Group	5
Steven Bos	1319671
Andy Man	1047612
Kenneth Touw Ngie Tjouw	1163256

Content

Introduction

This report describes the last step in the development process of the project Max the Robot, an educational application, which teaches children the basic concept of programming. The main focus in this step is the technical solution that fit the requirements and constraints that have been extracted during the design phase.

This brings us to the following problem definition: 'How can we create a technical realization of Max the Robot and how can we proof the concept'.

To answer the problem definition we split it in different components. We start with the technical solution, without a technical solution we are not able to proof the concept, because we cannot run tests and evaluations without a prototype. The technical realization of the application contains several aspects that are described in this report. These aspects are the platform, programming language, tools and frameworks. We also describe the problems that we encountered and the different solutions.

After the technical realization we evaluate the product against its initial design. After the product has been developed, we want to proof that the application can indeed teach children the basics of programming. The aspects that we have to deal with are the experiment setup, measuring relevant data and interpretation of the measured data.

The report is build up as follows: In chapter one a recap is given about the design phase. In chapter two we give a description of the implementation phase and in chapter three we continue with the test and evaluation phase. In the last chapter we draw a conclusion, followed by the Appendix.

1. About the product

1.1 Chapter summary

This chapter consists of short summaries of the goal, our target group definition defined in the Design Report and what the educational value of our project is.

1.2 Goal

The goal for this project is the development of a piece of educational software, which supports the gain in interest and basic knowledge in the field of ICT for primary school children (aged 11-12), by teaching children the basics of programming in a visual and interactive way.

To achieve this goal, the children should be willing to use the educational software. A way to encourage them to use the educational software is to develop it as a game with the Robot as a toy. Since children are easily distracted, this is a suitable way to keep them interested.

We have chosen to develop a tool called “Max de Robot”. The tool consists of a screen with limited options according to his or her level. When so called “block code” is dragged and dropped in the block code screen and the compile button is hit, the user made program will upload (wirelessly, by Bluetooth) and run on a Lego Mindstorm Robot. The robot will perform the instructions of the user made program in real life.

1.3 Target group

A short target group definition is mentioned below, for the full version please refer to the design report.

The target group is children in the last grade of Dutch primary schools. Characteristics of this group are that it consists of:

- boys and girls,
- children normally ranging between the age of 11 and 12,
- a huge diversity in intelligence,
- a diversity in reading skills,
- a diversity in computer using skills.

1.4 Educational Value

After using the tool Max de Robot, the children should have knowledge of the following aspects:

- What building blocks are necessary in software program?
- What steps have to be performed to create a piece of software to control a piece of hardware?
- How logical operators work?
- Make aware that creating a software program needs a sequential plan?
- What motors are and what is possible with it?
- How to create a piece of hardware by creating a piece of software?
- What Bluetooth is and that it can be used to transmit wireless commands to a hardware device?

We also hope to achieve to interest the children in technology, make aware that electronically devices around them are controlled by software, that programming can be fun and that imagination is the limit when combining software and hardware / a lot is possible.

2. Implementation

2.1 Chapter summary

This chapter consists of a description and motivation for the chosen platform and tools. We talk about the user interface and the implementation process itself.

2.2 Platform

In general when developing software applications, there are several possibilities for choosing a platform. The major platforms that are widely used around the world are Microsoft Windows, Apple OS and Linux. Of course it is possible to develop a platform independent application, for example with Java or using web technologies.

From our field research in April, we learned that all primary schools work with the Windows operating system. So for an application, who's target group is primary school children, it is clear that it must be able to run on a Windows platform. That means that we have a choice in developing solely a Windows applications or a platform independent application. The last option appears to be least achievable, because of issues that are related to properties of the NXT Lego Robot that we use in this project. We will come back to this later in this chapter.

2.3 Tools

2.3.1 Tools used

There are several tools we have used during the development of Max the Robot. For the sake of clarity we divided them in three categories.

1. Programming
    a. NXT# framework
    b. Visual C# programming language (based on .NET framework)
    c. Visual Studios 2005 (Development environment)

2. Communication
    a. NXT Bluetooth protocol

3. Hardware
    a. Bluetooth dongle
    b. NXT Lego Robot

2.3.2 Motivation chosen tools

In this paragraph a description is given of the motivation of the chosen tools in the last paragraph. It will also become clear what issues caused us to choose for a Windows application instead of a platform independent application.

We start with the motivations for the programming tools. When one starts developing a software application, there seem to be an endless offering of programming languages, Development environments and frameworks. Most of the time the properties of the application to be developed, limits the possible choices. In our case, the major issue is the ability to control the NXT Lego Robot. This meant that we had only a few choices left concerning

the programming tools.

Framework	Language	Open source	Difficulty	IDE
NXT#	C#	No	Intermediate	**VS 2005**
NXT J	Java	Yes	Easy	**Eclipse**
NXT C	C	No	Hard	**VS 2005**
Realterm	Matlab script/C	No	Hard	**Matlab**

There are several programming tools listed in the table above. From that table we chose NXT# as the framework of our application. NXT C and Realterm were dismissed, because they are very complex frameworks and we do not have the time to figure them out. NXT J is a frame that is built in Java and therefore it would make the application platform independent. This would be a very good choice, except for the fact that we had to replace the firmware of the NXT Robot. Since we only have one Robot and a limited amount of time, we cannot afford to take the risk that something goes wrong with the firmware, so also this attractive option have to be dismissed. This leaves us with the NXT# Framework.

NXT# has been built in C#. An Object Oriented programming language that uses the Microsoft .NET framework. The ideal development environment, probably the only one, is Visual Studio. This framework facilitates the control of the robot. It makes it possible to control the NXT Lego Robot from a Visual C# (meaning rich user interface) application.

