Mag.rer.nat. Michael Hubmann

Development and Evaluation of a Holistic Robotics Curriculum for School Education

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Gerald Steinbauer

Institute for Softwaretechnology
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Franz Wotawa

Graz, November 2020

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

In this thesis results from the development of a curriculum for a holistic robotics education were presented. The curriculum was developed especially for integrative teaching of robotics at a school in Austria, but the use on other schools and in other countries in the future was already envisioned. In the thesis we also analyzed the goals of educational robotics (ER) and how successfully these goals can be achieved with that approach. Robotics curricula and learning contents from schools and countries all over the world were analyzed and compared in this thesis. Moreover, a comprehensive study was conducted to answer the question which form of programming, graphical or textual, is better suited for novices in the field of robotics and programming.

In the study the positive influence of robotics education on the skills of computational thinking and problem solving, as well as on social aspects is shown. Additionally, the thesis also shows that graphical and textual programming is well suited for programming novices. But neither of the two forms is significantly better for beginners.

Based on that foundations a holistic curriculum for robotics education for the entire school career was finally developed. The learning objectives, teaching methods and materials were carefully designed and allow similar to literacy development in language and mathematics to build up in the development of skills on previous achievements. Principles from developmental psychology were also considered in the development. The successful development of the curriculum was shown in an evaluation with experts from the field of ER. In a qualitative research, the experts stated that the curriculum is suitable for use in schools, that it is suitably structured and that the requirements are appropriate for the respective age groups in each

learning year. Furthermore, all experts stated that the curriculum is suitable for use in schools in their countries.

Zusammenfassung

In dieser Arbeit wurde ein durchgängiges Curriculum für den Robotik-Unterricht an Schulen vorgestellt. Im speziellen wurde das Curriculum für den integrierten Robotik-Unterricht an einer österreichischen Schule entwickelt, soll aber in Zukunft auch in Schulen in anderen Ländern eingesetzt werden können. In dieser Arbeit werden unter anderem die Ziele des Unterrichts mit Robotern (Educational Robotics – ER) untersucht und wird diskutiert mit welchem Erfolg diese Ziele mithilfe von ER vermittelt werden können. Ferner werden unterschiedliche Lehrpläne für ER, sowohl von Schulen als auch von anderen Ländern im Rahmen dieser Arbeit betrachtet und verglichen. Darüber hinaus wurde eine umfassende Untersuchung durchgeführt, welche Form der Programmierung, grafisch oder textuell, für Einsteiger im Rahmen der Programmierung von Robotern besser geeignet ist.

Im Rahmen der durchgeführten Studie konnte ein positiver Einfluss von ER auf die Fähigkeiten Computational Thinking und Problemlösung gezeigt werden. Darüber hinaus wurde auch ein positiver Einfluss von ER auf soziale Fähigkeiten aufgezeigt. Ferner wurde in der Arbeit gezeigt, dass grafisches und textuelles Programmieren für Programmieranfänger gleichermaßen gut geeignet sind, jedoch keine der beiden Formen für Anfänger signifikant besser geeignet ist.

Auf Basis dieser Untersuchungen wurde ein kohärentes Curriculum für den einen durchgängigen Robotik-Unterricht im Rahmen der gesamten Schulkarriere entwickelt. Die Lernziele, Lehrmethoden und Materialien wurden sorgsam ausgewählt bzw. entwickelt und folgen den Ideen der aufeinander aufbauenden Entwicklung der Fähigkeiten in Sprachen und der Mathematik. Ferner wurden im Rahmen der Entwicklung des Curriculums

Aspekte der Entwicklungspsychologie miteinbezogen. Die erfolgreiche Entwicklung des Curriculums konnte im Rahmen einer Evaluierung, welche mit internationalen Experten aus dem Bereich ER durchgeführt wurde, gezeigt werden. Im Rahmen einer qualitativen Untersuchung gaben die Experten an, dass das Curriculum für den Einsatz an Schulen geeignet ist, dass es eine gute Struktur aufweist und, dass die Anforderungen für die jeweilige Altersgruppe in den einzelnen Schuljahren passend sind. Darüber hinaus gaben alle Experten an, dass das Curriculum für den Einsatz an Schulen in ihren jeweiligen Heimatländern geeignet ist.

Contents

List of Figures

1 Introduction

Our time is all about digitalisation, which do not stop at school. This is also shown by the fact that digital tools are playing an increasingly important role in teaching. (Ainslee, 2018)

Austria has addressed this development in recent years by introducing the subject of "Digital basic education" in schools. Within the scope of this subject, students should for example learn to work with operating systems (e.g. Windows), or to surf safely on the internet (Department of Education Austria, 2019). Besides that, students should also be taught skills in the field of computational thinking (Department of Education Austria, 2019). Therefore, robots are playing an increasingly important role in schools, since they can be used to teach computational thinking, among other aspects (see section 2.1).

In recent years, an increasing number of schools have started to use robots in teaching (Pahl and Stadler-Altmann, 2019, p. 160). Among other things, this thesis is going to address the question - how individual schools around the world have integrated working with robots into their lessons.
In this thesis we investigate which skills could be taught using educational robotics and whether this is done successfully. We will also determine which robotics contents should be covered in class and when.

As already stated, more and more digital tools, such as robots, are being used in schools. However, as reported in some articles like Tarkowski (2019) or (Brandhofer et al, 2018, p. 337 - 338), the mere use of digital tools in everyday school life is not sufficient to achieve improvements in the field of learning. The lessons must be based on a consistent and well thought concept. The main aim of the thesis is to develop a holistic curriculum for students aged 10 to 17, with respect to psychological aspects and previous scientific knowledge.

The curriculum will be integrated into the computer science lessons of a academic secondary school. It is the BG/BRG/BORG Köflach (Styria, Austria) which serves as a pilot school for the implementation of the curriculum. Besides, the author of this thesis works as a teacher at the pilot school. A strong focus is laid on the development of the skills from school level to school level and the question which kits should be used at each level. We also considered how robotics education can be integrated into a existing computer science curriculum, which in the case of the pilot school is prescribed by the Government of Austria. In addition, it needs to be determined at which point in the career of students it is useful to continue only with students who have specialised in the field of natural science or computer science.
As part of the development of the curriculum, we also investigate in the pilot school which type of programming, graphical vs. textual, is better suited for novices in the field of robot programming. This is important for the development of the curriculum, since in most cases students do not have any programming experience when they start their education at the academic secondary school in Austria. It should also be ensured that the curriculum is structured according to age of the students. Furthermore, the increase in knowledge in the context of skills (e.g. computational thinking) is also examined during the investigation as well.

1.1 Research Questions

On the basis of the previously mentioned considerations, the following research questions are going to be answered in the context of this thesis:

1. Which skills (e.g. computational thinking) can successfully be taught using educational robotics (ER)?
2. How are curricula structured for teaching robotics on various levels around the world? How do they differ in their structure?

3. Does modality (text-based vs. graphical programming) in the context of ER influence the learning success of the skills mentioned in question 1 for programming novices?
4. Based on questions 1 to 3, how should a curriculum be structured to ensure a holistic and broad robotics education over 8 years?

1.2 Structure of the Thesis

The second chapter investigates which skills should be taught with the help of robotics and whether there has been success in teaching the skills with ER in the past.
In addition, different schools and countries which use robots in their teaching are analyzed. The curriculum and teaching materials used by these schools/countries for robotics will be examined. A comparison between individual schools and countries will be presented as well. It is also assessed which technologies are used (e.g. robotic kits) and whether the schools considered also have achieved success in robotics education, e.g. in competitions.
Moreover, we will review publications that have examined modality (graphical vs. textual programming) in relation to learning success, especially for programming novices.

The following chapter shows the experimental setup for the comparison of textual and graphical programming for novices in the pilot school. The status quo of the pilot school in the context of computer science is described in more detail. The chapter also reports which robot kits are currently available at the school and to what extent robotics is currently taught. It is also explained how the learning success in connection with the skills, which should be taught with the help of ER, will be measured in the experiment. The chapter also explains how the students' knowledge increase was measured in comparison between textual and graphical programming.

The fourth chapter presents the results of the experiment mentioned in the last chapter. It is explained which learning successes the students achieved within the framework of the individual skills (e.g. computational thinking).

It is also presented whether there are significant differences with respect to graphical vs. textual programming for novices in the area of the individual skills.

In the sixth chapter the curriculum is presented. For each school year it is specified which contents are to be covered. In addition, the decisions for the contents are justified for each school year on the basis of developmental psychology and existing scientific literature. At the end of the chapter, recommendations for the integration of the robotics curriculum into an existing curriculum are also given.

In the penultimate chapter, the methodology for analyzing the curriculum is presented. Subsequently, the results of the qualitative research are presented. Furthermore, the experiences at the pilot school are reported.

In the final chapter the most important findings of the thesis are summarized and an outlook on possible future work will be given.

2 Robotics in Schools

In this chapter we will examine the goals and reasons related to using robots in schools. Moreover, we will summarize curricula that already integrated robotics. In the chapter we will also analyse literature that invesigated modality (graphical vs. textual programming) in relation to learning success, especially for programming novices.

2.1 Goals and Reasons for Using Robots in School

Educational scientists like Papert (1980) are convinced that robots can have a positive influence on teaching. According to (Benitti, 2012, p.986) and (Eguchi, 2014a) educational robotics can improve computational and engineering thinking skills, problem solving and social interaction/teamwork skills. In the next section we are going to take a closer look at these individual skills in the context of schools and how successfully they are conveyed with the help of ER.

2.1.1 Computational and Engineering Thinking Skills

What is computational thinking? The term computational thinking was first used by Papert (1980). In general, this term describes the professional way of thinking of computer scientists to solve problems. Computational thinking is very similar to thinking in mathematics. However, mathematics is more about to proof things. Computational thinking is more about achieving efficient solutions. Moreover, the results in the field of computational thinking are more descriptive. The solutions can be checked on a computer or visualized directly by means of robots. The results are thus brought to life. (Futschek, 2016)

In connection with schools, the following aspects were presented in Quick-Start Computing (2019), which is funded by the Department for Education of the United Kingdom and Microsoft. They describe the process of computational thinking to tackle a problem:

- **Logical reasoning:** This step is to find out why something happens and how it happens. Students should for example use logical reasoning to describe the behaviour of a program or detect and correct errors in algorithms or programs.
- **Algorithms:** At this step students should find a sequence of rules and steps to solve a problem. Students should for example draw a square with a robot on the floor. It also makes sense for students to write down their code in the form of a plan first, e.g. pseudo code. This makes it easier to give feedback.
- **Decomposition:** In this step a problem should be broken down into smaller problems, which are easier to solve. An example would be a robot that is not allowed to touch obstacles. One part of the problem would be to drive straight ahead to ensure accuracy, another one to detect obstacles and a third one to avoid them.
- **Abstraction:** Abstraction is the heart of computational thinking. Students need to think about what facts they need to focus on and what details can be hidden. In the context of robotics, students would initially ignore the ground and focus only on solving the problem, such as avoiding obstacles.
- **Patterns and generalisation:** Generalisation is about solving similar problems. In the context of patterns it is about recognizing general or recurring problems and finding rules for their solution. In the context of robotics, driving straight on is a recurring problem. So it makes sense to find a solution that generally works.
- **Evaluation:** The last step is to evaluate what has been done.

Despite a focus on educational robotics (ER) and computational thinking (CT) in recent years, there is still not much literature to investigate the influence of ER on CT on the age segment K-12 (Constantinou and Ioannou, 2018). Above all, there is a lack of valid instruments to measure the learning success of ER in combination with CT (Ioannou and Makridou, 2018).

In Constantinou and Ioannou (2018) the authors report that they had used validated instruments to measure the impact of ER on the development of CT skills. In two studies they found out that ER has a positive influence on the acquisition of CT skills. However, they also state that due to a small sample size and a bad gender balance a generalization of his results is only possible to a limited extent. Further studies must therefore be carried out.

According to the report Lucas et al (2014), the core idea of engineering thinking is: making 'things that work' or making 'things work'. Thus, for example, an engineer should be able to identify and examine interrelationships,to generate and evaluate creative solutions or to estimate costs. In context of schools, Council (2012) names eight engineering (and science) practices that are relevant for a student:

- **Defining problems:** Engineering begins with a problem to solve. This may also involve asking questions like: "What are the criteria for a successful solution?" (Standards, 2013). WaterBotics[1] is an example of a suitable degree of difficulty level for K-12 students. It is an open-ended problem, therefore students need to have attributes like perseverance, teamwork and also things like research skills in research facilities to solve it. (Eguchi, 2014b)
- **Developing and using models:** Models for example include diagrams or mathematical representations. In engineering such models can be used to analyze under which conditions our system has to work. (Standards, 2013)
- **Planning and carrying out investigations:** Students should also be able to determine which solution solves the problem best. In the course of time, students should get a better understanding for which purpose variables have to be used (input vs. output) or have to be changed during the experiments in order to get the desired results. (Standards, 2013)
- **Analyzing and interpreting data:** Students should expand their knowledge on the representation and interpretation of data. Using these data, students should be able to substantiate their conclusions. (Standards, 2013)

[1] https://waterbotics.org/ [Online; Last call 20.November 2020]

- **Using mathematics and computational thinking:** Mathematics is the key to understand science. Students should use mathematics to represent physical variables and their relationship. Computational thinking should be for example used to develop algorithms. (Standards, 2013)
- **Designing solutions:** The main goal of engineering is to develop solutions (Standards, 2013). According to Eguchi (2014b) the steps to develop such a solution consist of:
  - Design Task
  - Brainstorm
  - Design
  - Build
  - Test
  - Redesign
  - Share
- **Engaging in argument from evidence:** In this step, students should defend their ideas or solutions with the help of argumentation (Standards, 2013).
- **Obtaining, evaluating and communicating information:** Students should be able to read, understand and evaluate domain-specific texts. Students should also be able to communicate their ideas or solutions in different ways (e.g. diagrams, tables,...). (Standards, 2013)

According to Eguchi (2016), robotics or in particular the RoboCupJunior has an impact on engineering thinking. For example, 93% of students said that the RoboCup helped them improve their problem solving skills. Also Barker and Ansorge (2007) and Nugent et al (2009) show the positive influence of educational robotics on the development of engineering skills.

2.1.2 Problem Solving

Problem solving is a continuous process in which existing knowledge is used to discover hidden knowledge ((Fredericks, 2010, p. 148) ; Maheshwari, V.K. (2017)). According to (Fredericks, 2010, p. 148), problem solving consists of three basic functions:

- Information seeking

- Generating new knowledge
- Decision making

Problem solving can also form the core of a good school curriculum. Students can use their acquired knowledge to solve application-related problems from real life. This also allows deeper problems to be dealt with in class. (Fredericks, 2010, p. 148)

The author (Fredericks, 2010, p. 148 - 150) presents a five stage model that students can easily remember and use:

1. **Understand the problem:** The first important step is that students are able to formulate a problem or a goal in their own words. To facilitate this step, students can for example create a list of all known facts, list all conditions that are related to a problem or describe related problems.
2. **Describe any barriers:** Students should recognize the barriers that represent a problem in creating the solution. The recognition of these impediments also represents an important step in problem solving.
3. **Identify various solutions:** The next step for students is to find appropriate strategies to solve the problem. It is important that students find out that there are different strategies, but that not all are suitable for all problems. Here are some examples:
   - Create visual images
   - Guesstimate
   - Create a table
   - Use manipulatives
   - Work backwards
   - Look for a pattern
   - Create a systematic list
4. **Try out a solution:** If working with different strategies or combinations of strategies, it will be be important to:
   - Have a good documentation of the collected data, used strategies and predictions.
   - Work with different strategies until it becomes evident that a tried strategy does not work.
   - Carefully monitor and document steps as a part of their solution.

- Have no problem putting a particular problem aside for the moment.

5. **Evaluate the results:** It is important that students are able to evaluate their own problem solving skills and generate solutions. However, for students such self-assessment is often difficult. Thus, the teacher should support students with questions like "How do you evaluate your progress?" or "Why do you think your solution is appropriate?".

With regard to learning success in the context of problem solving, it shows that ER is a good tool for teaching these competences. In his thesis, Hussain et al (2006) shows an increase in problem solving skills, although this is not significant. In his thesis Nourbakhsh et al (2005) also reports an increase, which is statistically significant.

2.1.3 Social Skills

Social skills also play an important role at the workplace. There is nearly no job in which you have no contact with other people. The ability to work in a team, communication skills and flexibility are of particular interest. (Absolventa, 2019)

According to ebotics (2019), ER fosters the following social skills in school:

1. **Teamwork:** Children learn that by working together they can achieve better results.
2. **Discipline and compromise:** Students learn that their goals must be pursued persistently and patiently.
3. **Experimentation. Trial and error:** Students quickly see the results of their work. But they also recognize that mistakes are part of finding solutions.
4. **Enhance self-esteem:** In addition of finding out that mistakes are part of the learning process, students also develop skills to overcome fear of making mistakes and to cope with stress.
5. **Do it yourself empowering:** Students also acquire the ability to work independently.

Through an online questionnaire with 14 students who participate at the RoboCupJunior Eguchi (2014a) shows the positive influence of ER on soft and social skills. For example, 79% of respondents stated that collaborative skills were improved, 79% that they learned to be patient and 93% that they learned to be persistent. Furthermore, Owens et al (2008) show in their study the positive influence of using Lego robots when working with autistic children.

In the following chapter, robotics curricula from different schools and countries all over the world are presented and analysed.

2.2 Curricula from Different Schools and Countries

In this chapter curricula from different schools and countries all over the world from the continents North America, Asia, Australia and Europe are going to be see above. The first region to be considered is North America represented by the United States of America.

2.2.1 North America - USA

School systems are quite different all over the world. In order to be able to better classify the curriculum from the USA, an overview of the school system in the USA will be given.

2.2.1.1 School System

At around six years of age, children in the USA begin their school careers at primary school. They attending this school for five or six years. (Study in the USA, 2019)

Afterwards, children in the U.S. attend secondary school consisting of

"Middle School" or "Junior High School" and "High School". At the end of "High School" (12th grade) the students receive a certificate or diploma and can attend a university or college. (Study in the USA, 2019)

In Figure 2.1 one can get a short graphic overview of the school system in the USA and the typical age the students attend which type of school. In this figure the K-12 term is used. It is a short form for the publicly-supported school grades, where the grades are kindergarten(K) and the 1st grade through the 12th grades are 1-12 (Rose, 2020).

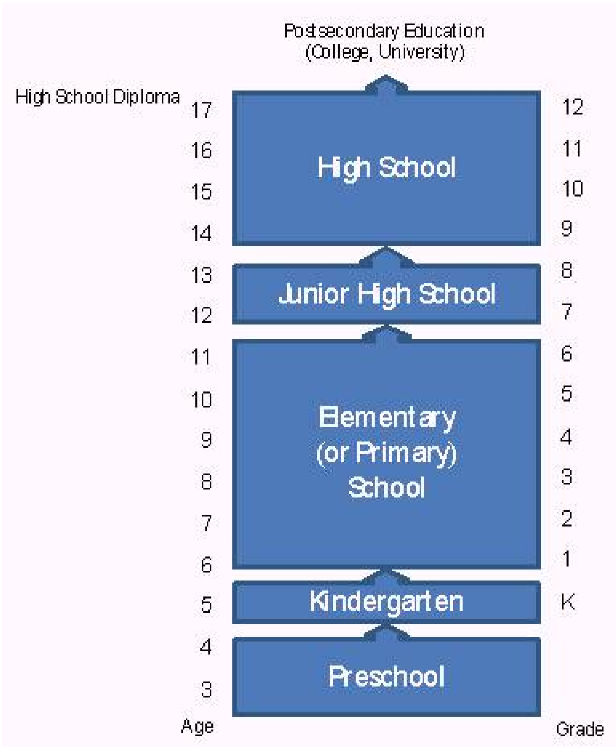


Figure 2.1: Overview School System USA [2]

[2]Source: http://www.konrad-fischer-info.de/school.htm [Online; Last call 21.July 2020]

2.2.1.2 Curriculum "Engineering[3]"

A closer analysis of an exemplary curriculum from the USA given now. The curriculum, which is available as a book, was composed for teaching in High Schools. It is called Engineering[3] and was written by Timothy Jump.

Field of Application and Facts

As already mentioned, the curriculum was developed for teaching in High Schools. The target group are students between 14 and 17 years of age. An exemplary school which uses this book is "The Hill School" or short "The Hill". This is the school where the author of the curriculum is teaching.

"The Hill" is a private High School, so the school is more likely to be attended by privileged students. Students from grade 9 to 12 attend this school. At the moment there are 522 students attending this school, about 75% of them are in boarding school. The school is located in Pottsdown, Pennsylvania and was established in 1851. In Figure 2.2 you can see a picture of the campus of the school. (Wikipedia, 2020b)

Figure 2.2: Campus of "The Hill School" [3]

Engineering[3] is a multi-year program and is set up in four parts. Each part requires a lot of time to complete it. To work through the whole curriculum

[3]Source: https://www.niche.com/k12/the-hill-school-pottstown-pa/ [Online; Last call 21.July 2020]

it usually takes four years. So students have to start in their first year of High School with Engineering^3 if they want to complete the course. The class meets every day in one school year. (Engineering^3 FAQ, 2020)

Engineering^3 is designed to be an elective course. It is not planned that Engineering^3 replaces a regular math or science class. It is also not recommended that Engineering^3 replaces any of these courses, as many universities consider these courses a prerequisite for admission. There are no prerequisites for taking Engineering^3 as a course, since all necessary content is taught as part of the course. During a mail exchange, the author and teacher Mr. Jump explained that the students have no previous experience from other schools. (Engineering^3 FAQ, 2020)

To work with the curriculum the Engineering^3-Kits and control elements are required (Jump, 2015, p. 686 - 800). The contents of the kits/packages differ depending on the part of the curriculum, a detailed description of the parts of the curriculum and used parts of the kits follows in the next section. So there are for example more Lego pieces in package E1. In Package E2 the RCM-Controller follows as control element of the robot. In packages 3 and 4 there are also sensors.

Detailed Curriculum

As already mentioned, Engineering^3 consists of 4 parts or modules. Part 1 deals mainly with the Introduction to problem solving structures, elementary mechanics, mechanics and engineering Design. In (Jump, 2015, p. 26-225) it can be seen that part 1 deals in particular with:

- **Designing and Problem Solving:** Students should be acquired with a fundamental problem solving model which is necessary for success with Engineering^3. In this context, students should learn to observe and question. Students should also get to know the design circle and an effective documentation of design concepts and outcomes.
- **Structures:** In the curriculum LEGO is used as a meaningful tool to study engineering. In order to have a higher level of success with LEGO, an engineer's understanding of the product is necessary. So

students will explore the Engineering[3] Part 1 Kit, establish elementary vocabulary in the context of LEGO elements and examine the mathematical relationship of LEGO elements with applications toward creating stable LEGO structures.

- **Forces- statics:** In this module students will focus attention on statics, evaluate structures in equilibrium in an attempt to understand why structures behave as they do, and build a basic knowledge foundation that will lead to design effective structures.
- **Forces- dynamics:** In this module students will focus on dynamics, evaluate structures experiencing acceleration in an attempt to understand why structures behave as they do when there is a change in motion; and continue to build a knowledge foundation that will lead to designing and prototyping effective structures.
- **Structures:** A very important part of engineering is focussed on structures that move. The Engineering[3] Kit 1 has many elements which are specially designed to generate motion/force. This module explores these elements and the LEGO vocabulary associated with these elements as well as their mathematical relationships to each other. Also the structural elements of the kit are explored. Associations of these elements are also explored with an eye on developing effective power transmission in mobile LEGO structures.
- **Machines:** In this module students will learn about basic engine structure and function. They will also have a look at an outline of many of the elements that must be considered and selected in the context of the design of a machine.
- **Engineering Challenge:** At the end of part 1 students have to do a challenge. In this challenge students have to apply what they have learned in part 1 in the context of a non-LEGO machine.

Part 2 of the curriculum is going to have its focus on Oneshape/CAD, custom parts design and explorations in systems development. In (Jump, 2015, p. 226-436) it can be seen that part 2 deals in particular with:

- **Designing parts with CAD:** In module 7 students are going to work with Oneshape CAD software and develop fundamental skills for designing parts and composite structures.

- **Component specific design:** An essential part of design is to make sure the parts fit together correctly. In this section several test parts are designed that highlight some of the more common errors, which are mainly made by beginners, to prevent these errors later in the design of robot parts.
- **Mechanical design:** In this module students again have to show all learned skills in the development of a test base machine, called Machine 1A_H. They also learn some basics about the used microcontroller called RCM-Controller. At the end students will control a robot with the help of the RCM-Controller manually.

In part 3 students get in touch with coding. They learn things about control systems, data collection and problem solving through code. In (Jump, 2015, p. 438-608) it is shown that part 3 deals in particular with:

- **Understanding control:** In module 10 students develop an introductory understanding of control system elements. Therefore, students are be introduced to the basic structure of control systems and the component groups that align to each control system. Students also learn basics about data (binary, hexadecimal,...).
- **Taking control:** Module 11 guides students through fundamental components, structures and writing of text-based computer programs. The focus is laid on the language C generally and picoC specially. This includes basic things like loops and queries, but also nested queries or operators. In this section students also get a more detailed view of the RCM-Controller, for example how students get their programs or code to the controller. They also learn to work with sensors and work with input and output of sensors.
- **Design for control- hardware and processes:** In this module students will explore different aspects of creating a successful use of control hardware. Students learn strategies for developing robot control algorithms (e.g. navigating through a maze) and how to work with sensors on the robot (e.g. data collection, placement).
- **Design for control - code:** In this module students experiment with writing programs to control a robot. The module leads students through code design processes including several focussed control challenges, leading to a fully autonomous robot able to attack the

fundamental challenges in the Engineering^3 Rescue Robot Challenge.

