analysis 41  
   5.2 Second User Study 45  

6 **Developer Guide** 50  
   6.1 Basis 50
1 Introduction

Technology becomes more and more important in children’s every day life. Statistics show that 72% of children in the United States in the age between 0 and 8 years have access to a computer. 59% of them have ever used a computer. Interestingly is that 68% of 5-to 8-year old children use the computer at least on a weekly basis [26]. Concerning about these statistics is that children aged 8 or younger spend more time playing games with media than using it for educational, or learning purposes[22].

1.1 Development Methodology

For the project, literature is reviewed to form a concept that connects a traditional toy with technology, and has children and their learning behavior in mind. This basic concept will be realized in a prototype which then will be evaluated.

The evaluation is going to take place in two cycles. For a first iteration shall test basic functionalities with children. Contextual inquiry and interviews with the focus group are used as examination methodology. Some of the tests should be run in groups so that it can be observed how children interact with each other when they are using the application. Interview shall be held to gather additional ideas.

As a result of this examination phase the requirements will be altered. Accordingly the prototype will be extended by additional functionalities. This prototype will then be
used for a final evaluation. Observation should take place to see how the children react on the system. With the help of two iterations, it should be considered in which extend an augmented reality application can enrich a children’s playground.

1.2 Aims and Objectives

My work is going to face the conception and creation of an educational system that uses augmented reality as technology, and building blocks as medium for interaction. This system should be both, fun to play, and at the same time teach children constructional skills. Hereby, I am going to focus on a certain user group, and apply theories of how children can learn the most effective on the guidelines for play environments. A prototype will be developed and an evaluation will take place in two iterations. With the help of the user studies the requirements of the software will be extended to form a profound concept for a new play environment for children.

In the following, chapter 2 covers a literature review that clarifies important taxonomy for the project, discusses related work, and explains technical aspects. Chapter 3 applies findings of the literature review to my project. Here, the concept of my application will be explained, and it will be covered in which extent it covers given theories and guidelines. Also professional, legal and ethical issues shall be covered in this part. In chapter 4 and 5 an iterative evaluation of the project takes place. Based on the first user study, the application will be improved or extended, and evaluated in a second iteration. In chapter 6 implemented code will be explained in an abstract way. Chapter 7 includes software tests that reveal strengths and weaknesses of the application. It also includes a small guide under which circumstances the application works the best. Finally, chapter 8 summarizes the main aspects of my work and suggests improvements for future work.
2 Literature Review

The literature review in this work covers three subjects: taxonomy, related works and technical aspects. Taxonomy (section 2.1) is necessary to define relevant terms and to create a common understanding of them. The section Related Works (section 2.2) then summarizes several projects that are in some way related to my work. Each project is followed by a critical assessment. The section Technical Aspects (section 2.3) includes a variety of technology that can be used for the augmented reality LEGO set. The final section in this chapter summarizes the findings to give an overview of my project.

2.1 Taxonomy

In this section relevant taxonomy will be defined to clarify the meaning of certain terms used in this report and also to declare abbreviations. I will cover virtual, mixed and augmented reality, tangible user interfaces, and child computer interaction. Additionally, theories of the components of an effective learning, and the rules for creating play environments will be stated in this section.

About Virtuality and Reality

The terms Augmented Reality (AR), and Mixed Reality (MR) can easily be confused. While Virtual Environments (VE) are clearly defined as a synthetic world, and real environment as the real physical world, MR can be found somewhere in between. It was Milgram [23] in 1994 who created the first taxonomy of MR. According to him, AR can be seen as a subfield of MR.

AR in the proper sense, is the combination of real and virtual world in real time. Whereby the merge of the two worlds is restricted to the visual representation of the virtual environment. One of the first papers that surveys AR was written by Azuma [3]

The confusion of MR and AR happens then, when AR does not only happen on displays, but also with the help of other medias such as audio and haptic. This definition in fact results in the given explanation of MR. For example in a more recent approach, Ritsos et al. [27] describe AR as a merge of virtual and real world just as Azuma, but they define more categories as in- and output for AR.

Inputs can be:

- visual (collecting data with a camera)
- auditory (i.e. voice commands)
- haptic (via input devices)
- kinesthetic (i.e. motion tracking and body recognition)
- sensory modalities (“to detect the user’s and the environment’s context” [27, p.3])

And outputs can be:

- visual (2D or 3D objects)
- auditory (feedback via sound)
- haptic (touch and feel, yet mostly realized through game controllers)

![Figure 2.1: Simplified Representation of a “Virtuality Continuum” [23]](image)
In the following, I will use MR and AR in the original sense, so that MR is the general fusion on virtual and real environments, and AR is the one that can only be shown on displays. Figure 2.1 shows the relationship between real environment, virtual environment, mixed reality and augmented reality.

However, it can be said that MR was and still is an evolving technology. For example, the kinesthetic input described by Ritsos et al. [27] without anything attached to the body is a reasonably new technology which mostly became popular because of Microsoft’s Kinect released in 2010. Examples like this show that one can expect more technical novelties to come.

**Tangible User Interfaces**

*Tangible User Interfaces* (TUIs) are definitely going to be in focus of my work. Therefore, a definition of this term should be found first. Ishii and Ullmer introduced in 1997 their idea of turning digital information into a touchable and manipulatable one [18]. They envisioned a world in which every physical object is represented as a bit in the digital environment, or in their words "cyberspace". As soon as the physical object is changed, moved, or manipulated in some way, the system reacts accordingly. Their vision can be seen as countermovement of using devices like mouse and keyboard. In their opinion, these devices are an alienation of the human intuitiveness. Whereby, intuitiveness can be defined as natural and instinctive behavior in this context. It is the natural interaction or reaction of humans with/on objects or other human beings.

The traditional mouse and the monitor are only used in two dimensions whereas humans live in a three dimensional world. This discovery led the authors to the definition of TUIs. With this idea Ishii and Ullmer started a new form of user interaction which is still, 15 years after, evolving and creates space for further research.

**Child Computer Interaction**

The field *Human Computer Interaction* (HCI)\(^1\) has been explored in depth for several years. Whereas Child Computer Interaction (CCI) in contrast, is a relatively new term, and needs to find its way to a common understanding.

\(^1\)HCI is the discipline that faces the interaction between human and machine.
The usage of technology is becoming more and more important for children in both, the educational and the entertainment area. Because of this, a growing demand for a guideline of how children interact with computers arose. This is hard to achieve, because there is a need to involve children in the design and evaluation process who are not always able to express their needs in a way that adults can understand and interpret [22]. It was Antle who created a framework for child-personas first. A child-persona is the interpretation of children’s psychology into an interaction design that is centered around the children in three dimensions: their needs, their developmental abilities, and their experiential goals [1]. The framework is based on theory and empirically gained data, and is meant to help developers in child-centered interaction design. In the following the three dimensions that need to be considered while creating a child centered design are summarized briefly:

• Children need:
  – to feel safe and secure while developing the ability to be independent.
  – to grow up in a positive environment to develop social abilities. This happens through observation and imitation.
  – to learn through active participation. Here, Antle defines two categories of responses of children in learning situations: helpless and mastery-oriented. A child that feels helpless in a learning situation needs guidance to develop the ability to master the situation.
  – to control their environment. Children have barely the power over the happenings in their environment. So, they need to develop the ability to be responsible for the consequences of their actions.

• Children’s developmental abilities:
  Children’s developmental abilities can not only be identified by the age of the child, because each child develops differently. Still, their developmental ability needs to be generalized. For this Antle suggests a categorization of motor, social, and cognitive abilities. Hereby, age-dependent classification is still possible.

• Experiential goals:
  On designing a product that is centered around children the experiential goal has
to be considered using so called "action modes". In other words, the product that is created for entertainment or learning purposes needs to set children a goal that can be achieved in making experiences.

The framework created by Antle is the basis of creating interaction between children and machines while considering their psychological backgrounds. For my work it represents the challenge that needs to be faced.

**Components for an Effective Learning**

To use new technologies for educational purposes, the knowledge about how children learn is important. For effective learning four basic factors need to be considered:

"(1) active engagement,
(2) participation in groups,
(3) frequent interaction and feedback, and
(4) connections to real world contexts"

– Roschelle et al. [28, p. 79]

First, active engagement is considered to be the countermovement to learning by heart. Children have to understand material and apply it. Second, participation in groups is a social component that increases the motivation for learning. Third, frequent interaction and feedback supports children in learning because they can directly see the fail or success of their work. Last, connections to real world contexts are the transition of theory into practice which causes information to stick in children’s memory.

All four components need to be considered on creating an educational system. Intelligent Tutoring System (ITS) is a research area that faces the integration of psychological concepts of how children learn into an educational system. Also the term Child Computer Interaction (CCI) that was covered before can not be ignored in this context.
Relation between Play and Game

Salen and Zimmerman describe the relation between play and game in a book called *Rules of Play: Game Design Fundamentals* [29]. They state that games are a subset of play, but at the same time, play is a component of games. These two statements seem contrasting in first glance, but can easily be explained. On the one hand, play can be seen as a looser form of games. It is less organized, and it is not necessary that rules exist to win or lose it. As soon as a play becomes more formalized, it is considered to be a game. On the other hand, play is a component of games because it is one of the elements (if not the main element) of a game concept. I will mainly concentrate on the term play, because the aim of my project is to create a play environment, and not a game environment. Following this, Salen and Zimmermann give the following definition:

“Play is free movement within a more rigid structure.”

– Salen and Zimmermann [29, p. 300]

This quote is useful in the sense that I will create an application that encourages children to play freely (meaning with not too many rules) and without a clear ending.

Meaningful Play

Salen and Zimmerman categorize meaningful play into two kinds: descriptive and evaluative. A play is descriptive if each action of the player results in a response of the system so that it can generate a meaningful experience to the player. When the action of a player does not only have an immediate effect on the play, but also on the later progress of the play, it is evaluative. Generally, it means that every action of a player has consequences, and the player needs to perceive these consequences, because otherwise s/he can do whatever s/he wants and it doesn’t has any effect. Overall, it is important to create a meaningful play, because its success depends on it.
Interactivity

One of the most important terms in play is interactivity. Without interactivity no play takes place. For this, the authors describe three levels on which interaction takes place:

• formal interaction which takes place between the player and the objects\(^2\),

• social interaction which takes place between the players, and

• cultural interaction which takes place “[…] beyond its space of play […]” [29, p. 58]

For a complete understanding, I would like to add Schell’s definition of (inter-)action in *The Art of Game Design* [30]. He describes two kinds of action: operative and resultant. Operative are these actions that the user can perform, and resultant are those with a certain goal or strategy behind the interaction.

In this project, children have to interact with LEGO bricks and the system reacts accordingly. Also, it should be possible to interact with the system collaboratively which would cover the second component of an effective learning at the same time.

Rules

“Following are the general characteristics that all game rules share:

• *Rules limit player action*

• *Rules are explicit and unambiguous*

• *Rules are shared by all players*

• *Rules are fixed*

• *Rules are binding*

• *Rules are repeatable*

”

– Salen and Zimmermann [29, p. 125]

\(^2\)Objects in play environments are things that can be manipulated. They have certain attributes whose value can be changed [30].

11
Rules are the fundament of a game which means that a game can not exist without rules. A play on the other hand doesn’t necessarily have to define rules. Nevertheless, they should be covered in this scope, because they will become important for the requirement analysis at a later point of the project.

The only statement that I would not completely agree with is that rules are fixed. It often occurred to me, that rules in a game were changed / added / deleted, if the game makes more sense or is more enjoyable with those changes. Following this, I would like to create a play with as few rules as possible, and the ones that are given are the ones that are required in order to play and interact with the system.

### 2.2 Related Works

In the last couple of years, several approaches have been created which face the combination of mixed reality with learning applications. In the following, I am going to present some of the work that has been done in this area. They are related to my work in that sense, that they cover different kinds of hardware and software technology, are centered around a specific user group, or give an important guideline that needs to be considered in my future work. I am going to summarize the examined literature, explain the basic concepts of them and asses them critically with regard to my topic. As a result, I hopefully can benefit from the strength and weaknesses of their work.

"Learning Using Augmented Reality Technology: Multiple Means of Interaction for Teaching Children the Theory of Colours" by Ucelli et al. (2005)

Ucelli et al. [36] created "The Book of Colors". This is a book that explains children the concept of colors. It contains several fiducial markers. These markers can be recognized, and their idea is it to augment a three-dimensional model of a chameleon from it which is colored in the combination of the colors shown in the book. They benefit from the traditional way of how children learn which is the physical book. Next to that, they add a virtual component to it which motivates children because they can learn in a playful way.
The positive findings of this project is the interactivity that comes along with the AR, the ease of use, because the children do not necessary need to know how to use a computer (secondary devices like mouse, keyboard etc. are not needed to use the system), and the active engagement in trying to guess what the color of the chameleon is going to be plus having a direct feedback. The system was realized with a head mounted display (HMD) which turned out to be a big obstacle. The given reason by Ucelli et al. is that it is too heavy for children and also does not suit their head well. In my opinion, also the cost of the HMD should be considered, because even seven years after their work was published HMDs are still not affordable for the simple household. Also, only as many children as HMDs are available can benefit from the AR technology of their project and even if every child has his/her own HMD there will be certainly barriers in the natural interaction with each other. Plus, health and safety aspects and the interaction in a group are not regarded in this article. HMDs cause eye-strain and cyber-sickness on wearing them for a certain time. These concerns will also be covered later on in section 2.3.

"PhoneGuide: Museum Guidance Supported by On- Device Object Recognition on Mobile Phones" by Föckler et al. (2005)

I could hardly find any references for mobile AR systems that are designed for children. Singh et al. created an AR comic book and AR notes for children using mobile phones [33], but their work is poorly documented and not evaluated at all. The report only describes the applications itself, but no concept behind it. Also, technical background is missing completely. Even though I found it was an interesting work, I can barely benefit from it and it is hard to use it as a reference.

One of the first mobile systems that uses AR is MARS (Mobile Augmented Reality System) created by Höllerer et al. from 1996 onwards [16]. The users wear an HMD through which they can perceive their environment with an augmented layer on top of it. Initially, the authors created testbeds with different user interfaces that can be used indoors and outdoors. The mobility is only possible if the users carry around a backpack system [16]. I don’t think that children can or should cope with a technology like this. Which is the reason for not explaining their system in depth. The researchers at Columbia University are still working on this project and are sure that it will find its
way into peoples every day life [9]. Even though, I don’t find their work suitable for my research at the very moment, it still deserves acknowledgement for being a milestone in the history of mobile AR computing.

Instead, I am going to focus on a mobile museum guide created by Föckler et al. in 2005 [10]. The system they developed is able to recognize objects on an exhibition via photographs. According to the authors, input from the user like entering a unique number, or recording a certain marker is unnecessarily complicated and inconvenient. So, they created an application that is able to identify an object from pictures with a recognition rate of over 90 percent in less than a second through color, intensity and shape. This is all calculated on the phone so that no data needs to be send to a server.

In their work, Föckler et al. explain the technological background in depth, and dwell on a discipline that is very important for AR systems – object recognition. The evaluation part is less centered around the fact how users responded to their system but focusses more on the technical aspect of the application. As a result, they were able to create a field survey under realistic conditions in which the system was able to distinguish between 50 different objects of an exhibition.

The authors considered the fact that all computation could be performed on the phone itself as a bonus because it does not bear any additional cost for the network connection. Nowadays, this factor does not necessarily need to be considered anymore. According to a study of the Office for National Statistics in 2011 the numbers for usage of internet technology on mobile devices in the UK increased significantly. Among 16- to 24-year-olds the rate grew from 44 percent to 71 percent within one year [24]. Also, they expected that a simple installation on a phone using the Symbian operating system (OS) would lead to a success. They expected that this would also reduce the expenses, "Since the visitor bring along their own phones" [10, p. 4]. Statistics show that the number of smart phones with Symbian OS continue to decline, whereas the sale iOS and Android systems increased significantly over the last couple of years [12]. Furthermore, for my research it needs to be considered whether a mobile application is suitable for children since there are ethical issues with children having access to mobile devices. I think this is also the reason why I could barely find any research on mobile phone apps using augmented reality with children as target user group. Also, it is clear that 3D rendering is computational expensive, and it is mostly necessary to transfer data to a server that performs the calculation.
"Design A Situated Learning Environment Using Mixed Reality Technology - A Case Study" by Yusoff et al. (2010)

Yusoff et al. [40] are trying to find guidelines to "design a situated learning environment using mixed reality technology" [40]. Therefore, they analyze a book similar to the one created by Ucelli et al. [36]. Again the book still functions as traditional textbook that explains scientific processes, but it also contains markers that can be tracked. This time the book is more meant to be used as educational tool that can also be introduced into secondary schools. The user of the system is in the position of a lab assistant who can interact with virtual objects such as petri dishes and text tubes. The learning effect is magnified through the animation of the virtual objects to i.e. demonstrate chemical reactions. In this paper, the authors describe the benefits of MR as a learning application in depth. Those are:

- **Attention** occurs because students are automatically attracted to the system (which is more based on the novelty of the technology),

- "**Constructivist learning environment**" [40, p. 888] which can be concluded from the immersion that virtual worlds create,

- **Sensory perception** caused by the interaction of the user,

- **Authenticity** that only occurs when real elements (i.e. objects, sounds, haptic) are reproduced virtually,

- enrichment through **realistic models**.

Next to that, Yusoff et al. map the learning outcomes of their use study on a model that is similar to the one created by Roschelle et al. [28] which consists of the four factors for a successful learning covered earlier. Here, active engagement is created through the interaction of the user with the system, in other words, s/he can move the real book and see the virtual object reacting. The participation in a group happens as soon as at least two students use the book together. In this system the virtual world is displayed on a monitor so that they share the same environment. This fact enables them to communicate, discuss and collaborate about the learning materials. Feedback is provided if the user accomplishes a given task. The connection to the real world is
given through the user being in the position of a lab assistant. The biggest challenge the authors faced in this article is according to them the lack of knowledge of setting MR technologies into a learning context.