The tools that are used for communication with the robot are very limited. Actually there are only two possibilities. The first one is by a wired USB connection. The second is wireless by the Lego Bluetooth Protocol. There are two main reasons why we dismiss the first option directly; we wanted the robot to be able to move freely. It is a must have, that the robot can move anywhere in a room or even in a building. A wired communication would make this completely impossible. Fortunate, the framework that we have chosen, communicates via the Lego Bluetooth protocol.

Choices concerning the hardware haven't been considered. A Bluetooth dongle (with WIDCOMM chipset) is necessary in order to be able to communicate via Bluetooth. An indispensable feature of this whole project is the NXT Lego Mindstorm Robot.

2.3.3 Positive / negative aspects of tools used

Though the choices made in what kind of tools to use during this project, there are some drawbacks. Below a summation is given of the pros and cons of the tools used.

NXT# framework

pros:

- Easy to integrate in our own application
- Solid communication facility with the robot

Cons:

- No complete version (only part of the functionality of the robot can be controlled from this framework.
- Lack of support
- Lack of documentation

Visual Studios

Pros:

- A very extensive development environment with an excellent functional support

Bluetooth

Pros:

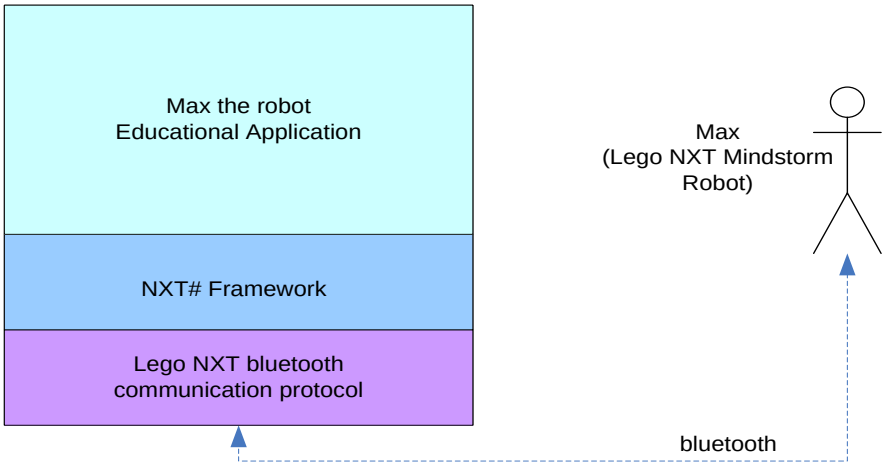
- Wireless, so the robot can move around freely within a range of 100 meters.

Cons:

- Interference with other Bluetooth stations, which can interrupt the communication
- Computer setup of Bluetooth communication is more difficult than USB setup

2.3.4 Model

After consideration of the tools that we have used to develop the educational application 'Max the Robot'. It's time to give a high level overview of the components of the application, before we start to talk detailed features, as we will do in the next paragraph.



2.4 Recap design

2.4.1 Paragraph summary

This paragraph will summarize the metaphors, functionality, interaction and education material discussed in our design report. Almost every discussed element had made it in the final program, except the integration of the AND, OR, NOT operators. They might be included in later versions.

2.4.2 Metaphors

To translate full or “for adult” programming to child programming we thought of the following metaphors to make it accessible, yet close to the real thing.

Instead of	We use
The programming language “c” or java,pascal,etc	The self-made programming language “blokcode”, referencing the Lego block and the fact that Lego block are build on top of each other, just like real code
Lego Mindstorm Robot model Alpha Rex	The name Max the robot, to make the robot more human like
Formal text, explanations	Child friendly text and more extensive help pointers
Common window theme	Build theme, with pipes, walls, Lego bricks and lots of color

2.4.3 Functionality

Functionality is discussed in our design report, therefore we pick some of the most interesting functionality and have the results of our tests integrated with them.

- Drag and drop programming (object orientated programming)
- User friendly pre compile messages
- Bright colors and “cool” user interface according to tests group.
- Easy and fun way to learn programming, boys/girls, computer experienced and non experienced. All children were able to comprehend the educational material and work with the application almost naturally.
- Four phase programming, so the user is always in control of the behaviour of the robot.
- No teacher assistance/attention required, so every user can work independently due to build in tutorials and user friendly help messages.

2.4.4 Interaction

Because interaction is extensively discusses in our design report, only key interaction is summarized here.

The user interacts to the computer as well as the robot. Interaction changes when the user gains experience with the application. When the user has a novice level (completed the first 9 lessons), the user has access to all interaction features (but not all functionality!!). One of these options we call “*Four Phase Programming*”. All important aspects of programming are represented in this new method; objects, properties of object and their values, debugging, code translation from abstract code to simplified Dutch, etc.

Other interaction is done to the robot. The user can program the robot to let his behavior depend on one of his three sensor inputs (if..else statement). The user has the choice to either let the environment produce input for the sensors or do it him/herself.