The fourth and last part of the curriculum consists of alternate considerations and challenges. In (Jump, 2015, p. 610-680) it can be observed that part 4 deals in particular with:

- **Control- a deeper look:** In this module the students learn some advanced control matrices(foundational control matrices, reflex matrices, decision matrices) and get a short overview over the world of Artificial Intelligence (AI).
- **Sensors - a deeper look:** In module 15 students will look at sensors in more detail. Thus, they look closer at structures of the input data streams and data types, so that students could build higher level control algorithms.
- **Advanced control processes:** In the last module students solve some challenges to get in touch with advanced control processes.

Conclusion

Within this curriculum, the approach to robotics is based on engineering science. Thus, the actual programming only begins in Part 3 of the curriculum. According to the author, robotics also "only" serves as a tool to better understanding the concepts of engineering. This also differentiates this curriculum from others, which choose a different approach away from engineering at the beginning.

The development process of the curriculum takes 20 years (Engineering^3, 2020). It was developed by Timothy Jump, as already mentioned. The success of the curriculum is shown by the fact that 75 percent of the graduates of the Engineering^3 have pursued engineering degrees and careers, the other 25 percent point out how the curriculum directs participants to break down problems and create solutions (Engineering^3, 2020). A team from "The Hill-School" also took part in the RoboCupJunior 2019 as part of the Rapidly Manufactured Robot Challenge (RMRC) in Australia (Hill School, 2020).

2.2.2 Australia

Now we will look at a curriculum from Australia. First of all, we will give an overview of the Australian school system in order to better classify the curriculum.

2.2.2.1 School System

In Australia all states and territories have their own department of education that is responsible for government-funded primary and secondary schools. There are 8 territories in Australia, these are New South Wales (NSW), Queensland (QLD), South Australia, Tasmania (TAS), Victoria (VIC), Western Australia (WA), Australian Capital Territory (ACT) and Northern Territory (NT). (Nuffic, 2018, p. 7f)

For children in Australia education is compulsory from 6 to 16. Depending on the territory children attend primary school for 6 or 7 years and for 5 or 6 years secondary school. The secondary school consists of 2 stages. The first step, the "secondary school" finishes with year 10 and the second step the "senior secondary school" makes up year 11 and year 12. Students who have completed secondary school have therefore attended school for a total of 12 years. (Nuffic, 2018, p. 7f)

Figure 2.3 shows an overview of the structure of primary and secondary school in each state.

2.2.2.2 Curriculum - Robotics and Mechatronics

Now we look into an exemplary curriculum from Australia. It was published by the Government of the Australian Capital Territory and is called "Robotics and Mechatronics".

<table>
<tr><th rowspan="2">School Year Level</th><th colspan="3">State</th></tr>
<tr><th>NSW, VIC, TAS, ACT</th><th>SA, NT</th><th>QLD, WA</th></tr>
<tr><td>12*</td><td rowspan="6">Secondary</td><td rowspan="5">Secondary</td><td rowspan="5">Secondary</td></tr>
<tr><td>11*</td></tr>
<tr><td>10</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td><td rowspan="7">Primary</td><td rowspan="7">Primary</td></tr>
<tr><td>6</td><td rowspan="6">Primary</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td></tr>
<tr><td>Pre-year 1*</td><td>Kindergarden (NSW, ACT)
Preparatory (VIC, TAS)</td><td>Reception (SA)
Transition (NT)</td><td>Pre-primary (WA)</td></tr>
</table>

Figure 2.3: structure of primary and secondary schooling by state of Australia [4]

Field of Application and Facts

The target group of the curriculum are students of the secondary school, more precisely those of the senior secondary school. The age of the students is therefore between 16 and 18. The curriculum came into effect in September 2019. According to Mr. Gerard Elias, four schools are currently start to implement the curriculum. One of these schools is St Francis Xavier College. This is also the school where Mr. Elias, who is a member of the Board of Trustees of RoboCup, teaches himself.

St Francis Xavier College is located in Florey, which is a suburb of Canberra and belongs to the Australian Capital Territory. It is a catholic secondary school, so students from year 7 to year 12 attend this school. The total

[4] Source: http://www.ibe.unesco.org/fileadmin/user_upload/archive/Countries/WDE/2006/ASIA_and_the_PACIFIC/Australia/Australia.htm [Online; Last call 22.July 2020]

number of students is 1200.

As already mentioned, the curriculum is designed for grades 11 and 12. The curriculum consists of different units, these are according to (BSS Australian Capital Territory, 2019, p. 9):

- Building & Programming Circuits
- Digital & Analog Interactions
- Robotics & Mechatronic Systems
- Applications of Robotics
- Negotiated Study

Each unit has a value of 1.0, these values are necessary to obtain the "ACT Senior Secondary Certificate[5]", and is delivered over at least 55 hours, which is usually one semester. Students can reach Minor (at least a minimum of 2 passed units) and Major (at least a minimum of 3.5 passed units) level. Units of the curriculum can be delivered in any order. (BSS Australian Capital Territory, 2019, p.44)

In this course the following materials are needed:

- Arduino
- Arduino IDE
- Fritzing

The curriculum provides the outline of what students should be able to do after the unit is completed and what content is taught. How these contents are taught is up to the teacher. It is also up to the teacher to check the content in the form of assessments. There are precise guidelines what an assessment has to look like. For example, there are specifications as, which content belongs to the design process (e.g. storyboard) and which belongs to solution design (e.g. prototyping). (BSS Australian Capital Territory, 2019, p. 10f)

[5] http://www.bsss.act.edu.au/information_for_students/act_qualifications [Online; Last call 13. November 2020]

Detailed Curriculum

Now we will take according to (BSS Australian Capital Territory, 2019, p. 18-43) a closer look at the individual units is taken:

- **Building & Programming Circuits:** In this unit students learn about the components of electronics and the design and construction of electronic systems. Design methodologies will be used to investigate, prototype, test strategies and critically analyse the construction of electronic systems. Students acquire the knowledge and skills necessary to apply a design process using electronics to create sustainable and innovative systems.
- **Digital & Analog Interactions:** In this unit students learn to respond to a real-world need and justify the creation of a complex control system. Microcontrollers and control systems are investigated and programmed by students. A design process is applied by the students to design interface circuits, prototypes, construct and test systems to receive input and collect data from sensors and provide a meaningful output.
- **Robotics & Mechatronic Systems:** In this unit students investigate the development of mechatronic systems and robotics. The impact of robotics and mechanised system on human society, built and natural environments and general well-being will critically be analysed by students. The design process will be used to create, test and control a product or solution incorporating mechanical, electrical and control systems.
- **Applications of Robotics:** In this unit the role of robots and other intelligent machines, including technologies such as, neuronal networks, Artificial Intelligence, machine learning. The design of a system, its construction and application of automated technologies will be investigated. A design process will be used to complete a project that includes prototyping, testing, constructing and evaluating an innovative system. Students will analyse their results and present them with justification.
- **Negotiated Study:** Before students could start with this unit, they have studied at least two standard 1.0 units before this course. In this unit, the learning content can be determined by a group of students or an

individual student in consultation with the teacher and the principal. It is important that this does not result in a duplication of previous learning content.

Conclusion

As explained above, the curriculum was published recently. Therefore "only" one unit is implemented at Xaver College at the moment. It is the lesson "Digital & Analog Interactions". Schools are currently in the process of implementing it into their teaching. According to an mail exchange with Mr. Gerard Elias, who teaches robotics at St. Francis Xaver College, his school supported other schools in implementing the curriculum. The special feature of the curriculum is that it is prescribed by an authority. The schools themselves then only deal with the implementation of the contents. This makes it different from the curriculum of most other schools, where setup is defined by the school itself or the teacher.

The pioneering role of the school in the field of robotics is also reflected in the RoboCup. The national RoboCup 2019 was held at the school. The school was also successful and was able to win the "Rescue Maze" competition, for example. (St. Francis Xaver College, 2020)

2.2.3 Europe - Austria

Now we will look at a curriculum from Austria. First of all, we will give an overview of the Austrian school system in order to better classify the curriculum.

2.2.3.1 School System

Compulsory education in Austria begins at the age of six. Children attend primary school for 4 years. At the age of 10 children then change to secondary school. They can attend a (lower) secondary academic school or a middle school. After four more years both school types end.

The obligation to attend school ends after 9 years. So most students have to add at least one more year. The following school types are available:

- **Secondary Academic School:** This type of school is a general education school also called grammar school. It lasts 4 years and ends with the school leaving examination (Matura [6]).
- **Secondary Technical & Vocational High School:** In Austria there are many schools that prepare students specifically for jobs in engineering, tourism or finance. Each of these schools is graduated with the school leaving examination and takes five years. The most important ones are listed here:
  - **Higher Technical School:** The HTL usually has a technical focus, for example on mechanical engineering, electrical engineering and computer science as well.
  - **Commercial School:** In this type of school, the focus is put on business, often in combination with languages and computer science.
  - **Secondary School for Economic Professions:** This type of school focuses primarily on tourism.
- **Secondary Technical & Vocational School:** This school type is similar to "Secondary Technical & Vocational High School", but lasts only 3 years and does not end with the school leaving examination.
- **Polytechnic school:** This type of school extends over one year and serves primarily as a preparation for vocational training within the framework of apprenticeship/vocational school. This school does not end with the school leaving examination either. However, students have the opportunity to take the school leaving examination parallel to their vocational training.

In Figure 2.4 one can seen an overview of the Austrian school system.

2.2.3.2 Curriculum - BRG Kepler

Now an exemplary curriculum from Austria is presented. It is a curriculum of a secondary academic school. The school is one of the pioneers in the

[6] The Matura entitles students to attend a university or a university of applied sciences.

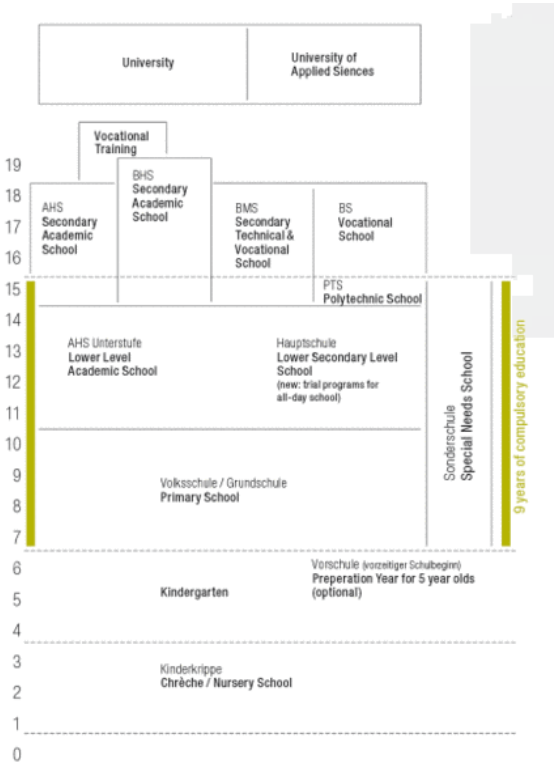


Figure 2.4: School System of Austria [7]

field of using robots for teaching in Austria.

Field of Application and Facts

The name of the school is BRG Kepler. BRG stands for secondary academic school with a focus on natural sciences. The school is located in the Styrian capital Graz. It is named after the physicist Johannes Kepler.

The school was founded in 1872. At the moment about 700 students are attending this school. The main focus of the school is put on natural science

[7]Source: https://www.innviertel.at/austrian-education-system [Online; Last call 25.July 2020]

and computer science. One of the special features of the school, apart from its own observatory, is the regular participation in competitions. These include competitions in the field of robotics, such as the RoboCupJunior or RoboLeague. In Figure 2.5 you see a picture of the school building. (Wikipedia, 2020a)

Figure 2.5: Picture of school building BRG Kepler [8]

The implementation of the curriculum began in the school year 2008/2009. Since this school year students have the possibility to deal with robotics from the age of 10 years up to the school leaving examination. Pupils meet different challenges in the fields of programming, physics, mathematics, electronics, electrical engineering, geometry and handicraft. (Kepler IT, 2020)

According to Kepler IT (2020) the curriculum in robotics has the following structure:

- **1st class (year 5):** Trial course for 7 half days
- **2nd class - 5th class (year 6 - year 9):** optional subject for two lessons (50 minutes) in a week
- **6th class - 8th class (year 9 - year 12):** elective course for two lessons in a week

All parts above mentioned are not compulsory and can be attended by students on a voluntary basis. However, according to the coordinator in

[8]Source: http://www.keplersternwarte.at/ [Online; Last call 26.July 2020]

charge, almost all students attend at least the trial course. There are also compulsory subjects that include learning content from the field of robotics. More on this in the following section.

Detailed Curriculum

The robotics curriculum of BRG Kepler in more detail is considered:

- **year 5:** Within the context of three trial afternoons and the possible complementary participation in four RoboDays, the students have the opportunity to get to know the programming of Lego robots with the programming language C. By means of tasks, they learn simple motion sequences by controlling the motors, as well as how to read in sensor values. The used hardware is the LEGO NXT robot. The programming Language NXC and the development environment BrixCC.
  The learning objectives for this year include building a robot, getting to know the BrixCC development environment and the basic structures of an NXC program (e.g. controlling motors, loops, display outputs).
- **year 6:** In this year, students learn the basics of the programming language C++. Using simple programming examples on LEGO EV3 robots, the students learn how to control the motors, read in and evaluate the sensors, as well as user input and graphic output on the EV3 brick. The used development environment is called "EVCDevelop[9]".
  After the course the students work independently on tasks of the KNAPP RoboLeague, where solutions to basic problems of robotic systems, such as navigation and route planning, the prevention of collisions or finding targets need to be addressed.
  Towards the end of the school year the pupils have the possibility to participate in the KNAPP RoboLeague competition.
  The learning objectives for this year include getting to know the development environment of EVC-Develop and the basic structures of the programming language EVC (e.g. controlling motors, if conditions, logical operators).
- **year 7:** In this year the focus is put on solving more complex problems, designing and building robots, such as extended program structures.

[9] http://www.evcdevelop.at/ [Online; Last call 13.November 2020]

Based on the task "Rescue Line" of the RoboCupJunior, mechanical constructions and software solutions have to be developed independently.
In spring, the students have the opportunity to participate in the RoboCupJunior Austrian Open.
The learning objectives for this year include, for example, getting to know the physical limits of sensors and time-controlled processes.

- **year 8:** Based on the developments and solutions of the past year and the experience gained during the participation in the RoboCupJunior Austria, the construction and the software of the Rescue Line robots will be further developed. Accompanying this, the students will learn a new approach to the processing of sensor values using algorithms in this school year.
  This year's learning objectives include debugging, getting to know arrays, documenting sensor values and the P-, PI- and PID-algorithm.
- **year 9:** After the students have partly reached the limits of the LEGO EV3 robotics system, they will now find extended possibilities to solve the mechanical or sensory problems which could not be solved with the old hardware. In the process, they will also learn the basics of electronics and a microcontroller. The students will work with Kepler-Brain[10]. The used programming language is C++, the development software is mbed.
  There is also the possibility to develop sensors themselves and integrate them in their own solutions.
  The goal is to participate in the RoboCupJunior Austrian Open with a robot design and software solution that completely solves the Rescue Line task.
  There is also the possibility to deal with new challenges in the context of the tasks "Rescue Maze[11]" or "Soccer[12]" while changing to a new robotic hardware.
  Goal is the participation in the RoboCupJunior Austria and, in the case of qualification, participation in the world championships.

[10] http://www.keplerrobotik.at/www/index.php?seite=keplerbrain [Online; Last call 13.November 2020]

[11] https://junior.robocup.org/rcj-rescue-maze/ [Online; Last call 13.November 2020]

[12] https://junior.robocup.org/soccer/ [Online; Last call 13.November 2020]

The learning objectives for this year include getting to know KeplerBrain, the development interface mbed[13] (e.g. creating projects, integrating libraries, transferring programs) and using the KeplerBrain library.

- **year 10:** The elective subject Robotics is intended to give students the opportunity to enter the world of robotics without prior knowledge or to take the step towards independent development based on previous knowledge. The main focus here is laid on the design and manufacture of 3D components using CAD software and a CNC portal milling machine or 3D printer. The same robotic system is used as in the year before. In case of CAD LiteCAD(2D Constructions), Designpark Mechanical (3D Construction), SheetCAM(CAM CNC-portal milling machine) and Cura (CAM 3D-printer) is used.
  The goal is again participating in the RoboCupJunior Austria and, in the case of qualification, participating in the world championships.
  The learning objectives include, for example, the creation of CAM data for a 2D CNC milling machine, creation of CAM data for a 3D printer, but also basic electronic networks (e.g. SPI sensors, I2C sensors).
- **year 11:** The students work and research rather independently and develop the hardware for their robots in the field of electronics as well as the mechanics. Accompanying this they learn the design of printed circuits with a layout software up to the production. Additional to the last year Sprint Layout and DesignSpark PCB are used. In addition, students can deal with object-oriented programming and digital image processing and object recognition in the field of software development.
  The goal is again the participating in the RoboCupJunior Austria and, in the case of qualification, participating in the world championships.
  In addition to getting to know the above mentioned software (Sprint Layout and DesignSpar PCB), the learning objectives include the independent further development of the old solutions of competition tasks.
- **year 12:** Students work independently on their development, supervision is individual and lessons are offered according to their needs. Finally, students who have attended the elective subject robotics for at least two years could do an oral exam in the context of school leaving

[13] https://os.mbed.com/ [Online; Lst call 13.November 2020]

examination in the field of robotics.
The goal is again the participating in the RoboCupJunior Austria and, in the case of qualification, participating in the world championships. The learning objectives of this school year are to improve the previous solution of the competition task and the preparation for the oral exam in the context of the school leaving examination.

Aspects of robotics are also integrated into compulsory lessons (taught from year 5 to year 9, one lesson per week):

- **year 8:** As part of the computer science lessons in year 8, students deal with the Arduino microcontroller. Programming is done with the graphical programming language "Scratch 4 Arduino". It is tried to build up interdisciplinary teaching with physics. (Kepler IT, 2020)
- **year 9:** The 9th school year is also the only one in Austria in which computer science lessons are compulsory. Therefore the contents of this curriculum must also be fulfilled. Nevertheless, students in the summer term deal with the Arduino again. This time they work with the textual programming language C. For example, the students program a traffic light with three LEDs, a stopwatch or estimate the capacity of a condenser. (Kepler IT, 2020)

Students can also choose the subject "Applied Information Technology" (taught from year 9 to year 12, two lessons a week). Here as well, contents related to robotics are taught:

- **year 10:** In this year students work with the Arduino-based KeplerFischBRAIN Controller. Programming is done with the language C++. Students learn, for example, how LEDs on the controller can flash, how texts can be displayed, how integer variables can be defined or how motors can be controlled on the controller. However, students also learn how to develop controls for more complex models such as a traffic light, a safe, or an ATM. (Kepler IT, 2020)
- **year 11:** The curriculum of this year is based on the knowledge gained over the past year. Work is done with the OpenBOT [14] system. Students will learn how to display static texts or variables on the OpenBOT

[14] http://www.keplerrobotik.at/www/index.php?seite=kepleropenbot [Online; Last call 13.November 2020]

main board, how to read values from sensors (e.g. light sensor) or how to use (control-)algorithms (e.g. P-algorithm) correctly. In the context of this course, programming skills are deepened in the context of the C/C++ language. (Kepler IT, 2020)

Conclusion

This curriculum is created for a period of 8 years. It is the only curriculum looked at in this thesis that contains compulsory content for all students, although most of the content can be attended on an optional basis.

BRG Kepler plays a leading role in teaching robotics in schools not only in Austria but also worldwide. On the one hand, this is shown by the long-term development work and the integration of robotics into the school profile and, according to Kepler Robotik (2020), by the following successes in the RoboCup:

- 39 x Austrian champion for teams of the BRG Kepler
- 1 x 1st place at world championships for a team of the BRG Kepler
- 5 x 2nd place at world championships for teams of the BRG Kepler
- 5 x 3rd place at world championships for teams of the BRG Kepler

2.2.4 Asia - China

Now we will look at a curriculum from China. First of all, we will give an overview of the Chinese school system in order to better classify the curriculum.

2.2.4.1 School System

Education in China is divided into three categories: basic education (School), higher education (Universities and Colleges) and adult education. Compulsory education lasts nine years, consisting of 6 years of primary school as well as 3 years of junior secondary education. (Kan, 2019)

Basic education consists of pre-school education (usually three years), primary education (six years, starting at the age of six) and secondary education (six years). The secondary education has two routes, the academic secondary education and specialized/vocational/technical secondary education. Academic secondary education consists of junior and senior middle school, each takes three years. Students of the junior middle school wanting to continue their education take a locally administrated entrance exam. On the basis of the exam they will then have the opportunity to continue in an academic senior middle school or enter a vocational middle school (or leave school at this point) for two or four years. Students who graduate from senior middle school and want to study at a university must take the national higher education entrance exam (Gao Kao). (Kan, 2019)

Figure 2.6 shows an overview of the school system of China.

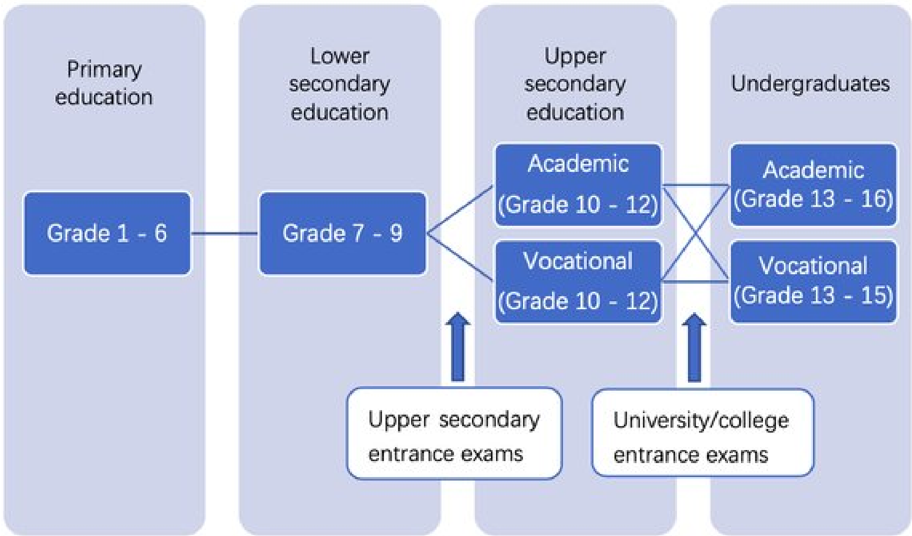


Figure 2.6: Overview of the Chinese School System (Dai and Martins, 2020)

2.2.4.2 Robot Design and Programming

The curriculum was designed by Mr. Jiaxiang Shi from Shanghai, who teaches robotics in school. According to Mr. Shi, Robot teaching in China has different forms and uses different equipment. An example of such a curriculum is presented in the next section.

Field of Application and Facts

The curriculum is used in the area of K-12 education. In order to be able to work with the curriculum an EV3 LegoMindstorm robot is required. Furthermore no previous knowledge is necessary for the work. According to Shi, Jiaxiang (2020) the curriculum follows 3 basic theories:

1. Follow the three dimensional goals of teaching in the context of robotics:
    - Emotional attitude values: for example, students should discover an interest in robotics during the course.
    - Process and methods: students should go through all steps from the simple identification of the bricks, to the assembly and the programming.
    - Knowledge and skills: students should acquire basic theoretical knowledge about robots and then apply it in practice.
2. Teaching difficulties: How to gain a sense of self-achievement in learning, improve the ability of space, logical thinking, and obtain the value orientation of self-improvement.
3. The following points/objectives are mentioned for working with the students:
    - Practical experience should foster interest in working with robots.
    - Cooperation among students should be encouraged.
    - Patience is needed during the assembly process so that the students can be creative.
    - Students should understand the necessary theory in the context of robotics. But it is also important that they are given the necessary freedom to experiment.

Detailed Curriculum

The curriculum of Shi, Jiaxiang (2020) is divided into 10 modules:

- **Meet the robot:** In this unit students learn how a robot is defined, what basic functions a robot performs and what basic conditions for the composition of robots exist (e.g. mechanical parts, the robot brain and so on).
- **The category and development of robots:** In this unit students learn the difference between industrial robots and service robots. They also learn about different areas of application. Students are also given an insight into the history of robotics and an outlook on further developments in the future.
- **EV3- hardware:** This lesson takes a closer look at the EV3. First of all, the screen of the EV3 is explained in more detail and the control options the user has. Afterwards, motors of EV3 kit are explained and how they differ. The EV3 brick's ports and their purpose are also explained. Finally, it is explained how the robot can be controlled via the brick (programming blocks on the brick) and how values measured using sensors can be displayed on the screen.
- **EV3- software:** This chapter introduces students to the EV3-G language development environment. The introduction is very basic: It includes an installation guide, a basic explanation of how the development environment works and a detailed explanation of the different building blocks or the different categories of building blocks (e.g. movement, measurement and so on).
- **Transmission:** This chapter deals with different gear structures. In addition, driven (turned or moved by driving gear) and driving(source of power) gear wheels are discussed in more detail. In the course of this, students will take a closer look at the "tooth number ratio". Students should also construct a model with different gears themselves.
- **Controlled movements:** This chapter deals with the basic movements that a robot should perform, including, for example, driving straight for a certain number of centimetres or taking curves. Work is done with the graphical development environment of LEGO Mindstorm.
- **Using sensor for curving:** In this session students meet the first sensor- the Gyro-sensor. Students already come across the problem of the

sensor's accuracy. Students solve more challenging tasks in this session, such as driving a square.
- **Driving along a line:** Line tracking is going to be implemented in this unit. In the course of this, students learn about the color sensor. Students extend a given program to improve the algorithm for line following.
- **Space Challenge - activate communication:** This chapter is based on the Space Challange [15] from LEGO . The goal[16] is to navigate a robot to a satellite and activate it with a push.
- **Space Challenge - assemble your Team:** The last lesson also deals with the Space Challenge. In this exercise[17] the flight commander has to be lifted up with the help of the robot arm and then taken to the base area and set down there.