"Towards Guidelines for Designing Augmented Toy Environments" and "Kingdom of the Knights: Evaluation of a Seamlessly Augmented Toy Environment for Playful Learning" by Hinske et al. (2008)

The usage of MR technology in a learning environment is a research area that Hinske et al. [15] already covered in 2008 (not cited by Yusoff et al.). Therefore, they created an augmented toy reality in using an existing playground and adding virtual components to it. The existing playground they used is the game Kingdom of Knights to which they added audio which is played in fixed situations for example on positioning a certain figure in a certain area [14]. With this toy, the authors performed a wide-ranging evaluation to compare the game once with and once without digitally processed media. Even though, MR only takes place through auditory feedback, they still attain knowledge that they could generalize into guidelines for the design of a toy benefiting from the MR technology [15]. One of their main principles is that "Ideally, the focus of games and toys should be towards interactivity and user experience: Fun, immersion, engagement, physical and mental challenges, emotional stimuli, socializing, and creativity have been identified as primary goals in game design" [15, p. 79], and end up in design guidelines for

(a) (traditional) toys:
A toy should be fun, challenging, age-appropriate, physical, reliable, durable, and safe. It should give the possibility to play it in groups, and to support children’s fantasy. It should be easy to understand and use, and give direct feedback. Also it should enrich children’s playground.

(b) integrating pervasive computing technology into (traditional) toys:
The technology must add value to the toy. It should be reliable, durable, safe, and not striking, it should give direct feedback and adapt to the usual playground. The
purpose of the technology should be easy to understand, and it should not be the main essence of the toy.

(c) tangible user interfaces in the context of augmented toys:

The TUI has to make use of the whole three dimensional space, and map both, physical and virtual objects spatially. It should give a range of different possibilities to interact with it, direct feedback, and the possibility to use it in groups. Also, "The physical appearance should be consistent and meet the children’s perceptual abilities and mental models" [15, p. 84].

(d) educational toys:

Educational toys have to give certain tasks and direct feedback. They should make children curious, hence support their motivation to use it, and stimulate their fantasy.

The authors clearly state that these guidelines are mostly derived from their experience and are not complete nor do they meet all expectations on a mixed reality game in general. Moreover, they see their work as a step towards the creation of a general guide to design such a game.

I assume that the developer of the Augmented Knight’s Castle, the MR toy that the authors analyze in respect to how well it meets the guidelines, created the guidelines before they performed the wide-ranging evaluation with it. This is not clearly identifiable, because both papers were created in the same year, but they mentioned in the paper that they are planning "to conduct a large user study on how well these guidelines are actually met" [15, p. 85].

The evaluation took place in an elementary school with more than 100 participants. The outcomes of this user study were mostly positive. One of the most significant findings was that children react differently on the played sounds. Some of them found it fun and incorporated them into their play, others found them distracting, and some completely ignored them. Also, it is questionable how much children like or dislike the sounds after playing the game over an extended period of time. Plus, it should be evaluated how and if the way of playing changes and to which degree the creativity of the playing children is influenced. The authors further explain that the almost invisible character of the MR technology in the Augmented Knight’s Castle is a positive feature, and if it becomes too tedious, it can be switched off. On doing so, the game turns into
the traditional toy Kingdom of Knights. I can not completely agree on that argument. The technological aspect of the game has to create added value to the traditional game, otherwise it does not make sense to implement it. It is very likely that children playing with this kind of MR toy are going to exactly know after a certain time which figure says what when it is placed in a certain area, which speaks for the big learning effect that the game creates. However, this fact can also bring along boredom. So, as soon as the children got used to the novelty of the game concept, it is not clear if they still benefit from its technology. This is a fact that Hinske et al. almost disregarded. It has to be kept in mind, that players that have never or barely experienced MR games before are first very attracted to the novelty of the game, and are not really aware of the underlying game concept [39].

2.3 Technical Aspects

This section almost completely relies on the article in [6]. Because of its actuality, and its good summary of the technologies used in the discipline AR, almost no other references were necessary. At this point, we already know that my project is going to concentrate on an AR toy that uses a LEGO set as tangible interface, and that children are the target audience. In the following, I am going to cover the technological aspects for such an AR system while having children as user group in mind.

Hardware

On developing an AR system, the hardware devices need to be considered first. There are choices to make that influence the whole implementation process fundamentally. In this scope, I will discuss (1) display, (2) tracking technology, and (3) computer issues. Input devices are not covered in this section because it is already clear the the input is going to be realized through the LEGO set.

Display

Possible displays are HMDs, mobile and stationary displays. HMDs are very immersive, while being very expensive at the same time. As already described before (see section 2.2), they are not very appropriate for children because
• they do not suit their head very well;
• they are too heavy for children;
• safety and security issues are too grave;
• they cause eye-strain on wearing them for a certain time;
• collaboratory play can only be realized if each of the children wears an own HMD.

Google is working on HMDs that should be less heavy, and also tackle all the other problems that come with HMDs. According to the New York Times, Google Glass is still in the prototype state [4]. The advantage of this technology is that it will only consist of a small see through glass that does not block the complete view. It is said that the interaction happens through voice commands, and eye-tracking. Apparently, Google Glass will be released within the next coming years. Since Google Glass is not available yet and the concerns of HMDs that are currently available predominate the advantages, they are ineligible for my project.

Mobile displays are either smart-phones, PDAs, or tablet PCs. Both, smart-phones, and PDAs are nowadays mostly equipped with integrated camera, GPS and accelerometer which makes them very attractive for AR systems. Nevertheless, it needs to be taken into consideration that the display is too small to cope with 3D graphics [6]. This problem does not exist for tablet PCs, but the fact that mobile displays need to be handheld is one that does not make them adequate for my application. Children should play with the LEGO set itself, and not with the device. This fact was already stated by Hinske et al. [15] who declared that the technology should not be in the center of the game mechanics.

Stationary displays is mostly covered by screens of computers, laptops, or TVs. They are very useful in this project because they are fairly cost efficient, available in almost every household, and in the case of laptops often come with an integrated camera. An additional feature of choosing monitors over mobile devices and HMDs is that all computation can be handled on the computer itself.

3Here, a cave should be mentioned as well. Because of its cost and its impracticability – it is hard to imagine it in a simple household – it is not discussed in this scope.
Tracking Technologies

A number of tracking technologies exists that are, depending on in which environment they are used – indoors, or outdoors, or both – either suitable or not suitable at all. For instance, techniques like GPS, and accelerometers are completely out of question. GPS works the best outdoors, and is mostly used to locate somebody’s position. This technology is not made to track little building blocks, and can not distinguish between distances below one meter. Accelerometers are made to measure acceleration which will not be used in this project. The only ones left worth to discuss are RFID tags and optical markers.

RFID stands for radio frequency identification and are associated with the category sensor based tracking technology. RFID tags are small microchips whose position can be detected via antennas. They can be categorized into active and passive tags. The latter ones do not need any energy supply which the active ones do. Instead, they reflect the energy from the receiver [25].

Optical or fiducial markers are landmarks that are easy to recognize by an algorithm either through their shape (passive tags) or through the light that they emit (active tags) [25]. Latter ones are ineligible for my project because the LEGO block would need to be transparent for that 4. The question about using either RFID tags, or optical markers is not trivial. For RFID tags additional antennas need to be integrated into the toy. Both techniques imply an adhesion of the tag or the marker to the blocks. For the optical markers it is very presumable that they can not be detected when they are scaled to fit on a LEGO block. RFID tags only track the position steadily. Figuring out the rotation angle can be very tedious.

A third solution tackling this problem exists. This solution is not associated to tracking technology in the traditional sense. With the help of depth and color information and computer vision as technology, the LEGO blocks can be recognized and augmented as well. Since the release of Microsoft’s Kinect in 2010 a reasonably cheap version of a 3D camera is available [21]. However, a system like this can become very unstable for example in situations with varying light conditions. Also in detecting objects through color information the risk case needs to be considered in which for example the playing child wears a sweater in the same color as a LEGO block, or the color is repeated somewhere in the background. The system must then be smart enough to distinguish between

\[4\] Those blocks exist, but are not the common color in a LEGO set.
the LEGO block and other objects. This task is not impossible, because a check on the shape can be implemented that uses the depth information, but creating an algorithm like this probably ends up in a system that is computationally demanding.

In theory, all three techniques are eligible for my project with a similar number of advantages and disadvantages. For the RFID tags, the antennas need to be setup exactly to make use of their technology. This can be quite difficult, if imagining the system would be used in everybody’s household. So that I came to the conclusion to not use RFID tags as tracking technology.

I had some test runs with Microsoft’s Kinect before I started this project. From experience, I know that the algorithms for separating certain objects from everything else in the camera picture can become very tedious and unreliable under varying light circumstances. As soon as not only one, but probably a hundred or several hundred blocks need to be recognized, the computer will not be able to cope with this amount of data.

The same problem will occur on using fiducial markers as tracking technology. However, I figured that changing the mechanics of the play environment could be a solution. Mechanics are the rules and aims of a game [30]. Since we now figured out that it is almost impossible to track every little LEGO block, we should make use of the interplay of several blocks together. On creating the rule that patterns need to be build with several LEGO bricks that then can be recognized by the system, I can avoid the recognition of individual bricks.

Computer
For the computer calculating the AR, all that it needs is a high-performance CPU and an adequate RAM [6]. I will use a MacBook Pro Intel Core 2 Duo for the development that runs on 10.6.8 Snow Leopard. The processor speed is 2.53 GHz and it has 4 GB Memory. The application will only be build for computers running under Mac OS X. Here, the application can just be dragged and dropped into the application folder and run just like every other application.

Technical Background
The video tracking in an AR system is realized using computer vision. Hereby, the object needs to be tracked first, so that in a second step the image can be reconstructed.
The object that is tracked is as discussed earlier a fiducial marker that needs to be detected in the camera image. This can happen through either "feature detection, edge detection or other image processing methods to interpret the camera image" [6, p. 344]. For the tracking a connection between the two dimensional video image and the three dimensional real world coordinate system needs to be found. In order to do that, either a feature-based or a model-based algorithm is used. All these steps are necessary to calculate the camera pose. With this data, it is then possible to reconstruct the three-dimensionality of the real world.

Fiducial markers are very common in AR systems. They are in demand because their geometry is known before. Hence, they can easily be detected in a camera image in real time using a technique called *Simultaneous Localization And Mapping* (SLAM). There exist also other techniques that are used if the environment for the augmentation is already known before. I will not go into that because in my project the camera will not be in a fixed-position for sure.

**AR Interfaces**

The interaction between the child and the virtual environment happens through the LEGO blocks. This kind of interaction is called *Tangible AR interface* or TUIs. TUIs have already been covered in section 2.1.

**Programming Libraries**

The programming library that I found the most appropriate for my purposes is the *ARToolKit*. It was found by ARToolworks, Inc., Seattle, WA, USA in 1999. It is open source and works on Mac OS X as operating system. The library is written in C, and can also be used in C++. It underlies the GNU General Public License [11], and so is freely available for non-commercial purposes [19].

The ARToolKit supports the recognition and tracking of squared markers using computer vision. The position of such a marker is calculated from the camera image. The reason, why I found ARToolKit very useful for my project was that it comes with a profound documentation and lots of sample code which facilitates the use of the library a lot. Utilities are available to calibrate the camera and to create new patterns. So, for me
it is not only suitable because of the operating system on which it runs, but also because of its features. When it comes to creating own patterns and "training" the library, so that it is able to recognize that pattern, ARToolKit becomes quite handy.

Another advantage of ARToolKit is that it can be extended by OpenSceneGraph. The combination of both is called osgART [2]. This enables the developer to create an AR application with a fast rendering of complex 3D model in real time. Without this extension the 3D models would need to be created in OpenGL, which can be very tedious and time consuming. Instead, a 3D modeling software such as Autodesk 3DS Max can be used and then read by the library.

However, even though osgART would facilitate my work a lot, I was not able to use it in the further course of my project. I was not able to install the library on my machine, so that I needed to use VRML to create 3-dimensional vector graphics. A big advantage is that ARToolKit is equipped with a sample application that reads VRML files and interprets them.

2.4 Summary

In the 1950’s the LEGO Group started the fabrication and commercialization of their product. Their concept was based on the idea learning trough play for children of all ages [34]. Using a traditional LEGO set and adding virtual interactivity to it is challenging. In this section, I have covered many aspects that need to be considered. In the beginning, some taxonomies are defined that are important to the project such as mixed and augmented reality, tangible user interfaces, and child computer interaction. Also theories are explained of the components of an effective learning and important criteria of play environments. In my work, I will try to connect the theories of leaning to a play environment in a TUI. It is my aim to create a toy that is fun to play, and at the same time teaches children constructional skills and strategical thinking.

Furthermore, I covered related work and assessed it critically. Ucelli et al. [36] used a traditional book and added an augmented chameleon to it that explains children the concept of colors. I could map their work on some of the main factors that help children to learn which were active engagement in guessing the color, frequent interaction and feedback in seeing the result virtually, and the context to the real world reflected by the chameleon. In my opinion, they disregarded the collaborative aspect. Children can only
use the book if each of them has an own HMD which involves a great deal of expenses. Yusoff et al. [40] had a better solution to apply the aspects of a successful learning to their system in simply using a computer display instead of HMD. They invented a mixed reality book similar to the Book of Colors, but clearly with the goal that it can find its way into the school routine. They tried to find some guidelines for creating an educational tool that uses the AR technology and could not find a solution. Hereby, they disregarded the work of Hinske et al. [15] that has been created two years earlier. The two projects are different in the selected target user group. While Yusoff et al. deal with students in secondary school, Hinske et al. create guidelines for toys used by children in elementary school. I would also identify my main target group with the latter one which is why I can more relate to their work. I can benefit from their guidelines inasmuch as I can apply some of their findings to my project. In summary, they state that a traditional toy with integrated AR technology and TUI needs to be reliable, durable, and safe, give direct feedback, and the possibility to use it in a group. The technology should add value to the toy, map the physical and virtual object spatially, and should not be the center of the toy. Furthermore, they expect that educational toys have to give certain tasks and direct feedback. They should make children curios, and hence support their motivation to use it and stimulate their fantasy. According to the authors, the guidelines are not complete and further research is necessary to generalize them. Even though I criticized some of their concepts severely, I think this is one of the works that I can most benefit from. One must not forget in the evaluation analysis that children probably more respond to the novelty of the system than to the underlying game concept, hence it is possible that the initial enthusiasm does not last long [39].

Another topic I dealt with was mobile AR systems. I doubted the use of mobile devices when applied to my work. Those doubts intensified later on as I inspected the technological aspects for my project. In the section that deals with the technological aspects, I figured that a stationary display such as a monitor would suit my project the most. RFID tags are not suited for the project because of the setup of antennas, so that I decided on fiducial markers. For this, the ARToolKit library will be used to realize the recognition of the markers.

For the future work, I will create a concept of a system that incorporates both, educational concepts, and criterions of play environments combined with mixed reality. I will create a prototype that is going to be evaluated to make further improvements.
3 Concept

Considering the literature review, a concept for an AR application shall be created that uses LEGO bricks as input device. This application should connect theories of the way children learn effectively with a play environment in a TUI. While working with and for children, legal professional, and ethical issues need to be regarded, which is why I am going to start this chapter in explaining them briefly.

3.1 Professional, Legal, and Ethical Issues

The growth of new media technology is often associated with a great benefit in the social and educational environment. However, at the same time, concerns in terms of bad influences that the new technology brings along increased. Most studies focus on the interaction of children with the internet. The anxieties and worries in this area are obvious: children having access to inappropriate content that includes sex and violence, see for example [38]. Having a closer look at the issues associated with children using technology reveals even more concerns: addiction, social isolation, loneliness and depression are terms that are often used in this context. Also, studies have shown that children have problems to distinguish between reality and fantasy [38]. Furthermore, health and security issues should not go unmentioned. It is said, that computer use can harm children’s eyesight and visual development. Ergonomic problems arise if children do not sit correctly in front of the computer [8].

Next to this, legal issues should be covered in this context as well. On conducting user studies with children, there are lots of requirements that first need to be fulfilled to do so. First of all, in Scotland there exist a administrative requirement that is needed to work with children, the Protection of Children (Scotland) Act from 2003 [32]. This is basically a list that keeps record of people who are disqualified from working with children. The aim of this act is the improvement of safeguard for children to protect
them from harm. For this, a disclosure is necessary. The disclosure Scotland is a police background check and contains information of criminal records of this person [31].

The user study in my project will be conducted as a voluntary informed consent, which means that the participants know about what kind of activity they will be doing. It is important that children and parents don’t think that they will benefit or learn from the application, when they participate. When working with children, permission of an adult responsible for them is necessary. It must be clear to the participants that they can withdraw the consent without giving a reason. In every part of this dissertation, the anonymity of the participants must be guaranteed unless they request otherwise [7].

All in all, technology can bring great benefits in being a fundamental educational and learning assistance, but at the same time many restrictions and requirements need to be considered. For this project, I will not only have to face ethical, and health and safety issues, but also issues that are regulated by law for the protection of children need to be kept in mind.

3.2 Concept Creation

When I started creating a concept for this project, I had to think about my childhood, and especially what I enjoyed about playing with LEGO. The concept of LEGO is to leave children space for creativity, they can build whatever they want in whichever way they want. I wanted to keep that idea, and expend it with technology. Augmented reality should be used to create an additional value to the already popular toy.

In section 2.3 I figured that RFID tags are not useful for the system because the rotation angle can not properly be recognized. Using computer vision would be too fuzzy, especially because the environment in which the system will be used is not fixed to a certain location (different rooms cause different lighting). The only technology left is using fiducial markers which will be hard to recognize if they are too small. So, applying fiducial markers on each of the building bricks would be inappropriate for the system. A workaround for this problem would be to create game mechanics, so that the user has to rebuild a certain pattern with the LEGO bricks which then can be recognized by the system.

Considering Hinske et al.’s guidelines for the design augmented toy environments, the requirements for a traditional toy are already fulfilled by LEGO itself. If used for educa-
tional purposes it should give certain tasks, on which the system gives direct feedback. As already mentioned the task in my play environment should be the reconstruction of certain patterns, which then can be recognized by the system, which then gives a direct feedback on whether they’ve build the pattern correctly. The question then was it, in which extent technology adds value to the toy, and why children would need it in their play environment. I came up with the idea to augment wall parts on the patterns, which then can be set together however children would want to. Two wall can for example be set next to each other or on top of each other, or can create a corner together. Hinske et al created the guideline that adding technology shall also add value to it, bit at the same time they have stated that if the audio that is added to their toy becomes too tedious, it can easily be turned off. In my concept turning the system off, means having the traditional LEGO bricks. This would lead children to build any random objects, rather than the given patterns. On the other hand, children could also build patterns from memory while the system is turned off. This would make sense, if they are in a place where they have no access to a computer. As soon as they get back to a computer and the system again, they can check whether they have build every pattern correctly. This would also speak for the learning effect. This is a similar theory that Uticelli et al already enunciated, when they expected children to guess beforehand which color the chameleon is going to turn into.