2.4.5 Educational material

The package we can provide any user of our target audience

- Application Max the robot which is a Computer Aided Learning tool with build-in knowledge to be learned. The list of Educational features can be found in our design report.
- One exercise ("Help Max through the maze") to train the user. Exercises are meant to let the user program the robot to deal with real life problems. When we continue with this project we can build a complete set of exercises and tests to create a complete course "interactive programming for children".
- A manual which covers all user interface controls

Beside the above material we are working on providing the following:

- In-dept background on programming for primary school teacher including weblinks so they can answer basic questions and show them the next logical step in programming for children (for example the Scratch language)

2.5 Implementation process

2.5.1 Paragraph summary

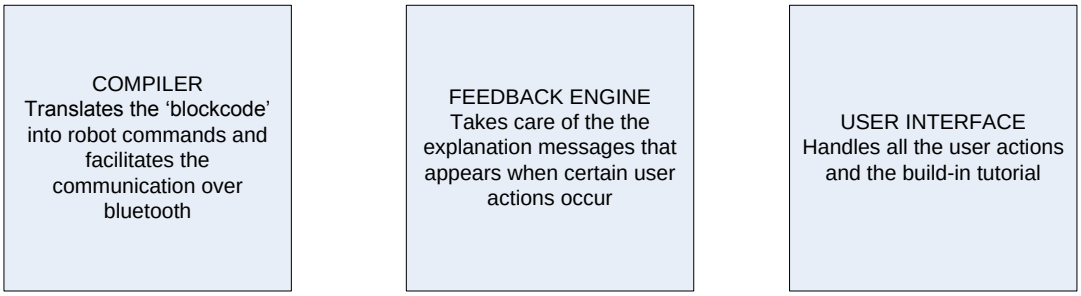
We faced various problems during this entire project. This paragraph will enlighten the reader what problems we dealt with and how. First, we present our general implementation process including an overview diagram. Second, all problems including the way we dealt with them are presented under categorized headings.

2.5.2 Implementation Process

Basically we divided the whole application in three components, the user interface, the compiler and the feedback engine. Every team member got full responsibility over one of these components. Of course as we are a team, we helped each other a lot to overcome the problems one faced. Since this project is a quite new and therefore unsupported area, we had to do lots of research to existing application and apply reverse engineering to get insight in the technology.

Thus during the implementation process lots of researched needed to be done on the fly since it was impossible to predict the problems we would encounter. The strategy used during this implementation process, is one of iteration. Small applications (off the different components) we build, integrated and tested. After a successful test session, the components were expanded with new functionalities and the process went all over again. This way we were able to have a stable version all the time, which was very beneficial during debugging.

Below a short overview of the three different components



Next we give an overview of the problems we have encountered during the implementation process.

General problems

Problem: Standard development issues like project management and team commitments

Solution: Regular phone/personal meetings, informal agreements and fixed role assignments

Problem: The project in this form –teaching young children programming with a robot, has never been done. Therefore all phases of the design, development and testing takes longer time than with standard projects.

Solution: No real solution; just commitment, hard work and effort to constrain the application (with continues new insight) to fit the deadline.

Project problems

Problem: All team members were not familiar with the Visual Studio 2005 IDE, aswell as the Visual C# language.

Solution: Intensive training and reading to quickly adapt to new environment. Other course- and private activities had to be postponed.

Problem: Unable to do risque analysis, due to the uniques of this project.

Solution: Brainstorm sessions with all three members when serious development problems rose. We contacted the author of NXT# (Bram Fokke) for technical support with his software.

Design problems

Problem: Designing build-in tutorials, due to deadline

Solution: One the our selling points of our application is that the user doesn't need any assistance/attention of a teacher. Therefor, the build-in tutorials must be integrated in our prelimanry test session. To quickly integrate them we made 9 tutorial lessons and made the exercise information window build-in for 5 lessons. The other four exercises were printed on plain paper. This way we could also test the information preference of the user (build-in or on paper). See Chapter 3 for the result.

Problem: Interface for mature audience is found to be "crowded"

Solutioni: No solution yet, but one of our future improvements would be two different backgrounds to suit both young and mature audiences.

Presentation problems

Problem: Robot volume is too soft without amplifier and 10 minutes time slot for presentation is short

Solution To make the presentation more attractive we planned to let the robot do various tasks like saying a few words and perform some real test on viewing audience.

Testing problems

Problem: Robot doesn't have much grip on smooth surfaces

Solution: We need to buy either a portable rubber surface or improve the grip mechanisms on the robot. For the test we used a roughly finished wooden table which was luckily available.

Future improvements

- Add a competition element for individuals or teams to make programming even more fun. At least two NXT robots are required.
- Two types of background to support both young and mature audiences.
- Improve user experience by adding more dynamic elements like short movies or flash controls.
- Add help functions for every control element
- Increase workspace from 800x600 pixels to 1024x786 or higher to allow future functions
- Improve tutorials by adding a lesson checklist
- Build-in tutorials for lesson 6-14
- Add a user registration form who keeps track of user progress
- Enable sensor support on robot. For this function multi threading is required.
- Create a course "interactive programming for children" which includes 14 tutorials lessons, a few practical assignments and a final test.
- Create a manual for teachers to make them feel comfortable with the application and robot.

2.6 Lessons learned

In this section of the report the lessons learned about the implementation, the teamwork and educational software are treated.

2.6.1 Implementation

Lessons learned during the implementation phase are:

- Make good agreements on variable names, this will save a lot of time when changes have to be made.
- Make more use of classes for better code and lesser debugging.
- Think before coding. Discuss rigorous and think good about important decisions before implementing them.

2.6.2 Teamwork

Lessons learned during the teamwork are:

- Arrive at time for working together for more productive days.
- Choose a better work environment earlier in the project than the canteen.
- Make a better planning and stick to it.
- Do not be over ambitious with projects, be more realistic.

2.6.3 Educational software

Lessons learned during the educational software lessons are:

- Children and adults have different ideas on color scheme.
- Children and adults have different ideas on an useful interface.
- Do not underestimate the ICT-skills of children.
- Children are less afraid of making errors than adults.
- Include children earlier in the project.