Conclusion

The present curriculum serves as an introduction to the field of robotics. It contains basic knowledge of the field of robotics such as "What is a robot?" and more challenging exercises such as the Space Challenge. To get started in programming, a graphical programming language is used. It also includes the integration of social skills as a learning objectives. This point was not explicitly mentioned in the other curricula.

2.2.5 North America - Canada

Now we will look at a curriculum from Canada. First of all, we will give an overview of the Canadian school system in order to better classify the curriculum.

[15] https://education.lego.com/en-us/lessons/ev3-space-challenge/ [Online; Last call 27.July 2020]

[16] https://education.lego.com/en-us/lessons/ev3-space-challenge/2-activate-communications [Online; Last call 27.July 2020]

[17] https://education.lego.com/en-us/lessons/ev3-space-challenge/3-assemble-your-crew [Online; Last call 27.July 2020]

2.2.5.1 School System

Most children in Canada attend the kindergarten for a year or two at the age of five or four. Then they attend school- the first grade becomes mandatory at an age of six. The education system in Canada depends on the province. Generally, the the education system has four levels: pre-elementary (kindergarten), primary, secondary and post-Secondary education (university or college). (University of the People, 2020)

The education system depends on the province in which the school is located. Figure 2.7 gives an overview of the different school systems of the different provinces.

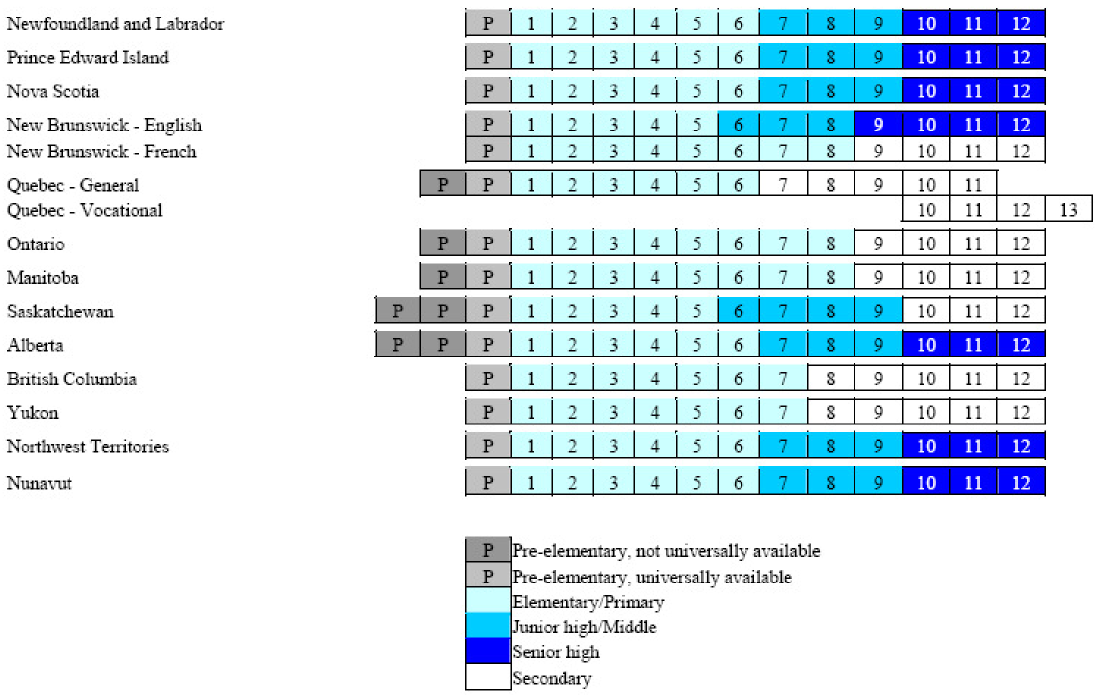


Figure 2.7: Overview of the Canadian school system, divided into provinces [18]

[18]Source: http://www.ibe.unesco.org/fileadmin/user_upload/archive/Countries/WDE/2006/NORTH_AMERICA/Canada/Canada.htm [Online;Last call 28.July 2020]

The curriculum, which will be presented, is used in the province of British Columbia (BC). The elementary school in BC includes the kindergarten and the primary school (grade 1 to 7). Secondary school is attended from 8 to 12 or 9 to 12. In districts with a large number of students there is a middle school (grade 6 to 9). Students finish school at the age of 18 and receive a graduation certificate. (British Columbia - Ministery of Education, 2020a)

2.2.5.2 Curriculum - Applied Design, Skills, and Technologies

The curriculum "Applied Design, Skills, AND Technologies (ADST)" has also been developed by the British Columbia Department of Education. The contents of the curriculum will now be discussed.

Field of Application and Facts

The curriculum covers the entire school education from kindergarten to grade 12. The ADST curriculum is an experiential, hands-on program of learning through design and creation which include skills and concepts from the existing disciplines of business education, information and communication, home economics and culinary arts, technology education and from new and emerging fields. The curriculum is designed to develop the skills students need to find creative and innovative answers to daily challenges. (British Columbia - Ministery of Education, 2020b)

The three main parts according to British Columbia - Ministery of Education (2020b) of the curriculum are:

- **Applied design:** The design process from concept to completion.
- **Applied skills:** Skills that support the design process, such as cooperation and collaboration or research skills.
- **Applied technologies:** Skills needed to gain access to technologies that support the design process and design thinking.

From kindergarten up to grade 5, the ADST has not defined any learning contents. The aim is to create interdisciplinary opportunities to develop foundational mindsets and skills in design thinking and making. (British

Columbia - Ministery of Education, 2020b)
Classes 6 to 9 are intended as exploration years. Students and teachers have different modules to choose from. Which modules they cover and how deeply the content is covered is left to the teachers based on the interests and passions of the students. (British Columbia - Ministery of Education, 2020b)
In the school years 10 to 12, students can dig deeper into certain areas. They are free to choose which areas(e.g. business education, information and communications technologies, technology education) they are interested in. (British Columbia - Ministery of Education, 2020b)

Contents from robotics are also included in the curriculum. In the school years 11 and 12 there are even standalone courses. In the following the contents of the curriculum related to robotics are going to be presented.

Detailed Curriculum

From grade 6 to 9, robotics is run as an independent module. Per year according to British Columbia - Ministery of Education (2019) the modules have the following contents:

- **grade 6/7:** In this module, students are taught that a robot is a machine that executes complex movements. Also the application areas of robotics, the main components of a robot (sensors, control systems and so on), ways how an object can move, programming and logic for robotic components and different platforms of robotics should be taught.
- **grade 8:** In this module students learn about the use of robotics in local contexts (e.g. use of robots in companies in the hometown), types of sensors, user and autonomous control systems, uses and applications of end effectors, movement- and sensor-based responses, program flow, interpretation and use of schematics for assembling circuits, identification and applications of components and various platforms for robotics programming.
- **grade 9:** This module is a combination of robotics and electronics. So students learn about the use of electronics and robotics, components

of an electric circuit, ways in which various electrical components affect the path of electricity, Ohm's law, tools for PCB (printed circuit board) production, basic robot behaviours using input/output devices, movement- and sensor-based responses, and microcontrollers, mechanical devices for the transfer of mechanical energy, mechanical advantage and power efficiency, including friction, force, and torque, robotics coding and various platforms for robotics programming.

In grade 10 robotics is combined with electronics. In grade 11 and 12 robotic is as already mentioned above as a standalone course. Per year according to the modules have the following contents:

- **grade 10**: As already mentioned, robotics is combined with electrical engineering in this module. This results in the following learning contents: design opportunities, Ohm's law, breadboard circuitry, production of simple circuits from schematic drawings, electronic diagnostic and testing instruments, function and application of components (e.g. light-emitting diode- LED), construction sequences involved in making a working circuit, function and use of hand tools and operation of stationary equipment (e.g. box and pain brake), cases (e.g. 3D printed) for enclosing a circuit, sequences involved in making a functional robot, robot elements, block-based coding or logic-based programming for robotics, programming platforms for robotics and flow charts related to robotics behaviour. (British Columbia - Ministery of Education, 2018a)
- **grade 11**: In this robotic course students learn about simple robotic design and production, interaction of robotic subsystems, relation of structure and power to motion, relation of sensors and control to logic, friction and traction, power and torque, developments in robotic technology, robotic technologies in the community and industry, similarities and differences between remotely controlled and autonomous robots, programming related to microcontrollers and design for the life cycle (British Columbia - Ministery of Education, 2018b).
- **grade 12**: After the course students will know advanced robotics design and production. This includes: sensors, robotic technologies in industry, research and education, syntax language related to robotics, flow charts, hierarchy charts and data sheets with standard symbols, feedback loops (e.g. position control), communication protocols, bat-

tery technology, wireless communication options, wiring and cabling, robotic systems working together to complete a challenge or task, design for the life cycle, future career options and opportunities in robotics design, production, and emerging applications and interpersonal and consultation skills for interacting with colleagues and clients (British Columbia - Ministery of Education, 2018c).

Conclusion

Similar to Australia, there is also a curriculum which was written by the authorities. However, it is designed for the whole school career. This means that pupils could already be taught the first robotics contents in primary school. But the contents are not obligatory in this case either. In terms of content, electrical engineering play an more important role alongside programming.

2.2.6 Overview of Curricula

Table 2.1 provides a summarised overview of the curricula considered. A more detailed analysis is given in Section 7.1.

2.3 Comparison of Graphical and Textual Programming

In this chapter the difference between textual and graphical programming languages will be introduced. A special focus is placed on the use in context of robotics. Moreover, based on related research it is discussed which programming language is more suitable for beginners.

Name of currilum	Country	Durration	Target age	Kits	Main contents
Engineering^3	USA	4 years	14 - 17	Engineering^3-Kits and control elements (4 parts)	Problem solving structures and through code, mechanics, engineering design, Oneshape/CAD, custom parts design, control systems, programming and data collection
Robotics and Mechatronics	Austrialia	2 years	16 - 18	Arduino UNO	Electronics (e.g. design and construction), programming of control systems and micocontroller, mechatronic systems, robotics impact on human society, design processes and AI.
BRG Kepler	Austria	8 years	10 - 17	Mindstorm EV3 and NXT, KeplerBRain, Arduino UNO, KeplerFichBRAIN controller, KeplerOpenBOT	Programming, focus on competitions, mechanical construction, sensors (e.g. construction), electronics and microcontroller, design and manufacture of 3D with CAD and digital image processing.
Robot design and programming	China	1 year	K-12	Mindstorm EV3	Logical thinking, structure of a robot, programming with EV3-G and working with sensors.
Applied Design, Skills, and Technologies	Canda	7 years	11 - 17	Up to school, e.g. Arduino UNO	Structure of a robot and field of application, programming, electronics mechanics, sensors, use of hand tools and operation of stationary equipment.

Table 2.1: Overview of considered curricula

2.3.1 Text-Based Programming

Within the framework of a text-based programming language, commands are expressed in the form of a formal language. The statements(syntax) based on a set of rules, which are called grammar (Sunitha, 2013). The two most popular programming languages in school are BASIC and Python (Bedford, Mike and Schischka, Sabine , 2018).

Both programming languages can also be used in robotics. The EV3 robot from LEGO can be programmed with both programming languages. Both BASIC[19] and Python[20] versions are available free of charge. Which of the two languages is used should be decided individually. A decision criteria could be, which programming language fits better into the curriculum of the respective school where it should be used to work with the robots.

Why are textual programming languages generally used in industry? One advantage of a textile programming language is that it offers a better overview than a graphical programming language, especially for larger programs(Merkel, 2020). Thus, subroutines can be used to structure the entire programme better and more effectively. Therefore, the use of a textual programming language is more effective in the context of more complex programs. Another advantage of textual programming languages is that they can be used in many more ways than a graphical programming language. These often have a very limited field of application (Citrin et al, 1995).

2.3.2 Block-Based Programming

The need to better prepare students for our digital world has led to the redesign of some computer science curricula. Many of these new curricula use a block-based programming language for beginners. In recent years, the visual approach to programming has become more and more widespread. The most popular language in this area is Scratch[21] (see Figure 2.8) (co-

[19] https://sites.google.com/site/ev3basic/ [Online; Last call 05.August 2019]
[20] https://sites.google.com/site/ev3devpython/ [Online; Last call 05.August 2019]
[21] https://scratch.mit.edu/ [Online; Last call 16.November 2020]

dakid, 2020). Other popular languages are Snap! and Blockly. (Weintrop and Wilensky, 2017)

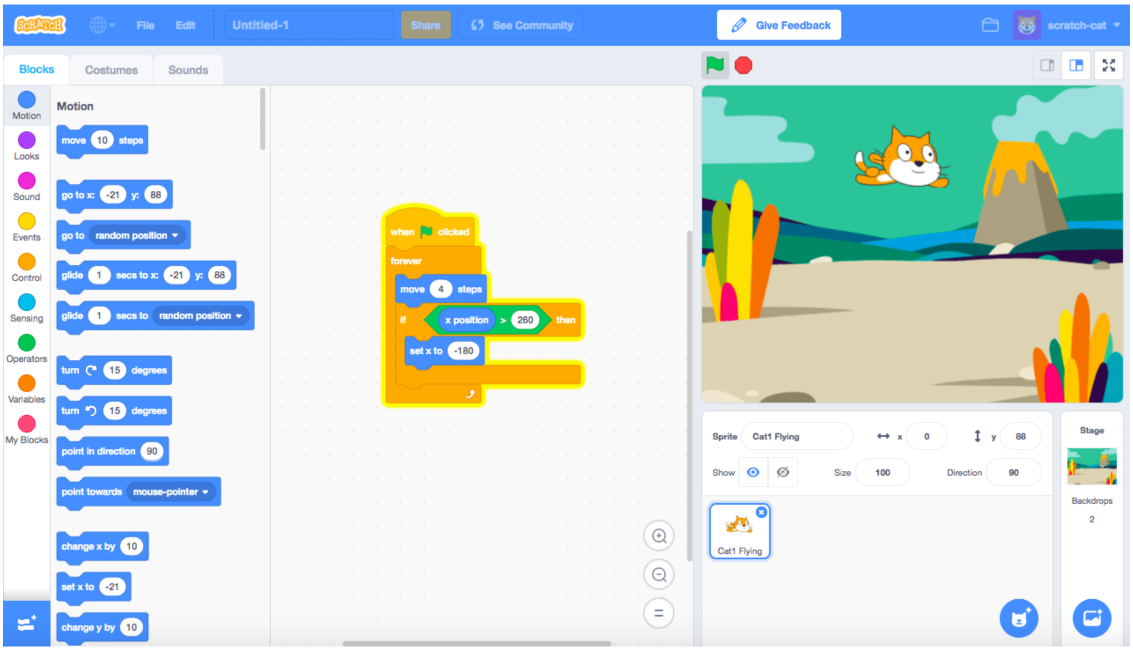


Figure 2.8: Graphical Programming - Scratch [22]

In Figure 2.8 you can easily see where the name graphical or block-based programming comes from. Programming is done with the help of building blocks. Only certain (matching) blocks can be connected to each other. This avoids syntax errors. To make it easier to differentiate between blocks in terms of their functionality, they have different shapes and colors. (Weintrop and Wilensky, 2017)

Graphical programming languages can also be used in the field of robotics. For example, Arduino has its own version of Scratch[23]. For the EV3 of Lego there is an official version of Lego EV3-G[24] and there is also a open source

[22]Source: https://guidoknaus.ch/?p=745 [Online; Last call 05.August 2019]
[23]http://s4a.cat/ [Online; Last call 05.August 2019]
[24]https://www.lego.com/de-de/mindstorms/learn-to-program [Online; Last call 05.Aug 2019]

alternative with MakeCode [25]. The advantage of the latter version is that it does not need to be installed. Programming can be done online as with Scratch.

2.3.3 Textual vs. Graphical Programming

With the introduction of graphical or block-based programming languages, the question arose whether a block-based programming language is a better entry point into programming for novices than a text-based programming language. Despite the increasing use of block-based programming languages, there are only a few studies that compare textual and graphical programming in the context of learning success (Weintrop and Wilensky (2015),Weintrop and Wilensky (2017)). There are even fewer studies in the field of ER. In contrast, there is already a growing number of studies which show that block-based programming is well suited for the entry into programming (Weintrop and Wilensky (2015), Franklin et al (2017), Weintrop et al (2018)).

Two studies will be reviewed in more detail in the following paragraphs. One study aim at a comparison of block-based and textual-based programming in general while the other one performs the comparison in the context of Arduino[26].

A general comparison was carried out in the research by Weintrop and Wilensky (2017). In one school two groups were formed, which dealt with the same curriculum. The programming environment Pencil.cc[27] (see Figure 2.9), which is a hybrid textbased/graphical environment, was used. One group used the text-based programming language while the other one used the block-based programming language. The study was conducted in an introductory course for programming. In the study, both groups improved in comparison to their pre-assessment. In both cases the improvement was significant, but the improvement was in numbers higher in the block group.

[25] https://makecode.mindstorms.com [Online; Last call 05.August 2019]
[26] https://www.arduino.cc/ [Online; 06.August 2019]
[27] http://pencil.cc/ [Online; 06.August 2019]

A t-test showed a significant difference between the groups. Those students who worked in the block group performed better. (Weintrop and Wilensky, 2017)

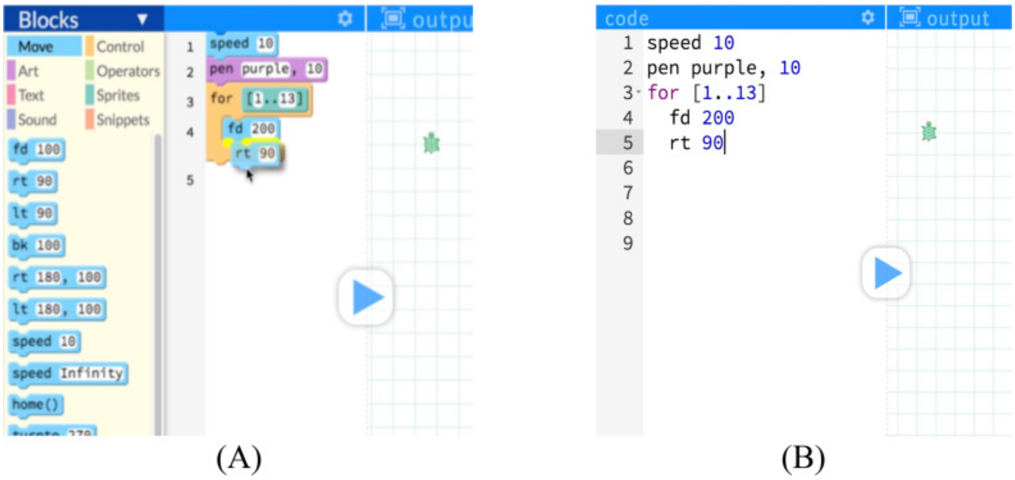


Figure 2.9: Block-based (A) vs. text-based (B) Programming - Pencil.cc (Weintrop and Wilensky, 2017)

In the article of Booth and Stumpf (2013), a comparison was made within the context of Arduino. The study was conducted with adults who had no programming experience with Arduino. In addition, all participants worked with both the block-based and the textual programming environment. The textual version was the default programming environment of Arduino. For the graphical environment the Modkit Alpha Editor (see Figure 2.10) was used. The study showed advantages in the context of the graphical programming language when modifying programs. The user experience (usability) was also higher in the context of the graphical programming language. In addition, the work load was perceived as lower and the feeling of success was greater. This study also concluded that a visual approach can be useful for beginners. However, it also states that further research is needed. (Booth and Stumpf, 2013)

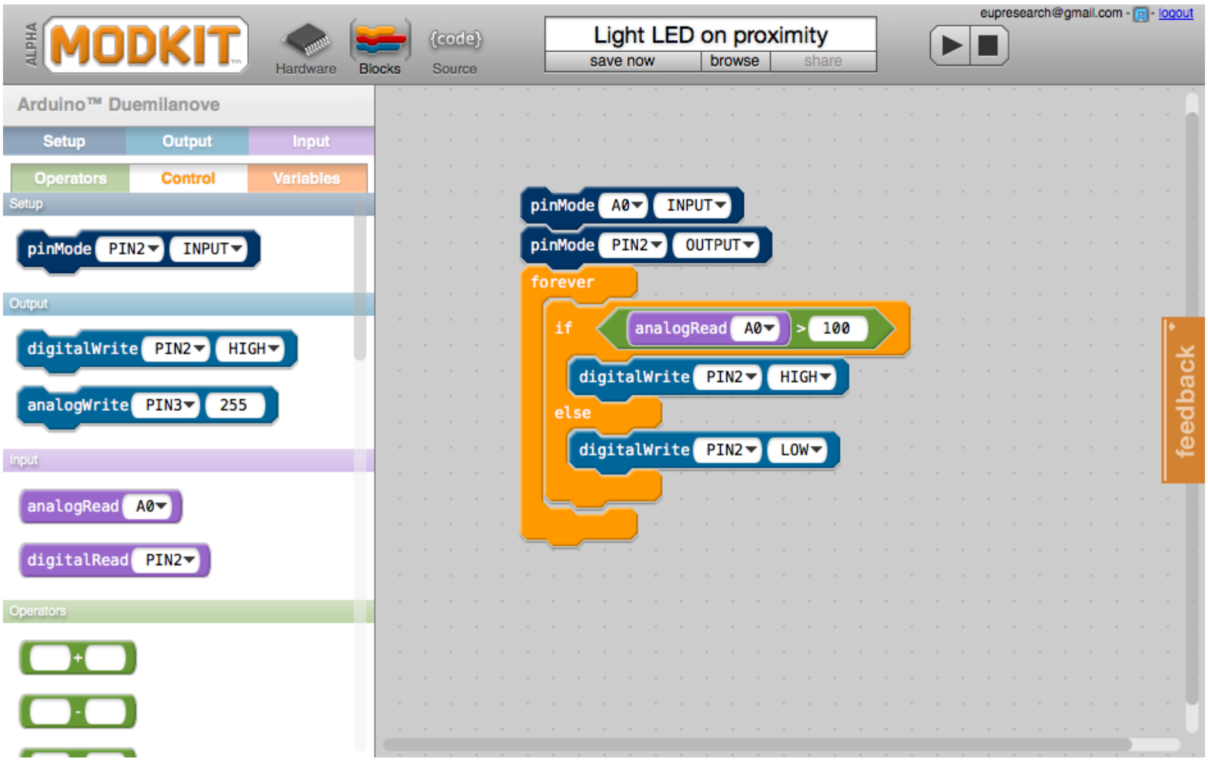


Figure 2.10: Block-based vs. text-based Programming - Modkit (Booth and Stumpf, 2013)

3 Experimental Setup

This chapter describe the test environment we will use to evaluate whether textual or graphical programming is more suitable for beginners in the context of Educational Robotics. The structure of the school and its actual situation in the field of robotics is described in more detail. Furthermore, the experimental setting is described in detail in the context of a comparison of graphical vs. textual programming and the assessment of learning success in the context of computational thinking and problem solving, as well as the impact on social aspects.

3.1 Test Environment- High School (BG/BRG/BORG) Köflach

The BG/BRG/BORG Köflach is located in the town of Köflach (Styria, Voitsberg) in Austria. For detailed information of the education system of Austria see section 2.2.3.1. Figure 3.1 shows the school building. At the moment about 1000 students attend the school and are taught by about 90 teachers. The school has 3 departments:

- **Languages (BG):** BG stands for a humanistic academic secondary school, i.e. a school whose focus is primarily on languages and general education. Students between year 5 and year 12 attend this department.
- **Natural Sciences (BRG):** BRG stands for a academic secondary school with a focus on natural sciences and general education. Students between year 5 and year 12 attend this department.
- **Sports (BORG):** In the BORG department of the school the focus is put on sports and again on general education. Students between year 9 and year 12 attend this department.

Figure 3.1: Picture of the campus of the BG/BRG/BORG Köflach [1]

3.1.1 Computer Science in Köflach

First an overview of the computer science lessons at the school in relation to different grades is given. The current content of the robotics lessons is going to be discussed later:

- **year 5:** In the first school year, computer science is integrated into the German lessons. The extent is one lesson per week. The focus lays mainly on learning the basics IT skills (e.g. sending mails) and word processing (e.g. Word).
- **year 6:** In the second school year the computer science lessons are again integrated into a subject. This time it is mathematics. The amount of lessons is half a lesson per week. Classes start with the summer semester so students have one hour per week of computer science in the second term. This year students work with spreadsheet programs (e.g. Excel) and mahtematic apps (e.g. GeoGebra).
- **year 7:** In the third year, students have to decide whether they want to attend the BG or the BRG department of the school. In the BG department there are no computer science lessons in the 3rd grade. In the BRG section computer science is obligatory with two lessons per week. In this school year, the main focus is laid on working with the explorer, word processing, presentation and spreadsheet programs. Students will also working with robots in class for the first time.

[1]Source https://www.koeflach.at/leben-in-koeflach/schulen/bgbrg-koeflach/ [Online; Last call 09.August 2020]

- **year 8:** At the moment there are no compulsory computer science lessons in the fourth school year. Students of the BRG are free to join a robotics club. This takes place every two weeks in the afternoon for 2 hours.
- **year 9:** The fifth year is the only year in which computer science is, based on the education curriculum of Austria, compulsory for all students (BG, BRG and BORG). In BG and BRG the students have 2 lessons of computer science per week, in BORG one lesson. In BG the focus is laid on working with the explorer, word processing, presentation and spreadsheet programs. In BRG the students learn programming with Visual Basic, work with Access, learn the basics of website design with HMTL and CSS and work with the Arduino UNO in the context of an experimental box. In BORG the students learn about the explorer and word processing.
- **year 10:** This year, BORG students again attend computer science classes for one lesson a week. Content of this year are spreadsheet and presentation programs. In the BRG, students must attend computer science classes for 2 lessons a week. The knowledge in programming (e.g. subroutines, algorithms), spreadsheet programs and web design is increased. Students of the BG can choose computer science as a compulsory optional subject for the next three years (2 lessons per week). The content always corresponds to the content of the BRG from the previous year. In this case, for example, the content of the 5^{th} class.
- **year 11:** From this year, computer science lessons are no longer compulsory for any of the 3 departments. However, students of the BRG can choose computer science as a compulsory elective subject (2 hours per week). The contents are cryptography, databases with dbSQL, network technology and robotics. This course was first offered in school year 2019/20.
- **year 12:** This course will be offered for the first time in the school year 2020/21. The planned contents are web programming (PHP, Javascript), Artificial Intelligence and robotics. The contents of this course will only be taught to students of the BRG.