Other than that, my concept meets important principle of the components of an effective learning, which is the connection to real world contexts. The representation of LEGO bricks as a virtual real looking building would teach children constructional skills. And since they are able to set the patterns together in whichever way they want, they are still able to be in control over their environment which is part of the child-persona created by Antle.

Another issue that was in the center of the literature review was the interaction of the user. The system needs to give feedback on every action of the player, and it shall be possible to use the application in groups which is also an important point to keep children motivated. A collaborative use of the application is definitely possible which was already considered when I decided of the hardware requirements. Nevertheless, a direct feedback is not completely given. The feedback is frequent, because the system can only react on the pattern, when it was finished. Also a recognition can not take place if the user did a mistake, and so it would not be able to give any feedback.
The purpose of the user evaluation is to “ensure functionality and usability from a user perspective” [5]. In this chapter, I will first cover the requirements analysis to give a deep understanding of the purposes of the application. This will also make the functionalities clearer that will be tested in the first user study. The first user study then will be covered in section 4.2.

4.1 Requirements Analysis

It is the goal to create a prototype of a system with an educational background. The user is suppose to interact with the system through LEGO blocks. The purpose of the system is the enhancement of constructional and creative thinking in abstracting from a 2-dimensional to a 3-dimensional representation. This chapter discovers the requirements of the system including user characteristics, and software requirements specification.

User Characteristics

The target user group is represented by children in elementary school. Their age ranges from 5 to 10 years. In this stage children learn new skills, and transfer them to the real world. It is the stage in which they require less from parental care, and start being more independent [13]. Children of a younger age can have difficulties to distinguish between the real and virtual world, and so are not applicable.

LEGO sets always come with information about the recommended age range on it (see [35]). Generally, it can be said that children

- $\geq 0$ are recommended to play with LEGO DUPLO,
• ≥ 5 are recommended to play with LEGO Bricks, City, and Cars,
• ≥ 7 are recommended to play with LEGO Creator, and
• ≥ 12 are recommended to play with LEGO Architecture.

LEGO DUPLO are LEGO bricks that are bigger than the usual ones, so that children
under 5 are not in risk to swallow them. LEGO Architecture is a LEGO set to create
more complex buildings (i.e. the White House). In this work, we will more focus on the
groups in between. Children between 5 and 10 are recommended to play free without
any instructions in order to create their own buildings or cars, or to rebuild simple shapes
such as a space ship. This is exactly what this project aims to do: The user has to build
certain shapes and to set them together to create a building of their imagination.

Concluding the target user group from the age range given by the traditional toy was
also a strategy that Hinske et al. had for their concept. It ensures that the system is
age-appropriate.

Software Requirements Specification

The functions that the system will perform is augmenting walls of a building on top of
real world walls built of LEGO blocks. The buildings can be categorized by a certain
theme to achieve different kind of buildings, i.e. castles, houses, etc. This mode needs
to be selected first. The patterns made of LEGO bricks can be recognized by the system
as soon as it is completed. The user must know about the shape of the pattern that can
be recognized by the system and about how the virtual representation would look like.
As soon as a pattern is completed, the system maps a virtual wall on top of the pattern
in the camera image. In putting several patterns together, the user can create a complete
virtual building.

External interface requirements

The system needs to be started by a mouse. The system itself provides an interface
that does not require any other input device than the LEGO bricks and a camera. The
monitor is the only output device.
Use Cases

Figure 4.1 shows a use case diagram for the interaction between user, input interface, and the system.

Figure 4.1: Use Case Diagram for the Basic Functionalities

In **step 1**, the user starts the program. The system will show different modes to choose from, i.e. castle, house, etc.
The mode is selectable with a mouse (step 2). The system will show the camera image and also different patterns and how they will look like in a virtual representation.

In step 3 the user starts to rebuild a pattern with LEGO bricks. As long as the pattern is not complete, the system will not recognize the pattern and will not show the virtual representation.

Step 4 shows a complete pattern made of LEGO bricks. The system will still display the camera image, but with a virtual representation mapped on top of it. The representation is a wall of a building with certain characteristics, for example a window or a door.

In step 5 the user starts to create a new pattern, but has not finish it yet. Therefore, the system will still display the same picture as in step 4.

Step 6 shows that several different patterns can be recognized. Patterns that are the same are virtually represented by walls with the same characteristics. Patterns that are different have different virtual characteristics.

**Functional Requirements**

The use cases already cover most of the functional requirements of the system. Here, they will be listed again to generalize them.

- The system shall be able to augment building walls in different themes.
- The interface mainly consists of the camera image.
- The interface shall give information about the shape of patterns that can be recognized by the system.
- The user shall know which kind of pattern will be represented by which virtual model.
- The system shall be able to recognize several patterns at the same time.
- Two brick walls with the same pattern shall be represented by virtual wall with the same characteristics.
- Brick walls with different patterns shall be represented by virtual walls with different characteristics.
• It is desirable that the pattern that is represented by a wall with a window has similar attributes (kind of looks like a wall with a window), and that the pattern that is represented by a wall with a door also has similar attributes (looks like a wall with a door).

• A help section shall always be available.

4.2 First User Study

In this scope user evaluation is not used to identify software bugs, it rather helps to conform whether the application „satisfies its intended use and user needs“ [17]. This method is also called software verification and validation according to IEEE Std 1012-2004. The first part of this section explains which of the requirements are realized in my application. The second part covers the task that the participants have to accomplish, and gives an overview of the user study. And the third summarizes the results.

Practical Implementation

The minimum of the requirements were implemented for the first user study. For this, the walls of two different buildings, a brick building and a caslte, were designed and realized in VRML. VRML stands for Virtual Reality Modeling Language and was originally created for 3D graphics on the internet. The models are represented in vector graphics, and typically saved in *.wrl files [37]. The *.wrl files can be read and interpreted by the system that I created and can then be rendered in real time. Figure 4.2 and 4.3 show the walls that have been coded for the AR system.

Each building has a neat wall, a wall with a window, and a wall with a door available. The castle additionally has a battlement (Figure 4.3 right).
Each of these walls needs to be assigned to a certain physical wall that is build in LEGO (note, that if you would print the pattern on a paper the system would still recognize it; which means that the recognition takes place independent from the material). The patterns that can be recognized by the ARToolKit need to meet certain requirements [20]:

- The pattern needs to be squared.
- The pattern needs to have a black boarder.
- The image inside the boarder needs to be 50% smaller and centered in the middle.
- The image can be colored.
- The image should not extend too far into the boarder.
• The image must be unambiguous, so that it is always clear which side is top, bottom, left and right.

Figure 4.4 shows a sample pattern. With LEGO bricks it is difficult to create a squared pattern\(^1\). But the ARToolKit also works fairly well with rectangles that are almost square sized. Figure 4.5 and 4.6 show the patterns that I came up with.

\(^1\)I tried so, and ended up with a pattern sized 19.2x19.2 cm as the smallest. If each pattern was that big, it would take too much space and also too many bricks as well.
The order is the same as before for the VRML models. Each pattern that is associated with a wall can be found in the same place. So that Figure 4.5 shows the pattern associated with the neat wall of the brick building on the left, the one next to it stands for the window and the one on the right for the door. The same accounts for Figure 4.6. The idea was it, to create patterns with similar attributes as the 3D models that they represent. Therefore, I abstracted the shapes and colors of the walls. For example the color of the bricks is red, so that the main color for the brick building patterns is red as well. The window is small and blue, and the door is a bit larger, and brown. The white boarder around the patterns guarantees a better recognition but is indeed ignored by the system. It becomes necessary as soon as two patterns are placed next to each other, so that they can clearly be distinguished from each other. The image shown in Figure 4.6 was also presented to the children in the first user study. Small vertical lines indicate the height of each pattern and the knobs on top showed the width.

For the first user study, the system is able to recognize the given patterns. It is possible to select a certain mode at the startup (at least in the developer version). This is regulated through Xcode’s console.

On running the application, only the camera image is shown. Depending on the selected mode (brick building or castle), the corresponding patterns can be recognized. So for the brick building three and for the castle four patterns can be distinguished from each other. The system is able to augment the same 3D model onto the same pattern. If for example two of the window patterns are shown towards the camera, the application will show two windows accordingly. The system is able to recognize up to eight patterns at the same time.
User Test

The user test was performed in week 9 of this project with two participants aged 8 and 12. They were using the system together with one parent always present. The session took one hour, in which the participants were asked to perform three tasks, plus a little bit of free play in the end. The user study was conducted as an informed consent. The children were told that they will have to rebuild patterns with LEGO bricks on which the application can react. Pictures of the patterns that they were given can be found in Appendix B. On doing so, they will be recorded from three different angles, which were two video recorders plus a screen recording. Each of the cameras also included an audio recording. While they were using the application, they were asked questions about their experience with LEGO and computer games, and also about their experience with the system itself. The transcript of the session can be found in Appendix A.

Task 1

The participants were asked to reconstruct the second left pattern of Figure 4.6. The pattern was presented to them printed on paper and in full-size (which means exactly the same measurements like the actual pattern).

The participants were able to accomplish the first task in less than five minutes. While they were playing, they were asked about their experience with LEGO. One participant stated that he really enjoys playing with LEGO, and does it every day. The other one enjoys it less.

The participants proceeded in a very structured way from the beginning. They counted the knobs on top of the pattern first to calculate the width of it. Then they counted how high the pattern will have to be. Obviously the number of knobs and the lines that indicate the height of a pattern were very helpful. They communicated the small mistakes they’ve made fairly well. Those mistakes were mostly related to the height of the pattern which is good because it shows that the task in building the pattern is challenging but not too hard to accomplish. One of the participants talked a bit more about what is wrong, and what they had to change to make it right; The other one rather changed it without really communicating.

As soon as the participants hold the pattern towards the camera, they became very euphoric, but were also a little bit confused about what is happening. For example the
participant holding the pattern sees the augmentation of the virtual wall on the screen and turns around the actual pattern that he was holding to see whether there was something on it. The other participant asked whether the augmentation was what the system was meant to be.

**Task 2**

The participants were now given the same pattern with the door pattern next to it. This time the image was only shown in a digital version on the computer screen. The purpose was to test, whether it would make a difference for them to build the pattern from a physical version or a digital. During the second task the children were asked about their relationship to computer games. Only the older one answered that he would play them sometimes. He stated that he even liked playing LEGO themed games on the computer.

They have finished the task in round about the same time as the first pattern. It is recognizable that they talk about the mistakes they’ve made more than in the first task. This might also be caused because they were a little bit more shy in the beginning. Also in the first task the mistake they have made was not that grave.

As soon as they finished they directly wanted to see what the pattern is going to augment on the second pattern. Their reaction was very positive:

“

**Interviewer:** How was that for you so far?

**User A:** It was fun […].

**User B:** It’s cool because it’s just LEGO but then it projects some picture.

”

On asking the participants whether they found it easier to build the pattern from the printed version or from the digital, they both answered that it did not much matter to them.

**Task 3**

In task 3, the participants had to reconstruct the roof pattern. It is much smaller and so it took them only two minutes to build it. They already tried guessing what might be augmented on top of it as soon as they saw the pattern on paper.
This time, it can be noticed that the social aspect is influenced in a positive way. The older participant was much faster in finding suitable blocks. After seeing that the other participant was slower than him, he told her what she can do to help.

As soon as the participants finished the roof pattern and hold it towards the camera, they could not quite identify it as a parapet. They recognized it as "another bit of wall".

**Free Play**

In the free play, I wanted to see whether the participants come up with some own ideas of how to put the patterns together. It was meant to be a test to check whether they got the idea that they can put the pattern just anywhere they want and the system would still be able to recognize it. Unfortunately we ran short on black and white LEGO bricks, so that they were barely not able to build more patterns.

The participants were very reserved, and wouldn’t just tear apart what they’ve just created. So, I told them to take one of the patterns and stick it to another place. They liked the fact that the system was able to react on it, and they started planning building more parapets to stick them along the wall patterns. They stated that if they had more time and bricks they would like to "build a whole house. [...] An then you can [...] build other houses around".

One of the participant started a modified version of one of the patterns, which was bigger in width and smaller in height. She was curious what was going to happen if she held it in front of the camera.

The fact that we ran short on bricks influenced the last session a lot. When they didn’t have enough bricks to build another pattern, they started building random objects and were wondering what was going to happen by holding them towards the camera.

**Summary**

"**User A:** I enjoy playing with LEGO anyway. Any type of LEGO. I think it (the application) just adds more fun!"

Overall, the reaction of the participants was very positive. It was not hard for them to
reconstruct the patterns that they were given, neither from the paper print nor from the
digital version shown on screen. This finding is important when it comes to creating the
interface. I was not sure whether they might have problems reconstructing the pattern
when they only have a digital version available. This finding most probably means that it
would be okay if the application comes with small patterns displayed next to the camera
image which indicate which ones would be recognized by the system. An additional
handbook containing the patterns in real size would of course be useful as well.

From this session it can be said that the social interaction was influenced in a positive
way. The users were communicating the mistakes they have made a bit less at the
beginning of the session, but then more after some time when they lost their shyness. In
the last session they created a plan together on building several parapets to stick along
the wall patterns. The older participant (the one that likes to play with LEGO every
day) quickly saw that he was much faster on putting the pattern together and often let
the other participant some time to find a suitable brick.

The participants were curious what would happen if they build modified versions of
the patterns or just completely other objects. They didn’t say it explicitly, but it was
very obvious that they wished that every single brick would have a representation in the
virtual world. They probably would have liked to stick some bricks together and see
what the system would interpret in that.

The fact that we did not have enough bricks in the end, was very unfortunate and it
held the participants from realizing their own ideas. As already mentioned in the Free
Play section, the participants would have liked to build a complete house or a block of
houses if they had enough time and bricks. This fact led me come to the idea to extend
the application, so that users would still be able to create a complete complex building
or even a city, even if they don’t have enough bricks. The extended version of the system
will be explained in the next chapter.

**How the study should have been conducted**

The participants were very keen on accomplishing the tasks, and it can be concluded
from the session that the system actively engaged the children. The usage in groups
was not a problem at all for the two participants. Nevertheless, it needs to be said, that
only very little can be drawn from the study and nothing can be generalized. For a
generalization of my results, more participants would be needed. A study similar to the one done by Hinske et al. with 100 participants would be impossible for me to conduct considering the time. If I had more time, I would have taken nine participants, with six of them using the system in groups of two. All age groups shall be represented and the distribution of different genders shall be balanced.

For a user centered design, it would have been necessary to include children in the design process from the very beginning. I didn’t do that on purpose, because the AR technology would have been completely new to probably most of the children, so that it would have been hard for them to come up with their own ideas. It would have been hard to create paper prototypes, because it takes a lot of imagination to understand, that an object in the real world can be turned into something completely different, and the children would probably be more confused than helping. I can hardly degree on Antle’s theory on seeing children as equalized design partner. This might be possible when children are already familiar with the technology, but in my case they are most probably not. Also on asking the two participants whether they would have an idea for the system, that uses the same technology, they came up with very little. I would have liked them to draw a picture of what they would use the technology for, but since I got very little reaction on what they can imagine this would have been inappropriate.

Of course similar as described by Hinske et al. a long time study would have been needed to see, whether children enjoy the system that much that they still want to play with it after more than an afternoon. With a long time study could be detected whether the behavior of their play changes in general and whether their creativity might be influenced. A long time study would also be necessary to see how children interact with the system when they got used to the novelty of the technology. As Yusoff et al. already stated the added AR would draw a lot of attention, and children would always be very fascinated by the technology in the beginning.
5 Evaluation: Second Iteration

Based on the finding of the first user evaluation, the system was extended. The participants stated that they would like to build a whole building or several buildings next to each other. This statement was based on the fact that we didn’t have enough time and bricks. Therefore, it was clear, that the application needs a feature to realize this request. This feature was going to be a copy function, with which it should be possible to copy walls of a building and to have them available virtually. The extension will be explained in detail in the next section on which a second evaluation follows.

5.1 Extended Requirements Analysis

For the copy function a concept needed to be found. Since I wanted to pull this project into the direction of finding an alternative way of using the application in a tangible way, I didn’t want to create a feature which makes the user go back to the traditional way of using the computer, meaning giving input only with a mouse-/keyboard-interface. Nevertheless, for an easier interaction, the keyboard was necessary to make a copy. Also, I added a base pattern (Figure 5.1) to which the walls will be attached when they are copied.

To explain the process of making a copy, I created some additional use cases and functional requirements.
Use Cases

The copy of the wall will be attached to a base pattern and moves around in relation to the base pattern. The base pattern is attached to a 32x32 knobs base plate from the manufacturer LEGO. With the base plate the user is able to stick patterns onto it, which facilitates the interaction of handling camera, wall patterns, base pattern and keyboard at the same time.

Figure 5.2: Use Case Diagram for the Copy Function
Figure 5.2 shows a use case diagram for the interaction to copy a virtual wall.

In step 1, the user holds one wall pattern that sticks on the base plate towards the camera. Both, the wall pattern and the base pattern need to be "seen" by the camera. The system will recognize both patterns, and gives feedback in augmenting the virtual wall on the selected pattern and a green borderer around the base pattern. If both augmentations happen at the same time, the system is ready to make a copy.

Step 2 shows the copy process. Here, both patterns are recognized by the system and the user has to press the space bar¹ to tell the system that s/he would like to make a copy of the virtual wall.

Step 3 basically shows that as soon as the copy was made and the wall pattern was removed from the base plate, the system will display a copy of the virtual wall. The user only has to show the base pattern towards the camera. When moving the base pattern around, the copied wall moves in the same manner. It is always augmented in relation to the base pattern.

In step 4, the user copies several patterns at the same time. For this each pattern that the user wants to copy, needs to be recognized by the system in addition to the base pattern. When the user presses the space bar, the system shall be able to make a copy of all virtual walls that are shown towards the camera, and to put them to the right place.

In step 5, the system has already copied some walls and the user wished to add more to it. For this the base pattern is recognized by the system and the already copied virtual walls are augmented in relation to the base pattern. This should help the user to have an orientation on where to put the wall pattern s/he would like to add. If the user has stuck the pattern to the desired spot and the system recognizes the base pattern and the wall pattern at the same time, the user can press the space bar, and the wall will be added. The virtual walls that have already been copied before are then still available and displayed in the same place as before.