3. Testing Data

3.1 Chapter summary

In order to test the prototype version of our tool, we contacted a primary school Klimop -Hoeve from the village Bleiswijk. The test will be performed with children from the 8th grade. For the first test 5 children will individually be put in front of a computer with our tool and with the Lego Robot connected wirelessly. The choice for 5 children is that it is a primal number and that it is not too small, not too large to control the test session and when taking conclusions from the test data, the choices for Yes and No would not be equal. The total amount of time of the test session should not take more than half a day. So with preparations and final dialogues with the teacher and principal only 2-3 hours is left to test with the children.

Furthermore in this chapter the process of testing our tool, which is developed by the guidelines of chapter two, is discussed. First we have the experiment set up, after conducting the experiment, the results are evaluated. Data gathered will be categorized and finally interpreted to form conclusions and draw suggestions from it.

3.2 Experiment setup

In the experiment we want to get answers revolving around the problem definition: 'How can we create a technical realization of Max the Robot and how can we proof the concept'. With the focus on the part how we can proof the concept.

In order to answer this, the following questions need to be answered during the experiment:

- Is the target group able to perform the tasks in the lessons?
- Is the positioning of the element clear for the target group?
- Does the target group understand the concepts we try to educate them throughout the lessons?
- Are there any items we missed, when developing the tool?
- Is the difficulty level of the tool for appropriate for the target group?

A test session for a child consists of:

Test parts	Duration
an explanation of what the tool is and what the child have to perform	5 min
the execution of the test assignments	10 min
a final questionnaire.	5 min

The total time of a test session has to be less than half an hour, due to a decrease of interest and concentration of the child, when performing such a session with strangers looking at him/her. The ideal length of the session should come near to 20 minutes, due to concentration time spans of children are approximately 5 minutes. The explanation will roughly take 5 minutes, the questionnaire should be short and concrete, otherwise the child will not be willing to perform it, or just mark randomly at the end, and this will be limited to 5 minutes again. So the most interesting part, and the part where we think we can catch the concentration of the children longer is the testing part, will have a duration of 10 minutes. The application is of course build to entertain and teach a child for longer than this period.

Within two hours the testing has to be performed, so the class is not distracted too long. Afterwards a 30 minute discussion with the teacher and principal will be conducted.

Below the three parts of the test session will be explained.

3.2.1 The explanation of the project "Max the Robot"

In a children friendly version the project is being explained, this explanation can be found in appendix A,B and C. The tasks the children have to perform are also explained. The children have to follow a list of assignments and perform them as best as possible.

3.2.2 The test assignments

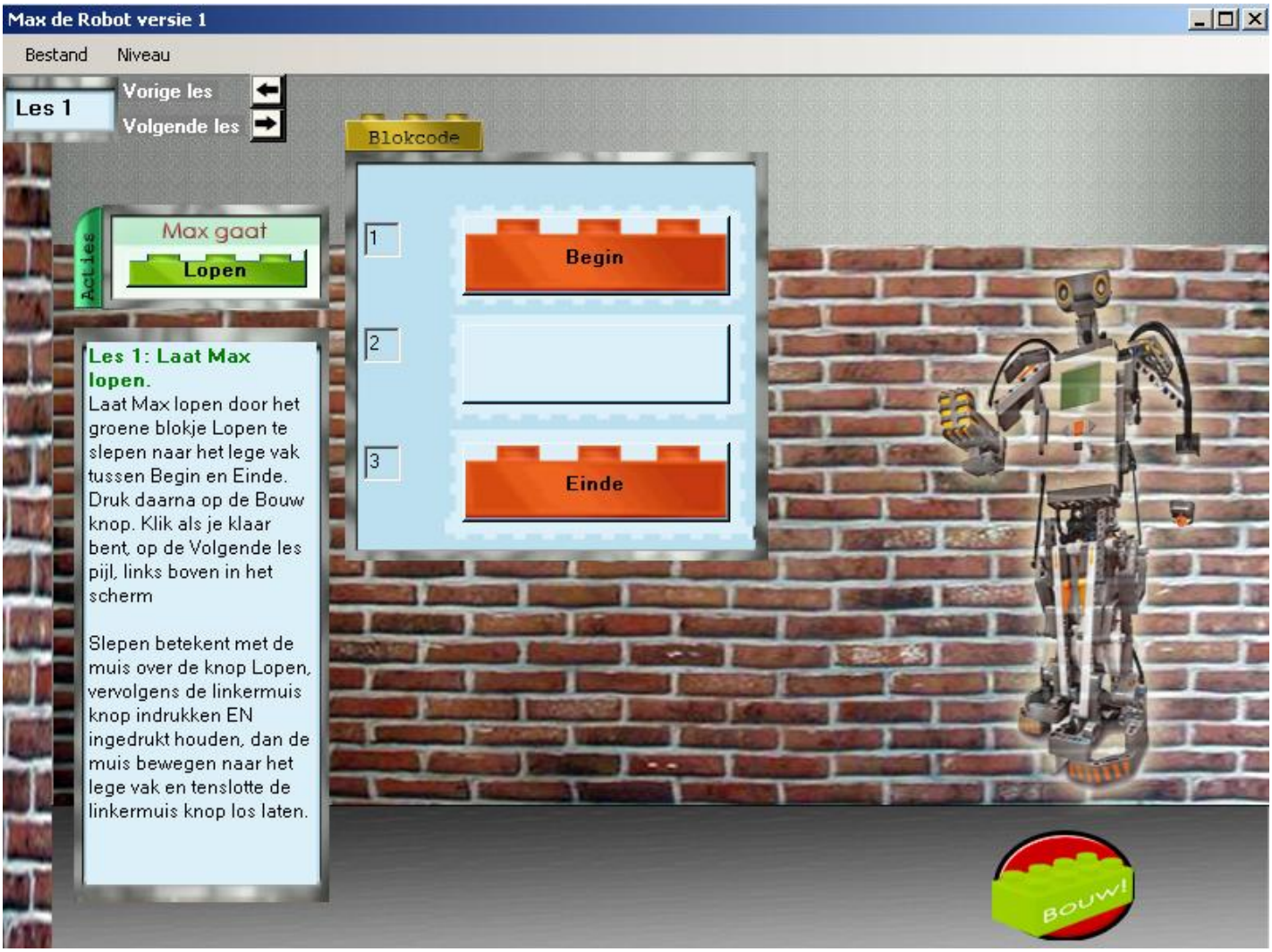
After the brief introduction, the user is placed in front of a computer which already runs the Max the Robot application. We prepared 9 lessons and an assignment, but we predict that our test users will only complete half of the lessons in 10 minutes. All lessons are meant to familiarize the user with the controls of the application and will teach the basics of programming. After the lessons the user could be given an assignment like “let Max walk through a simple maze”, to utilize all his thought knowledge of the controls and programming logic.