3.1.2 Robotics at BG/BRG/BORG Köflach

This section describes the state-of-the-art of Robotics at BG/BRG/BORG Köflach. First an overview of the available robot kits is given.

3.1.2.1 Used Robot Kits

The school is currently provides Ozobots, EV3 Lego Mindstorms and Arduino UNOs.

Ozobots

The first robot which is described in more detail is the Ozobot. The school bought 9 Ozobots in the school year 2017/18. At the moment there are two different models of the Ozobot, the Ozobot Bit 2.0 and the Ozobot Evo. The school owns the model Ozobot Evo. The two models differ in the provided features. In addition to the functions described below, the Ozobot Evo has distance sensors, programmable LEDs and speakers. Also users have the possibility to connect the robot via bluetooth with an app and control the robot. Both models of the Ozobot can be programmed with colors and lines as well as blocks. Figure 3.2 shows the Ozobot Evo. (Ozobot Deutschland, 2020)

Both models use light sensors to follow lines and to detect the different colours. The Ozobot is able to follow a black line. Along this black line different color codes can be placed. These are the colors red, green and blue. For example, the robot can be told in which direction it should turn at the next crossroad or with which speed it should move. On the website[2] of Ozobot there is an overview of all possible combinations of color codes which the robot is able to recognize. (Ozobot, 2020a)

[2] https://files.ozobot.com/stem-education/ozobot-color-codes.pdf [Online; Last call 10.August 2020]

Figure 3.2: Figure of OzobotEvo [3]

The second possibility of programming is by using blocks. Ozobot provides an online development environment[4] for this. This environment offers different blocks, for example to control direction and speed, control structures such as loops or possibilities to control LEDs. To load the program onto the robot, the Ozobot must be held on a white spot on the screen and then be calibrated. Afterwards, the program is transferred to the Ozobot by means of light signals. This has the advantage that the robot does not need to be physically connected to the computer and no software needs to be installed on the computer. Figure 3.3 shows the development environment with an example code and the area where the robot must be held to transfer the program. (Ozobot, 2020c)

[3]Source https://ozobot-deutschland.de/ozobot-evo/ [Online; Last call 10.August 2020]

[4]https://ozoblockly.com/editor?lang=en&robot=evo&mode=2 [Online; Last call 10.August 2020]

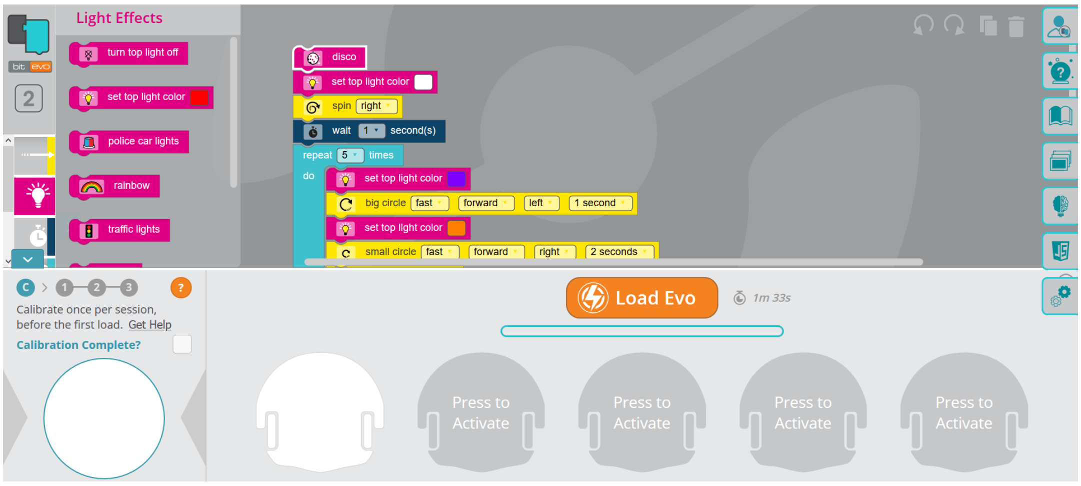


Figure 3.3: Figure of the development environment OzobotBlockly

Lego Mindstorm EV3

The second robot kit which is used at the school is the Lego Mindstorm EV3. The school bought 8 robots of this model in the school year 2018/19. At the school the EV3 Education Set is used. The set includes the programmable EV3 brick, sensors and motors. Using the Lego component contained in the package you can build different models from a simple car to a robot on two legs. Figure 3.4 shows an exemplary setup in which sensors such as the touch sensor or light sensor are included. (LEGO Education, 2020)

The programming possibilities are manifold. There are different programming languages and platforms including EV3-G (graphical), LeJOS (Java), MonoBrick (C#, F#, IronPython), c4ev3 (C, C++), ev3dev (Python, JavaScript, Java, Go, C++, C, Vala), Scratch (ScratchX), NEPO(graphical), ROBOTC (C) and EV3-Basic (Small Basic, Python). At BG/BRG/BORG Köflach the languages EV3-G and EV3-Basic are used at the moment. (Deitelhoff, 2017)

Let us take a closer look at the two programming languages used in school. We will first consider EV3-G. The G stands for graphical, because this language is a graphical or block-based programming language. It was developed by LEGOEducation. Programs can be transferred directly to the robot.

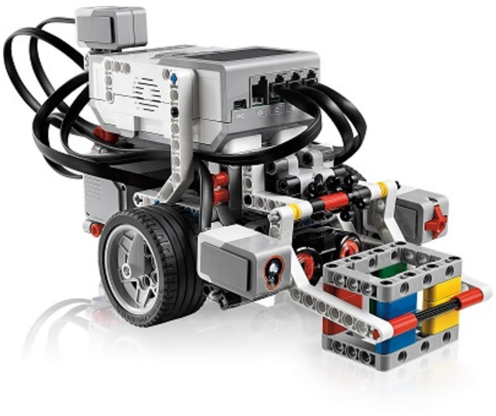


Figure 3.4: Exemplary setup for LegoMindstorm EV3 [5]

The programming language can be used to control all motors, sensors, buttons and screens of the EV3. In addition to blocks for control structures such as conditionals and loops, there are also blocks for mathematical operations or variables. Figure 3.5 shows the development environment with examples of placed and connected blocks and the basic part of the GUI for motor control, screen output, sound output and lighting control. The figure shows a section of an example. The goal here is to keep the distance to a wall. If the distance is between 10 and 20 cm, the robot moves straight, if the distance is greater than 20 cm, the robot corrects towards the wall and if the distance is less than 10 cm, the robot moves away from the wall.

EV3 Basic is used as an textual programming language. This programming language is based on Microsoft Small Basic, which was extended with EV3 functionality. With the extension it is possible to write Small Basic programs that can interact with the EV3 motors, sensors, speaker, screen and buttons. If the EV3 brick is directly connected to the PC, the program can be executed directly on the robot. If this is not the case the EV3Explorer is needed. The program can then be loaded directly onto the EV3 and executed there. Figure 3.6 shows the development environment. (EV3 Basic, 2020)

[5]Source https://www.channel-e.de/nachrichten/article/ti-bausteine-im-roboterbaukasten-lego-mindstorms-ev3 [Online; Last call 10.August 2020]

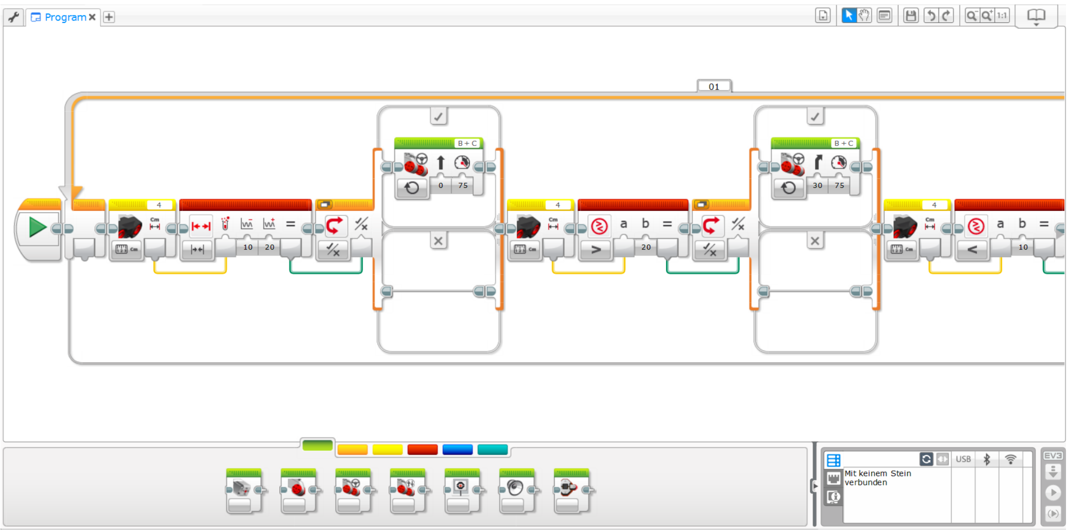


Figure 3.5: Exemplary setup for LegoMindstorm EV3

Arduino UNO

The third model used at the school is the Arduino Uno microcontroller. The model was bought in the school year 2017/18. The microcontroller is used in the context of an experimental box[6] (measure, actuation, control) and is also integrated in a car, the Smart V3.0 Robot Car. The experimental box contains sensors for different measurement like temperature sensors and ultrasonic sensors, components for actuation like a traffic light or an LED cube, and components for controlling like a fan. The robot car (see Figure 3.7) consists of 4 motors and sensors for distance measurement (ultrasonic sensor) and sensors for line tracking.

The official development environment of Arduino is the Arduino IDE[7]. It is based on the programming language C/C++. Scratch for Arduino is a graphical alternative. However, at the school only the official Arduino IDE is used in regular lessons. Programs can be transferred directly to the Arduino connected via USB-cable.

[6] http://msr.leo-edv.com/ [Online; Last call 17.November 2020]
[7] https://www.arduino.cc/en/main/software[Online; Last call 10.August 2020]

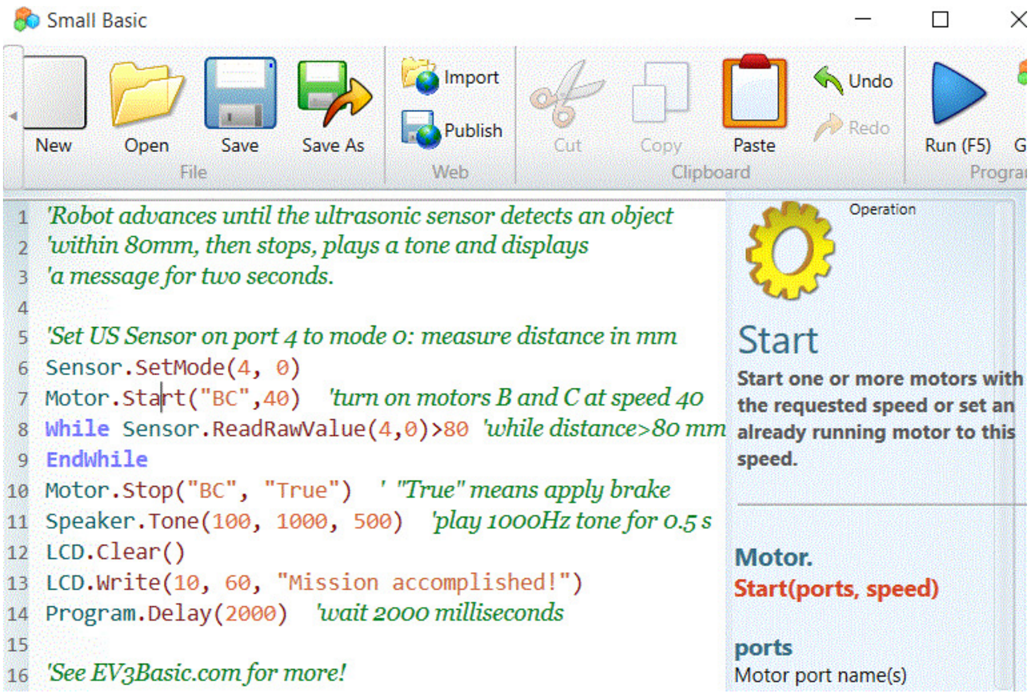


Figure 3.6: Development Environment EV3-Basic (EV3 Basic, 2020)

3.1.2.2 Use of Robots in Lessons

Robots have been used in the classroom in BG/BRG/BORG Köflach since 2018. The following list provides an overview of the use in the last two school years:

- **year 5 - 6:** So far, the Ozobot has been used in the first two school years as part of projects. The colour codes are used in regular lessons. As a part of the summer school Ozoblockly have been used. A first use in regular classes for Ozoblockly was planned for the school year 2019/20. However, due to COVID19 measures, this was not possible in a comprehensive and complete way.
- **year 7:** In the third year, the Lego-Mindstorm robots were already used in regular lessons in the school year 2018/2019. All contents planned within the curriculum (details in Chapter 5) for graphical programming were covered. In the school year 2019/2020, the research

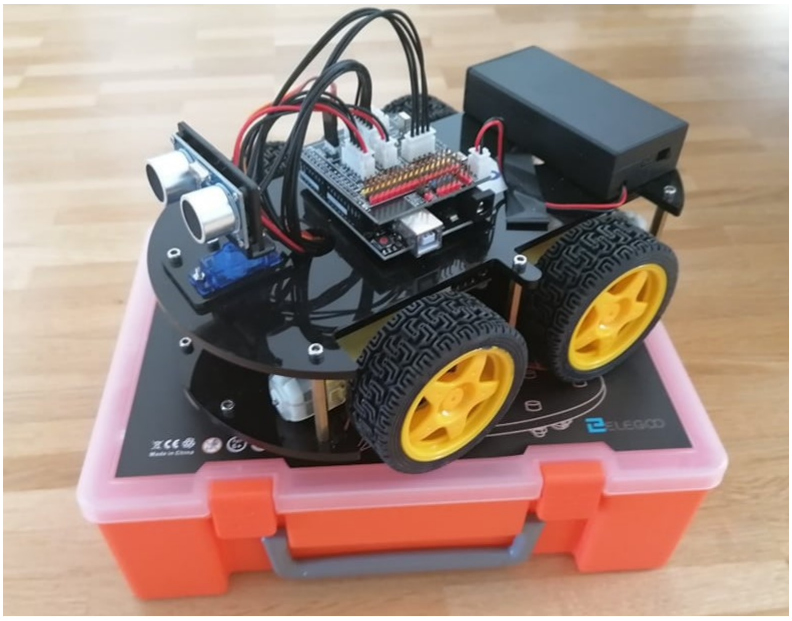


Figure 3.7: Figure of Smart v3.0 Robot Car

for the comparison of graphical vs. textual programming as well as the measurement of learning success within the framework of the skills mentioned in Section 2.1 was carried out (see Chapter 4).

- **year 8:** In the year 2020 the robotics club has been offered to students since 2 years. In this club students learn textual programming and work with the EV3 LegoMindstorm. The focus is laid on the preparation for KNAPP Robo League. For details see Section 5.2.4.
- **year 9:** In the school year 2018/19, the fifth class of BRG worked with the experimental box (see Arduino) is used for the first time. During this process, contents from the curriculum presented in this thesis were also be addressed. This was not possible in the school year 2019/20 because of COVID19 measures.
- **year 10:** In the school year 2018/19, the students worked with the robot car (Arduino Uno). The contents of the course were partly in line with those of the presented curriculum in this theses. In the past school year, the contents could not be taught because of COVID19.
- **year 11 - 12:** So far, no contents on robotics have been introduced in these classes. For the school year 2019/2020 contents (according to the

curriculum see Section 5.2.7) on Artificial Intelligence in connection with robotics had been planned. Because of COVID19 measures these had to be postponed to the following school year.

Given this, it can be concluded that there has been a lack of a consistent structure in the context of working with robots so far. The curriculum, which will be presented in this thesis, is designed to counteract this situation.

3.2 Experimental Setting

This section describes the experimental setting in the context of learning success and the comparison of graphical vs. textual programming. The purpose of the comparison was to determine which type of programming is more suitable for newcomers to robotics or where the greater learning benefit can be achieved especially in the context of the skills mentioned in section 2.1 and whether a possible difference is significant. In this context, learning success and effects in the areas of computational thinking, problem solving and social aspects have also been examined.

3.2.1 General Conditions

In order to carry out the comparison, a class of the BG/BRG/BORG Köflach was divided into 2 independent test groups(one groups using graphical programming, the other one using textual programming). This was a class of the BRG and therefore a class of the natural scientific department.

It was a 3rd class (grade 7) and thus a class that had no (or few) experience in robotics or programming at that time. The class consisted of 30 students. It was therefore divided into two equal groups of 15 students each. The groups were divided randomly according to the alphabet. Normally the groups are coached by different teachers. This could be a confounding variable. So the teachers found an exception so that the lessons in robotics could be taught by the same teacher.

3.2.2 Teaching Material

In order to make the results valid, it was ensured that the learning materials only differed in the programming languages. The material was divided into 8 units:

- Basics of the EV3
- Motion of curves
- Control structures
- Ultrasonic sensor
- Sounds and screen output
- Colour and light sensor
- Touch sensor and logic
- Gyrosensor

The content of the lessons is discussed in Section 5.2.3. In this section, unit 6 (colour and light sensor) is used as an example to show that the lessons for graphical and textual programming are only distinguished by the programming language used.

In both cases students work with the LegoMindstorm EV3. Both units start also with the same running example. The aim is to repeat the contents as described in section 5.2.3. During the review in this lesson, the use of the ultrasonic sensor and the output of sounds will be consolidated. In both cases (graphical and textual) the robot should beep louder and louder as it gets closer to the obstacle (e.g. a wall).

The theoretical input is the same. In both cases the students learn the theory of the light sensor. The contents were taken from the official manual[8].

This is followed by an explanation of how data (here the reflection) can be read in using the light sensor. Both units (graphical and textual) explain this with a short code example (see Figure 3.8). In both cases the amount of the reflection is read from the sensor and then displayed on the screen. With textual programming the reflection is read in with Sensor.ReadRawValue

[8] https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf [Online; 17.November 2020]

and write to the screen of the robot with LCD.Write. With graphical programming, the reflection is read in with the third yellow brick and the following green brick writes the data on the screen of the robot. In both programming paradigms the code is inside an endless loop.

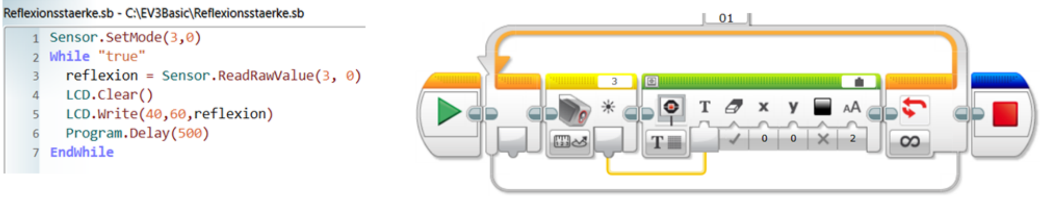


Figure 3.8: Textual (left) vs. Graphical (right) content in the context of unit 6

In the next step, the tracking of a black line with a light sensor has to be implemented.
In both programming paradigm the first step is to determine the reflectivity of the different surfaces (black vs. white). Moreover, the idea behind line tracking with one light sensor is explained (driving at the edge of the line). In both programming paradigm the solution is given step by step. First the robot should only move when it is on the black line, as shown in Figure 3.9. In the figure one can see the textual code above. The nested if ensures that the robot only moves when it is at the edge of the line (the values of the reflection must be adapted to the local conditions). In the case of graphical programming (below) the block range (red) is used. If the measured value is within the range, i.e. 15 and 55, this block returns the value "True". With the help of the block "switch" it is checked whether the returned value is true or false. If the value is true, the robot moves straight.

Figure 3.10 shows the extension of the program from Figure 3.9. The textual code (above) was extended by 2 else statements. These ensure a correction(right or left) if the robot moves away from the edge oft line. With graphical programming (below) the code is extended by the block "comparison" among some other blocks. This block determines whether the measured value is above or below the respective comparison value. If the reflection value is too high, the correction is made to the right (white area), if it is too small, the correction is made to the left (black area).

```
Sensor.SetMode(3,0)
While "True"
  reflexion = Sensor.ReadRawValue(3,0)
  If reflexion > 10 Then
    If reflexion < 55 Then
      Motor.Start("BC",60)
    EndIf
  EndIf
```

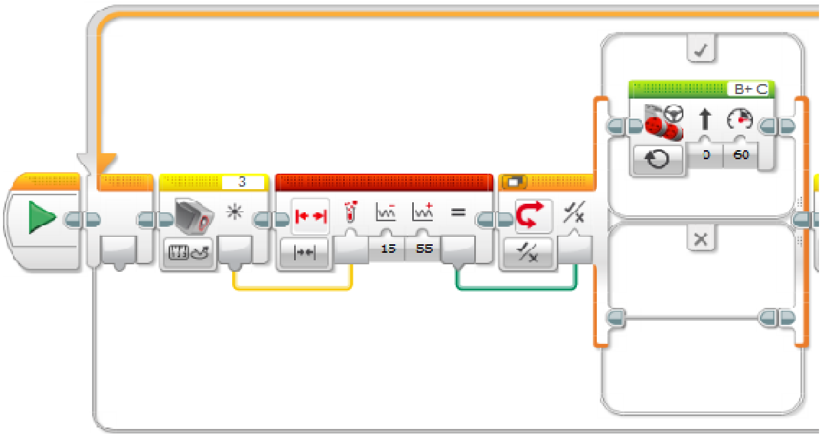


Figure 3.9: Textual (above) vs. Graphical (below) contents in the context of unit 6 follow the line part 1

```
Sensor.SetMode(3,0)
While "True"
  reflexion = Sensor.ReadRawValue(3,0)
  If reflexion > 10 Then
    If reflexion < 55 Then
      Motor.Start("BC",60)
    Else
      Motor.StartSync("BC",40,20)
    EndIf
  Else
    Motor.StartSync("BC",20,40)
  EndIf If
EndWhile
```

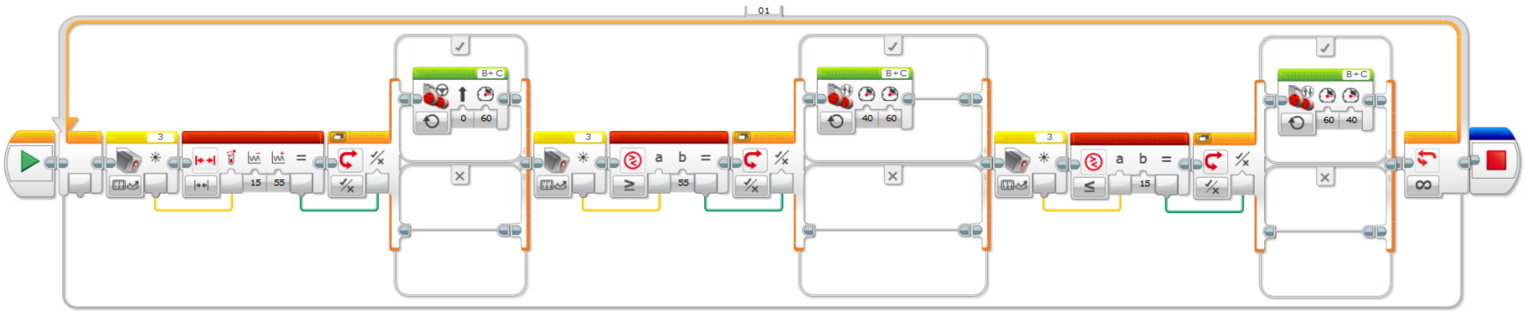


Figure 3.10: Textual (above) vs. Graphical (below) contents in the context of unit 6 follow the line part 2

In the next step the line tracking will be improved in both cases. Then, students are informed that the robot should correct using different strengths depending on the measured reflection. It is obvious that the only difference is the programming language. The idea of implementation of the code is also very similar, even if - in this example with nested ifs - it differs slightly in the implementation. Only a small advantage of graphical programming is used in this case. The advantage was that due to the "range" block it was not necessary to use nested conditions in the context of the graphical language.

After the exercise above students start to solve exercises independent. In both programming paradigm students solve the same exercises. In one example, students are to implement line tracking with two sensors, in the other they are to use reflection-based speed.

3.2.3 Evaluation Method

An online questionnaire was developed (see Appendix A and B) to measure the learning success and to measure and compare the improvement in performance. In order to develop a valid test parts of already validated tests were used. Since the new test also consisted of questions developed by the author himself, a trial-test was necessary. This step additionally ensured that a valid test was used in the actual testing process. Since the focus of the evaluation was to measure the performance increase and subsequently compare this performance increase between the test groups (graphically and textually), a pre- and post-test was carried out (Salkind, 2010). In the pre-testing, the students' state of knowledge at the beginning of the testing was assessed. In the post-testing, the state of knowledge at the end of the study was assessed. The questionnaires used were identical. To ensure anonymity and comparability of the individual results, IDs were given to the students. The test was divided into 3 parts. Each part focuses on one of the skills mentioned in section 2.1. One of the reasons for this division was to allow the students to focus on one of the skills in each part.

3.2.3.1 Robotics and Computational Thinking

The first part (see Appendix A.1) deals directly with the learned contents from the field of robotics. In addition, the focus is also put on the field of computational thinking. The type of question and its structure is based on tests already carried out in Weintrop and Wilensky (2015) and Román-González (2015).
There are two types of questions. At type 1 questions a task is given, the user has to choose the appropriate code from 4 possible answers. Figure 3.11 shows such a question from the textual test. The aim of this question is that the robot moves forward a certain distance, then stops and only moves half the distance backwards again.