Step 6 shows the "undo" function. For this, the user simply has to press the [z] key². It doesn’t matter which patterns are in the camera image. This function even works when no pattern at all is recognized by the system.

¹I chose the space bar because it is the biggest key on the key board. When the user has to handle all patterns and the camera, and watch the system at the same time, it makes sense to choose the biggest key.

²The [z] key derived from the command [CTRL]+Z which in many application is used to undo the last action.
Functional Requirements

Again, the functional requirements cover a list of operations that shall be performed by the system.

- The system shall give feedback, when it is ready to make a copy.
- It shall be possible to make one or more copies at the same time.
- The copies are made, when the system recognizes the base pattern and at least one other pattern at the same time and the user presses the space bar.
- The copies of the virtual walls that were made are in the virtual world displayed in the same spot as the patterns in the real world.
- On moving the base pattern in front of the camera, the copies of the virtual building move in the same manner, as if the according wall pattern was still stuck on the plate.
- The user shall be able to add copies of virtual walls to the ones that s/he already made.
- It shall be possible to copy a finite number of virtual walls, caused by the computer capacity.
- The system shall be able to "lock" the position of the patterns, which means that when a copy was made, walls are arranged so that they adjoin to each other. Also if two walls build a corner the copies are arranged so that the corner is always 90°, since it is only possible to create 90° corners with the traditional LEGO bricks.
- The system shall give an opportunity to undo the last action with the \[ z \] key.
- It shall be possible to undo several times.
- It would be desirable to remove selected copies of virtual walls.
- It would be desirable to remove all copies at the same time.
- It would be desirable to save the virtual building or to export it.
5.2 Second User Study

For the second user study most of the requirements stated before were realized. I will again, explain the operations that were actually implemented first, and then go into the details of the second user study. In the end of this section the findings will be summarized.

Practical Implementation

The feedback that is given by the system when it is ready to copy is realized by a green rectangle around the base pattern, plus the augmentation of the wall pattern that is recognized by the system. This means if the user shows a pattern to the camera, that for any reason (i.e. light circumstances, covered by other objects) is not recognized by the system, it will not create a copy of that specific virtual wall. So, each pattern that is recognized by the system will get copied and represented in the virtual world at the same place as the one in the real world as soon as the user presses the space bar. Figure 5.3 shows a screenshot of the application during the execution of making a copy. In this picture, the copy was just made, and the pattern was removed from the base plate and is held by the user. The base pattern is surrounded by a green border which means that it is recognized by the system. Therefore, the copy of the virtual wall is shown in the camera image as well. It is identifiable that the wall pattern was placed right behind the marker while it was copied and before it was removed from the plate. If the user would move the plate, the copy of the wall would move accordingly, as if the wall pattern was still stuck on the plate.

The calculation of where the pattern is placed in the real world works the best, when the wall pattern is held so that it is aligned with the camera and appears squared in the
camera image. Also, holding the pattern as close as possible to the camera, makes the calculation more precise.

Adding more copies to the building is possible. The copies are always made in relation to the base pattern and will not change their position towards the base pattern when walls are added. To this point, it is possible to create up to eight copies.

The locking function is mostly implemented. The copies of the walls are always placed either in the same angle or in a $90^\circ / 180^\circ / 270^\circ$ angle. Calculating the position of the patterns in relation to each other, does not work steady enough. The reason for that is that the depth coordinate can not properly be calculated from the 2 dimensional camera image.

The system is able to undo the last action when the user presses the [z] key. Pressing the [z] key multiple times, undoes earlier actions than the last one.

**User Test**

The implementation of the copy function unfortunately had a lot of bugs (see chapter 7 for that). Those bugs would just confuse children in a user test, and the only conclusion that can be drawn from it would be that the code needs improvements.

To still collect children’s reaction on the copy function despite the fact that it is not working properly, a video\(^3\) has been created that explains the process. It was originally the plan to collect responses through the comment function on YouTube. We E-mailed the link around, but could not gather any reaction. Because of the fact that not much time was left, I walked around in the city center to find some participants. I showed this video to four children and collected wanted to record their reactions. The sessions were organized as a field study with different focus groups.

The first session included a boy that was too young for my focussed age group. This session took place in a park on a sunny day. I did not realize how difficult it is to watch a video on a computer screen outside. Additionally, I figured that a session with three children at the same time is very hard to handle. So that I went to a coffee shop next, that specializes on children as their guests. I asked the owner for permission, and she told me that the children that come in are mostly under 5 years old. Nevertheless, I was able to find a participant for a short session.

\(^3\)http://www.youtube.com/watch?v=lU2Qm_9muUE
After watching the video, the children always requested to play for a short time with it. This shows that children are quite interested in a play environment like this in general. I was prepared for that request and brought the LEGO bricks and the base plate with the base pattern taped on it.

The reaction of the augmented walls was very similar to the ones in the first user study. All of the children liked the interactivity of the system. Their enthusiasm for the application can as already covered also be interpreted as the curiosity that was drawn by the novelty of the system. The augmentation appears to them kind of like a magic trick, so that they show a lot of interest in the application at least in the beginning.

Test 1

The first test was the one with three children. In this study the children were aged 3.5, almost 6 and almost 9. Two of them were boys, one of them was a girl. The children were not shy at all from the very beginning. They responded very positive on the video and immediately asked whether they can play with it. The oldest one of the children understood the concept of the application very quickly and already summarized the functionality while watching the video.

As soon as they started playing, they were very excited about the augmentation. Only the oldest one understood the concept of creating a house in whichever way they want. For the copy process she had a lot of questions, but when her younger siblings started fighting over the LEGO bricks, she quickly lost her interest.

Test 2

For the second test, a 10 year old boy was conducted. He stated that he likes playing with LEGO a lot. He was very quiet and reserved over the whole time, but in the end was able to use the copy function successfully. After watching the demo video, he was able to repeat most of the functions in his own words. When I gave him the LEGO and the base, the LEGO bricks were already stuck together to build 3 complete patterns. This made it easier to test only the reaction of the children on the copy function only. The participant was very reserved in his reaction on the augmentation. He understood the concept of setting the patterns together very well, and set the patterns up in many different ways. For the copy process, the participant had more questions. But after an
explanation, he was able to create a copy.

**Results**

Only very little can be drawn from the two tests, and nothing can be said in general. The responses on the video were always very positive. The participants were got very curios, and wanted to see by themselves how the system works. This either means that the video is well made, or it can be interpreted that an AR application can draw a lot of attention.

The child in the first test that showed a good understanding of what was shown in the video lost her interest too quickly when she used the application. The boy in the second test was too reserved so that almost no data could be collected from him. The 5 year old participant in test 1 was not focussed on the application at all.

It might be possible that the application is too challenging for younger children. Especially the copy process seems to be very abstract for them, and also very tedious to handle. However, as Antle already stated, each child develops differently so that its developmental ability can not be identified by the age.

**How the study should have been conducted**

To draw conclusions on the age range, more participants would be needed. At least two in every age group and preferably of different gender, would be appropriate for a user test.

For me, it was hard to get any reactions on the copy function of the application, mainly because it didn’t work properly. Disregarding the fact, that the application had too many bugs, a longer session would be needed to see whether children find this function helpful. The short sessions that I conducted also didn’t tell much about the natural behavior of the children. To see whether children really enjoy the system, an observation in their natural environment would be necessary. For that, the system needs to be made available to the children at their home. It would be appropriate to let the children experience the play environment for at least a week. In that time slot, children would also get used to the novelty of the technology and it would be possible to see whether children really like the system, and are not only curios to try it out.
It seemed to me that children would not want to play with LEGO and having certain rules and guidelines on how to structure the bricks. They rather enjoy it when they can create anything that comes into their head. To check whether a child really understands the idea of setting up the different walls in whichever way it wants, it needs the possibility to play more often with it and to get the inspiration of what could be possible.
6 Developer Guide

This chapter is divided into Basis and Extension. In Basis, I will explain the basic recognition of the markers. This is the part of the implementation that was tested in the first user study. The application was then extended by a copy function, which will be explored in detail in Section 6.2. Not covered in this chapter is the implementation of the three dimensional walls, which I found redundant. The code of my application can be found in Appendix C.

6.1 Basis

ARToolKit comes with a lot of sample code. I was able to use most of the code from simpleVRML.c. It is able to interpret VRML files and to display them in an OpenGL window. The recognition of multiple patterns is already realized. It needs a text document that tells which pattern should be assigned to which *.wrl file and where it can find the files. Thus the application is able to differentiate between different patterns and displays the augmented 3D model even if multiple patterns are shown at the same time. However, the application is not able to augment 3D models when several of the same pattern are shown to the camera. It will only create an augmentation on the one, that is seen as the most "confident". Each marker that is recorded by the camera is assigned a confidence factor. The one with the highest confidence factor will be taken for the augmentation. Since I want for my application that every marker will be represented by its 3D model even if two of the same are shown at the same time, I had to change the code in the Idle() function. This was realized in a two dimensional array. Since the objects were already organized in an array, it now makes sense to turn it into a two dimensional array, that has the size of the number of different objects times the number of how many of the same patterns can be recognized at the same time. I have set the number of same patterns that can be interpreted at the same to 9 for each pattern. Since the code for the
recognition of several of the same patterns is pretty similar to the code that was given by the example code, I will not go too deep into the alterations that I performed. Just a small note, if every recognized marker is represented by a virtual building, the system often displays 3D models on objects that are indeed no markers. For this, the confidence factor helped to constrain the number of augmented walls. I considered 0.75 in a range of 0 to 1 to be a good factor. So that every recognized pattern with a recognition rate higher that 0.75 will be interpreted as a virtual wall.

6.2 Extension

In this chapter, I rather want to concentrate on the implemented copy function because it took much more effort to realize it than the interpretation of several of the same markers.

To make a start, we need a function that is called, when the space button is pressed. I named this function `createCopy()`. When the function is called it grabs the video image, and checks how many marker it can find in this image. If there is none, it will not execute the rest of the function. If there is at least one, all markers with a confidence factor higher than 0.75 will be stored in an array. Also it will be checked whether the base pattern is available in the image. If more than one base pattern is present, only the one with the highest confidence will be taken into consideration. Nevertheless, the base should not be the only marker that is present in the camera image. If all those requirements apply, `arGetTransMat()` is called. This is a function given by the ARToolKit that takes the marker information and transforms it into the object information. This basically means that information such as size, position, distance, etc. of the marker is calculated, so that the object (in other words the virtual model) can be displayed relatively.

We need to store the information of the wall pattern in positional relation to the base pattern. The reason for this is that in each frame after the copy process, the position of the copied wall needs to be calculated depending on where the base pattern was found in the camera image. In summary, this means:

1. Get the position of the base and the wall at the time point where the user creates the copy.
2. Transform the base to the origin of the OpenGL coordinate system.
3. Move the virtual wall accordingly.

4. Each following frame, find the base marker in the camera image.

5. Find the transformation matrix, that moves the base from the origin to the position 
in 4.

6. Transform the virtual wall accordingly.

To make this process more clear, I created an illustration that can be found in Figure 
6.1.

Since the first three steps have to happen for each copy, it is useful not to calculate 
this transformation in every frame. Instead, the position of the virtual wall should be 
stored in relation to the base pattern when it is notionally moved to the origin. This
does not only save time for the calculation it also saves a lot of capacity, because we don’t need to remember the position of the base every time the user made a copy. It will always be represented by the following matrix

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Consequently, we only need to remember the matrix of the virtual wall. As already mentioned, we need to move it in the same manner as the base when moving it to the origin. Hence, the transformation matrix of the base needs to be found, so that it can be applied to the copy. Again, this transformation matrix moves the base from the position in which it was found during the copy process to the origin (step 2):

\[
\text{Trans} \cdot \text{Base}_{\text{prev}} = \text{Base}_0,
\]

with \(\text{Trans} = \begin{pmatrix} t_{0,0} & t_{0,1} & t_{0,2} & t_{0,3} \\ t_{1,0} & t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,0} & t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,0} & t_{3,1} & t_{3,2} & t_{3,3} \end{pmatrix}\) and \(\text{Base}_{\text{prev}} = \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}\).

To find the transformation matrix something similar to a matrix division needs to be found. Mathematically, matrix division does not exist, so that I had to find a workaround.

To explain this, we need to recapitulate the process of transformation in a 3D coordinate system. The order for the transformation is scale first, rotate next, and translate last. We do not need to change the size of the 3D model, which means that the base needs to be rotated first, and translated afterwards. The same should happen for the virtual wall. Again, an illustration shall help, to make the process more clear (Figure 6.2).

\[\text{The last row in this matrix stands for the homogeneous coordinate, which is only needed for the matrix multiplication, since it is not possible to multiply a 3x4 matrix with a 3x4 matrix.}\]
Considering matrix multiplication, we get the following equation:

\[ \text{Base}_{\theta,i,j} = \sum_{k=0}^{3} t_{i,k} \cdot b_{k,j} \]

Together with the knowledge about the matrix of \( \text{Base}_0 \):

\[
\begin{pmatrix}
    t_{0,0} \cdot b_{0,0} + t_{0,1} \cdot b_{1,0} + t_{0,2} \cdot b_{2,0} + t_{0,3} \cdot b_{3,0} & \cdots & t_{0,0} \cdot b_{0,3} + t_{0,1} \cdot b_{1,3} + t_{0,2} \cdot b_{2,3} + t_{0,3} \cdot b_{3,3} \\
    \vdots & \ddots & \vdots \\
    t_{3,0} \cdot b_{0,0} + t_{3,1} \cdot b_{1,0} + t_{3,2} \cdot b_{2,0} + t_{3,3} \cdot b_{3,0} & \cdots & t_{3,0} \cdot b_{0,3} + t_{3,1} \cdot b_{1,3} + t_{3,2} \cdot b_{2,3} + t_{3,3} \cdot b_{3,3} \\
    1 & 0 & 0 & 0 \\
    0 & 0 & -1 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

This can be transformed into 16 linear equations to calculate \( \text{Trans} \):

\[
\begin{align*}
\text{Base}_{\theta,0,0} &= t_{0,0} \cdot b_{0,0} + t_{0,1} \cdot b_{1,0} + t_{0,2} \cdot b_{2,0} + t_{0,3} \cdot b_{3,0} = 1 \\
\text{Base}_{\theta,0,1} &= t_{0,0} \cdot b_{0,1} + t_{0,1} \cdot b_{1,1} + t_{0,2} \cdot b_{2,1} + t_{0,3} \cdot b_{3,1} = 0 \\
&\vdots \\
\text{Base}_{\theta,3,3} &= t_{3,0} \cdot b_{0,3} + t_{3,1} \cdot b_{1,3} + t_{3,2} \cdot b_{2,3} + t_{3,3} \cdot b_{3,3} = 1 
\end{align*}
\]
We can set $t_{3,0}$, $t_{3,1}$, and $t_{3,2}$ to 0 and $t_{3,3}$ to 1, because of the homogeneous coordinate.

Since we are calculating the rotation matrix first, we can also conclude that the last column in a rotation matrix equals 0, so that we can set $t_{0,3}$, $t_{1,3}$, and $t_{2,3}$ to 0. Hence, there are 9 linear equations left, to calculate the remaining 9 values.

Let’s have a close look at the first row of the $Base_0$ matrix. The linear equations that we get here are:

\[
\begin{align*}
Base_{0,0} &= t_{0,0} \cdot b_{0,0} + t_{0,1} \cdot b_{1,0} + t_{0,2} \cdot b_{2,0} + t_{0,3} \cdot b_{3,0} = 1 \\
Base_{0,1} &= t_{0,0} \cdot b_{0,1} + t_{0,1} \cdot b_{1,1} + t_{0,2} \cdot b_{2,1} + t_{0,3} \cdot b_{3,1} = 0 \\
Base_{0,2} &= t_{0,0} \cdot b_{0,2} + t_{0,1} \cdot b_{1,2} + t_{0,2} \cdot b_{2,2} + t_{0,3} \cdot b_{3,2} = 0
\end{align*}
\]

With $t_{0,3} = 0$ and some shifting we get:

\[
\begin{align*}
b_{0,0} \cdot t_{0,0} + b_{1,0} \cdot t_{0,1} + b_{2,0} \cdot t_{0,2} &= 1 \\
b_{0,1} \cdot t_{0,0} + b_{1,1} \cdot t_{0,1} + b_{2,1} \cdot t_{0,2} &= 0 \\
b_{0,2} \cdot t_{0,0} + b_{1,2} \cdot t_{0,1} + b_{2,2} \cdot t_{0,2} &= 0
\end{align*}
\]

These equations can be solved with Cramer’s rule. $t_{0,0}$, $t_{0,1}$, and $t_{0,2}$ can be found as follows:

\[
t_{0,0} = \frac{\det \begin{pmatrix} 1 & b_{1,0} & b_{2,0} \\ 0 & b_{1,1} & b_{2,1} \\ 0 & b_{1,2} & b_{2,2} \end{pmatrix}}{\det \begin{pmatrix} b_{0,0} & b_{1,0} & b_{2,0} \\ b_{0,1} & b_{1,1} & b_{2,1} \\ b_{0,2} & b_{1,2} & b_{2,2} \end{pmatrix}}
\]
This needs to be calculated for every row, to find the rotation matrix. This rotation matrix is needed to multiply it with the position of the virtual wall, so that it rotates in the same manner as the base.

Next, we calculate the translation matrix. In a homogeneous coordinate the translation matrix looks as follows:

\[
T = \begin{pmatrix}
1 & 0 & 0 & t_{0,3} \\
0 & 1 & 0 & t_{1,3} \\
0 & 0 & 1 & t_{2,3} \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

We only need to find the values for \(t_{0,3}\), \(t_{1,3}\), and \(t_{2,3}\). If we insert all numbers into the 16 linear equations described before, we get the following equations:

\[
\begin{align*}
Base_{0,3} &= b_{0,3} + t_{0,3} \\
Base_{1,3} &= b_{1,3} + t_{1,3} \\
Base_{2,3} &= b_{2,3} + t_{2,3}
\end{align*}
\]
These equations can easily be solved:

\[ t_{0,3} = -b_{0,3} \]
\[ t_{1,3} = -1 - b_{1,3} \]
\[ t_{2,3} = -b_{2,3} \]

If we multiply this transformation matrix with the copy, it is translated relatively to the origin. This way we calculated step 3 of Figure 6.1.