The lesson are structered as follows:

- The first 4 lessons introduces the 4 actions the robot can perform; walk, talk, move arm/head, and show a drawing on his belly. The actions talk and drawing are in this prototype version not integrated.
- Lesson 5 introduces the blokcode explanation facilities. This information screen translates abstract blokcode to simplified Dutch.
- Lesson 6 introduces the blocks “Begin“ and “Einde”.
- Lesson 7 upgrades the amount of empty tiles from 6 to 20, to give the user 18 possible slots for sequential action commands.
- Lesson 8 gives the user the possiblity to switch tiles from above or beneath.
- Lesson 9 introduces the property screen. This screen will teach the user that every building block (object), except “Begin” and “Einde”, has properties and these properties have values.

After these 9 lessons, the user is familiar with our Four Phase object orientated programming method. When further developed, the program will facilitate 14 lessons which teaches the user the If..Else statement and the use of three sensor inputs. All lessons and assignments can been done without any teacher input.

The user will start with lesson 1 and sees this screen:



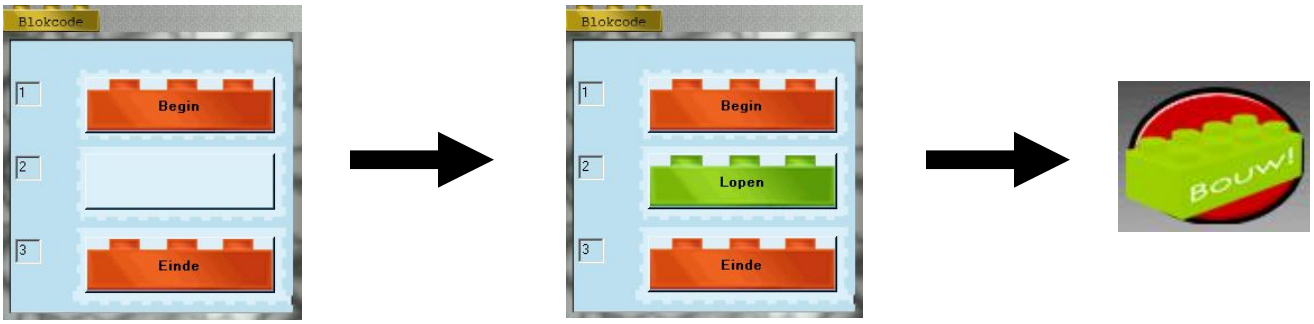
As mentioned in our design report, the user must be able to use the application without any teacher input. In this prototype version we introduce an information screen to inform the user of the controls. Though the principle of drag and drop is visibly very easy to show, it is not so easy to explain the same in text. In possible later versions of this application we might use a 3 second short video to show how easy it is to perform it.

The goal of lesson 1 is to show the user:

- basic controls of the application
- How easy it is to make to robot do something (this makes it fun!)

The user makes the robot walk by simply drag an action button and drop it to a free tile and hit the “Bouw” button. All intelligence, like which direction and how many steps are pre-programmed. In lesson 9 the user will learn to

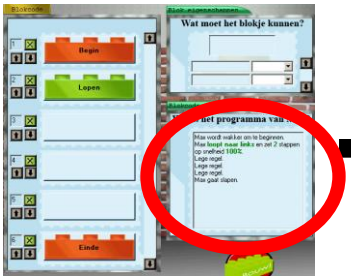
modify these properties him/herself.



Step 1 Step 2 Step 3

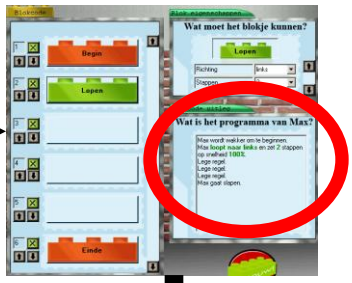
All lessons are built in such a way that the user can concentrate on the goal of the lesson and can complete the lesson in very little steps. Even one of our most advanced lessons, lesson 9, requires but 5 steps to complete. In this prototype version, lesson 6-9 are supported by a small paper handout with these steps, because the information screen used in lessons 1-5 is to small when the “Begin” and “Einde” building block are introduced. In a possible future version smarter ways, like step by step instructions, will be used to inform the user.