For type 2 tasks the code is given. Students then have to choose the appropriate result of the code from 4 possible answers. Figure 3.12 shows an example question from the textual test. In this question students must specify the value of the variables after the code has been executed.

As with the learning materials, it was ensured that the questions only differ in the code used. This was evident in the tasks as well as in the possible answers for both question types. The possible answers were exactly the same, so possible "traps" were the same for both tests. Figure 3.13 shows the comparison of a type 1 question. The aim of that question was to make the right wheel turn twice as fast as the left wheel, where x is the speed of the left wheel. Figure 3.14 is a comparison of type 2, in both cases the students had to fill in the gap appropriately. The question was, how often the loop has to be executed at least, so that the robot reaches the speed of "100".

3.2.3.2 Problem Solving

The next part of the evaluation focused on problem solving. On the one hand, tasks with a focus on problem solving had to be solved and on the other hand, the students answered self-assessment questions in the context of problem solving. The same questions were used for textual and graphical assessment.

Question1 *

Which of the code sections shown below solves the following task? Your robot should move forward a certain distance, then stop. Afterwards it should only drive backwards for half of the distance and stop.

Option A

```
Motor.Start("BC",50)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",50)
Program.Delay(2000)
Motor.Stop("BC","True")
```

Option B

```
Motor.Start("BC",50)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(2000)
Motor.Stop("BC","True")
```

Option C

```
Motor.Start("BC",50)
Program.Delay(2000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(4000)
Motor.Stop("BC","True")
```

Option D

```
Motor.Start("BC",100)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(4000)
Motor.Stop("BC","True")
```

- Option A
- Option B
- Option C
- Option D

Figure 3.11: Example question robotics and computational thinking type 1

The questions that test the students problem-solving skills all have been taken from the ”beaver computing challenge[9]”, the questions asked (see Appendix A.2) from different competition. The 1st question dealt with the Josephus problem and algorithms (Biber der Informatik, 2013, p. 42f). In the context of this task, cases are placed on a conveyor belt which runs in a circle. The suitcases are always placed on the third free space. Students must now select the correct position/sequence of the cases when all 5 have been placed on the conveyor. The 2nd question also came from the field

[9] https://www.ocg.at/de/biber-der-informatik [Online; Last call 14.August 2020]

Question 4 *

Look at the programme in the figure below. What is the value of the variables x and y after the programme has finished?

```
x = 10
y = 5
y = x
x = x + 5
```

- x = 15; y =15
- x = 5; y = 10
- x = 15; y =10
- x = 10; y =5

Figure 3.12: Example question robotics and computational thinking type 2

of problem solving and algorithms (Biber der Informatik, 2012, p. 16). In this task a robot moves over a field with obstacles. If it hits an obstacle or the border of the field, it turns 90 degrees to the right and continues its journey. Every field which the robot passes becomes an obstacle. The pupils have to find the field where the robot does not reach its target (green). The 3rd question was a question about problem solving competence and optimization (Biber der Informatik, 2012, p. 32). A graph is given for this task. It is indicated how long the journey from one node to the other takes. Students have to find the fastest way to the goal. Question 4 dealt with partial problems (Biber der Informatik, 2011, p. 23). In this task, woods correspond to a certain monetary value, depending on their size. The bigger the wood pieces are, the heavier they are. The beaver can only carry a maximum weight. The sticks must now be combined so that the beaver earns a maximum amount of money. The last question to test the problem solving competence of students was about a decision problem (Biber der Informatik, 2009, p. 19). There are three colours (yellow, green, red) which can also be combined to other colours. Now eggs will be dive into the colours. Students must now determine the possible colouring of an egg among the existing ones.

The self-assessment questions (see Appendix A.2 on page 166) were taken from Nugent et al (2010) (cited in Kandlhofer (2017)) and translated into German. Here pupils were asked 5 self-assessment questions. Within these questions they had to indicate whether they use a step-by-step approach to solving problems, whether they plan their approach, whether they use new methods to solve the problem when the old ones don't work, whether they analyse a problem before they trying to solve it and whether they divide bigger problems into smaller ones. A likert scale was used for evaluation. This can be used to measure meninges, for example, and thus offers more answer options than simple yes/no questions (SurveyMonkey, 2020). In the specific case the following answer options were available: "Strongly Agree", "Agree", "Uncertain", "Disagree" and "Strongly Disagree".

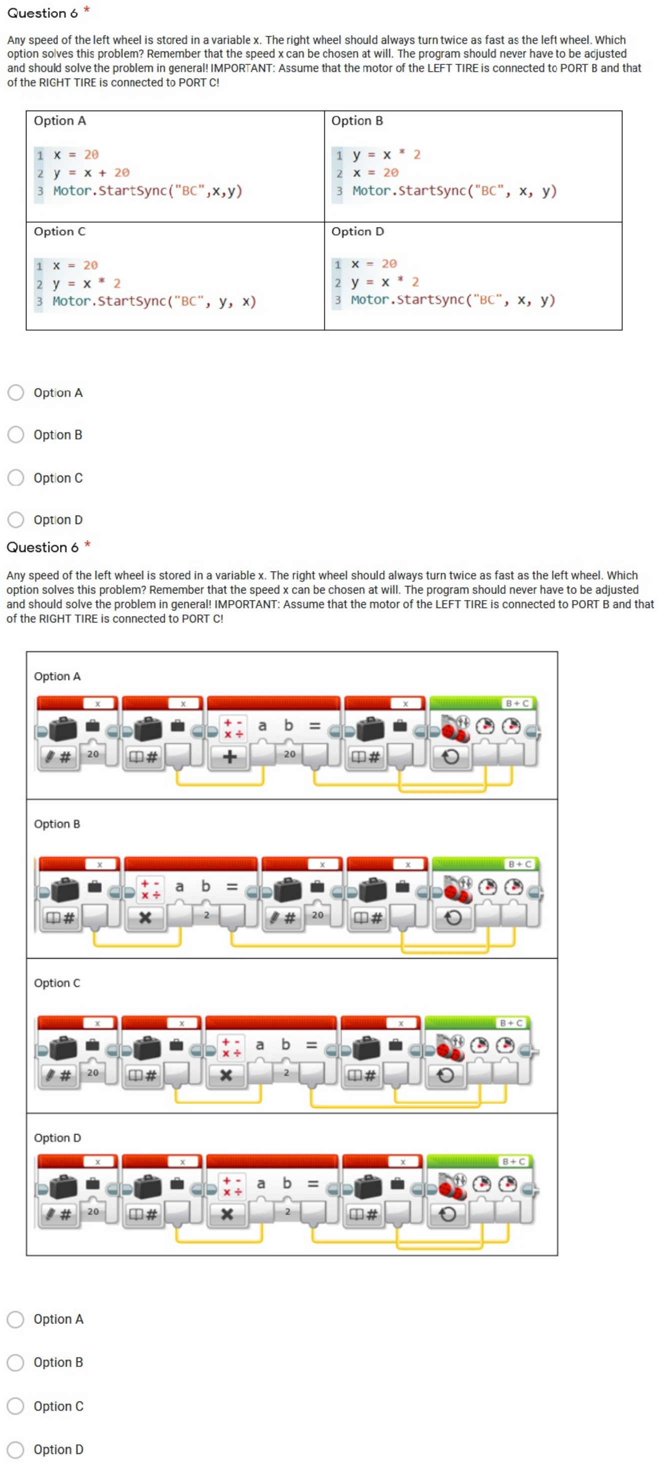
3.2.3.3 Social Aspects

The last part of the test focuses on social aspects. The same questions were used for both groups and questions were again taken from existing surveys.

Thus, the questions 1 to 4 (see Appendix A.3) were taken from Nugent et al (2010) (cited in Kandlhofer (2017)) and translated into German. These questions mainly aimed at teamwork. Students had to indicate whether they listened to others when solving problems, whether they liked working in a team, whether they asked team colleagues for help if they didn't understand something and whether they liked solving tasks together with others. A likert scale was again used for the answers. The possible answers were: "Strongly Agree", "Agree", "Uncertain", "Disagree" and "Strongly Disagree".

Questions 5 to 8 (see Appendix A.3 on page 170) were taken from Ralph B. McNeal et al (2004) (cited in Kandlhofer (2017)) and translated into German. These questions dealt with social skills. Students had to indicate whether it is easy to persuade friends of their opinion, whether it is easy to make new friends for them, whether it is easy to ask friends for a favour and whether it is easy to get along with others. A likert scale was again used for the answers. The possible answers were: ”Strongly Agree or Very easy”, ”Agree or easy”, ”Uncertain”, ”Disagree or hard” and ”Strongly Disagree or Very hard”.

In the next chapter we will present the findings of the comparison between textual vs. graphical programming for novices. In addition, the learning outcomes in the areas of computational thinking and problem solving, as well as the impact on social aspects will be shown and analysed.



Question 6 *

Any speed of the left wheel is stored in a variable x. The right wheel should always turn twice as fast as the left wheel. Which option solves this problem? Remember that the speed x can be chosen at will. The program should never have to be adjusted and should solve the problem in general! IMPORTANT: Assume that the motor of the LEFT TIRE is connected to PORT B and that of the RIGHT TIRE is connected to PORT C!

Option A

```
x = 20
y = x + 20
Motor.StartSync("BC",x,y)
```

Option B

```
y = x * 2
x = 20
Motor.StartSync("BC", x, y)
```

Option C

```
x = 20
y = x * 2
Motor.StartSync("BC", y, x)
```

Option D

```
x = 20
y = x * 2
Motor.StartSync("BC", x, y)
```

- Option A
- Option B
- Option C
- Option D

Question 6 *

Any speed of the left wheel is stored in a variable x. The right wheel should always turn twice as fast as the left wheel. Which option solves this problem? Remember that the speed x can be chosen at will. The program should never have to be adjusted and should solve the problem in general! IMPORTANT: Assume that the motor of the LEFT TIRE is connected to PORT B and that of the RIGHT TIRE is connected to PORT C!

Option A

Option B

Option C

Option D

- Option A
- Option B
- Option C
- Option D

Figure 3.13: Comparison type 1 test questions textual vs. graphical

Question 7 *

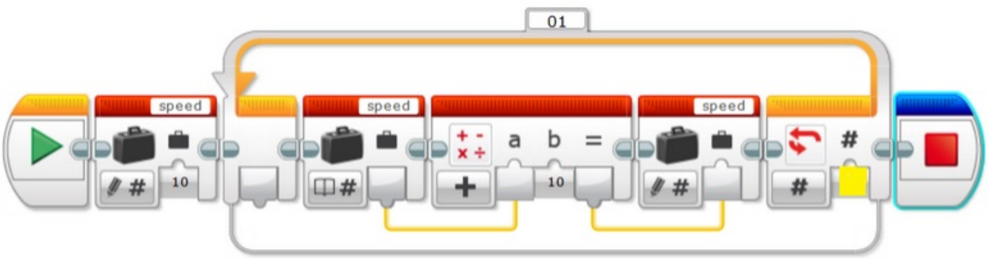
In the following programme the variable speed is always increased by 10 within the loop. How many times must the loop be executed at least, so that the variable speed is increased by 100?

```
speed = 10
For counter = 1 To ???
  speed = speed + 10
EndFor
```

- ◯ 10 times
- ◯ 11 times
- ◯ 9 times
- ◯ No answer is correct

Question 7 *

In the following programme the variable speed is always increased by 10 within the loop. How many times must the loop be executed at least, so that the variable speed is increased by 100?



- ◯ 10 times
- ◯ 11 times
- ◯ 9 times
- ◯ No answer is correct

Figure 3.14: Comparison type 2 test questions textual vs. graphical

4 Evaluating the Learning Success of Textual vs. Graphical Programming

In this chapter the knowledge acquisition using graphical and textual programming is compared. A special focus is laid on knowledge acquisition in the context of problem solving and computational thinking. Possible changes in the social field are also considered. At first only an overview of the collected data is given. A more detailed analysis of the individual performance of the candidates is going to follow in the next section.

4.1 Data

Both groups consisted of 15 students. In the group using graphical programming there were 5 female students and 10 male students. In the group using textual programming there were 4 female and 11 male students. No further information about the students were gathered. First, the data for robotics and computational thinking are considered.

4.1.1 Robotics and Computational Thinking

First, the percentage of correct answers per question is compared. We start with the group using graphical programming. Figure 4.1 shows a comparison between the scores obtained in pre-test (blue) and post-test (orange). The figure shows the percentage of correct answers for each of the 17 questions in the robotics and computational thinking part of the test.

Figure 4.2 depicts the average percentage of correct answers per question for the group using textual programming.

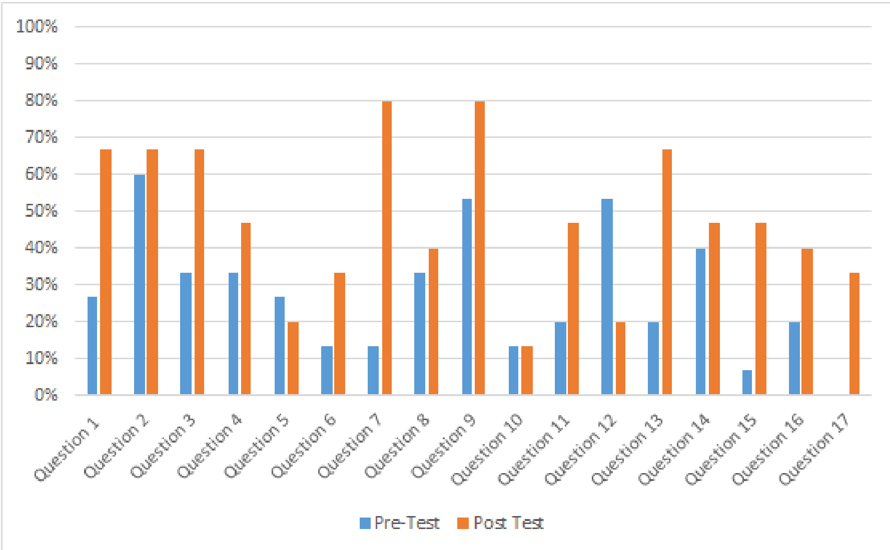


Figure 4.1: Average correct answers per question at pre-test and post-test of the group using graphical programming (pre-test-std = 2.58; post-test-std = 3.09)

In the following, the performance per test candidate in pre-test and post-test is compared. Figure 4.3 shows the evaluation of the group using graphical programming. The blue bar represents the percentage of correct answers in the pre-test for the test candidates, while the orange bar represents the number of correct answers in the post-test. Figure 4.4 shows the same of the group using textual programming.

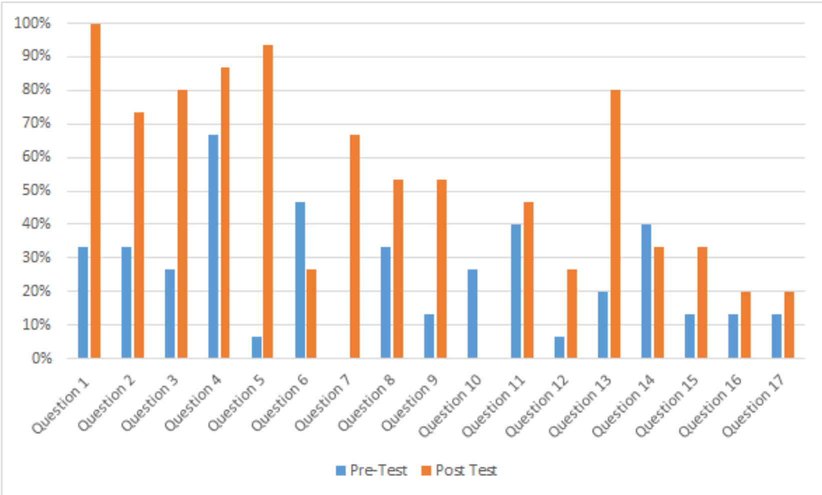


Figure 4.2: Average correct answers per question at pre-test and post-test of the group using textual programming (pre-test-std = 2.58; post-test-std = 4.47)

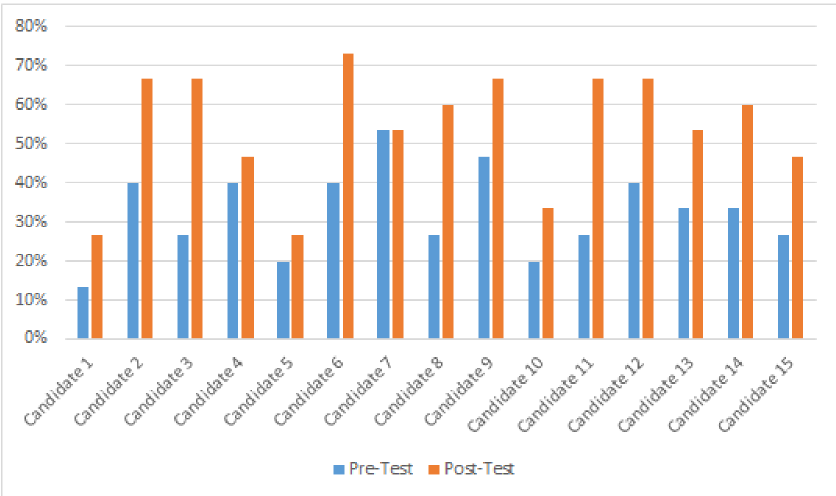


Figure 4.3: Average correct answers per candidate using graphical programming at pre-test and post-test (pre-test-std = 1.71; post-test-std = 2.29

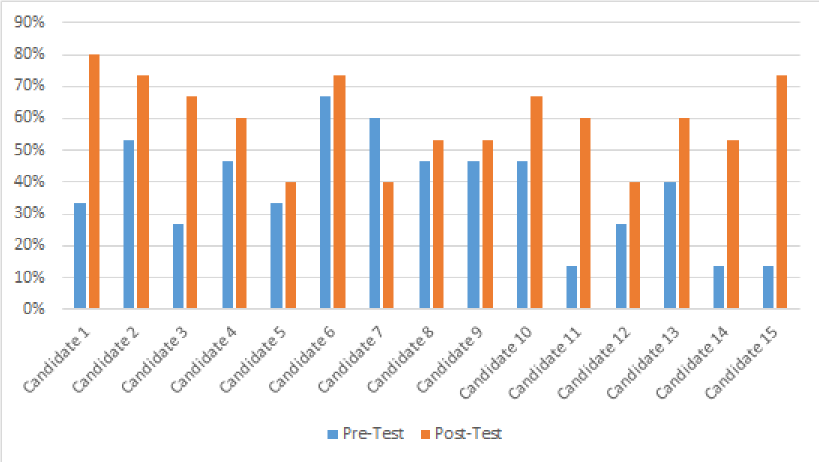


Figure 4.4: Average correct answers per candidate using textual programming at pre-test and post-test (pre-test-std = 2.53 ; post-test-std = 1.94

4.1.2 Problem Solving

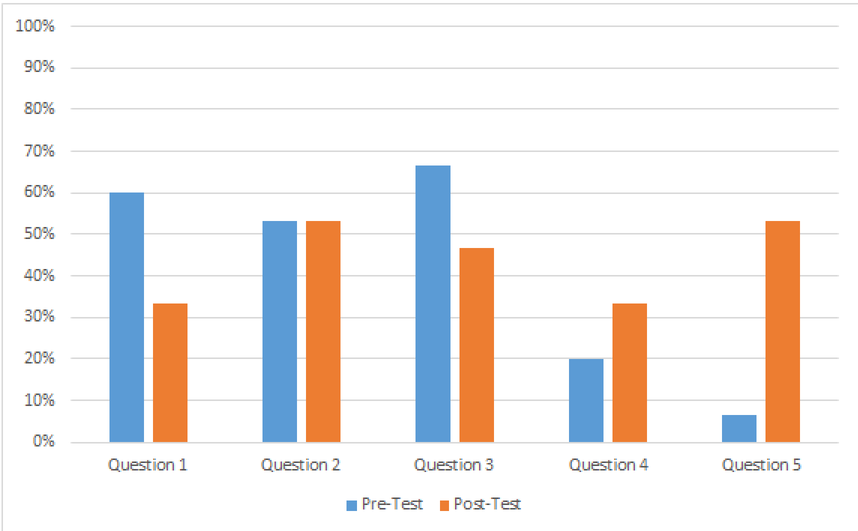


Figure 4.5: Average of correct answers per question of knowledge in the part for problem solving for the group using graphical programming (pre-test-std = 3.65 ; post-test-std = 1.52)

First of all the data for the questions of knowledge are going to be provided. Figure 4.5 now shows a comparison per question between pre-test and post-test of the group using graphical programming. Figure 4.6 shows the same of the group using textual programming.

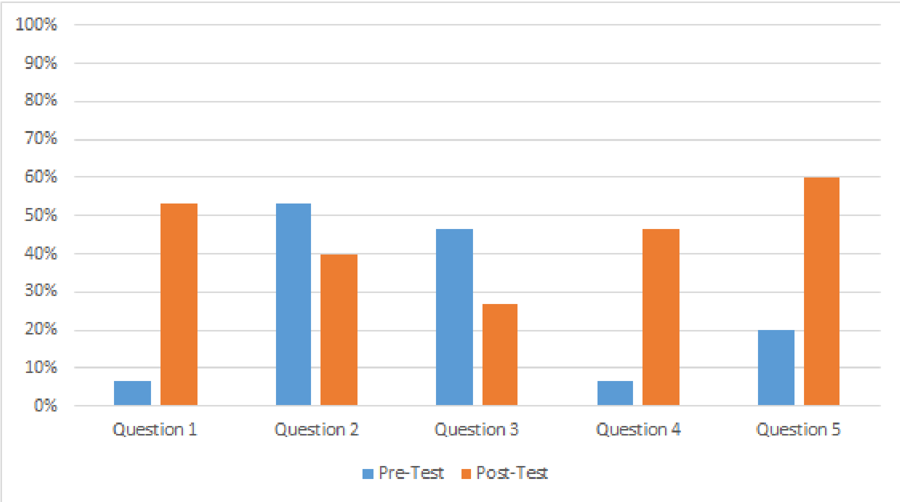


Figure 4.6: Average of correct answers per question of knowledge in the part for problem solving for the group using textual programming (pre-test-std = 3.32; post-test-std = 1.92)

Figure 4.7 shows an individual evaluation per candidate within the questions of knowledge for the group using graphical programming. Figure 4.8 shows this for the group using textual programming.

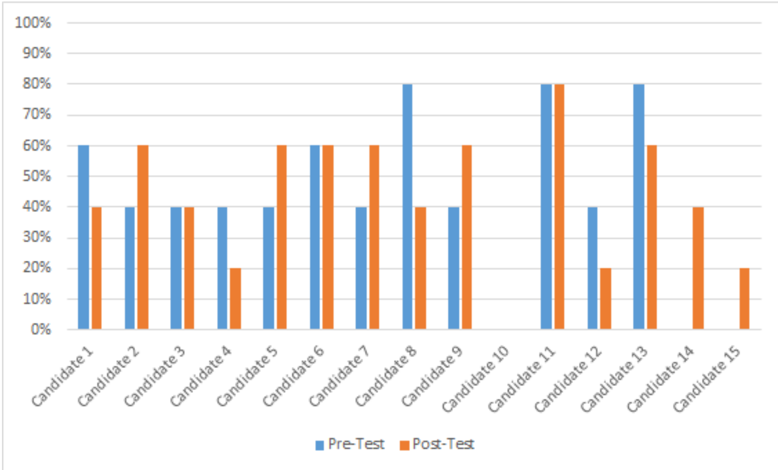


Figure 4.7: Average of correct answers per candidate for the part of problem solving of the group using graphical programming (pre-test-std = 1.36; post-test-std = 1.08)

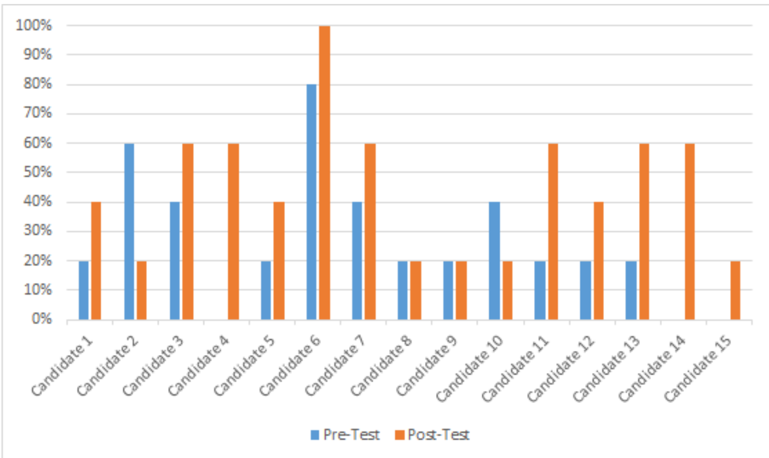


Figure 4.8: Average of correct answers per candidate in the part of problem of the group using textual programming (pre-test-std = 1.11; post-test-std = 1.16)

In the following the results of the self-assessment questions are shown. Figure 4.9 shows the results for the pre- and post-test of the group using graphical programming. The figure shows for each question the summarized answers of the candidates in percent in the context of the likert scale for the self-assessment questions in the framework of problem solving. Figure 4.10 does this for the group using textual programming.