Step 4, 5, and 6 are now easy to explain. To move the copy relatively to the point where the base is found in each frame, we need to find the rotation matrix of the base first, and multiply it with the position of the virtual wall, that was just calculated. Afterwards we need to find the translation matrix, and multiply that with the virtual wall. To calculate this, exactly the same process as I just explained needs to be repeated. This time the process needs to be calculated in each frame.

An important point that was also implemented was the "0° / 90° / 180° / 270° – locking". This is regulated in rounding the all values of the copy matrix to 0, 1, or -1 except for the last row. This is also done in step 3.

The next step was the rounding of the position to lock it to certain positions. This is hard to generalize, mostly because the camera records a two dimensional image from which the depth coordinate is calculated. The ARToolKit guesses the distance from the size of the patterns that are available in the camera image. If two patterns have the same size in the camera image, it assumes that they are held from the same distance. This fact also means that if the user was holding a big pattern from a large distance, and a small pattern from a small distance, it assumes that both are located in equal distance. In my system, this causes a lot of confusion on where exactly the wall is places towards the base pattern. The further away the LEGO model is held from the camera, the more imprecise the calculation becomes. The calculation is the most precise if the pattern is held parallel to the camera, so that the pattern appears squared. Since, the base and the wall pattern are setup in a different angle, it is not possible to hold both parallel to the camera. I tried to find a workaround for that, in holding only the wall pattern parallel to the camera, and calculate the rest. This makes the calculation more precise, but still not exact enough to generalize the positional locking.
7 Testing

For the testing we can go through the use cases created in section 4.1 and 5.1. In this chapter, I will go through some bugs of the application and explain the reason, why I could not fix them in the short amount of time. In the end, I will cover some guidelines for the ideal setup of the application.

7.1 Testing Basic Functionalities

The following table lists the use cases of Figure 4.1 again.

<table>
<thead>
<tr>
<th>Step</th>
<th>User Interaction</th>
<th>System Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>To start the program, double-click the application in the Applications folder</td>
<td>A mode select screen shows up</td>
</tr>
<tr>
<td>2</td>
<td>To select a mode, type in 1 for the castle mode, and 2 for the brick building mode, and press enter</td>
<td>The application runs in the selected mode</td>
</tr>
<tr>
<td>3</td>
<td>Use the LEGO bricks to create a first pattern, that can be recognized by the system</td>
<td>The system shows no augmentation as long as the pattern is not complete</td>
</tr>
<tr>
<td>4</td>
<td>Complete the first pattern with the LEGO bricks</td>
<td>The system shows an augmented wall</td>
</tr>
<tr>
<td>5</td>
<td>Use the LEGO bricks to create a second pattern, that can be recognized by the system</td>
<td>The system still augments only one wall</td>
</tr>
<tr>
<td>6</td>
<td>Create 3 complete patterns where 2 are the same</td>
<td>The system shows 3 augmented walls where 2 are the same</td>
</tr>
</tbody>
</table>

**Step 1** is not realized as it should be. The application runs in Castle mode by default.
when it was started from the Applications folder. A mode is selectable, when the application is started from the Terminal. The process of how to start it from Terminal is explained in the attached user guide.

The system’s reaction in step 2 is also not the same as described. Here, a panel shows up, in which the camera can be calibrated. This calibration is important to run the application under different light circumstances. Nevertheless, it is hard to find the right settings for each light condition, for further information see section 7.3.

Step 3 – 6 are realized exactly as described. For this I created a screen shot for each of the steps shown in Figure 7.1.
7.2 Testing the Copy Function

The use cases of the second iteration (Figure 5.2) are listed in the following table:

<table>
<thead>
<tr>
<th>Step</th>
<th>User Interaction</th>
<th>System Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show wall and base pattern towards the camera without covering any part of it</td>
<td>The system recognizes both patterns and shows a green frame around the base and an augmented wall around the wall pattern</td>
</tr>
<tr>
<td>2</td>
<td>Press [SPACE] while both are recognized</td>
<td>The application creates one copy of the wall pattern</td>
</tr>
<tr>
<td>3</td>
<td>Remove the wall pattern from the plate and move it around</td>
<td>The system shows the AR copy in relation to the base pattern</td>
</tr>
<tr>
<td>4</td>
<td>Make a copy of two patterns at the same time</td>
<td>The system creates 2 copies</td>
</tr>
<tr>
<td>5</td>
<td>If two copies are already created, add another one</td>
<td>The system shows 3 AR walls</td>
</tr>
<tr>
<td>6</td>
<td>Press z to undo</td>
<td>The last added copy is removed</td>
</tr>
</tbody>
</table>

I have covered most of these steps in the demo video\(^1\) presented in the second user study. For further explanation, I will add some screenshots of the video at this point.

**Step 1** works fairly well. It is a little bit tricky to place all items at the same time in the camera image without covering any part of it, but at least 2 walls plus the base pattern can easily fit into the camera image. As described and also shown in Figure 7.2 the base pattern is surrounded by a green frame, and the wall pattern is covered by the augmented 3D model when both of them are recognized by the system. It would be desirable to have a text output to show the user, that the system

\(^1\)http://www.youtube.com/watch?v=IU2Qm_9muUE
is ready to make a copy, so that the user does not have to check the base pattern and the wall pattern, to see whether they are recognized.

**Step 2 & 3** are shown in Figure 5.3. It might happen that on copying the wall, the image that is grabbed is from an earlier time point than was pressed. This is because the method `arVideoGetImage()` often returns `NULL`. So that I had to find a workaround in saving the last image that could be produced and didn’t return `NULL`. Figure 7.3 shows the movement of the base and hence the base pattern. It shows that the copy moves in the same manner, always in relation to the base pattern. As already mentioned, the position of the copy is not always precise enough, which is linked to the mentioned mistake that is made when the system estimates the distances. Find further information on how the system works the most precise in section 7.3.

**Step 4** works exactly as described with the same bugs as described for the copy of one pattern. Placing two of more patterns inside the camera image together with the base pattern is not always easy, especially when the patterns are additionally arranged in a different angle. In that case, it is also not possible to hold both parallel to the camera which is a weakness of the system. Nevertheless, the copy process still works, if both of
them are not aligned with the camera. But this means at the same time, that the position of the virtual walls are not precise.

The process of step 5 is shown in Figure 7.4. The image on the left shows that two copies are already made. In the middle a wall pattern is placed on the base plate. The image on the right shows the copied walls after [SPACE] was pressed, and the pattern was removed from the plate. The virtual walls that were already copied before the window was added, are still in the same relative position to the base pattern.

In step 6, the [z] key has to be pressed to undo the last action. Every time the [z] key is pressed, only one wall is undone even if several walls were copied at the same time. If for example two walls were copied at the same time, the [z] key has to be pressed twice to undo both. The first wall that disappears, is the one with the lower confidence factor. In my testing, I found this to be a better solution than deleting both of them at the same time. The reason for this is that the wall with the smaller confidence factor is often also the one, that is set less precise. Of course, the selective deletion of virtual walls would be much more effective, but to realize that, there was not enough time left. The [z] key can be pressed whenever the user desires. For that, it is not even necessary that the system recognizes the base pattern. This might be confusing for the user, when it comes to interaction design, because as soon as the user has pressed the [z] key accidentally, and a copied wall is missing, s/he might get confused.

7.3 Additional Guidelines

The amount of bugs in the application created a need for an additional guideline that explains how it can be used the most efficiently and precise.

Light Conditions

The recognition of the patterns is very dependent on the light conditions. Direct sunlight only works when the sun beams come from behind the camera. If the sun light shines into the camera, there are too many shadows on the pattern to recognize it. In general, sun light does not work well, since the computer display reflects too much.

No light or very little light sources influence the recognition of the patterns badly. If it is too dark, shapes and colors are not recognized anymore.
For a constant recognition a setup of lights that creates very little shadows would be the perfect environment to run the application. In my tests, fluorescent lamps worked the best for a constant recognition. It created the least shadows, and is bright enough, but at the same time not too bright. The light bulb itself should not be in the camera image though. The reason for this is the so called white balance. If the light bulb is present in the camera image, the system assumes the bulb to be white. This results in a shift of colors. The white frame around the patterns for example, is then not perceived as white, so that the pattern can hardly be recognized.

Additionally the fabrication of the LEGO bricks is not ideal for a recognition. If they are completely new, the surface of the brick reflects a lot of light, which again changes the perception of colors.

**Setup for the Copy Process**

If the user presses the [SPACE] bar, a copy is not always created. This bug is always present no matter under which circumstances. It could be solved, in changing the code, so that the image is saved every time that the method `arVideoGetImage()` does not return `NULL`. This needs to be done in every iteration that produces the camera image.

The position of the copy is set the most precise, if the wall pattern is placed parallel to the camera, so that it appears squared. It should be placed in a distance of ~50 cm to the camera. With this setup, the rounding of the distances and angles work the best. Unfortunately, I was not able to find a generalization of distances to correct the mistake of the distance calculation.

**Performance**

The system runs mostly fluent under the described light circumstances. If the light is not ideal, a lot of flickering can be perceived. I ran the application one hour without stopping it, and it worked fine.

The camera image is a little bit delayed, which can be recognized when the user moves very quickly. In that case, patterns can also not be recognized anymore. I ran the application on four different Macs, and came to the result that the video is shown the most fluently on the 13- and 15-inch MacBook Pro. The other two Macs were iMacs that showed a huge delay in displaying the camera image. The following table contains
further information of the different systems:

<table>
<thead>
<tr>
<th></th>
<th>MacBook Pro 13-inch</th>
<th>MacBook Pro 15-inch</th>
<th>iMac 21.5-inch</th>
<th>iMac 27-inch</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>10.6.8</td>
<td>10.7.4</td>
<td>10.6.8</td>
<td>10.6.8</td>
</tr>
<tr>
<td><strong>Processor</strong></td>
<td>2.53 GHz Intel Core 2 Duo</td>
<td>2.6 GHz Intel Core i7</td>
<td>2.66 GHz Intel Core 2 Duo</td>
<td>2.5 GHz Intel Core i5</td>
</tr>
<tr>
<td><strong>No. of Processors</strong></td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>4 GB 1067 MHz DDR3</td>
<td>8 GB 1600 MHz DDR3</td>
<td>2 GB 800 MHz DDR2</td>
<td>4 GB 1333 MHz DDR3</td>
</tr>
<tr>
<td><strong>Graphic Card</strong></td>
<td>NVIDIA GeForce 9400M</td>
<td>Intel HD Graphix 4000 &amp; NVIDIA GeForce GT 650M</td>
<td>ATI Radeon HD 2600 Proc</td>
<td>AMD Radeon HD 6750M</td>
</tr>
</tbody>
</table>
8 Conclusions and Future Work

This chapter shall summarize my work, draw conclusions, and give an overview of the future work. This includes suggestions on which parts of the system need to be improved, and where more research is needed.

8.1 Summary

In the introduction, I explained the topic of my master project and its significance.

The literature review clarified some important taxonomy, discussed relevant literature followed by a critical assessment, and technical aspects. Every part of the literature review is connected to my project, and most of the theories could already be applied to it. All discussed aspects of the literature review are reviewed and summarized in the end of the chapter.

In the concept chapter first professional, legal, and ethical issues were covered that need to be considered when working with children. Afterwards, a concept is created, and many theories presented in the literature review were revisited. Hereby, I mostly relied on the guidelines that Hinske et al. created. The most important part of the concept is that it connects to real world contexts. In putting walls of a building together, that then is represented by a real looking virtual building, children can refine their constructional skills. Also, it was important to me, that the application provides an interface in which traditional input devices such as mouse and keyboard are almost replaced. Instead, the input shall mostly come from the LEGO bricks that can be used in all three dimensions.

The project was evaluated in two iterations. The user studies were always very small, but helped in defining the requirements of the application and extending them. The first user study showed a lack in the concept. As soon as the children run out of suitable LEGO bricks, they lose their interest, and refuse to build more patterns. For that matter a copy function was created. This function gives children the possibility, to create copies
of virtual walls, so that they can keep on playing with the environment even if they don’t have enough bricks available.

The prototype was extended to realize the copy function. Unfortunately, the copy function had too many bugs, to evaluate it accordingly. Instead a video of the functions of the system was created, and shown to children of a different age range. The children always reacted quite positive to the video, but this user test also revealed that the age range might need to be changed.

The developer guide covers the implementation of the application. For this, the implementation of the first prototype is barely covered. Instead the copy function is explained in detail. A lot of matrix operation is needed to understand the process of the copy function. The homogeneous coordinate is needed to calculate transformation matrices, as well as a modified version of a "matrix division". It is important to remember the order in which an object is transformed in a 3D space: 1. scale, 2. rotate, 3. translate.

The testing demonstrates which part of the application work well, and when errors occur. It also includes a guideline that explains under which circumstances the application works the best.

8.2 Conclusions

I was able to create a concept of an AR application that replaces most of the interaction with mouse and keyboard by LEGO bricks. Most of the basic functionalities of this application have been implemented. Initially, it was planned to evaluate the application in two iterations, which was not realized in the way that I would have wanted to if I would have had more time.

In general, it can be said, that children always become curious when they get to experience a technology that they have never seen before. This is also the reason why AR in many literature is assumed to be a good method to use in an educational system. It draws a lot of attention, so that the users automatically become actively engaged. Nevertheless, many authors forget about the fact that it might only be a short time reaction, and that it can happen that after a while children get bored with the system.

This project showed that it is possible to join Roschelle et al.’s components of an effective learning with Hinske et al.’s guidelines for the design of an augmented toy environment. The effectiveness of the application and how it influences the play behavior
I would say that my work distinguishes from Hinske et al.’s project in the sense that children can set a building together in whichever way they want. Every time they use the application, something new can be created. In the Kingdom of the Knights audio is produced depending on where the toy is placed on the playground. The audio is then repeated over and over again. Their work is much more evaluated in depth, so that they were able to create some guidelines. Of course, for my project much more evaluation cycles would be needed to reveal weaknesses of my concept. More time would also give me the chance to eliminate the software bugs which would make an appropriate user study possible.

8.3 Future Work

The application that I implemented contains some errors that would need to be fixed. A concept for the user interface was completely not covered in this project. For this user centered design approaches would be needed to do so. As I have already stated, involving children in the design process is difficult because children might not always be able to phrase what they think, or adults are not able to interpret their words correctly. For this more research is needed, especially when it comes to technology that is completely new to them. In that case, the interaction would need to be explained to them without influencing their ideas.

Furthermore, the application will need to be extended by additional functionalities. Since the system works inaccurate in setting the copied walls to a certain position, a move function can be appropriate. This can also include a rotation of the walls. The selection of certain copied walls and removing them can also be helpful.

As already stated in this report, more user studies are necessary to draw generalized conclusions. For a full working application, the evaluation will need to be conducted in more cycles. With this, more functionalities can be found that would be beneficial for the use of the AR application. I also made assumptions on the age range that will need to be proven in sessions with more participants of every age.

An installation at children’s home will help to conclude whether the aim of this project is actually fulfilled which was to create a play environment that is fun and at the same time rewarding for constructional skills. In this way, it can be shown whether
children are still willing to play with the system even after getting over the first excitement that the technology of AR naturally draws.

In my opinion the concept of the application that I created is not ideal, because it does not give constant feedback and even worse, it can not tell children that they made a mistake in building the pattern if they have made one. This can frustrate the user, even though I think that the patterns are simple enough to accomplish the task. At the moment it is computationally almost impossible as covered in the literature review to track every single brick and create a reaction of the system on that. Nevertheless, a concept similar to mine, in which the application can react on every object that users create with LEGO bricks would be much more enjoyable.

In general, I question the learning effect that the system creates. In the beginning I stated that the aim of this project is to create an educational application that uses AR as technology. Now, I would say that the application is not ideal for teaching constructional skills. The patterns are very simple, and can be reconstructed with LEGO bricks quite fast, as the first user study has shown. However, a great benefit was the observation of the social interaction between the two participants. In this study, the children were very kind and supportive. They even started to create a plan and assigned tasks to each other to realize it. The fact that the application supports social interaction again needs to be proven in further user studies, because it is also very likely that children can get into a fight over the LEGO bricks.
Appendix

A First User Study

**Interviewer:** Hello and welcome to my first user study session. Do you already know what is going to happen today?

**User A:** No...

**User B:** I only know that it has something to do with LEGO.

**Interviewer:** Yes, that is right. I will ask you to accomplish some tasks. You will have to rebuild some patterns with the LEGO bricks, and afterwards you can see a connection with the computer. But first, I'd like to ask you whether you rather like to play together or separately.

**User B:** Together.

**Interviewer:** Alright. So, this is the first pattern that I would like you to build.

**User A:** Okay...

[is counting the number of knobs on top] One, two, three, four, five, six, ..., eight, ten, twelve.

**User B:** Oh and we will need one, two three, four, five, six, ..., eight in height.

**Interviewer:** I can give you a second copy of the pattern if that helps.

**User A:** Thanks.

**User B:** Six there.
Next, umm...

User A: We got to do the black ones first.

User B: Oh yeah. Umm...

Interviewer: So, do you like to play with LEGO.

User A: Mhm (Yes). Yeah, I’ve got loads of it.

Interviewer: You’ve got loads of it. How often do you play with LEGO?

User A: Every day.

Interviewer: What do you like to build when you play with LEGO?

User A: I like building robots and stuff sort of. Um, I’m not sure how to explain it. I like building vehicles and things like that.

Interviewer: What about you, User B?

User B: Um, I quite like it, but I’m not as big of a fan as User A. Okay, I start on the yellow (bricks) now.

User A: That’s it.

User B: Oh yeah, this is it. Oh no, this doesn’t fit.

User A: You need to get another one. Just use this.

User B: I just take that one.

[Some bricks fell on the floor]

User A: Got it. [Sees that User B has put a white brick on top and thought that was wrong.] No, it’s just one.

User B: But User A, we need to put a black row on top.

User A: Oh yeah, yeah, sorry.

User B: And then take that one.
After 4:26 minutes they finished the first pattern.

User B: Cool.

Interviewer: Can you hold it towards the computer?

[They do so.]

User B: That gonna be a house.

User A: Cooooool.

User B: That is cool.

Interviewer: Maybe you have to hold it a bit closer to the camera.

[The system augments a wall on User A’s thumb.]

User B: It makes your thumb look also.

User A: Oh, that’s cool.

[turns the pattern around, to see if there is actually something on the pattern.]

Interviewer: You need to be careful so that you don’t put your fingers on top of the pattern.

User B: Can I hold that [pattern]?

User A: Yep.