The 5 steps of lesson 9 are as follows:



Step 1

Step 2

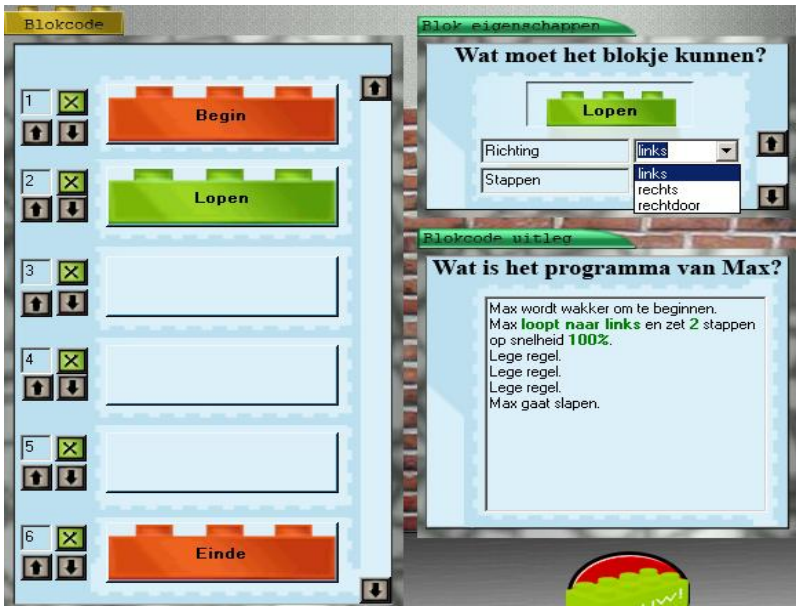


Step 3

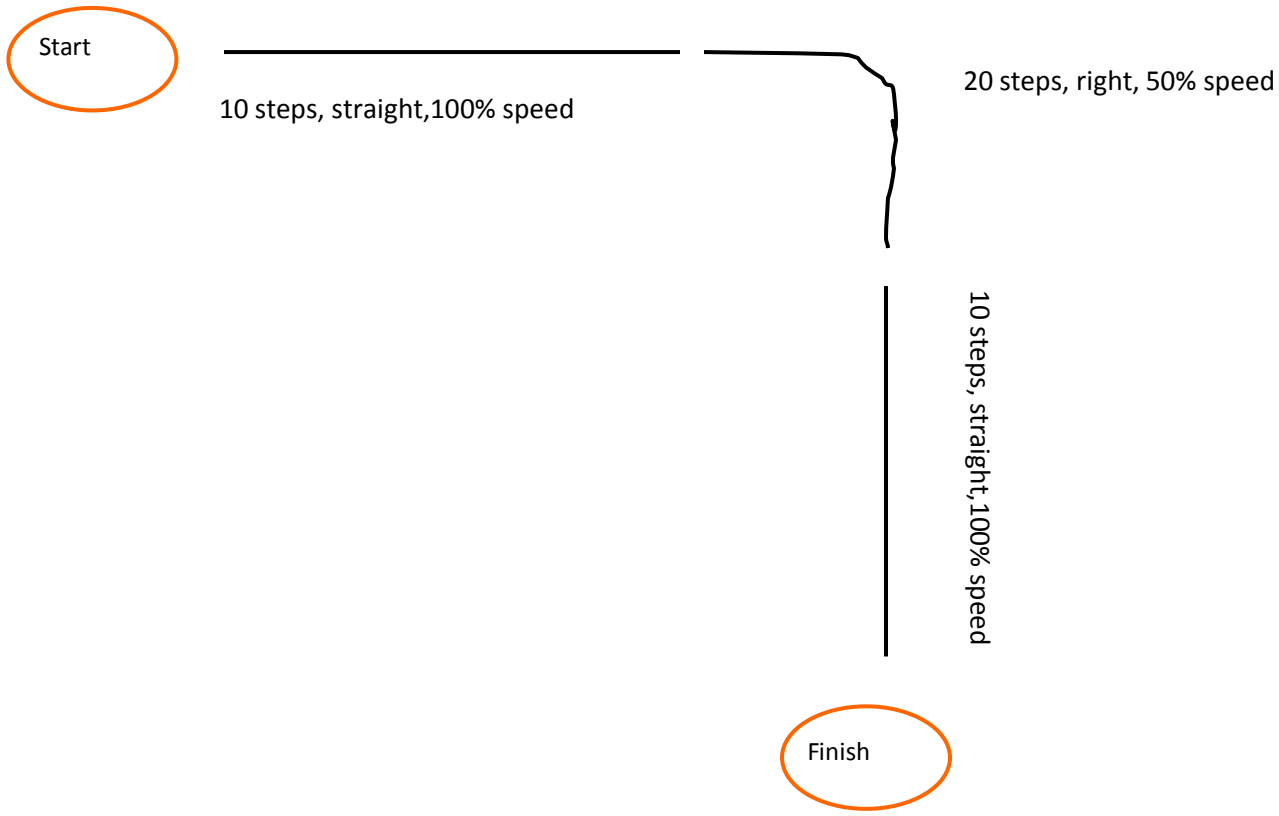
Step 4

Step 5

Beneath an enlargement of step 4:



When a test user manages to finish all 9 lessons within 10 minutes, he or she is allowed to try their first assignment. The first assignment is called “Help Max through the maze”. The assignment goal is to let the user use the properties of the block “Lopen” effectively. The robot has to walk the following pattern and when reached must wave his arms.



To assist the user in this challenge we provide the user with the following facts:

- First let Max walk 10 steps in a straight line with 100% power, then make a turn and let Max walk 10 steps in a straight line with 100% power to the finish. When done let Max wave is arms.
- To let Max make a turn, you need to set its speed property to 50% and its amount of steps to 20. You also have to set the direction property to left or right.

After completion of the lessons and the assignment or after 10 minutes of following the lessons, the user will be interviewed for 5 minutes to extract his or her findings of the application and the lessons/assignment.