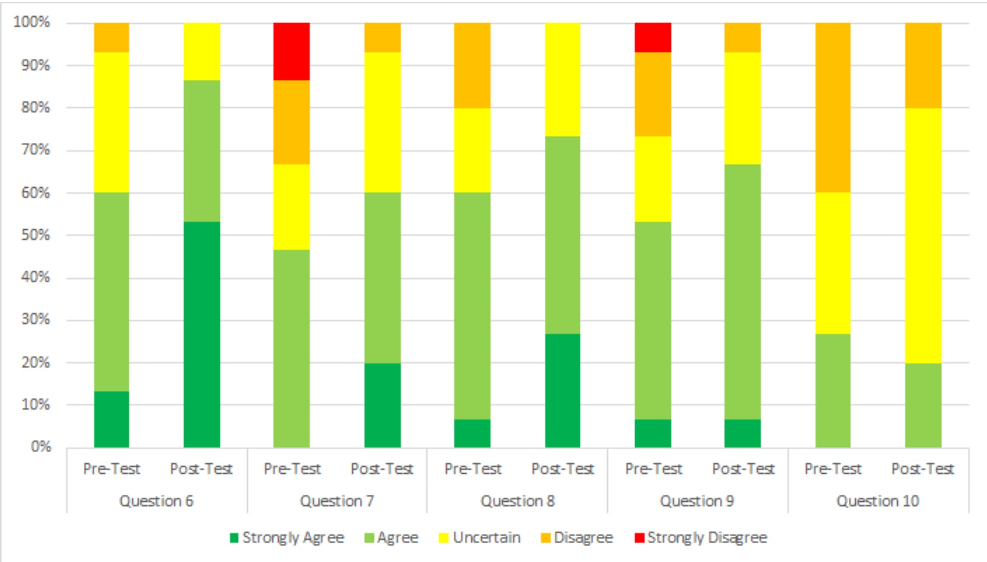


Figure 4.9: Results of self-assessment part per question for problem solving of the group using graphical programming (pre-test-std = 4.92; post-test-std = 6.83)

Figure 4.11 shows a comparison per candidate for pre-testing and post-testing of the group using graphical programming in the context of the self-assessment questions for problem solving. To make it easier to compare the difference between pre-test and post-test between candidates, the answers of the individual questions are summarised for each candidate. Figure 4.12 does this for the group using textual programming.

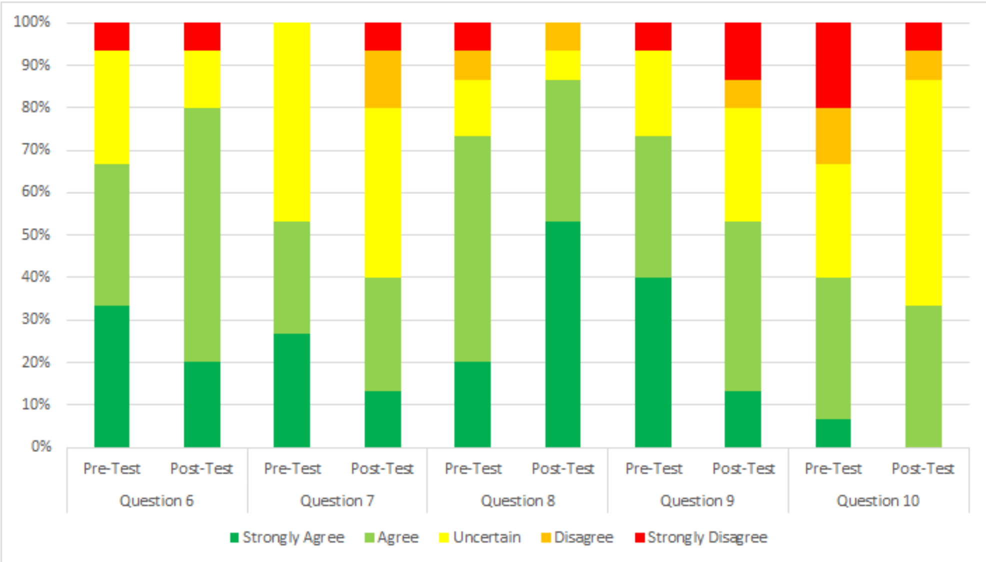


Figure 4.10: Results of self-assessment part per question for problem solving of the group using textual programming (pre-test-std = 6.42; post-test-std = 7.53

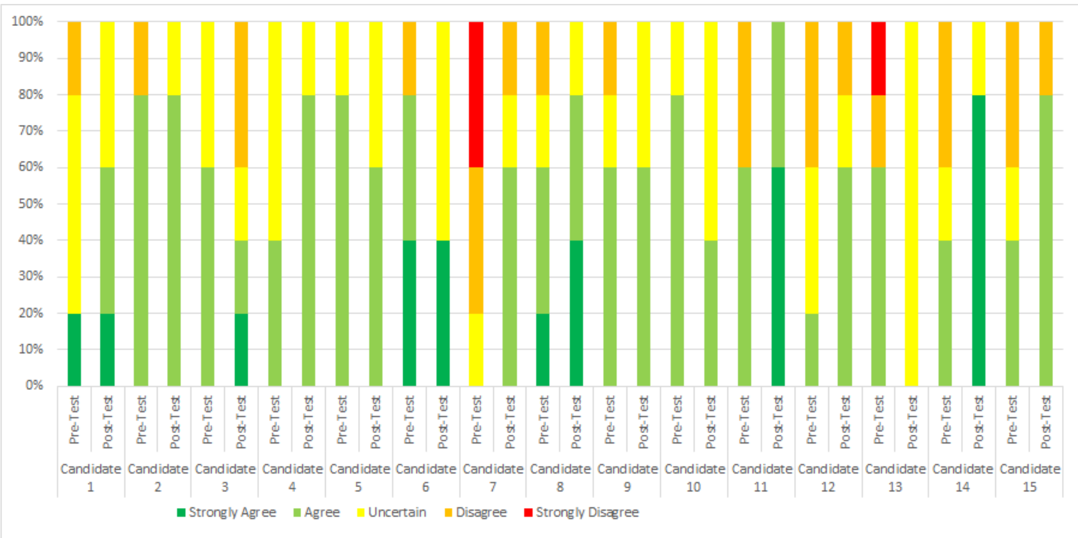


Figure 4.11: Comparison self-assessment question per candidate for pre-test and post-test of the group using graphical programming (pre-test-std = 2.60; post-test-std = 2.29)

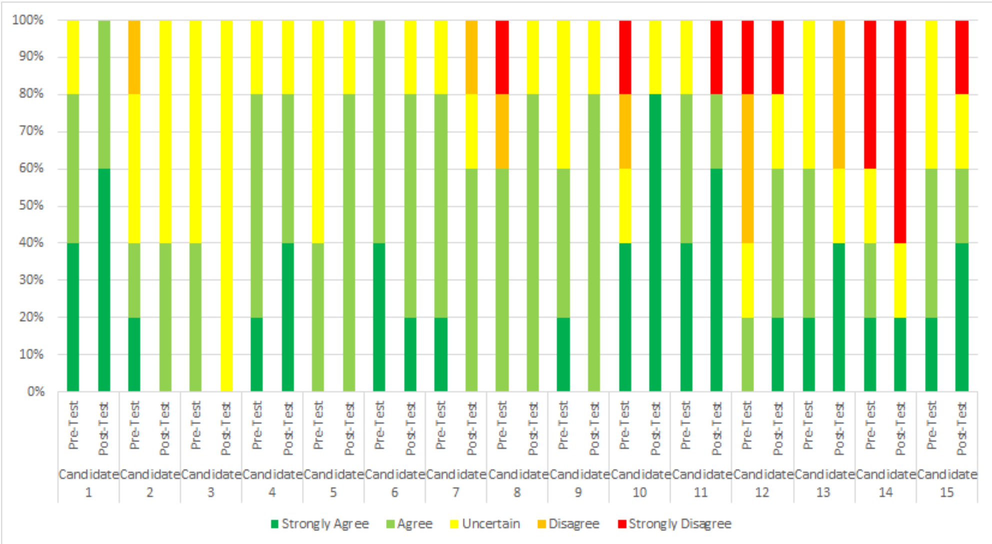


Figure 4.12: Comparison self-assessment question per candidate for pre-test and post-test of the group using textual programming (pre-test-std = 3.04; post-test-std = 2.84)

4.1.3 Social Aspects

In this section results in the relation to social aspects are presented. As already mentioned, these aspects are related to teamwork and social skills. To make it easier to compare the difference between pre-test and post-test between candidates, the answers of the individual questions are summarised for each candidate. Figure 4.13 now shows the summarized results per question for the group using graphical programming. Figure 4.14 shows the same for the group using textual programming.

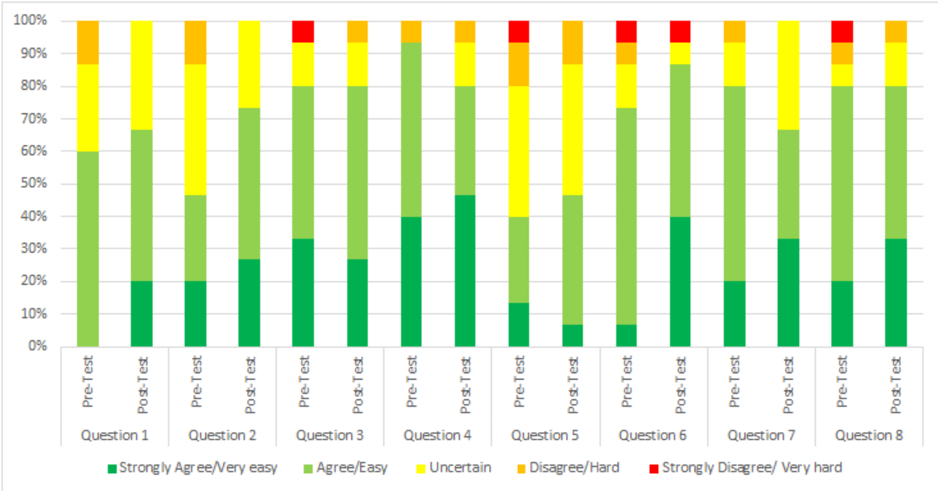


Figure 4.13: Comparison per question of social aspects of the group using graphical programming (pre-test-std = 5.01; post-test-std = 3.74)

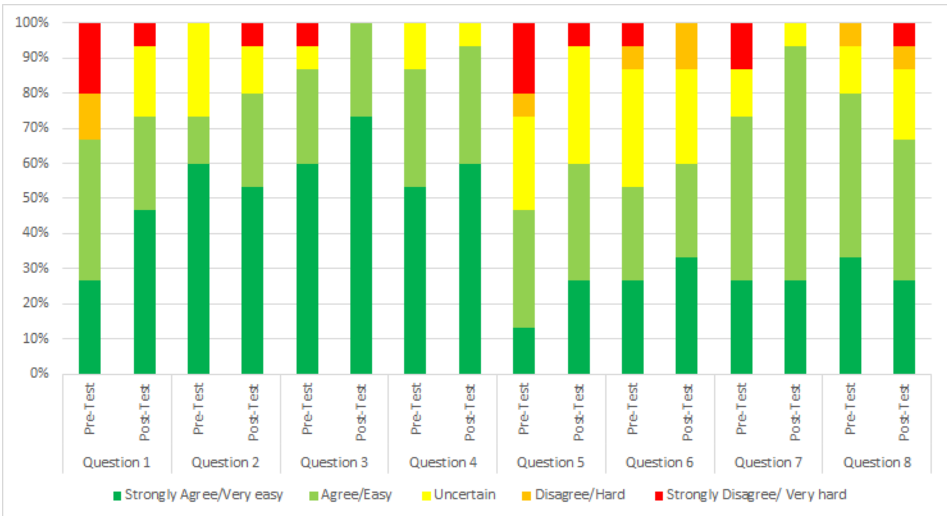


Figure 4.14: Comparison per question of social aspects of the group using textual programming (pre-test-std = 6.93; post-test-std = 4.93)

Figure 4.15 shows the comparison per candidate for pre-testing and post-testing of the group using graphical programming. Figure 4.16 shows the comparison of the group using textual programming.

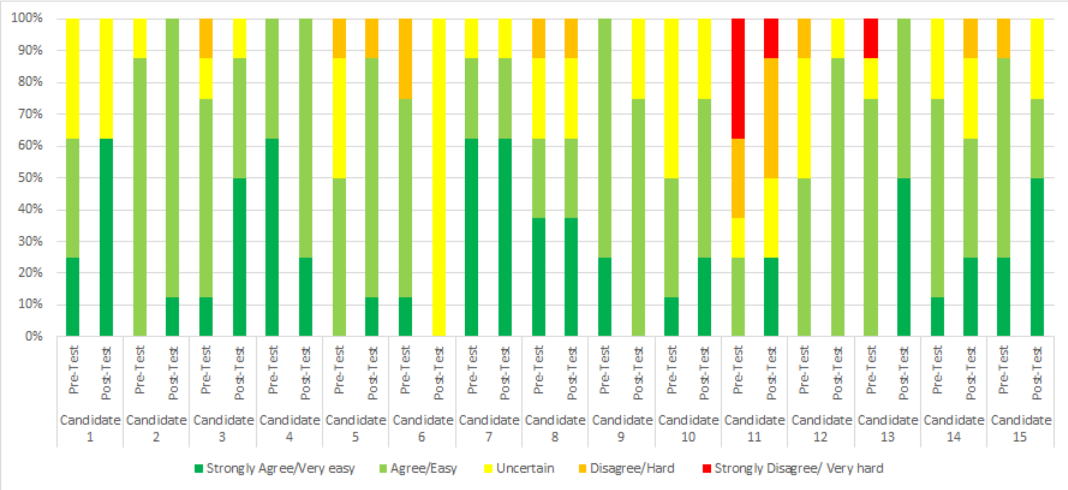


Figure 4.15: Comparison per candidate of pre-test and post-test of the group using graphical programming (pre-test-std = 4.43; post-test-std = 3.94)

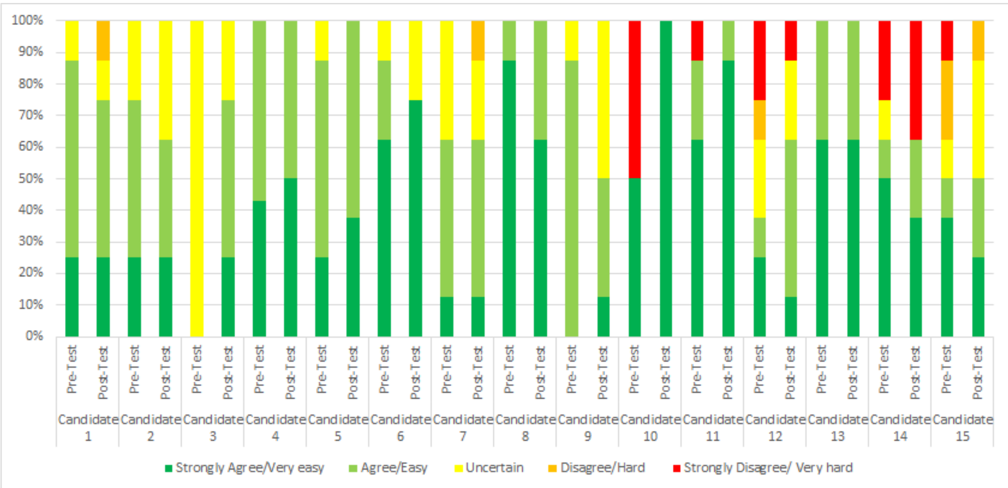


Figure 4.16: Comparison per candidate for pre-test and post-test of the textual test group (pre-test-std = 4.72; post-test-std = 4.39)

4.2 Discussion

In this section the results presented in the previous section will be discussed and analysed. In both cases the study lasted about 2 months. The long duration of the study will play a role, especially in the social aspects. This will be discussed in more detail later. In the first part we analyse the testing results in relation with computational thinking and programming in the context of robotics.

4.2.1 Computational Thinking and Robotics

The two groups using textual respectively graphical programming are analysed individually. A comparison between textual and graphical programming is given in Section 4.2.4.

Figure 4.1 shows that the number of correct answers in the comparison between pre-testing and post-testing has increased in the case of 14 questions out of the asked 17 in the test. This can also be seen in absolute numbers. So, 122 correct answers were given in the post-test compared to 71 in the pre-test. Figure 4.3 also shows that all candidates have improved or at least achieved the same result as in the pre-test. Only Candidate 7 reached the same result as in the pre-test. Unfortunately it is not possible to give a reason for the stagnation, as the experiment was anonymous. It can be assumed that the student was among those who missed units (e.g. due to illness). Due to the temporary increase an improvement in performance is evident in this area. A paired t-test is used to measure the statistical variance of the increase. SPSS[1] is used as a tool for evaluation. The analysed data is normally distributed(pre- and post-test), although this is not a necessary condition for the use of the test, because the paired t-test is robust against violations of the normal distribution (Pagano, 2012). The increase in performance is confirmed by a *paired sample t-test*. The result is a value of $T = -6.730$ and a significance of $p < 0.001$. The performance improvement of the group using graphical programming is therefore highly significant.

[1] https://www.ibm.com/analytics/at/de/technology/spss/ [Online; Last call 20.November 2020]

For the textual test group, Figure 4.2 shows an increase in correct answers compared to pre-test and post-test. An improvement was achieved in 14 of 17 questions. Here was an overlap with the group that used a graphical programming language. Both groups could not improve on question 10. This question should therefore be reviewed in more detail. One possibility is that the used code is too unstructured. In total numbers the increase could also be seen, as 85 questions were answered correctly in the pre-test and 134 questions in the post-test. Figure 4.4 shows the extent to which the individual candidates have improved. With one exception, all candidates have improved their performance. The assumption, why one candidate could not improve his performance is, that the student was absent from too many units. The group using textual programming also shows therefore a clear improvement in performance. This is again confirmed by the *paired sample t-test*. The result is a value of $t = -4.016$ and a value of $p = 0.001$. The increase is therefore highly significant, too. The pre-test and post-test data are normally distributed again.

In both test groups an increase in knowledge can be seen. Since all questions were asked in the field of computational thinking, the positive impact of robotics on this skill can be observed and the results of Constantinou and Ioannou (2018) which found an improvement of this skill in the course of robotics lessons could be supported. Whether this positive effect was caused by ER or programming cannot be said with certainty. Further studies would have to be conducted. A significant increase in performance has taken place in both test groups, regardless of the programming language used.

4.2.2 Problem Solving

The analysis of the results in the area of problem solving is presented next. Starting with the questions of knowledge and the group using graphical programming. Figure 4.5 shows that during the pre-test individual questions were answered correctly to a very different extent. In the context of the post-test, the difference between the correct answers of the questions was not so large, which is also evident from the decrease in standard deviation (see Figure 4.5). Why the difference it the post-test was much bigger can unfortunately not be said. All questions were taken from the beaver of computer science and had the same level of difficulty. Figure 4.7 shows an individual evaluation per candidate. It shows that the performance of 10 candidates has remained the same or improved, whereas 5 candidates showed an decreasing performance. The reason for the decrease has to be investigated further. To be able to make a more precise statement, several more questions from the area of problem solving must be asked. One assumption, why candidates answerd less quesitons correct, is that the focus of the participants was more on part 1 than on the pre-test. Since they were already familiar with the concepts of programming and guessing (as is often the case with the pre-test) was no longer necessary for part 1. Thus, a lot of energy was already used up at this point compared to the pre-test. The overall performance remained slightly the same. In the pre-test a total of 32 questions could be answered correctly compared to 33 questions in the post-test. Both pre-test and post-test data are not normally distributed. However, the paired t-test is robust to the non-normal distribution(Pagano, 2012). The increase is only minimal and therefore not significant with *paired sample t-test*: $t = -0.235$ and $p = 0,818$.

Figure 4.6 shows the results of the group using textual programming and a similar picture as in the which used graphical programming. In the context of the post-test, the difference between the correct answers of the questions was not so large, which is also evident from the decrease in standard deviation (see Figure 4.6). In the context of the individual evaluation per candidate, see Figure 4.8, in contrast to the graphical group a far greater individual increase in performance can be observed. Thus, 13 candidates were able to achieve the same or better performance, only 2 achieved a lower performance. The absolute numbers also confirmed this. Thus, 20 correct answers were given in the pre-test, 34 in the post-test. The increase in performance is also confirmed by the textitpaired sample t-test. There is a significant increase in performance with $t = -2.709$ and $p = 0.017$. In the current view, the data of the post-test are normally distributed, those of the pre-test are not normally distributed.

Now the self-assessment questions will be evaluated. Figure 4.9 shows a comparison per question for the group using graphical programming. In general, for each question, it can be observed that students are now increasingly using problem solving concepts. From Figure 4.11, which shows an evaluation per candidate for pre-testing and post-testing, one can again seen that most students are now using concepts from problem solving to a bigger extend. The evaluation by *paired sample t-test* confirms the increase. With $t = -2.844$ and $p = 0.013$ it delivers a significant result. The data are normally distributed during pre-testing and post-testing.

Figure 4.10 shows the an evaluation per question of the self-assessment questions for the group using textual programming. From the figure it can be seen that there were also questions (questions 7 and 9) where students, in contrast to the pre-test, indicated that they now avoid using appropriate methods in problem-solving. Overall, however, the decrease was minimal. From the available data, it is not possible to say why this is in particular in contrast to the significantly better performance (compared to the pre-test) in context of the questions of knowledge for this group. Figure 4.12, shows that candidates who initially indicated to use suitable concepts to a small extent now tend to use them much more (e.g. candidate 10 or 12). Thus, in general, the result remained very similar. This can also be determined by means of the *paired sample t-test*. It does not provide a significant difference with $t = 0.598$ and $p = 0.559$. The data are normally distributed in the context of the pre-test and in the context of the post-test.

In the context of the questions of knowledge an increase can be seen in both groups. However, only the group using textual programming shows a significant increase. As far as the application of concepts for problem solving is concerned, the group which using graphical programming used them to a significantly higher extent after the course units. The results of the group using textual programming remained more or less the same. The reason for this result in contrast to the question of knowledge can unfortunately not be answered at this point. Anyway, a positive influence of robotics on the field of problem solving could be observed, even if these concepts were rather secondary in the context of the research, since the focus was laid on the teaching of basics. Therefore the results of Hussain et al (2006) and Nourbakhsh et al (2005) which show a positive influence of robotics on problem solving competences can be confirmed in the context of the research. At this point it must also be stated again that there should be more detailed research, whether the success is really based on ER and not on programming.

4.2.3 Social Aspects

Finally, the results regrading social aspects will be analyzed. As already mentioned, these aspects cover the areas of teamwork and social skills.

Again, the group using graphical programming is analysed first. Figure 4.13 shows an evaluation per question and compares pre-testing and post-testing results. It can be seen that the overall picture has changed slightly to the positive. For example, students now increasingly indicated that they were more willing to work with other people. The individual evaluation, see Figure 4.15, also shows an improvement for many candidates. The *paired sample t-test* yields $t = -1.968$ and $p = 0.069$, the increase is therefore slightly insignificant. All data are normally distributed.

The group using textual programming shows a quite similar result. Again, Figure 4.14 shows an improvement in social aspects. Also in the individual comparison, an improvement can be seen for many candidates (see Figure 4.16). The *paired sample t-test* does not show a significant increase with $t = -1.472$ and $p = 0.163$. All data are normally distributed again.

Summarizing, a positive result can be found as well as in Eguchi (2014a) and Owens et al (2008), even if it is not significant. For this thesis it must be critically noted, that the study was extended over 2 months. Thus, other factors that have posively influenced the class community, such as a excursion, cannot be excluded.

4.2.4 Comparison of Graphical vs. Textual Programming

The main aspect of the study is to determine which type of programming is more suitable for beginners in the context of robotics. Thus a special focus was put on programming and thus part 1 of the testing. This part is going to be analysed first.

To perform the comparison between the treatment groups (graphical vs. textual) the *Repeated Measures Anova* is used. The *Box's Test for Equivalence of Covariance Matrices*, which is necessary key factor to use the Repeated Measures Anova, returns $p = 0.146$. This is bigger than 0.05, the condition (variance homogeneity) is thus fulfilled and all data are beside this normal distributed. Therefore the results of the *Repeated Measures Manova* could be used. First, the method confirms (across the groups) the results of the t-test that there is a significant increase in performance. The effect strength could additionally be calculated, since eta-squared is given with $\eta^2 = 0.634$ (Cohen, 1988, p. 284 - 287). The effect strength can facilitate a comparison in further research (e.g. further studies). Also we can make a more precise classification of the effect (weak, medium, strong). It follows that the effect size is $f = \sqrt{\frac{\eta^2}{1-\eta^2}} \approx 1.31$, which is bigger than 0.4 and therefore results in a strong effect size (Cohen, 1988, p. 284 - 287). The comparison between graphical and textual groups shows a different increase in absolute numbers. In the graphical group, 51 more questions were answered correctly in the post-test compared to the pre-test, in the textual test group 49 more correct answers were given. This means an increase of about 72% of the graphical test group and about 58% of the textual group. In comparison, the increase of the graphical group is higher. The test resulted in a $p = 0.890$ when comparing the test groups, as well as no effect strength with $\eta^2 = 0.001$ and $f \approx 0.032$ (Cohen, 1988, p. 284 - 287). Therefore, there is no significance in the context of performance depending on to which treatment group a student belongs to.

Consequently, as in the research of Weintrop and Wilensky (2017), it could be shown in the context of ER that both groups have significantly improved. Although the performance in absolute numbers was better in the graphical group. In contrast to the study carried out by Weintrop and Wilensky (2017), no significant increase in performance could be proven in the graphical group compared to the textual group in the context of robotics. In the context of this research, it can be determined that both types of programming are equally suitable for novice students in the context of ER. As a consequence, it also shows that both types of programming are well suited for the development of computational thinking skills for novices.

In the context of problem solving part, it is now analyzed whether one group has performed significantly better than the other one. The *Repeated Measures Anova* is used again. First the questions of knowledge are considered. The variance homogeneity is full field, considered data are not normally distributed for pre-testing and post-testing. However, the comparison can still be performed because *Repeated Measures Anova* is robust to violations of the normal distribution (Blanca et al, 2017). The significant effect of robotics teaching on problem solving is confirmed with a significance of $p = 0.033$, as well as an eta-squared of $\eta^2 = 0.152$ and thus a strong effect size of $f \approx 0.423$ (Cohen, 1988, p. 284 - 287). At this point it is again pointed out that the actual success of ER must be shown again in comparison with the programming. In absolute numbers, the group using graphical programming has increased in sum by one correct answer, the textual group by 14 correct answers. In percentage, this means an increase of about 3% of the graphical group and 70% of the textual group. In contrast to the first part of the test, the textual group has increased more noticeably in this case. However, the *Repeated Measures Manova* with $p = 0.062$ does not provide a significant difference here either. In the context of (variance homogeneity full field and all data are normal distributed) the self-assessment questions, a significant difference with $p = 0.021$, as well as $\eta^2 = 0.176$ and thus a strong effect strength of $f \approx 0.462$ can be seen (Cohen, 1988, p. 284 - 287). With $p = 0.114$ no significant difference could be found across group boundaries. Furthermore, the result is a middle effect strength with $\eta^2 = 0.087$ and $f \approx 0.309$ (Cohen, 1988, p. 284 - 287).