User B: Cool. Is that what it’s meant to be?

Interviewer: Yeah, how do you like that?

User B: Yeah, that is really cool.

[At 5:18 they started with the second task]

Interviewer: Okay, the next step is to build this pattern. This time we don’t have it printed on paper. You will have to build it from the screen.
User A: Shall we just build it onto this bit (the one that they’ve built before)?

Interviewer: Yeah. You can also use this plate if you want.

User A: Oh yeah, that is helpful.

Interviewer: So, are you also playing computer games?

Both: Yeah.

Interviewer: What kind of computer games?

User A: Um, i like playing LEGO Star Wars.

User B: Um, User A, you can take that.

Interviewer: Oh, what is LEGO Star Wars?

User A: Um, you know what Star Wars is, right?

Interviewer: Yes.

User A: It’s within that except in LEGO. Yeah, you can get LEGO Indiana Jones, or LEGO Harry Potter, ... You can get loads of LEGO things.

Interviewer: Oh cool.

Parent: They’ve got this little LEGO characters, you know these LEGO figures, and they’ve took whatever the main characters are [...]. He plays it on the PS3.

User A: And online, yeah.

Interviewer: And how often do you play computer games?

User A: Um, not as often as some other kids. But umm...

Parent: It’s not as often as he wants to.

[The kids are searching for more bricks to match with the color.]

User B: You will need to put it on the top of it, not there.
Interviewer: Oh yeah, maybe we will need some more of your bricks now.

User A: Oh yeah. They (the boxes) are always very hard to open.

User B: Yeah, he was making flats of different so...

[Sees that something’s wrong.] User A. Haven’t... oh no.

Actually yeah. I think you’ve build it like one too high, cause you gonna be put it over, over the top up there.

User A: Nah, that’s all right.

Interviewer: You can just hold it next to it if you want.

User A: I know what I’ve done wrong. That was stupid. We forgot the black bit.

User B: [Thinking] Oh yeah.

Why are you doing that (it’s a gap between the yellow and the white bricks)? Cause it doesn’t has a space there.

I see what you’ve done wrong. [User A changes it without User B telling what was wrong.] Oh yeah, and then you need to put a black one in there. [User A searches for one.] User A, I got one here.

User A: [searching for more black bricks.]

User B: [finds small bricks that build up the right brick all together.] You don’t need to find any more. User A, I’ve got.

User A: Why did you put all the black ones in the bottom?

User B: I don’t know.

User A: You get some black bricks.

User B: Got one (it’s 1x1).

User A: You need four.
User B: Yeah.

Parent: You can stop searching. I guess, I can do that.

User A: Right.

User B: I need a white one. Oh, I don’t. Actually no, this is...

User A: Leave that, leave that.

User B: Oh yeah.

[After 10:30 min they finished the second pattern.]

Interviewer: What do you expect next, when you hold it towards the camera?

User A: Another building.

User B: Two houses.

Interviewer: Let’s see. Hold it a bit higher.

User B: Up. Oh cool, that ones got the door and that ones got the window.

User A: [Tilts the model up and down.]

User B: Getting there. It doesn’t really want to do it.

Interviewer: Nah, it doesn’t.

User B: You have to hold it straight. You’ve put that one upside down.

User A: [Holds on to the base plate.]

User B: User A.

Uh, we’ve got something... It’s not coming.

Interviewer: Hold on a second. I’ll restart it.

[Switches the light on.]

User B: We’ve got some... We’ve got it!
User A: Oh oh, yes, yes.
User B: We did see the full picture for a second.
User A: Yeah. Down, down.
Yaay. Cool.
User B: Can I stop now? Oh look, you see that blue bit? In the window, and then there’s this yellow bit.
User A: Yeah.
User B: Alright, I’m gonna put it down now.
User A: That’s cool.
Interviewer: Cool. How was that for you, so far?
User A: It was fun. It’s so cool because you can see it like this [Holds hands as if holding the base plate towards the camera.]
User B: It’s cool, because it’s just LEGO but then it projects some picture.
User A: Yeah.
Interviewer: What did you prefer? Building it from the printed version or from the digital version?
User B: Um, I don’t mind.
User A: Yeah, I think it was about the same.
Interviewer: It was about the same. Cool. Because, I’ve got another pattern for you that you can add if you want. Let me find it.

[14:54 minutes left so far.]

Interviewer: Here it is.
User A: Cool.
User B: I think that might be like another window maybe or something.
User A: Let’s find out.
User B: Oh, I start doing the black bit. [Searching for black bricks and figuring out that User A already did the black bit.] Oh...
User A: Here, you can do that.
User B: [Grabbing bricks.] I’ve got this one, and I’ve got this one. Okay. We’ve got on a full square two full squares.
Interviewer: So, you would think that is a window?
User B: Yeah.
User A: Um, I really don’t know what it’s gonna be.
User B: I will just get this (a yellow brick)...
User A: User B, you can just get the black ones cause we need some more. Actually, no we don’t. Do we? No.
User B: Okay, I’ve got this (white on top of a yellow brick). [Sees that User A already did it.] Alright, you’ve already got it. Oh man. Okay, then this, and we’re done.
Interviewer: You’re getting faster and faster, huh?
[It’s been 16:47 minutes. The kids are holding the model up towards the camera without any request.]
User B: Yeah. Eh, it’s a wall.
User A: It’s another bit of wall. Oh god, so cool.
User B: I think it might end up as a school, or a castle, or something.
Parent: [Comes over to have a look at it.] You know what I think is really quite cool?
User B: What?
Parent: Well, if you look at the picture over there...

User B: Yeah?

Parent: ... you didn’t quite build it right.

User B: Huh.

Parent: Cause this bit is on top of the blue one. [Points on the picture]

User B: Ooohh...

Parent: But it still worked.

User B: Oops, oh well. Shall we still change it?

User A: No, we’re fine. Oh, that is so cool.

Interviewer: This bit is actually a part of a roof. And when you build several of them next to each other, then you can build a full parapet as a roof.

User B: [Moves hands up and down.] The camera is actually slower than I am. Cause when I do that, it doesn’t. [Moving arms up and down in a fast manner.] Look.

Interviewer: Oh yeah.

User B: [Giggling.] It’s cool.

Interviewer: Um, so how did you find it?

User B: It’s really cool.


Interviewer: Okay, and how did you like the virtual building that you saw on the screen?

User B: Umm, it’s cool.

User A: [Nodding.]
Interviewer: Would you like to change anything about it?

User A: Not really.

User B: What is there to change?

Interviewer: Just if you could. Is there anything you’d like to add?

User A: Um, I don’t think so. I think it’s good anyway.

User B: I would like to make it like build more.

User A: Yeah, like more roof parts, I think. I’ll just put it (the model) on the table.

Parent: What are the things that you’d like to actually add to it?

User A: Maybe, ... I guess that is quite complicated, but maybe a bird on the roof would be quite cool.

User B: I know, a carrot.

All: A carrot?

User B: [Giggling.]

User A: Seriously?

Interviewer: Cool, a carrot, right. Can you imagine putting the second pattern that you made somewhere else?

User A: Which one? That?

Interviewer: The blue one. Just take it off.

User B: What would it look like, if you like... put that one there?

Interviewer: Put this one to the side.

User B: Alright, like that.

Interviewer: Like this. Like a corner of a building.
Both: Alright, oh yeah.

User B: I wonder what that would look like. [Holding up the model towards the camera.]

Interviewer: Can you turn it around? Move it all the way around.

User B: To the back.

User A: Ah, oh, that’s cool.

[Some bricks fell from the base plate. They picked it up. Parent came over to see it.]

User B: Hold on. [Puts the walls back together.] Right. There.

Parent: Wow.

User B: That is cool. What would happen if I do that? [Covering pattern with finger. Figured out that the virtual model is not shown anymore, when the pattern is covered.] Oh, that’s cool.

Interviewer: So, what can you imagine now, what you could build with it if you had more time?

User B: You could maybe build like a whole house.

User A: Yeah, at least the front of a house.

User B: And then you can make it like build other houses around it.

User A: Yeah that’d be cool.

User B: So, does it also like work with any pattern. Does it not only work with that pattern?

Interviewer: Right now it only works with these patterns. And sometimes it sees patterns where there is none, like here. But yeah, it actually only works with these given patterns. And maybe in future with a few more. Would you rather like to build something yourself and then see what it’s like?
Both: Yeah.

User A: So like if you build like...

User B: A carrot.

Interviewer: Would you like to play a bit further, and see what you can build?

User A: Yeah, can do.

Interviewer: Unfortunately you don’t have that man bricks left, but you can just break apart the other ones if that helps.

User B: We have some of User A’s. Are you making a carrot? Yay.

Interviewer: Well, I can already promise you that the system is not going to recognize the carrot.

Parent: No, it will only work if you make things out of those patterns. So, you can drop those together in different ways.

User B: I’m just gonna try it with a carrot.

Interviewer: You can for example make more of these roof bits.

User B: Oh yeah, uh um, I’m just gonna come down.

Um, would it recognize this but in different colors?

Interviewer: No.

User B: Uh.

User A: [Turns around the patterns.] I’m very inventive.

User B: And then, I make few more of them.

User A: [Holds the model up towards the camera.] So it (the roof part) works just everywhere.

User B: Cool.
User A: Alright, so you can just put it anywhere.

User B: Right, so you know how this is just a bit of roof. So... if... Okay User A, you start making more of them, so like all the way down. And I make something. I’m gonna make like... [Irritated by the system.] It seems to be scanning me.

Parent: Maybe, it turns you into a carrot.

User B: I’ll stop making a carrot.

Interviewer: Do you like carrots?

User B: No.

User A: So, why are you obsessed with them?

User B: Cause they’re cool and they are random. I’m gonna try like... want to go like this. You can always see a little bit... Uh, if we make like another one of that and then we stick it along there, and then you can stick that one over there, and that one over there. [Pointing to the roof bits.] Okay... Can you make like some different sizes?

Interviewer: Different sizes? What do you mean by different sizes?

User A: Can you make like that pattern bigger?

Interviewer: Yes definitely. You can for example make it double as big if you want to, yeah.

User B: Okay. I’m gonna make a big one.

Interviewer: Maybe you will need to take that wall, cause otherwise you won’t have enough bricks.

User B: Okay, I make this, and then I just need to put a yellow blog in there. I can make the blue... six...

User A: Yes.
User B: So I need to put... six... act... yeah. Oh it needs to be so high. No we don’t put 4.

User A: I don’t know what you’re trying.

User B: What if, I make that three with the same... This isn’t gonna work. I’d need a lot more pieces to do it.

Interviewer: Yeah, unfortunately.

User B: I’ll just make another one with that. And then it would be another one. That one, that one, ...

User A: [Built another roof pattern and holds it up towards the camera.] Cool.

User B: Do you want me to do it (holding the model)? Yay. Cool. You can make that one... [See that the pattern is not complete from the back side.] Oo User A.

User A: Yeah, I know.

Interviewer: Okay so what do you enjoy about the system?

User B: It’s cool, and fun, and something to do with your spare time.

User A: Yeah, I enjoy playing with LEGO anyway. Any type of LEGO. I think it just adds some more fun.

Interviewer: Is there something you didn’t like about it?

User A: Not really. I enjoyed all of it. There wasn’t anything I did not like.

User B: Oh, I need more white. Four of these.

What would happen if there were gaps.

Interviewer: It probably wouldn’t recognize it as a pattern.

Oh, that is very smart, User A.

User B: What are you doing?
User A: I only made the front, the back is not important.

User B: Ohh, oh.

Parent: User A, just take that one for the front (it’s a 1x1 brick). It doesn’t matter if it’s just one.

User A: Oh oh yeah, I never thought about that.

Parent: Can you think of anything else you’d like to make?

User A: Not really, cause I’m not really imaginative. Most of the time I just make it up when I go along.

Interviewer: Would you like to build another building?

User A: Yeah, maybe like two buildings. So that you can have one building here, and one building over here. Two separate things. That would be cool.

Parent: What type of building?

User B: Castle.

User A: Just any type yeah. Umm..., yeah, I really don’t know.

Parent: And apart from carrots, User B, what would you like to do?

User B: Apples. I’d probably like to do people or something.

How did you make it able to do that?

Interviewer: Well, it first searches for the black bit and as soon as it finds the black square, it checks what is inside, what shapes and colors are inside.

User A: So it just ignores the white part?

Interviewer: Actually yes. The white part is just because if you have something black in the background then it can not distinguish between...

User A: Yeah.
User B: [Tries another version of a pattern.] This may work.

Parent: You’ll need to have another row of the red one.

User B: I know. Actually, it’s going to be like flat. Just to see if it’s going to work.

User A: [Has built a third roof bit.] Yaay. It works.

Interviewer: See, as soon as you’ve got you finger in front of the pattern, it’s not going to recognize it. You can also put your finger just in front of the black bit, and it doesn’t see the pattern anymore.

User B: Why is it that if you put your finger on the white part it can see, but if you put it on the black part it can not see?

Parent: It just sees the black and ignores the white.

User B: Oh yeah.

Parent: So that’s why it doesn’t really matter for the white.

Both: Oh yeah, yeah.

User A: Yeah the last one was very... (corrupted), but it still worked from the front.

Interviewer: I’m curious what is going to happen with that one (the flat one that User B is building.)

Parent: Just put one of these.

User B: I know. And then I need to put the white. Umm... yeah.

[Mumbling. User A realized that there are not enough bricks left to build another pattern, so User A helps finding bricks for User B. Then User A doesn’t want to interfere User B’s play and start building a mannequin.]

Interviewer: Are you building a robot now?
User A: I don’t really know what it is. Sort of...
User B: ... Pattern? Thing? That looks like feet.
Parent: You’re a little bit off task, User A.
User A: Sorry right.
User B: [Finished the flat one. It’s like the door pattern but more squeezed. The proportions are more like the one of the window pattern.]

[The system shows the window upside down.]

User B: Alright, I’m holding it upside down.
Parent: Oh, wasn’t that the right way?
Interviewer: Well actually, it was the right way, but it shows it upside down on the screen.
User A: Oh, it shows it as a window. Can you move it up? Now it’s a door.
Interviewer: That’s interesting.
User B: Oh my gosh, this is really really fun.
Interviewer: Now it’s time for your carrot?
User B: Yeah, nice carrot.
Interviewer: Cool.
User B: [Holds the carrot model towards the camera. User A takes the model. User B turns around.] What are those cameras for?
Interviewer: I’m filming you, so that I can analyze what you did afterwards. Alright, we’ve got it all.
User B: [starts over again.]
Interviewer: What are you making?
User B: I’m gonna try make that one (roof part). Lots. I don’t know how.
User A: So what are you making?
User B: You’ll see.
User A: Okay...
User B: [Returned working on the mannequin.] This thing has strange arms skin.
User A: I usually use yellow for skin.
Parent: User B, you can use these small ones.
User B: Okay... Is it going to work just without it (the white border).
Interviewer: Try it. You can try holding the white paper behind it.
User B: [Holding up the new pattern, but the system doesn’t react on it.] No, it doesn’t work... [Turns the model around.] I just put this here, to hold on to it. Oh no, it doesn’t. Oh well. But I think that is really cool (the brick to hold on to). [Looking at User A’s model, which is the mannequin.] Is he gonna have hair?
User A: Yes, and it’s a she. Cause otherwise this terrible pun wouldn’t work.
User B: It’s not me, is it? Just do it without the hair.
User B: User A, that is connecting to its eye.
User A: Stop breaking... Very nearly there... [Finished the mannequin and puts it on the table.] What do you think her name is?
User B: Um, something with LEGO?
User A: Aileen.
User B: I don’t get that.
User A: Aileen, because she’s leaning.

Interviewing: Thanks a lot for playing with me.

User A: You’re welcome.

Parent: Did you enjoy that?

Both: Yeah. [Holding the up the building towards the camera again.]

Parent: What would you like the system to turn Aileen into?

User A: Umm, I don’t know. [Holds up the mannequin towards the camera. There’s something augmented in the face.]

User A: Wait, wait.

Interviewer: There’s something on top of her nose, huh?

User B: I think that’s a mouth.

User A: No, that’s a nose. She doesn’t has a mouth, her parents got fed up from her talking all the time.

Parent: Cool. Shall we go home?

User A: Okay.

Parent: Was that everything that you needed?

Interviewer: Yes, that was everything. We just need to sort out the bricks. But thank you so much for your time.
B Patterns for the First User Study

The following presents the patterns that the two participants of the first user study had to reconstruct (dimension 1:2).

**Task 1**

![Task 1 Image](image1)

**Task 2**

![Task 2 Image](image2)
Task 3
C    Code

The following shows the code of arLego.c which is an altered version of simpleVRML.c [19]:

```c
/*
 * arLego.c an altered version of simpleVRML.c
 * Demonstration of ARToolKit with models rendered in VRML.
 * Press '?' while running for help on available key commands.
 * Copyright (c) 2002 Mark Billinghurst (MB) grof@hitl.washington.edu
 * Copyright (c) 2004 Raphael Grasset (RG) raphael.grasset@hitlabnz.org.
 * Copyright (c) 2004-2007 Philip Lamb (PRL) phil@eden.net.nz.
 *
 * Rev Date   Who    Changes
 * 1.0.0  ????-??-??  MB  Original from ARToolKit
 * 1.0.1  2004-10-29  RG  Fix for ARToolKit 2.69.
 * 1.0.2  2004-11-30  PRL Various fixes.
 *
 */
/*
 * This file is part of ARToolKit.
 * ARToolKit is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 */
```
* ARToolKit is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with ARToolKit; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston,
* MA 02111-1307 USA
*
*/

// ==============================================================
// Includes
// ==============================================================

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <GLUT/glut.h>
#include <AR/config.h>
#include <AR/video.h>
#include <AR/param.h>  // arParamDisp()
#include <AR/ar.h>
#include <AR/gsub_lite.h>
#include <AR/arvrml.h>

//@import "Foundation/Foundation.h"
#include "object.h"
#include "homogenius.h"
// Constants
// ==============================================================
#define VIEW_SCALEFACTOR 1.0 // 1.0 ARToolKit unit becomes 1.0 of my OpenGL units.
#define VIEW_DISTANCE_MIN 10.0 // Objects closer to the camera than this will not be displayed.
#define VIEW_DISTANCE_MAX 10000.0 // Objects further away from the camera than this will not be displayed.