3.2.3 The questionnaire

The questionnaire consists of 9 multiple choice questions and 2 open questions. Ranging from what the difference

is in the impression of the children from before and after using the program to if they will want to work more with the program and if they will suggest this program to other children.

3.3 Evaluation

In this section we evaluate the results of the test and whether we have reached our goal by this test session. Secondly we pay attention to the weaknesses of this test and ways to improve it to overcome these weaknesses in future test sessions.

So since our goal was to get more insight in what effect the application has on children and how easy it is for them to work along with, we prepared a test setup as has been described in previous sections. According to the data we gathered, we can see that is was overall quite easy for different type of children to work with the application.

Since we only tested it with 5 children, it is of course wishful to expand this amount. Though it is enough to see how the children react on the interface and the concept of the application. Since different children with different profiles have been tested, it was also possible measure the influence of the background of a child.

Further looking at the interaction between the child and the application, we measured a lot of enthusiasm and fun. Though this is a biased measurement, because at that point the fact that the child was chosen to participate in the test, could be an initiator of that enthusiasm. Taking that effect into account we observed the child thoroughly during the experiment. Based on our observations and the questionnaires, we measured enthusiasm and further interest.

Recommendations for the next test sessions are:

- A thorough preparation included with a research in basic cognitive properties of our target group, the children. Probably it gives us better understanding in how we can compose lessons in such a way that in will not be boring or complex.

- Expand our tests with more schools in different neighborhoods. This way we can test with more profiles and with profiles that differ more from each other.

- Since we have planned to implement more functionality in the application, it is possible to make more complex and exciting exercises to test with.

3.4 Gathered data

Below we summarize the data we gathered during the test session with five children of the primary school.

General	
Gender	2 Male, 3 Female
Age	11 - 12
Computer literacy	4 high, 1 low
What they do on computers	MSN, internet games, power point, downloading music and movies
Interest in technology	All have interest in technology and one even wanted to advance in technology or ICT

Test session	
Duration spend on the 9 lessons	10 – 12 minutes
Lesson remarks	The children went pretty fast through the lessons, but we gave them some directions, due to the fact that this is a non-100% functional prototype. In the future, the child should be able to do it alone.
Opinions about the program	Almost all of the children had no problems with the interface, only one was confused with the arrows.

Questionnaire	
Question	Result
1. What is your first impression when you saw the program?	2.4*
2. What is your impression now?	2*
3. Would you like to play more with this program?	5 times Yes
4. Would you suggest this program to your friends?	5 times Yes
5. Would you like to see more projects like these for your school?	5 times Yes
6. What was your impression of the lessons?	2*
7. Could you find everything to perform the lessons?	4 times Yes, 1 time No
8. Would you like to play longer with this program?	5 times yes, ranged from 10 to 20 minutes, average of 11 min
9. Do you prefer the lessons on paper than in the computer?	1 time Yes, 4 times No
10. What do you wish Max the Robot can do?	Singing, dancing, talking, drawing on the chest, parrot a person
11. Do you still have any remarks?	No remarks

* On a scale of 1 to 5

3.5 Interpretation

In order to setup the test we first made some questions that define what we wanted to test and what insight we wanted to gain from the experiment. Below again a the questions:

1. Is the target group able to perform the tasks in the lessons?
2. Is the positioning of the element clear for the target group?
3. Are there any items we missed, when developing the tool?
4. Is the difficulty level of the tool for appropriate for the target group?

When we look at the gathered data in the see that the children went through the test quite fast so (1) is answered with a clear yes. Secondly remarked that the arrows caused some confusion at a certain point, so (2) is answered with no, the position of the arrows that control the lessons should be adapted and also the size of the arrows for scrolling purposes. (3) Is not answerable because it seems that the prototype hasn't got that much of functionality. The items that are there are enough for the current state. (4) is answered with a clear yes. The children ran through the lessons without any problems.

3.6 Conclusion

Overall we can say that the application is usable for our target group. From the test it appears to be necessary that we expand the functionality of the application and the complexity of the assignments. That way it will be more challenging for the children and probably the user interface will then be more severely tested. So we will have more valuable test data.

4. Conclusion

In this section we present the conclusion in which we try to answer the problem definition defined in the introduction.

'How can we create a technical realization of Max the Robot and how can we proof the concept'.

In this report a process has been described in which we show how to realize the application 'Max the Robot'. The proof of concept, in which the concept is to learn children the basics of programming in a very interactive and fun way by using a Lego Mindstorm NXT Robot. The proof of concept has been partial given by the test we did with the children. Since we observed that they understood the certain basic concepts of programming they applied during the test

A comment can be raised that more experiments are needed to create a complete overview.

The implementation phase of this project was the most exhaustive phase. Creating applications that communicate with the Lego Mindstorm NXT Robot, is a very small and unsupported area. That means that we had to do lot's of low level technical research, sometimes we applied reverse engineering on existing applications, which was a very time consuming task. Our strategy was to build the application iteratively. So we built small components and tested them thoroughly. After it was clear a component could stand the test, it was integrated into the application. This way we created a modular application, which is beneficial for future developers.

The test results in general show a very interesting phenomenon, namely when looking at the profiles of the children we see that they spend lots of time using a computer. Especially for communication (MSN), games, but also surf the internet for their school assignments like small papers and reports they have to write. It seems that children are making use of computers and internet more than ever before.