Since the difference in the context of questions of knowledge was barely insignificant, this should be investigated again in future studies, which will focus on whether the chosen programming language has an influence on the problem solving skills of the participants. However, the difference in the self-assessment questions was significant. Unfortunately, it cannot be explained with the available data why the group, which has actually improved to a great extent, rates itself approximately equal in the post-test. Therefore, the self-assessment questions should also be part of the study mentioned above. It would be important to have a larger sample in future studies (candidates and questions).

It is also going to be examined whether the changes in the area of social aspects differ significantly between the groups. All key factors (normal distribution and variance homogeneity) again allow the use of the results of the *Repeated Measures Manova*. In contrast to the comparison within the group, this method now shows a significant change in social aspects across the groups with $p = 0.027$ and $\eta^2 = 0.163$ and thus an strong effect strength of $f = 0.441$ (Cohen, 1988, p. 284 - 287). In comparison, however, there is no significant difference with $p = 0.932$ and $\eta^2 = 0$ (Cohen, 1988, p. 284 - 287).

Within the groups, no significant change could be observed in the area of social aspects. However, this was the case,as shown above, beyond group boundaries. It should also be noted at this point that the research was conducted over a long period of time (around 2 month) and therefore other influencing (e.g. excursions that support the teamwork) factors cannot be excluded. Summarized it can be said that the chosen programming language has no influence on the social aspects of the test group.

5 Robotics - Curriculum

In this chapter the developed curriculum for teaching robotics in schools is presented. As already mentioned, this curriculum was developed for the use in a full school career from K5 to K12. This corresponds to the 8 years that students in Austria attend a secondary academic secondary school. The curriculum is designed in such a way that it can be adapted for other countries and school types.

The curriculum was developed over the last 2 years. The curriculum was developed step by step according to the school year to ensure a common thread in the structure. The idea behind the development was to create a unified concept for teaching robotics. Schools often have a lack of uniform approaches. This makes the coordination within a school location very difficult. Official curricula, which do not even exist for robotics in Austria, for example, are often structured in such a way that the teacher is given a large degree of freedom as what content should be taught and, in particular, how this content should be taught. The curriculum for robotics presented in this thesis allows freedom for implementation, but also provides more concrete guidelines/ideas on how the contents of the individual years should be taught. In general, the goal was to develop a curriculum that can serve as a template for schools around the world and to implement robotics education in more schools.

First an overview of the all years content is given. Next each year is described in detail. The contents of the curriculum are selected and justified in according to developmental psychology and and previous findings from related research.

5.1 Overview

Table 5.1 on the following page shows an overview of the curriculum. For each year, the school grade, a rough overview over the contents and the learning objectives is given.

5.2 Detailed Content According to School-Year

A detailed presentation of the content in the curriculum per school year follows. For better orientation an exemplary unit per school year is also presented. In addition, it will be explained why the content is the appropriate one for the respective school year. A detailed view of the used materials could be found online[1]. Materials for year 12 are not part of the cloud, because EDLRIS[2] follows a train the trainer approach. Trainer who obtained and finished a training for an EDLRIS course get access to the training material. The training materials are not publicly available to maintain a certain quality in teaching. We persent example teaching material form EDLRIS in this thesis that is already published. Trainers who want to use the material are recommended to complete the related trainer courses.

5.2.1 Year 5 - 1st Year of Robotics Lessons

The table 5.2 shows a brief overview of this year.

In year 5 students are between 10 and 11 years old. According to Ozobot (2020b), the devices can be used in the entire K-12 education. The guide of (Hunsaker, 2018) becomes more specific when he recommends the use of ColorCodes within the Ozobot for an age range of 7 - 12 years. According to Geier and Ebner (2017), the Ozobot is, based on their research, the best tool to build up basic computer knowledge and problem oriented thinking. Furthermore, it was determined that working with the Ozobot motivated

[1] https://drive.google.com/drive/folders/1dCMuBIfe6RVqlABDIv53TSDmpeil_3ov?usp=sharing [Online; Last call 09.September 2020]

[2] https://www.interreg-athu.eu/edlris/ [Online; Last call 22.November 2020

grade	contents	learning objectives
year 5	-) Using color codes to solve tasks with increasing difficulty.	-) Students combine color codes to solve simple tasks.
year 6	-) Tasks are solved with OzoBlockly.	-) Students know that there is an alternative programming language in the form of blocks and can solve tasks with OzoBlockly
year 7	-) Students program the LegoMindstorm EV3 with the help of a EV3-G. -) Students learn basic programming concepts such as variables and control structures.	-) Students can solve practical tasks of varying degrees of difficulty using a graphical programming language. -) Students know the most important concepts of programming.
year 8	-) Students learn the same concepts as in the previous year. This time using a textual programming language.	-) Students now using familiar methods of solution in the context of a textual programming language.
year 9	-) Student program the microcontroller Arduino UNO with the help of ArduinoC.	-) Students have knowledge about microcontrollers and can program the Arduino UNO with the help of ArduinoC.
year 10	-) Working with Smart v3.0 Robotic Car in combination with Arduino UNO.	-) Students can solve tasks (e.g. line tracking)with the Smart v3.0 Robotic Car.
year 11	-) Students learn the basics about AI and combine AI with robotics.	-) Students can define AI and know important areas of applications and can solve independently project with combines AI with robotics.
year 12	-) Getting to know robotic arms, as well as the associated mathematical concepts.	-) Students know basics about robotic arms and associated mathematical concepts.
Robotics club	-) Students are prepared for competitions like the RoboCupJunior.	-) Students succeed in solving the problems of the competitions.

Table 5.1: Overview of the robotics curriculum

contents	technologies	learning objectives	amount of time
-) Making first experiences with robots. -) Observe the behavior of the robot. -) Using color codes to solve tasks with increasing difficulty.	Ozbots with color sensors	-) Students first fears of contact with robots are taken away. -) Students know important color codes and the basic behavior of the robot. -) Students combine color codes to solve simple tasks.	2 school hours of 50 min

Table 5.2: Brief overview of year 5

the students. In addition, French and Crouse (2018) (cited in Mayerová et al (2019)) could show that the Ozobot increases girls interest in computer science. Finally, Fojtik (2017) reported that the Ozobot is one of the most appropriate tools to teach the basics of programming and algorithm. In summary, it can be said that the Ozobot offers a good introduction for the desired age group. A possible alternative choice to the Ozobot would be to use the BeeBot[3]. However, it is recommended to use the Ozobot, because it has more functionalities and can be used in future years. Also, the Lego-Mindstorm EV3 could be used at the beginning. It is recommended to start more gently so that students can concentrate on the robot and do not have to deal with programming and construction. From the point of view of developmental psychologists, a flatter entry can also be justified. At the beginning of the first year, for example in the case of mathematics, familiar topics from elementary school (e.g. basic arithmetic) are repeated. New topics (e.g. arithmetic with decimal numbers) follow after a familiarisation time in the summer semester. According to school curricula (e.g. mathematics[4]), the networking of skills also takes place at a later stage. Therefore, from a developmental point of view, it makes sense to work only with the robot at first and to start the construction and programming at a later time.

Let's have a look at lesson 1 as an exemplary unit. In the context of the development of the materials , a part of the already existing documents[5]

[3]https://beebot.at/ [Online; 21.November 2020]

[4]https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568&FassungVom=2017-08-31[Online; Last call 22.November 2020]

[5]https://storage.googleapis.com/ozobot-lesson-library/ozobot-workshop-2/

of Ozobot was used. During the unit, students first get to know the robot. Therefore, the first task is to activate the robot and subsequently calibrate it on a black spot. Students are then given the task of drawing a cross with a black pen on a white sheet of paper. The goal is to make the robot follow the black line and to observe the robot's behaviour at the cross. The main goal at this point is that after a few attempts, the students report that the robot always follows a random path.
The next big step is to discuss with students how to tell the robot what to do. The colour codes are then to be brought into play. Students should then experiment independently with colors and black lines.
Once the students have developed an understanding of how the colour codes work, they are receiving the first exercise sheet. As part of the task, the robot must move from a starting point (its home) to its destination (the school). The worksheet (see Figure 5.1) contains black lines with empty fields for colors. Students must draw the appropriate colors in the fields. It is recommended at this point that extra sheets are provided in case students make a mistake or alternatively, suitable colour dots are handed out.

The following lesson is based on the acquired knowledge. Students must then solve further tasks on the sheet like the example above. The main goal in the first year is, as already mentioned in the overview, to give students their first contact with robotics and thus take away their first fears or even better, to motivate them for the following years. For robotics in this school year 2 school lessons (2 times 50 minutes) are calculated. However, the content can be deepened with additional material, such as material from the official website[6] from which also exercises within the context of the curriculum were taken.

5.2.2 Year 6 - 2nd Year of Robotics Lessons

Table 5.3 shows you a brief overview of this year.

ozobot-workshop-2.pdf [Online; Last call 19.August 2020]

[6] https://classroom.ozobot.com/lessons [Online; Last call 19.August 2020]

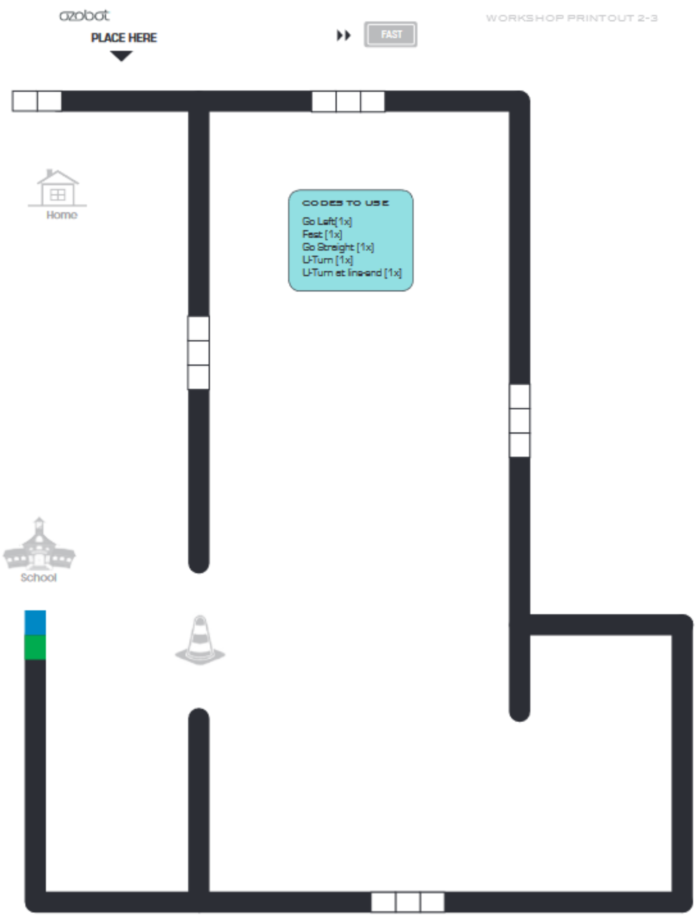


Figure 5.1: Example Worksheet for the Ozobot [7]

The students are between 11 and 12 years old in the sixth grade. This school year, students are going to work with the Ozobot again. But this time with Ozoblockly. According to (Hunsaker, 2018), the Ozobot in connection with Ozoblckly is suitable for students between 6 and 17 years. According to May (2020), the suitability of a graphical programming language is also supported by the developers of the similar programming language Scratch,

[7] https://storage.googleapis.com/ozobot-lesson-library/ozobot-workshop-2/ozobot-workshop-2.pdf [Online; Last call 19.August 2020]

contents	technologies	learning objectives	amount of time
-) Get to know alternative forms of programming in the form of blocks. -) Transferring programs to the Ozobot. -) Online provided tasks are solved with Ozoblockly. -) Students solve physical tasks (on paper) like the "Slot Car Challenge" and "Find the black Box" with Ozoblockly	Ozbots with Ozoblockly	-) Students know that there is an alternative programming language in the form of blocks. -) Students know how to transfer a program to the Ozobot. -) Students can solve practical tasks of varying degrees of difficulty using Ozoblockly.	2 - 3 school hours of 50 min

Table 5.3: Brief overview of year 6

who state that their language is suitable for children from 8 years of age. The arguments for using the Ozobot at a young age remain the same as in the previous section. The goal of this year is to get in touch with the next higher level of programming. This next developmental step can again be justified within the framework of developmental psychology with the development of skills within the framework of mathematics curriculum. Students learn in year 6 how to calculate with fractions. Here, for the first time, students no longer work with natural numbers. They now work with rational numbers and must additionally apply the previously learned basic arithmetic within the framework of these numbers. In the concrete case the robot is now connected to a programming language for control, which is more abstract than working with color pens.

Also in this year the first Unit/lesson is presented as an example. In this unit, students first learn how to transfer a program to the Ozobot. Part of this transfer is to calibrate the Ozobot on the screen. This is sometimes a little bit tricky. See Section 6.1 for some suggestions. Once the Ozobot is calibrated, the transfer itself is very easy. Students only need to hold the robot over the screen on the appropriate transfer field. The program is then transmitted using light signals. The Ozobot flashes during this transmission. As part of the practical implementation, students now have to solve tasks in Ozoblockly's online editor[8]. These tasks have been provided by the producer of the Ozobot and should be suitable for grade 2+. As part of the tasks, students have to make sure that the Ozobot runs figures, such as a parallelogram (see Figure 5.2), and lights up in the appropriate colors. The behaviour of the Ozobot is simulated on the screen. In order to get a feeling for the real world, students should test each program with the Ozobot itself.

Figure 5.2: Example for coding with OzoBlockly

In the next lesson(s), students have to solve further tasks online. Furthermore, they will also receive paper tasks[9] provided by Graz University of Technology. The total amount of time required is about 2 to 3 school lessons of 50 minutes, depending on whether all tasks are to be solved by all students. In this year contents can also be deepened, if enough time is available. Material can also be found on the official website of Ozobot.

5.2.3 Year 7 - 3rd Year of Robotics Lessons

Table 5.4 shows you a brief overview of this year.

In the third year of learning, the change to the LegoMindstorm kit takes place. This is again a development step, because the LegoMindstor has much

[8] https://games.ozoblockly.com/shapetracer-basic?lang=en&level=1 [Online; Last call 19.August 2020]

[9] https://learninglab.tugraz.at/informatischegrundbildung/wp-content/uploads/2017/08/ozobot_einheit_8.pdf [Online; Last call 19.August 2020]

contents	technologies	learning objectives	amount of time
-) Theoretical basics and application areas of robotics, as well as historical information about robotics. -) Theoretical information about the Lego Mindstorm EV3 like background knowledge about the brick and knowledge about the function of the sensors. -) Programming using a graphical language. Solving tasks with an increasing degree of difficulty. -) Application of all available sensors in different practical situations. -) Students learn basic programming concepts such as variables and control structures.	Lego Mindstorm EV3 with EV3-G	-) Students know what a robot is, where they are used and roughly their development path. -) Students know how each sensor works and can use it in different practical situations. -) Students can solve practical tasks of varying degrees of difficulty using a graphical programming language. -) Students know the most important concepts of programming such as variables and control structures.	9 - 10 school hours of 2 times 50 min

Table 5.4: Brief overview of year 7

more functionality and expandability than the Ozobot (García-Peñalvo et al, 2016). According to the manufacturer the kit is suitable for children aged 10+ (LEGO, 2020). The positive influence of LegoMindstorm robots has been investigated and tested in many studies. Cavas et al (2012) showed for example the positive influence within the framework of a robotic club after school, for the sixth and seventh grade. This influence is not only shown in the academic area, also the interest in technology and science has increased Cavas et al (2012). The positive influence of the robot and its deep and versatile applications (also in competitions like RoboCupJunior) show that it is a good tool for advanced teaching. As already mentioned in Section 3.1.2.1, Lego Mindstorm offers different possibilities for programming. Basically it can be distinguished between graphical and textual languages. Part of the research was to find out which kind of programming is more suitable for beginners at a young age. It was shown that both types of programming are equally well suited, but none is significantly better. On the basis of a study by Weintrop and Wilensky (2017), which indicates that the graphical version provides the better results, the decision was made to use this type. The changeover to the textual variant is going to take place in the following year. In the following year, it is also stated why a changeover makes sense from the point of view of the mathematics curriculum in year 8. In the context of this year we also want to refer to the curriculum of mathematics to show that the change of the robtic kit at this point makes sense from the point of view of developmental psychology. The range of numbers will be extended by the negative numbers in this school year. In the context of robotics, construction is now added to the previous skills. The level of abstraction remains the same with the chosen form of programming language.

In the following, one unit is going to be presented in more detail. This time it is the last unit of the module. The subject of this unit is the gyrosensor.
Each unit has the same structure. It consists of one part which repeats contents already studied, one part offering new input and one part fostering independent work in the form of exercises. The structure of the lessons is based on best practice. One of them is constant repetition. According to Bayrisches Staatsministerium für Unterricht und Kultus (2000) this is of great importance. Moreover, according to Guter Unterricht (2020), independent practice should be part of a well structured lesson. This strategy has already been followed in the units of the previous years of this curriculum. The only difference is that it has been made more visible to the students structuring handouts in that way.
Since working with the touch sensor was part of the last unit, a task must first be solved with its help as a repetition. For this purpose, one touch sensor is attached to the front and one to the back. If the front sensor is pressed, the robot moves backwards, if the back sensor is pressed, the robot moves forward. The next step is a theoretical input to the gyrosensor itself. The theoretical part was taken directly from the manual of LegoMindstorm EV3. Next the sensor is used for the first time. The robot should turn 90 degrees to the right. Once this task has been completed, the robot should turn to the left in sequence(see Figure 5.3). Students are informed that the robot is now turning in the negative direction and that the degrees are therefore also negative. This is followed by practical tasks which have to be solved by the students themselves. Within the scope of these tasks the

robot has to drive a square with a side length of 5 cm and, if this task is successfully completed, it has to drive an equal-sided triangle with a side length of 20 cm.

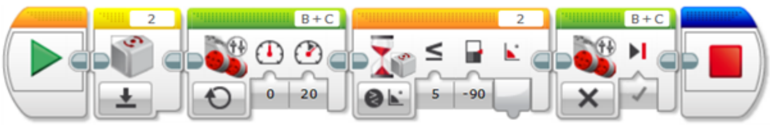


Figure 5.3: Example grahical Programming with gyro sensor in the context of LegoMind-storm EV3

It is suggested by the author to divide this module or teaching year into 9 units of about 2 times 50 minutes:

- **Theoretical Input (Unit 0):** Students learn the basics of robotics. This includes historical and application areas.
- **Basics (unit 1):** This unit is dedicated to the basics. This includes the basic structure of the EV3 (e.g. ports), available sensors and first steps in the development environment of EV3-G including program transfer. Within this framework, the robot should perform very simple movements like driving straight ahead.
- **Motion of curves (unit 2):** The aim of this lesson is to extend the knowledge of controlling the motors and thus to allow the robot to perform different movements, such as driving curves.
- **Control structures (unit 3):** In this lesson the knowledge about motor control is further deepened. First control structures like the loop are used. Furthermore, possibilities of calculations within the framework of the program is shown. In the context of this lesson, the students should also learn the concept of variables.
- **Ultrasonic sensor (unit 4):** In this lesson students learn about the first sensor - the ultrasonic sensor. In the course of this lesson students are also learn if-then-statements (e.g. stop at distance x).
- **Sounds and screen output (unit 5):** The purpose of this lesson is to teach students how to display values on the screen, for example. They should also be able to output sounds after the lesson, as well as make the buttons light up.

- **Colour and light sensor (unit 6):** In this unit theory about the light sensor (e.g. different applications) is taught. The unit is limited to reflection. The core of the unit is to implement a line tracking.
- **Touch sensor in connection with logic reasoning (unit 7):** This unit is intended to provide knowledge about the touch sensor in connection with logic (pressed vs. not pressed).
- **Gyrosensor (unit 8):** Details in the paragraph above (detailed view of one lesson).

All units are designed to take approximately 2 lessons of 50 minutes. Ideally, these lessons are held as double lessons. As a rule, however, 10 lessons should be included if a test is to be carried out at the end. However, more units can easily be performed if there is enough time available. For example, the light sensor could also be used as a color sensor and for measuring light intensity.

5.2.4 Year 8 - 4th Year of Robotics Lessons

Table 5.5 shows you a brief overview of this year.

The fourth learning year is very similar to the previous year. It only differs in the programming language used and in a project at the end. The programming language used now is EV3-Basic. The decision for this programming language is based on the following reasons: According to EV3 Basic (2020), the structure of the language Small Basic is relatively easy to learn, the transfer to the EV3 is simple, the language is free of cost and the pilot school for which the curriculum is specifically designed uses Visual Basic in the lessons of the higher classes. The reasons for switching to a textual programming language can be found in the state-mandated curriculum of mathematics lessons[10]. Students get into contact with algebra for the first time in grades 7 (basics) and 8 (advanced). This is very abstract for students, especially at the beginning. After a lot of practice they get along with it much better in the 8th school year. A textual programming language can also be very abstract at first. But since students in this school because of

[10] https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568&FassungVom=2017-08-31 [Online; Last call 20.August 2020]

contents	technologies	learning objectives	amount of time
-) Students learn the sameconcepts as in the previous school year. This time they work with a textual programming language. -) Furthermore, tasks of the KNAPP Robo League are solved.	LEGOMindstorm EV3 with EV3-Basic	-) Students now using familiar methods of solution within the framework of a textual programming language. -) Students solve complexcompetition tasks using a textual programming languageand LEGOMindstorm EV3 (-) Students take part in the KNAPP Robo League competition.)	12 school hours of 2 times 50 min, >12 school hours of 2 times 50 min for students who participate at the competition (students join robotics club)

Table 5.5: Brief overview of year 8

algebra year have already got practice in this abstract way of thinking, it is a good time to start with a textual programming language. The child's cognitive abilities are in the middle of phase 4 (lern psychologie, 2020). The child is thus also in an extended phase of his problem-solving competence (lern psychologie, 2020). It makes sense from the point of view of developmental psychology that the child now tries to solve extensive problems independently, for example in the form of a larger project (details later).

As already mentioned, the contents of the year are very similar to the previous year. In contrast to the previous year, this time a project is being carried out. For the first time, students are to solve a larger task independently. Students also have to work on the project as a team. Thus, it is also a goal that the teamwork of the students improves. It is also the goal that for the first time students consciously use concepts of problem solving (e.g. splitting the problem into smaller parts). At this point, the project at the end is going to be described in more detail. Within the project, students have to solve a task of their choice of the KNAPP Robo League[11] competition. The company KNAPP comes from the logistics sector but is also active in the field of robotics. Now based on KNAPP (2020) an overview of the competition tasks:

- **Follow the Line:** Within this task the robot must follow a black line. As an additional difficulty, the line is interrupted in some places. In the competition, the time the robot needs to follow the entire line matters.
- **Dont't touch anything:** Here the robot moves in a bordered arena. There are obstacles (bottles) in the arena. The robot has to avoid them, it is also not allowed to touch the walls of the arena. The field (150 cm x 120 cm) is additionally divided into small squares (30cm x 30xm). The winner of the competition is the robot that manages to move in the arena for the longest time without touching an obstacle and passing as many squares as possible. Figure 5.4 shows a picture of the arena.
- **Find the brick:** The arena has a size of 150 cm x 120 cm and is divided into squares of 30 cm x 30 cm. But the arena is not surrounded by a wall. Instead, the arena is surrounded by a black line. In one of the squares in the middle area, a block or book is placed. The goal is that the robot, which starts in one of the corners, finds the block or the book as fast as possible and emits a signal when the target is touched and also starts flashing.
- **Leave the labyrinth:** The fourth task is a labyrinth. The structure of this labyrinth is known. It is therefore also known where the "exit" is located. Only the starting position (4 possibilities) is random. The labyrinth is surrounded by a wall. The walls of the labyrinth itself are represented by black lines. The goal is to reach the end field, i.e. the exit, as quickly as possible.

The structure of the units before the project is the same as in the previous year to the already mentioned difference of the programming language. The time for the units themselves should be about 8 school lessons with 2 times 50 min (unit 0 of the previous year is left). For the project itself at least 4 school lessons should be included. This part can also be omitted. However, this is not recommended, since students have to solve more complex problems for the first time and thus problem solving and team

[11] https://www.roboleague.at/ [Online; Last call 20.August 2020]

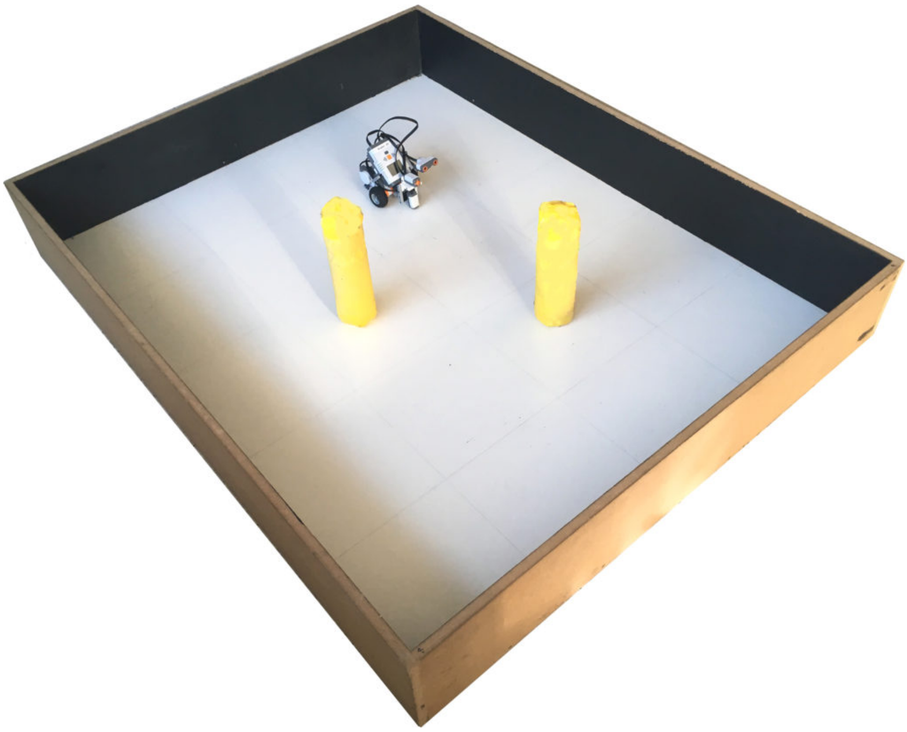


Figure 5.4: Arena of KNAPP RoboLeague for dont't touch anything[12]

working skills are properly promoted to a greater extent for the first time. Omitting the project would risk the achievement of important learning objectives for this year. If the time resources are not sufficient, the project can also be outsourced to a "Robotics club" (details see Section 5.2.9). Thus, it would be also possible to solve several or all tasks.