// ==============================================================
// Global variables
// ==============================================================

// Preferences.
static int prefWindowed = TRUE;
static int prefWidth = 640; // Fullscreen mode width.
static int prefHeight = 480; // Fullscreen mode height.
static int prefDepth = 32; // Fullscreen mode bit depth.
static int prefRefresh = 0; // Fullscreen mode refresh rate. Set to 0 to use default rate.

// Image acquisition.
static ARUint8 *gARTImage = NULL;

// Marker detection.
static int gARTThreshhold = 100;
static long gCallCountMarkerDetect = 0;

// Drawing.
static ARParam gARTCparam;
static ARGL_CONTEXT_SETTINGS_REF gArglSettings = NULL;

// Object Data.
static ObjectData_T *gObjectData;
// > MY CODE ...
static ObjectData_T myObjectDataArray[9][9];
static int gObjectDataCount;

// Mode
static int mode;

// Data for Copy.
static int atLeastOneCopySet = 0;
static double myBase_prev[3][4] = {
    { 1, 0, 0, 0 },
    { 0, 0, -1, 0 },
    { 0, 1, 0, 0 }
};
static ObjectData_T myCopies_prev[20];
static ObjectData_T myCopies_cont[20];
static int noOfCopies = -1;

static ARMarkerInfo *save_m_info;
static int save_m_num;
// < ... MY CODE

// =====================================================
// Functions
// =====================================================

static int setupCamera(const char *cparam_name, char * vconf, ARParam *cparam)
{
    ARParam wparam;
    int xsize, ysize;

    // Open the video path.
    if (arVideoOpen(vconf) < 0) {
        fprintf(stderr, "setupCamera(): Unable to open
            connection to camera.\n"");
        return (FALSE);
    }
// Find the size of the window.
if (arVideoInqSize(&xsize, &ysize) < 0) return (FALSE);
fprintf(stdout, "Camera image size (x,y) = (%d,%d)\n", xsize, ysize);

// Load the camera parameters, resize for the window
and init.
if (arParamLoad(cparam_name, 1, &wparam) < 0) {
    fprintf(stderr, "setupCamera(): Error loading
    parameter file %s for camera.\n", cparam_name);
    return (FALSE);
}
arParamChangeSize(&wparam, xsize, ysize, cparam);
fprintf(stdout, "*** Camera Parameter ***\n");
arParamDisp(cparam);
arInitCparam(cparam);

if (arVideoCapStart() != 0) {
    fprintf(stderr, "setupCamera(): Unable to begin
    camera data capture.\n");
    return (FALSE);
}

return (TRUE);

static int setupMarkersObjects(char *objectDataFilename,
ObjectData_T **objectDataRef, int *objectDataCountRef)
{
    // Load in the object data - trained markers and
    associated bitmap files.
    if ((*objectDataRef = read_VRMLdata(  
        objectDataFilename, objectDataCountRef)) == NULL)
        {  
            fprintf(stderr, "setupMarkersObjects():

```
read_VRMLdata returned error !!\n"; return (FALSE);
}
printf("Object count = %d\n", *objectDataCountRef);
return (TRUE);

// Report state of ARToolKit global variables
arFittingMode, // arImageProcMode, arglDrawMode, arTemplateMatchingMode,
// arMatchingPCAMode.
static void debugReportMode(const ARGL_CONTEXT_SETTINGS_REF arglContextSettings)
{
if (arFittingMode == AR_FITTING_TO_INPUT) {
    fprintf(stderr, "FittingMode (Z): INPUT IMAGE\n");
} else {
    fprintf(stderr, "FittingMode (Z): COMPENSATED IMAGE\n");
}

if (arImageProcMode == AR_IMAGE_PROC_IN_FULL) {
    fprintf(stderr, "ProcMode (X) : FULL IMAGE\n");
} else {
    fprintf(stderr, "ProcMode (X) : HALF IMAGE\n");
}

if (arglDrawModeGet(arglContextSettings) ==
    AR_DRAW_BY_GL_DRAW_PIXELS) {
    fprintf(stderr, "DrawMode (C) : GL_DRAW_PIXELS\n");
} else if (arglTexmapModeGet(arglContextSettings) ==
    AR_DRAW_TEXTURE_FULL_IMAGE) {
    fprintf(stderr, "DrawMode (C) : TEXTURE MAPPING (FULL RESOLUTION)\n");
} else {
    fprintf(stderr, "DrawMode (C) : TEXTURE MAPPING (HALF RESOLUTION)\n");
}
if (arTemplateMatchingMode == AR_TEMPLATE_MATCHING_COLOR) {
    fprintf(stderr, "TemplateMatchingMode (M) : Color Template\n");
} else {
    fprintf(stderr, "TemplateMatchingMode (M) : BW Template\n");
}

if (arMatchingPCAMode == AR_MATCHING_WITHOUT_PCA) {
    fprintf(stderr, "MatchingPCAMode (P) : Without PCA \n");
} else {
    fprintf(stderr, "MatchingPCAMode (P) : With PCA\n" );
}

static void Quit(void)
{
    arglCleanup(gArglSettings);
    arVideoCapStop();
    arVideoClose();
    #ifdef _WIN32
    CoUninitialize();
    #endif
    exit(0);
}

//> MY CODE ...
void createCopy(void)
{
    double transformationMat[3][4];
    double difference[3];
    double roundedRotMat[3][3];

    ARMarkerInfo *marker_info; // Pointer to
array holding the details of detected markers.

int k[5], i, j, l, h;
int s, t;
int hiroExists = 0;

/**
 ** Check all the markers. The one with the highest confidence factor
 ** is marker_info[k]. Also, we check whether the base is available
 ** in the image. If so, it can be found in marker_info[l].
 ** (In gObjectData[gObjectDataCount-1] is the base marker)
 ***/
for (h = 0; h < 5; h++)
  k[h] = -1;

h = -1;
l = -1;
for (i = 0; i < save_m_num; i++)
  if (save_m_info[i].id != gObjectData[gObjectDataCount-1].id) {
    if (save_m_info[i].cf > 0.75) {
      h += 1;
      k[h] = i;
    }
  } else { // if marker is base, store its id in l
    if (l = -1) l = i;
    else if(marker_info[i].cf > marker_info[l].cf){
      l = i;
    }
  }
}
/**
 ** The base should not be the only detected marker.
 ***/
if (l != -1 && h != -1) {
hiroExists = 1;
else {
    hiroExists = 0;
}
if (hiroExists == 1){
    printf("Copy has been created!\n");
atLeastOneCopySet = 1;
    for (i = 0; i < gObjectDataCount-1; i++){ // we
        only need to check
            // for the Lego patterns
            // not for the base pattern
        for (j = 0; j<=h; j++){
            if (save_m_info[k[j]].id == gObjectData[i].id)
                {
                    /** arGetTransMat writes the properties of
                        the marker into
                        ** gObjectData. We need these information
                        to save the
                        ** position of the base and the copy to
                        that time point.
                        **/
                        arGetTransMat(&save_m_info[k[j]],
                                      gObjectData[i].marker_center,
                                      gObjectData[i].marker_width,
                                      gObjectData[i].trans);
                        arGetTransMat(&save_m_info[l],
                                      gObjectData[gObjectDataCount-1].
                                      marker_center, gObjectData[
                                      gObjectDataCount-1].marker_width,
                                      gObjectData[gObjectDataCount-1].trans);

                        if ( save_m_info[k[j]].area != 0){
                            if (noOfCopies < 8) {
                                noOfCopies = noOfCopies + 1;
                            }
                            myCopies_prev[noOfCopies] = gObjectData[i] ;
                        }
for (s=0; s<3; s++){
    difference[s] = gObjectData[i].trans[s][3] - gObjectData[gObjectDataCount-1].trans[s][3];
}

// rotate base (gObjectData[gObjectDataCount-1]) to origin position (myBase_prev)
matrixDivision(myBase_prev, gObjectData[gObjectDataCount-1].trans, transformationMat, 0);
matrixMultiplication(transformationMat, gObjectData[gObjectDataCount-1].trans, gObjectData[gObjectDataCount-1].trans);

// also rotate copy
matrixMultiplication(transformationMat, myCopies_prev[noOfCopies].trans, myCopies_prev[noOfCopies].trans);

// we will need the rounded transformationMat to shift the x, y, z values in the end
rounding(transformationMat);

for (s=0; s<3; s++){
    for (t=0; t<3; t++){
        roundedRotMat[s][t] = transformationMat[s][t];
    }
}

// translate base to origin, from now on gObjectData[gObjectDataCount-1].trans = myBase_prev
matrixDivision(myBase_prev, gObjectData[gObjectDataCount-1].trans, transformationMat, 1);

// also translate copy
matrixMultiplication(transformationMat, myCopies_prev[noOfCopies].trans, myCopies_prev[noOfCopies].trans);
rounding(myCopies_prev[noOfCopies].trans);

// rounding the x, y, and z values for
locking to certain positions
if (difference[0] < -25) {
    if (difference[0] >= -75) {
        difference[0] = -50;
    } else if (difference[0] >= -125) {
        difference[0] = -100;
    }
} else {
    if (difference[0] <= 25) {
        difference[0] = 0;
    } else if (difference[0] <= 75) {
        difference[0] = 50;
    } else if (difference[0] <= 125) {
        difference[0] = 100;
    }
}

if (difference[1] < -75) {
    if (difference[1] >= -125) {
        difference[1] = -100;
    } else if (difference[1] >= -175) {
        difference[1] = -150;
    }
} else {
    if (difference[1] <= 0) {
        difference[1] = -50;
    }
}

if (difference[2] < -70) {
if (difference[2] >= -150) {
    difference[2] = -50;
} else if (difference[2] >= -200) {
    difference[2] = -100;
} else {
    if (difference[2] <= -40) {
        difference[2] = 0;
    } else if (difference[2] <= 10) {
        difference[2] = 50;
    } else if (difference[2] <= 60) {
        difference[2] = 100;
    }
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
static void Keyboard(unsigned char key, int x, int y) {
    int mode, threshChange = 0;
    switch (key) {
    case 0x1B: // Quit.
    case 'Q':
    case 'q':
        Quit();
        break;
    case 'C':
    case 'c':
        mode = arglDrawModeGet(gArglSettings);
        if (mode == AR_DRAW_BY_GL_DRAW_PIXELS) {
arglDrawModeSet(gArglSettings, AR_DRAW_BY_TEXTURE_MAPPING);
arglTexmapModeSet(gArglSettings, AR_DRAW_TEXTURE_FULL_IMAGE);
} else {
    mode = arglTexmapModeGet(gArglSettings);
    if (mode == AR_DRAW_TEXTURE_FULL_IMAGE)
        arglTexmapModeSet(gArglSettings, AR_DRAW_TEXTURE_HALF_IMAGE);
    else arglDrawModeSet(gArglSettings, AR_DRAW_BY_GL_DRAW_PIXELS);
}

fprintf(stderr, "*** Camera - %f (frame/sec)\n", (double)gCallCountMarkerDetect/arUtilTimer());
gCallCountMarkerDetect = 0;
arUtilTimerReset();
debugReportMode(gArglSettings);
break;
case '-':
    threshChange = -5;
    break;
case '+':
case '=':
    threshChange = +5;
    break;
case 'D':
case 'd':
arDebug = !arDebug;
    break;
case '?':
case '/':
    printf("Keys: \n");
    printf(" q or [esc] Quit demo. \n");
    printf(" c Change arglDrawMode and 
        arglTexmapMode. \n");
    printf(" - and + Adjust threshold. \n");
    printf(" d Activate / deactivate debug 
        mode. \n");
    printf(" ? or / Show this help. \n");
    printf("\nAdditionally, the ARVideo library \n");
supplied the following help text:

```c
arVideoDispOption();
break;
// > MY CODE ...  
// added two keys

case ' ': // make copy
createCopy();
break;

// < ... MY CODE

default:
    break;

if (threshChange) {
    gARTThreshhold += threshChange;
    if (gARTThreshhold < 0) gARTThreshhold = 0;
    if (gARTThreshhold > 255) gARTThreshhold = 255;
    printf("Threshold changed to %d.\n", gARTThreshhold);
}
}

static void Idle(void)
{
    static int ms_prev;
    int ms;
    float s_elapsed;
    ARUint8 *image;

    ARMarkerInfo *marker_info; // Pointer to 
    array holding the details of detected markers.
    int marker_num; // Count of 
    number of markers detected.
    int i, j;
    int m, s;
```
int k[9];
int idOfHiroMarker = -1;

// Find out how long since Idle() last ran.
ms = glutGet(GLUT_ELAPSED_TIME);
s_elapsed = (float)(ms - ms_prev) * 0.001;
if (s_elapsed < 0.01f) return; // Don’t update more
          often than 100 Hz.
ms_prev = ms;

// Update drawing.
arVrmlTimerUpdate();

// Grab a video frame.
if ((image = arVideoGetImage()) != NULL) {
gARTImage = image; // Save the fetched image.

gCallCountMarkerDetect++; // Increment ARToolKit FPS
        counter.

// Detect the markers in the video frame.
if (arDetectMarker(gARTImage, gARTThreshhold, &
        marker_info, &marker_num) < 0) {
    exit(-1);
}

save_m_info = marker_info;
save_m_num = marker_num;

// > MY CODE ...

for (j = 0; j < marker_num; j++) {
    // we need to remember where the hiro marker is
    if (marker_info[j].id == gObjectData[
            gObjectDataCount-1].id) {
        if (idOfHiroMarker == -1) idOfHiroMarker = j;
        else if (marker_info[j].cf > marker_info[
                    idOfHiroMarker].cf) {
            idOfHiroMarker = j;
        }
    }
s = -1;
// Check for object visibility.
for (i = 0; i < gObjectDataCount; i++) {
    // Check through the marker_info array which
    // markers are
    // compatible to the object’s pattern.
    for (m=0; m<9; m++) {
        k[m] = -1; // k[] stores all compatible markers
    }
    for (j = 0; j < marker_num; j++) {
        if (marker_info[j].id == gObjectData[i].id) {
            if (marker_info[j].cf > 0.75) {
                s = s+1;
                if (s != 9) {
                    k[s] = j;
                }
            }
        }
    }
}

// s is only -1 if no marker found in the video is
// compatible to the object pattern
if (s != -1) {
    for (j=0; j<9; j++) {
        if (k[j] != -1) {
            if (myObjectDataArray[i][j].visible == 0) {
                arGetTransMat(&marker_info[k[j]],
                               myObjectDataArray[i][j].
                               marker_center,
                               myObjectDataArray[i][j].
                               marker_width,
                               myObjectDataArray[i][j].trans);
            }
            else {
                arGetTransMatCont(&marker_info[k[j]],
                                   myObjectDataArray[i][j].trans,
myObjectDataArray[i][j].
    marker_center,
    myObjectDataArray[i][j].
    marker_width,
    myObjectDataArray[i][j].trans);

    myObjectDataArray[i][j].visible = 1;
  }
else {
    myObjectDataArray[i][j].visible = 0;
}
}

if (atLeastOneCopySet == 1) {
  if (idOfHiroMarker != -1) {
    for (m = 0; m <= noOfCopies; m++){
      myCopies_cont[m] = myCopies_prev[m];
  
      calculateCopyCont(myObjectDataArray[
          gObjectDataCount-1][idOfHiroMarker].
      trans, myBase_prev, myCopies_prev[m].
      trans, myCopies_cont[m].trans);

      myCopies_cont[m].visible = 1;
    }
  }
else {
  for (m = 0; m <= noOfCopies; m++){
    myCopies_cont[m].visible = 0;
  }
}

  // if none is compatible, then the objects should
  not be visible.
else {
  for (j = 0; j < 9; j++){
myObjectDataArray[i][j].visible = 0;
}
for(j=0; j<=noOfCopies; j++){
myCopies_cont[j].visible = 0;
}

}  // < ... MY CODE

// Tell GLUT the display has changed.
glutPostRedisplay();
}
}

// This function is called on events when the visibility of the
// GLUT window changes (including when it first becomes visible).

// static void Visibility(int visible)
{
    if (visible == GLUT_VISIBLE) {
        glutIdleFunc(Idle);
    } else {
        glutIdleFunc(NULL);
    }
}

// This function is called when the GLUT window is resized.

// static void Reshape(int w, int h)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

// Call through to anyone else who needs to know about
// window sizing here.
}

} //// This function is called when the window needs
// redrawing.

static void Display(void)
{
    int i, j;
    GLdouble p[16];
    GLdouble m[16];

    // Select correct buffer for this context.
   glDrawBuffer(GL_BACK);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //
    // Clear the buffers for new frame.

    arglDispImage(gARTImage, &gARTCparam, 1.0,
                    gArglSettings); // zoom = 1.0.
    arVideoCapNext();
goARTImage = NULL; // Image data is no longer valid
    // after calling arVideoCapNext().

    // Projection transformation.
    arglCameraFrustumRH(&gARTCparam, VIEW_DISTANCE_MIN,
                        VIEW_DISTANCE_MAX, p);
glMatrixMode(GL_PROJECTION);
glLoadIdentity(p);
glMatrixMode(GL_MODELVIEW);

    // Viewing transformation.
    glLoadIdentity();
    // Lighting and geometry that moves with the camera
for (i = 0; i < gObjectDataCount; i++) {
    for (j=0; j<9; j++){

        if ((myObjectDataArray[i][j].visible != 0) && (myObjectDataArray[i][j].vrml_id >= 0)) {
            //printf("Check [%i][%i].visible = %i\n", i, j, myObjectDataArray[i][j].visible);
            // Calculate the camera position for the object and draw it.
            // Replace VIEW_SCALEFACTOR with 1.0 to make one drawing unit equal to 1.0 ARToolKit units (usually millimeters).
            arglCameraViewRH(myObjectDataArray[i][j].trans, m, VIEW_SCALEFACTOR);
            glLoadMatrixd(m);

            // All lighting and geometry to be drawn relative to the marker goes here.
            //fprintf(stderr, "About to draw object %i\n", i );
            arVrmlDraw(myObjectDataArray[i][j].vrml_id);
        }
    }
} for (i = 0; i <= noOfCopies; i++) {

    if ((myCopies_cont[i].visible != 0) && (myCopies_cont[i].vrml_id >= 0)) {
        // Calculate the camera position for the object and draw it.
        // Replace VIEW_SCALEFACTOR with 1.0 to make one drawing unit equal to 1.0 ARToolKit units (usually millimeters).
        arglCameraViewRH(myCopies_cont[i].trans, m, VIEW_SCALEFACTOR);
    }
glLoadMatrixd(m);

// All lighting and geometry to be drawn relative
to the marker goes here.
//fprintf(stderr, "About to draw object %i\n", i);
arVrmlDraw(myCopies_cont[i].vrml_id);
}
}

// Any 2D overlays go here.
//none
glutSwapBuffers();

int chooseMode(void){
  int modeInput = 0;
  printf("Select a Mode: \n [1] Castle \n [2] Brick
Building \n");
  scanf("%d", &modeInput);
  getchar();
  return (int) modeInput;
}

int main(int argc, char** argv)
{
  int i;
  char glutGamemode[32];
  const char *cparam_name = "Data/camera_para.dat";
  //
  // Camera configuration.
  //
  #ifdef _WIN32
  char *vconf = "Data\WDM_camera_flipV.xml";
  #else
  char *vconf = "";
  #endif
  mode = chooseMode();
  int loop = 0;
  while (mode != 1 && mode != 2){
    printf("Illegal Input! Try it again! \n");
mode = chooseMode();
loop++;
if (loop == 200) {
    mode = 1;
}
char castleData[] = "Data/object_data_castle";
char brickData[] = "Data/object_data_brick";
char objectDataFilename[25];
if (mode == 1) {
    strcpy(objectDataFilename, castleData);
} else if (mode == 2) {
    strcpy(objectDataFilename, brickData);
}

// Library inits.
//
glutInit(&argc, argv);

// Hardware setup.
//
if (!setupCamera(cparam_name, vconf, &gARTCparam)) {
    fprintf(stderr, "main(): Unable to set up AR camera.
     .\n");
    exit(-1);
}

#ifdef _WIN32
    CoInitialize(NULL);
#endif

//
// Library setup.

// Set up GL context(s) for OpenGL to draw into.
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA |
                   GLUT_DEPTH);
if (!prefWindowed) {
    if (prefRefresh) sprintf(glutGamemode, "%ix%i:%i@%i",
                               prefWidth, prefHeight, prefDepth, prefRefresh);
    else sprintf(glutGamemode, "%ix%i:%i", prefWidth,
                               prefHeight, prefDepth);
    glutGameModeString(glutGamemode);
    glutEnterGameMode();
} else {
    glutInitWindowSize(prefWidth, prefHeight);
    glutCreateWindow(argv[0]);
}

// Setup argl library for current context.
if ((gArglSettings = arglSetupForCurrentContext()) ==
    NULL) {
    fprintf(stderr, "main(): arglSetupForCurrentContext()
    returned error.\n");
    exit(-1);
}
dbgReportMode(gArglSettings);
glEnable(GL_DEPTH_TEST);
arUtilTimerReset();
if (!setupMarkersObjects(objectDataFilename, &
    gObjectData, &gObjectDataCount)) {
    fprintf(stderr, "main(): Unable to set up AR objects
    and markers.\n");
    Quit();
}
int s,t;
for (s = 0; s<gObjectDataCount; s++){
    for (t = 0; t<9; t++){
        myObjectDataArray[s][t] = gObjectData[s];
    }
}
// Test render all the VRML objects.
fprintf(stdout, "Pre-rendering the VRML objects..."
    );
fflush(stdout);
glEnable(GL_TEXTURE_2D);
for (i = 0; i < gObjectDataCount; i++) {
    for (t = 0; t < 9; t++){
        arVrmlDraw(myObjectDataArray[i][t].vrml_id);
    }
}
glDisable(GL_TEXTURE_2D);
fprintf(stdout, " done\n");

// Register GLUT event-handling callbacks.
// NB: Idle() is registered by Visibility.
glutDisplayFunc(Display);
 glutReshapeFunc(Reshape);
 glutVisibilityFunc(Visibility);
 glutKeyboardFunc(Keyboard);

 glutMainLoop();

return (0);
The following shows the code of the class homogenius.c:

```c
/*
 * homogenius.c
 * This class realizes the matrix operations performed by arLego.c
 * when a copy is created. It uses the homogeneous coordinate
 * to calculate the movement of the copied wall.
 * Created by Tolan Druckmiller on 30.07.12.
 */

#include "homogenius.h"

#include<stdio.h>
#include<stdlib.h>

/**
 * This function performs the matrix multiplication and writes the result into
 * the variable "product[]\[]".
 *
 * var factor1[]\[] - matrix one which is multiplicated with
 * var factor2[]\[] - another matrix, both of the size 3x4
 * The homogeneous coordinate is added in this function to perform the
 * multiplication
 * var product[]\[] - the result of the multiplication
 */

void matrixMultiplication(double factor1[3][4], double factor2[3][4], double product[3][4]){
    int i,j;
    // here we need the homogeneous coordinate to perform the matrix multiplication
```
double productCalc[4][4], factor1Calc[4][4],
factor2Calc[4][4];
for (i=0; i<3; i++) {
    for (j=0; j<4; j++) {
        factor1Calc[i][j] = factor1[i][j];
        factor2Calc[i][j] = factor2[i][j];
    }
}
for (j=0; j<3; j++) {
    factor1Calc[3][j] = 0;
    factor2Calc[3][j] = 0;
}
factor1Calc[3][3] = 1;
factor2Calc[3][3] = 1;

// Note: factor1Calc * factor2Calc = productCalc
int k;
for (i=0; i<4; i++) {
    for (j=0; j<4; j++) {
        productCalc[i][j] = 0;
        for (k=0; k<4; k++) {
            productCalc[i][j] += factor1Calc[i][k] * factor2Calc[k][j];
        }
    }
}

for (i=0; i<3; i++) {
    for (j=0; j<4; j++) {
        product[i][j] = productCalc[i][j];
    }
}

/***
** This function calculates the quotient in a matrix division (which by
** definition doesn’t exist). The matrices are 3x4, whereas the matrix
** multiplication would need the following: \( X \times A = B \)

with \( X,A,B \) as 4x4 ** matrix (HOMOGENEOUS COORDINATE!). We ignore the forth row in this case.

** The function just needs to know whether a translation or a rotation takes place.

** var translation = 1 for a translation

** = 0 for a rotation

**

** x00 x01 x02 x03 a00 a01 a02 a03 b00 b01 b02 b03
** x10 x11 x12 x13 * a10 a11 a12 a13 = b10 b11 b12 b13
** x20 x21 x22 x23 a20 a21 a22 a23 b20 b21 b22 b23
**

** ie. \( x00\times a00 + x01\times a10 + x03\times a20 = b00 \)

**

** var dividend[][] - the dividend matrix of the division

** var divisor[][] - the divisor matrix of the division

** var quotient[][] - the result of the division

** var translate - equals 1 for a translation, equals 0 for a rotation

**/

void matrixDivision(double dividend[3][4], double divisor[3][4], double quotient[3][4], int translate){
    // B / A = X
    // dividend / divisor = quotient
    // X * A = B
    // quotient * divisor = dividend
    int i,j;

    /**
    ** If translate is set to 1 then quotient is:
    ** 1 0 0 *
    ** 0 1 0 *
    ** 0 0 1 *
    ** Only * must be calculated
    /**

    **/
if (translate == 1) {
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            if (i != j) {
                quotient[i][j] = 0;
            } else {
                quotient[i][j] = 1;
            }
        }
    }
    /**
    ** b03 = a03 + x03
    ** b13 = a13 + x13
    ** b23 = a23 + x23
    **/
    quotient[i][3] = -divisor[i][3] + dividend[i][3];
}
/**
** Here the rotation will be calculated
**/
else {
    /**
    ** for the rotation the last column is 0 0 0
    **    * * * 0
    **    * * * 0
    **    * * * 0
    **/
    for (i=0; i<3; i++) {
        quotient[i][3] = 0;
    }
    /**
    ** The rest is calculated through the determinant D
    ** and
    ** complicated linear algebra.
    **/
    double D;
    D = ( divisor[0][0] * divisor[1][1] * divisor[2][2]
    + divisor[1][0] * divisor[0][2] * divisor[2][1] + divisor[2][0] * divisor[0][1] * divisor[1][2] );
\[\begin{align*}
- (\text{divisor}[0][0] & \* \text{divisor}[2][1] & \* \text{divisor}[1][2] \\
& + \text{divisor}[1][0] & \* \text{divisor}[0][1] & \* \text{divisor}[2][2] \\
& + \text{divisor}[2][0] & \* \text{divisor}[1][1] & \* \text{divisor}[0][2] ) \}
\]

quotient[0][0] = ((\text{divisor}[1][0]*\text{divisor}[2][2]*
- \text{dividend}[0][1]+\text{divisor}[2][0]*\text{divisor}[1][1]*)
- \text{dividend}[0][2]-\text{dividend}[0][0]*\text{divisor}[2][1]*
\text{divisor}[1][2])-(\text{divisor}[1][0]*\text{divisor}[2][2]*
- \text{dividend}[0][1]+\text{divisor}[2][0]*\text{divisor}[1][1]*
\text{divisor}[0][2])}/D;

quotient[0][1] = ((\text{divisor}[0][0]*\text{divisor}[2][1]*
- \text{dividend}[0][2]+\text{divisor}[2][0]*\text{divisor}[0][2]*
\text{divisor}[0][1]*)
- \text{dividend}[0][1]-\text{dividend}[0][0]*\text{divisor}[2][1]*
\text{divisor}[2][2])-(\text{divisor}[0][0]*\text{divisor}[2][2]*
- \text{dividend}[0][1]+\text{divisor}[2][0]*\text{divisor}[0][1]*
\text{divisor}[0][2])}/D;

quotient[0][2] = ((\text{divisor}[0][0]*\text{divisor}[1][2]*
- \text{dividend}[0][1]+\text{divisor}[1][0]*\text{divisor}[0][1]*
\text{divisor}[0][2])-(\text{divisor}[0][0]*\text{divisor}[1][1]*
\text{divisor}[2][2])-(\text{divisor}[0][0]*\text{divisor}[2][1]*
\text{divisor}[1][2]))}/D;

quotient[1][0] = ((\text{divisor}[1][0]*\text{divisor}[2][2]*
- \text{dividend}[1][1]+\text{divisor}[2][0]*\text{divisor}[1][1]*)
- \text{dividend}[1][2]-\text{dividend}[1][0]*\text{divisor}[2][1]*
\text{divisor}[1][2])-(\text{divisor}[1][0]*\text{divisor}[2][1]*
\text{divisor}[2][2])-(\text{divisor}[1][1]-\text{dividend}[1][0]*\text{divisor}[1][1]*
\text{divisor}[2][2]})}/D;

quotient[1][1] = ((\text{divisor}[0][0]*\text{divisor}[2][1]*
- \text{dividend}[1][2]+\text{divisor}[2][0]*\text{divisor}[0][2]*
\text{divisor}[2][1])-(\text{divisor}[0][0]*\text{divisor}[2][2]*
- \text{dividend}[1][1]-\text{dividend}[1][0]*\text{divisor}[0][2]*
\text{divisor}[2][2])-(\text{dividend}[1][2]-\text{dividend}[1][0]*\text{divisor}[2][1]*
\text{divisor}[2][2]})}/D;
divisor[0][2]))/D;

quotient[1][2] = ((divisor[0][0]*divisor[1][2]*-
  dividend[1][1]+divisor[1][0]*divisor[0][1]*
  dividend[1][2]-dividend[1][0]*divisor[1][1]*
  divisor[0][2])-(divisor[0][0]*divisor[1][1]*-
  dividend[1][2]+divisor[1][0]*divisor[0][2]*-
  dividend[1][1]-dividend[1][0]*divisor[1][2]))/D;

quotient[2][0] = ((divisor[1][0]*divisor[2][2]*-
  dividend[2][1]+divisor[2][0]*divisor[1][1]*
  dividend[2][2]-dividend[2][0]*divisor[2][1]*
  divisor[1][2])-(divisor[1][0]*divisor[2][1]*-
  dividend[2][2]+divisor[2][0]*divisor[1][2]*-
  dividend[2][1]-dividend[2][0]*divisor[2][2]))/D;

quotient[2][1] = ((divisor[0][0]*divisor[2][1]*-
  dividend[2][2]+divisor[2][0]*divisor[0][2]*-
  dividend[2][1]-dividend[2][0]*divisor[2][2]*
  divisor[0][2])-(divisor[0][0]*divisor[2][2]*-
  dividend[2][1]+divisor[2][0]*divisor[0][1]*-
  dividend[2][2]-dividend[2][0]*divisor[2][1]*
  divisor[0][2]))/D;

quotient[2][2] = ((divisor[0][0]*divisor[1][2]*-
  dividend[2][1]+divisor[1][0]*divisor[0][1]*
  dividend[2][2]-dividend[2][0]*divisor[1][1]*
  divisor[0][2])-(divisor[0][0]*divisor[1][1]*-
  dividend[2][2]+divisor[1][0]*divisor[0][2]*-
  dividend[2][1]-dividend[2][0]*divisor[1][2]))/D;

}  
}  

/**
** This function rounds the values in a 3x4 matrix to 0 or 1.
**
** var mat[][] = the rounded matrix
**/
void rounding(double mat[3][4]){
    int i,j;

    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            if (mat[i][j] >=0) {
                mat[i][j] = mat[i][j] + 0.5;
            } else {
                mat[i][j] = mat[i][j] - 0.5;
            }

            mat[i][j] = (int) mat[i][j];
        }
    }

    /**
    ** This function is called in every idle as soon as at least one copy is set.
    ** It uses functions in this class to calculate the position of the copied walls.
    ** base_cont[][] - position of the base in the current image
    ** myBase_prev[][] - the position of the base to the time point when the copy was created.
    ** it is most likely parallel to the ground at the origin, in a 90 degree angle to the camera
    ** myCopies_prev[][] - the position of the wall pattern to the time point when it was copied. Here, the copy is transformed in relation to the base
    ** myCopies_cont[][] - the result of this calculation
    **/
    */

    void calculateCopyCont(double base_cont[3][4], double myBase_prev[3][4], double myCopies_prev[3][4], double myCopies_cont[3][4]){
/** Translate base to origin
 ** transformationMat is the matrix that, when you
 ** multiply it
 ** with myBase_prev, sets the base to the origin.
 ** The copy needs to be translated in the same
 ** manner. This basically
 ** means that the copy is still in the same
 ** position relative to the
 ** base, but the base is translated to the origin.
 ***/
double transformationMat[3][4];
int i, j;

/** Rotate the base
 ** We need to find the matrix that rotates
 ** myBase_prev into the position
 ** of myBase_cont. This matrix is calculated in
 ** transformationMat. The
 ** rotation is also performed by the copy.
 ***/
double baseOrigin_cont[3][4];
for (i=0; i<3; i++){
    for (j=0; j<4; j++){
        baseOrigin_cont[i][j] = base_cont[i][j];
    }
}
for (i=0; i<3; i++){
    baseOrigin_cont[i][3] = 0;
}
matrixDivision(baseOrigin_cont, myBase_prev, 
               transformationMat, 0);
matrixMultiplication(transformationMat, myCopies_cont, 
                    myCopies_cont);

/**
 ** Translate back to the position where the base is
 ** now.
 ** Same as before. We Translate the base to where
 ** it is now and need to
** find the matrix when multiplied with brings it into that position.**
** The copy needs to be multiplied with this matrix as well.**
/**
matrixDivision(base_cont, baseOrigin_cont, transformationMat, 1);
matrixMultiplication(transformationMat, myCopies_cont, myCopies_cont);
}

The following shows the code of the header homogenius.h:

```c
/*
 * homogenius.h
 *
 * Created by Tolan Druckmiller on 30.07.12.
 *
 */

#include<stdio.h>
#include<stdlib.h>

void rounding(double mat[3][4]);
void shiftValues(double factor1[3][3], double factor2[3][3]);
void matrixMultiplication(double factor1[3][4], double factor2[3][4], double product[3][4]);
void matrixDivision(double dividend[3][4], double divisor[3][4], double quotient[3][4], int translate);
void calculateCopyCont(double base_cont[3][4], double myBase_prev[3][4], double myCopies_prev[3][4], double myCopies_cont[3][4]);
```
D User Guide

Getting started:
Drag and drop the application into your Applications folder. Double click the application to start it. On doing so, the application will be loaded in the castle mode by default, which means that the augmented 3D models will be walls of a castle. If you don’t want to run the application in castle mode, you will have to start the application from terminal. To do so, start the terminal (press [COMMAND] + [SPACE] to open Spotlight, type "Terminal", and press [ENTER]), and type in:

    cd /Applications/ARLego.app/Contents/MacOS

Now run ARLego from command prompt:

    ./ARLego

On doing so, the terminal will ask you to select a mode. Press "1" for the castle mode and "2" for the brick building mode.

Camera configuration:
The application loads the according VRML files, and will display a video configuration panel. Here, you can change the settings of your camera image, and also select another source, if you have more that one camera connected to your computer. Usually the default settings work fine, under normal (not too bright and not too dark) light circumstances. Press [ENTER], after you have configured the camera settings.

Create your AR world:
As soon as you see one window with the camera image, you can start building the patterns. You can find pictures of the patterns in Figure .1 and .2. If you want to know which pattern will be represented through which virtual wall, you can just print out the according pattern and hold it towards the camera. You can also just build it with the LEGO blocks and see what happens.
Start reconstructing the patterns and set them together in whichever way you want: Next to each other, on top of each other, in a $90^\circ$ angle, etc. Figure .3 shows an example setup. The left picture shows the real world construction, while the right one shows the augmented walls. Watch your fingers! If you cover a part of a pattern, the system will not be able to recognize it.
Extend your creation:

If you haven’t finished your building yet and you’re running out of suitable LEGO bricks, you have the option to copy virtual walls. You can copy as many walls at the same time as you can fit into the camera image. To do so, place the LEGO patterns whose representation you want to copy anywhere behind the base pattern, shown in Figure 4. Move it, so that all patterns are shown in the camera image, and not covered by anything. The base pattern should be surrounded by a green boarder. Every virtual wall that you see in the camera image augmented on the patterns will be copied, when you press the [SPACE] bar.

Remove the patterns from the base plate and you will see, that each virtual wall will move depending on where you move the base pattern. It looks as if the patterns would still stick on the plate. You can always add more virtual walls to the building you create in just repeating this procedure. If you have to add a virtual wall on top of an already copied virtual wall, you can just build a wall as high as a pattern with any bricks (don’t use the same pattern again, or cover it in the copy process), and stick the pattern whose representation you want to copy on top of it.

If you want to undo your last action, press the [z] key. Actions older than the last, can be undone by pressing the [z] key multiple times.

Figure 4: Copy Process
Bibliography