When looking at the value of educational software, it is very clear that it is becoming a very powerful tool to use in today's educational system. Since children have lots of affinity with computer on a very young age and experience a totally different development then older generations did, because of the fact that they play games, chat and surf. Studies have shown that today's children can process far more data at the same time and in a totally different way, in comparison with their predecessors (Like their parents and grandparents). Educational software has the flexibility to adapt to this new way of processing data (or learning if you like). Children can learn new concepts by playing the so called serious games (games that have the purpose to train or learn the user a certain concept or skill). It seems that our time demands this extra tool in education. Also from this project we have seen how children have learned the simple basics of programming in only a few minutes. Just by playing with the application and experiencing the influence of their handlings on the robots actions, which is a very interactive way of learning, they understood the concept and even saw the potential of being used in their classes.

As time goes by more and more technical solutions will provide more possibilities for people to communicate with each other and therefore more ways of gathering and distributing information will be provided. Also the mind of people changes over time as in the cognitive profile. This means that also the way people process information (learning) is changing. Various studies show that for certain cases the use of educational software leads to a more sustainable and faster way of learning then in comparison with conventional learning methods. For example the use of animated stories, simulation and web lectures can in some cases lead to better result. Therefore our vision of the future contains a world where all the study subjects have there own way of teaching, there is no real standard way of teaching as today with lectures and books (from math to programming courses it's generally the same). Next people will gather information in different ways, with pod-cast like applications, web lectures and total learning environments. All these developments will have a huge impact on the society and probably the whole economy will be rearranged.

Appendix: Manual

Extra:
Please include a cd-rom with the educational software that you have developed (if applicable).

Appendix A

Vragen	
1. Wat was je eerste indruk toen je het programma voor het eerst zag?	Makkelijk ○ ○ ○ ○ ○ Moeilijk
2. Hoe vind je het programma nu?	Makkelijk ○ ○ ○ ○ ○ Moeilijk
3. Zou je vaker met dit programma willen spelen?	Ja ○ Nee ○
4. Zou je dit programma willen aanraden aan je vrienden?	Ja ○ Nee ○
5. Wil je meer van zulke projecten zien op school?	Ja ○ Nee ○
6. Hoe vond je de lessen?	Makkelijk ○ ○ ○ ○ ○ Moeilijk
7. Kon je alles vinden om een les te doen?	Ja ○ Nee ○
8. Wil je nog langer doorgaan met het programma?	Ja ○ ..... minuten Nee ○
9. Wat wil je dat Max de Robot nog verder kan doen? …… …… ……	
10. Heb je nog opmerkingen? …… …… …… ……	

Appendix B

We are three students and your teacher thinks you and four other class mates can help us improve our project by participating in the test project. We plan to create a program teaching children the concept of programming in a child-friendly and attractive way.

This is Max the Robot, today he walks a bit awkward, due to a small accident with the transportation, and nonetheless his legs make walk movements. Max also can move his arms and heads at the same time. Max can speak and show a drawing at this chest, but at this time the functionality is not finished yet.

You now are going to perform nine lessons, the first five lessons are in the computer and lessons six till nine are on paper. When you are ready you can start and when you have problems or question you can ask.

Afterwards there is a short questionnaire.

Appendix C Participant (test) data

Name	Nick
Gender	Male
Computer literacy	MSN, downloading music and movies, internet, internet games
Hobbies (non-computer)	Soccer
Interest in technology	Yes
Future profession	No idea yet
Lesson duration	10 minutes
Actions	Fast
Button removal	Fast
Arrows for row swapping	Fast
Property window	Moderate
Begin … End	Fast
Lesson remarks	Performs the lessons pretty fast. Does not see the down button at the property window for action Walk, due to the set up environment. The sun was reflected on the laptop and blocked the view.
Opinion about program	No problems with the interface. Conveniently arranged layout.

Name	Remco
Gender	Male
Computer literacy	Low, makes almost no use at home with computers
Hobbies (non-computer)	Soccer, hammering, helping in the garden
Interest in technology	Yes
Future profession	Working in glasshouses like his father
Lesson duration	12 minutes
Actions	Moderate
Button removal	Moderate
Arrows for row swapping	Fast
Property window	Moderate
Begin … End	Fast
Lesson remarks	In the introduction, it was told that the action Talk was not implemented and Remco skipped it himself. He needed more time for all the lessons. He forgot begin and end in lesson 6, but solved it later himself.
Opinion about program	No problems with the interface.

Name	Lotte
Gender	Female
Computer literacy	High, uses the computer a lot, creates powerpoint presentations herself. Makes use of MSN, internet games.
Hobbies (non-computer)	None
Interest in technology	Yes
Future profession	Technology, computers
Lesson duration	11 minutes
Actions	Moderate
Button removal	Fast
Arrows for row swapping	Fast
Property window	Moderate
Begin … End	Fast
Lesson remarks	Performs the lessons very fast. Hesitation when adding End button herself, but in the end managed to fix it herself.
Opinion about program	No problems with the interface.

Name	Melanie
Gender	Female
Computer literacy	High, makes use of internet games and MSN
Hobbies (non-computer)	Singing and gymnastics
Interest in technology	Yes
Future profession	Singer or veterinarian
Lesson duration	10 minutes
Actions	Fast
Button removal	Fast
Arrows for row swapping	Very fast
Property window	Very fast
Begin … End	Fast
Lesson remarks	At the beginning it was not clear for her, but afterwards, she performed the lessons very fast.
Opinion about program	No problems with the interface.

Name	Shirley
Gender	Female
Computer literacy	High, makes use of internet games and MSN
Hobbies (non-computer)	Singing and tennis
Interest in technology	Yes
Future profession	Maternity help
Lesson duration	10 minutes
Actions	Fast
Button removal	Fast
Arrows for row swapping	Moderate
Property window	Moderate
Begin … End	Fast
Lesson remarks	Performs the tasks fast
Opinion about program	The arrows of the programs confused her.