5.2.5 Year 9 - 5th Year of Robotics Lessons

Table 5.6 shows you a brief overview of this year.

Since students have already gained experience with abstraction last year

[12]Source: https://www.roboleague.at/teilnahmebedingungen/ [Online; 21.November 2020]

contents	technologies	learning objectives	amount of time
-) Theoretical basics about the microcontroller and in particular the Ardunino UNO. -) Students get to know the development environment and take their first steps in the Arduino C language. -) Students work with different sensors from the experimental box and solve different application tasks in the fields of measuring, acuation and control.	Arduino UNO with experimental box and Arduino C	-) Students know what a microcontroller is and where it is used. They have knowledge about the Arduino UNO, such as its construction. -) Students can solve various practical application tasks in the fields of measuring, acuation and control with sensors and use suitable components of the experimental box. They are also able to establish connections to the subject of physics.	6 school hours of 2 times 50 min

Table 5.6: Brief overview of year 9

in the context of the programming language in the previous year, this is now applied in the context of the robot kit. This school year the Arduino UNO is used. The Arduino offers more possibilities for self-construction and combines robotics with electrical engineering very well. The positive effects of the Arduino have already been demonstrated in several studies (also in comparison to other kits) such as JUNIOR et al (2013), Jang et al (2015) or Martín-Ramos et al (2017). It makes sense to start with the kit at this point, as students already have experience in physics, and in some cases also (like BG/BRG/BORG Köflach) in a physics laboratory. This includes knowledge from the field of electricity (e.g. motors) or thermodynamics (Department of Education Austria, 2017b). The contents can be therefore practically applied or deepened. In the specific case a kit for ”measure-actuation-control”[13] is used. It was developed by Leo Köberl, who works at the teacher training college of Styria (Köberl, 2020a). In contrast to the author, the programming language ArduinoC is used in this curriculum because students already have gained experience in textual programming. From the point of view of developmental psychology, the choice of the more abstract path in context robotics kits can again be justified with the curriculum of mathematics. In the context of mathematics, students must think more abstractly then before in the field of trigonometry for example this year. The bridge to

[13] http://msr.leo-edv.com/ [Online; Last call 21.August 2020]

physics is also evident in mathematics. In preparation for the school leaving exam, examples often have a physical context. Thus, a further step in the development of the skills of this curriculum is shown. To the more abstract programming language a more abstract robotics kit is added and a bridge to other subjects is built.

Since from now on the units are structured as scripts in order to be more flexible in lessons with older students, a section of the script will be described in more detail. The entire content follows in the next step. The following content is part of the "actuating" section. At this point students are already familar with the theoretical aspects of the Arduino system. Therefore, the first practical exercises are now done. In the context of this suitable components of the kit are used. In addition to various connections (e.g. for Bluetooth), one component of the kit consists of three LEDs in the colors red, yellow and green. At first the LEDs should light up one after the other. A further step would be the programming of a traffic light (see Figure 5.6). The final exercise is to control an RGB-LED. In the course of this exercise, students will learn about the RGB color model (more precisely they will repeat what they have learned in physics), and the difference between digital and analog ports.

Over all, the learning contents can be divided as follows:

- **Introduction Arduino and microcontroller:** In this chapter the students are taught what a microcontroller is and where it is used or integrated. Students also learn more about the Arduino and its construction (e.g. ports, power connections). Furthermore, the individual sensors/components of the construction kit are explained. The students also learn about the development environment and how the transfer to the Arudino works.
- **Actuating:** In the chapter "Actuating", the knowledge described above (detailed description of content) is teached. In addition, students learn about output options such as the OLED display and, in the course of this, how to integrate libraries. Furthermore, they learn about output via the console, as well as output via a LED cube and a seven-segment display.
- **Measuring:** In this chapter students work with different sensors. One of these is a temperature sensor. With the help of this sensor the

```
#define red 13
#define yellow 12
#define green 11

void setup() {
 pinMode(13,OUTPUT);
 pinMode(12,OUTPUT);
 pinMode(11,OUTPUT);
 digitalWrite(13, LOW);
 digitalWrite(12, LOW);
 digitalWrite(11, LOW);
}

void loop() {
  digitalWrite(13,HIGH);
 digitalWrite(12, HIGH);
 digitalWrite(11, HIGH);
 delay(1000);
  digitalWrite(13,LOW);
 digitalWrite(12, LOW);
 digitalWrite(11, LOW);
 delay(1000);

}
```

Figure 5.5: Code of the traffic light in ArduinoC

temperature has to be measured, suitably converted and written on the OLED for example. Subsequently, the LEDs should also start to light up, depending on the temperature. This combines knowledge from the previous section and from physics (e.g. conversion). In further consequence students also work with the sound sensor and develop a noise traffic light. As an alternative project a breathalyser can also be developed.

- **Control:** In this chapter the previous knowledge is combined. Students should build up a thermal control loop. This consists of a candle, a temperature sensor, a fan and a screen. The speed of the ventilator should depend on the temperature. The idea is based on Köberl (2020b).

For this module about 6 lessons with 2 times 50 minutes should be planned. Again, there are various expansion options, such as working with the ultrasonic or light sensors. Depending on progress and time, these content can also be covered.

5.2.6 Year 10 - 6th Year of Robotics Lessons

Table 5.7 shows you a brief overview of this year.

contents	technologies	learning objectives	amount of time
-) Introduction to the Smart v3.0 Robotic Car in combination with the Arduino UNO. -) Solving different motion tasks with the help of the car and ArduinoC. -) Working with the ultrasonic and line tracking sensor (with P-algorithm) in the context of practical tasks	Smart v3.0 Robot Car with ArduinoUNO and ArduinoC	-) Students know the construction of the Smart v3.0 Robtic Car including the most important connections. -) Students perform basic movement with the car using ArduinoC. -) Students use the available sensors to solve practical tasks such as tracking a black line and know the concepts used.	6 school hours of 2 times 50 min

Table 5.7: Brief overview of year 10

This school year, the Arduino is going to be used in combination with a model car, the Elegoo Smart Robot Car, based on duckytown[14]. The suitability of the Arduino for teaching has already been discussed in the previous section. In the study Page et al (2017) the suitability of the Robot Car in particular could be shown. From the perspective of developmental psychology, the combination of the more abstract robot with a car can now be justified using vector calculation. A 3rd dimension will be added to the vectorial calculation this school year. This also increases the abstraction in the context of mathematics. In the context of robotics, a further dimension of action is now added by the car to the more abstract robot.

A part of the script is going to be examined more detailed. At the end of the module a project should be completed again. Within this project the line tracking for the car will be implemented. For this purpose, the 3 tracking sensors (see Figure 5.6) of the car will be used. In this context the students are introduced to the idea of the P-only-algorithm. The PID algorithm is not part of the lessons. At the current time the mathematical

[14] https://www.duckietown.org/ [Online; 21.November 2020]

basics are still missing. Thus, from a developmental psychological point of view, in relation to the mathematics curriculum, it can be said that students do not yet have sufficient cognitive skills to understand the concept of the PID algorithm. However, it can be seen as a possible extension and can be taught to students who attend the robotics club.

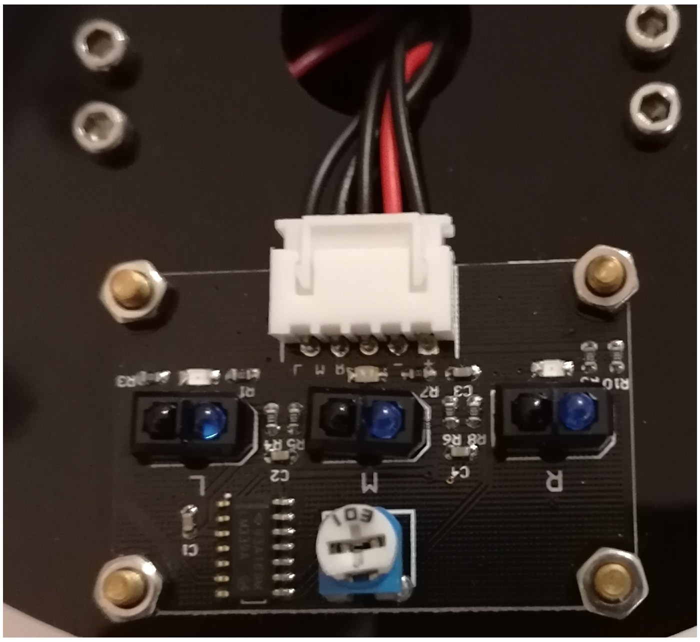


Figure 5.6: Picture of the 3 line tracking sensors of the Elegoo Smart Robot Car

The module is divided as follows:

- **Repetition of Arduino and introduction of Smart Robot Car:** In this chapter important foundations about the Arduino and microcontrollers are repeated. Then the concept of the Robot Car is explained in more detail. The functions of the individual components (e.g. motors) are also explained in more detail.
- **Movement:** This chapter explains how the motors can be controlled. It is explained which possibilities of control (analog vs. digital) there are and how they differ. Afterwards, exercises for controlling the motors is done, from simple straight driving, over simple driving of figures to exact speed control. Furthermore, the application of stepper motors is

also explained.
- **Ultrasonic sensor:** This chapter explains how the turnable ultrasonic sensor works. Moreover use of libraries is repeated. Different tasks related to the ultrasonic sensor are solved, such as slowing down and stopping in front of obstacles or avoiding obstacles.
- **Line Tracking:** In this chapter the line tracking explained in the previous paragraph is implemented.

As in the previous school year, about 5 lessons with 2 times 50 minutes should be calculated. If, for example, the idea of the PID-algorithm should be discussed and implemented, it is recommended to take 1-2 additional lessons into calculation. When there is more additional time available, tasks from the KNAPP Robo League (see school year 4) could be solved with the help of the Arduino.

5.2.7 Year 11 - 7th Year of Robotics Lessons

Table 5.8 shows you a brief overview of this year.

contents	technologies	learning objectives	amount of time
-) Students learn the most important basics about Artificial Intelligence, such as what Artificial Intelligence is and where it is used. -) Students get to know different chatbots and program their own chatbot in succession. -) Students connect the AI with robotics. With the help of an EV3, the fastest way out of a maze is determined.	LEGO Mindstorm EV3 with EV3-Baisc for the maze and the GaitoBot for programming the ChatBot	-) Students can define what Artificial Intelligence is and know important areas of application. -) Students know what a chatbot is and how it works. -) Students independently solve a project on the topic of AI in connection with robotics.	8 school hours of 2 times 50 min

Table 5.8: Brief overview of year 11

In the previous two school years, robotics was mainly combined with physics, or more precisely, electronics. In this school year a connection between Artificial Intelligence and robotics should be established. Since there is no state-mandated curriculum (in Austria) that includes Artificial

Intelligence to a sufficient extent, the necessary knowledge must be built up in the course of the robotics lessons. Thereby, the proven material from the EDLRIS (European Driving License for Robots and Intelligent Systems) program AI-Basics[15] will be partly used. Due to changes in curriculum and economic requirements, schools are beginning to integrate AI more and more into their curriculum (HTL Kaindorf, 2020). The first contact is done in the 11th grade (HTL Kaindorf, 2020). This shows the relevance of teaching of AI foundations. The general aim of this module is to give students a first insight into advanced concepts of robotics in combination with AI using a simple framework. According to lern psychologie (2020), the development of cogitative skills, also in terms of problem solving, is completed with this school year. With the beginning of the differential calculus in mathematics, another step towards abstraction is made. From the point of view of developmental psychology, it is therefore a good time to introduce the concept of AI and to combine it with robotics within the framework of a project (problem-solving skills).

The section that connects the topics robotics and AI is introduced in more in detail. The LEGO Mindstorm EV3 with 4 color sensors and the programming language EV3-Basic are used. There also exists a physical maze, which is represented by black lines. The maze is constructed in such a way that the robot finds the exit of the maze with the help of the left-hand rule (or right-hand rule). The end of the maze is marked with a black area. Figure 5.7 shows the structure of a very simple maze. The project consists of two parts. In the first part, the tracking of the lines is implemented as well as the finding of the exit using the left-hand rule (or right-hand rule). During this process the travelled path is stored in an array. In the second part a simple form of an Artificial Intelligence approach is implemented. For this purpose the path stored in the array is used. If the robot has to turn around, the path in the array is simplified (e.g. left-back-left is changed to straight). The algorithm used is dead-end filling (Norbert-Brendan and Cristian Marius, 2019). At the end the fastest way to the exit is stored in the array. Thus, the robot first finds its way out of the labyrinth and saves the path it has taken. By pressing a button on the robot, the second run starts and the fastest way is determined in the background based on the available information and is

[15] https://edlris.ist.tugraz.at/ai-basic/ [Online; Last call 21.August 2020]

also followed in sequence. The idea of the project is based on Marcelo Rovai (2020).

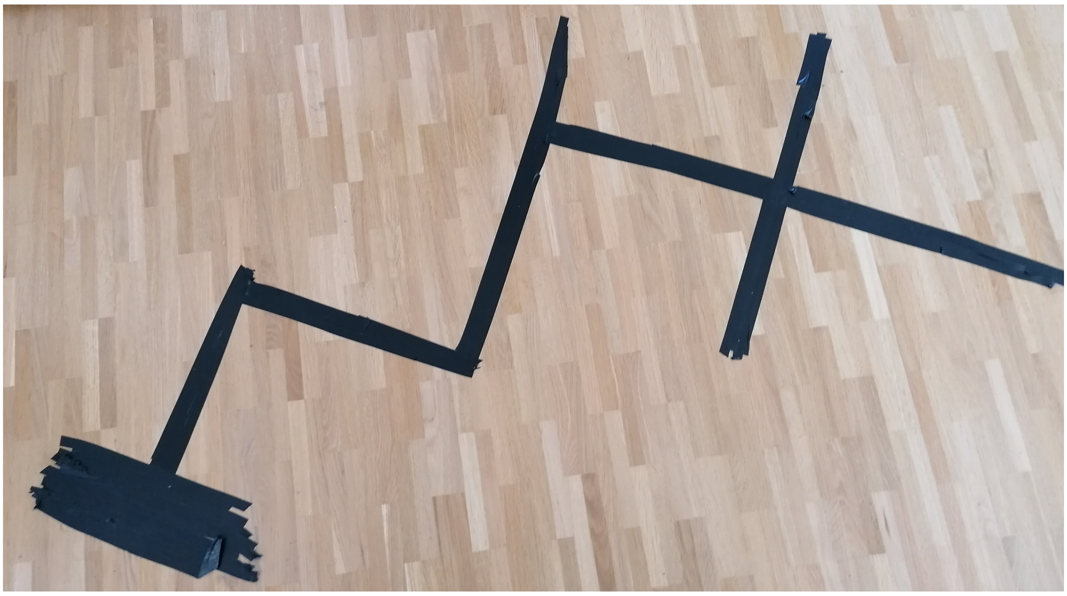


Figure 5.7: Example of a simple maze

The script includes the following contents for the school year:

- **Foundations of Artificial Intelligence:** This chapter deals with the basics of Artificial Intelligence. Students are taught what Artificial Intelligence is. They learn about different application areas of artificial intelligence. The basic features of neural networks are also explained.
- **Chatbots:** Chatbots are an easy way to understand AI concepts. Students get to know different chatbots from different websites. In the course of this they get explained how chatbots work, especially using the example ELIZA. Also machine learning is explained to the students and possible language barriers between humans and computers are discussed (e.g. problems due to natural language). Finally, a chatbot is implemented with the help of GaitoBot.
- **Connect robotics and AI:** After the students have understood the basics of AI, the connection between robotics and AI described in the paragraph above is realized. Through this project, the skills with

regard to problem solving will be trained again. This can be seen especially in the way the students should approach the solving of the tasks (e.g. breaking into sub problems).

As time expenditure at least 8 school lessons of 2 times 50 minutes should be planned, because students will work very much independently in this module. Possibilities for expansion are also offered here. More content from the EDLRIS project AI Basic can be covered, or even parts of AI Advance(e.g. advanced machine learning) [16] can be covered in class.

5.2.8 Year 12 - 8th Year of Robotics Lessons

Table 5.9 shows you a brief overview of this year.

contents	technologies	learning objectives	amount of time
-) Getting to know different robotic arms. -) Acquisition of theoretical (mathematical) basics necessary for programming such as matrices, homogenous transformations, translated frame base (TFB) approach and the geometrical model. -) Solving practical tasks using the programming language Python in a given framework.	Python and framework of EDLRIS-Course	-) Students know how robtic arms of robot's work and how they are constructed. -) Students know theoretical basics to be able to solve and understand practical tasks. -) Students know the most important structures and commands of the programming language Python to be able to solve tasks of the given framework in connection with the learned theories.	8 school hours of 2 times 50 min

Table 5.9: Brief overview of year 12

This year marks the end of the training under this curriculum. Here students will be introduced to more in-depth concepts. This will happen again based on the EDLRIS material. This time contents of module Robtics Advanced[17] are taught. The aim is to give students an idea of how a robotic arm works. The reason why this content is covered in the 8th class is the necessary

[16] https://edlris.ist.tugraz.at/ai-basic-2/ [Online; Last call 21.August 2020]
[17] https://edlris.ist.tugraz.at/ai-basic-3-2/ [Online; Last call 21.August 2020]

mathematical foundation such as matrices. For example, necessary contents are covered at a higher technical school in the 4th year (year 12) (Department of Education Austria, 2015). Since the necessary mathematical basics are generally not part of the mathematics curriculum of a secondary academic school, it is included into the robotic curriculum. Furthermore, the framework provided by EDLRIS uses the Python programming language. A brief overview of the most important contents of this language which are needed to solve the tasks also given to the students. In the context of the mathematics curriculum, students have reached the end of their education with integral calculus and hypothesis testing. These topics already require a high degree of mathematical abstraction. This is also necessary for learning the mathematical skills to control robot arms within the context of this chapter. Also, as already mentioned, the development of cognitive skills is completed. From the point of view of developmental psychology, students should therefore be able to cope with the very abstract topic of robotics in this school year. This year represents also the highest level of abstraction within the robotics curriculum.

In detail, the sections "Homogenous transformations" and "Translated frame-based approach" will be considered. Students are first taught what the homogenous transformation is needed for and what it represents. Next, the approach for the 2 dimensional cartesian coordinate system follows. At the beginning only the position of a point is described. Then it is shown how the orientation is described mathematically and how the two concepts (position and orientation) are combined. After some exercises the attempt is made to derive the homogeneous transformation mathematically. In the next step the translated frame-based approach is explained. This is followed by examples of six possible homogeneous transformations(revolute and prismatic joint for x,y and z). After some exercises solved together, students should be able to solve them independently. See Figure 5.8 for one example of modelling a robotic arm in this context. At the end, the extension to 3 dimensions is done.

The entire learning content is divided up as follows:

- **Basics about robotics arms:** Students first learn about different robotic arms and where they are used. They also learn about different joint types and what the term degree of freedom means.

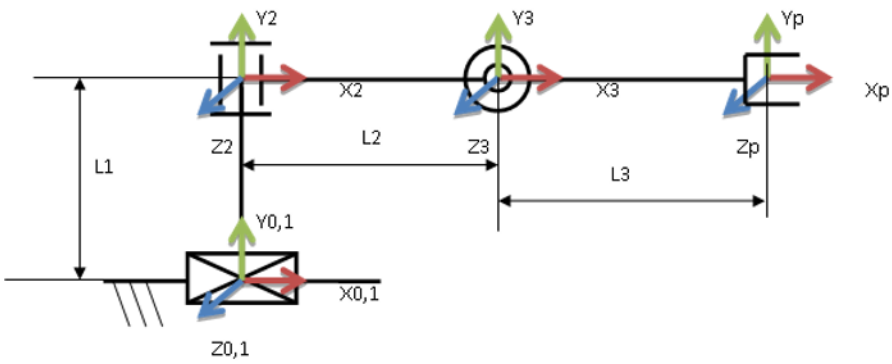


Figure 5.8: Example for modelling a robotic arm (Kandlhofer et al)

- **Basics of Python:** Students have to learn basics about python. These are user inputs, variables, data types and math, if-else-conditions, lists, loops and functions.
- **Homogenous transformations and translated frame base approach (TFB):** For details please read the paragraph above.
- **Geometric model:** The last basic theoretical knowledge the students learn is the geometric model. The idea of the geometric model as a geometric description of a kinematic process is needed for the programming of exercises.
- **Exercises in the framework:** The students should now complete basic exercises in the python framework provided by EDLRIS. These are concerned with different positions of robotic arms(2 and 3 joints). They apply concepts from the mathematical basics, Python and the learned theories of robotic arms.
- **Ethics:** Here, different aspects will be discussed with the students, such as the changes in the workplace situation caused by robotics.

Estimating a number of lessons needed for this module is difficult because it depends heavily on previous knowledge of the students. If all contents have to be worked out, at least 8 school lessons of 2 times 50 minutes have to be expected. The mathematical concepts will be difficult to handle at the beginning. Thus, it is important to give students enough time to understand the basic mathematics concepts. It is important that the majority of the EDLRIS materials of Robotics Advanced have been omitted due to the high mathematics level. Therefore, at this point there is also a far-reaching possibility of expansion, such as inverse geometry or the kinematical model.

5.2.9 Robotics Club

Table 5.10 shows you a short overview of the robotics club.

contents	technologies	learning objectives	amount of time
-) In the free subject robotics club students are specifically prepared for competitions. These are for example the KNAPP Robo League (year 8) or the RoboCupJunior (year 9 - 12).	LEGOMindstorm EV3 with EV3-Basic	-) Students succeed in solving the problems of the competitions independently and take part in competitions in consequence.	2 times 50 min every two weeks for one school year

Table 5.10: Brief overview of robotics club

In the framework of the robotics club students will be prepared for competitions. One competition is the KNAPP Robo League, which has already been explained. The target group here are the students of the 8th grade. Another competition is the RoboCupJunior[18] for the older students (grades 9 -12).

The LegoMindstorms EV3 will be used as a robot form. Working with self-constructed kits is not planned at the moment. This is an open point for the further development of the curriculum. At the moment, the goal for students is to deal with complex tasks of RoboCupJunior and are using the learned concepts of the previous years. A perfect solution or a good placement is not a primary goal. Research-based learning where mistakes are allowed (as in all parts of the curriculum) is the focus. In the school year 2020/2021 only one participation in the Resuce Line[19] competition is planned, as it fits well into the structure of the curriculum. Knowledge, which is necessary to solve the competition tasks and was not covered in the regular lessons, will be taught in the Robotics Club.

[18] https://robocupjunior.at/ [Online; Last call 21.August 2020]

[19] https://junior.robocup.org/rcj-rescue-line/ [Online; Last call 21.August 2020]

5.3 Integration of the Curriculum

An important aspect is how the curriculum could be integrated into the normal curriculum of a school. In very few cases there is a dedicated subject called robotics. It is also not common that the subject of computer science is taught continuously. This curriculum is also not primarily designed as a stand-alone curriculum. The goal is more an integration into existing lessons (e.g. computer science). In this section we will discuss how the individual sections of the curriculum can be integrated into the regular school curriculum.

For this purpose, the integration of the curriculum into the pilot school BG/BRG/BORG Köflach will be analyzed as an example. Details on the respective subjects can be found in Section 3.1.1. In the course of this, alternative integration possibilities of the curriculum, are also mentioned:

- **year 5 - 1st year of robotics lessons:** At the BG/BRG/BORG Köflach, the units are going to be integrated in the subject "German-Computer science". This subject is part of the "Digital basic education" mentioned in Chapter 1. As already mentioned, "German-Computer science" is a compulsory course for all students of the school. This means that every student of the school going to be taught in robotics in the first year. If it is not possible to integrate these units into the regular lessons, which is recommended, the units can also be taught in the context of project days (e.g. science day).
- **year 6- 2nd year of robotics lessons:** At the sample school the units of the 2nd year are integrated into the subject "Mathematics-Computer Science". This subject is again part of the "Digital basic education". "Mathematics-Computer Science" is also compulsory for all students. As the planned number of units is small, these lessons could also be taught on a project day.
- **year 7- 3rd year of robotics lessons:** From this year robotics at the BG/BRG/BORG Köflach is only taught to students of the natural sciences department (except year 9 and those students of the BG who choose Computer Science as a compulsory elective subject from the 6th school year). The robotics lessons are taught within the subject computer science. Unfortunately, it is not possible to teach robotics

in a larger scope due to the computer science curriculum, because other topics (e.g. word processing) have to be covered this year. As an alternative to integration, separate robotics lessons can also be offered. One hour per week (or e.g. blocked 2 lessons every 2 weeks) would already be sufficient, then it would be possible to look at some topics in greater depth. Many schools in Austria currently have the subject "Digital Basic Education" in their curriculum. Robotics could also be integrated into this subject (like in year 5 and 6 in Köflach). This could be justified by the contents of computational thinking that have to be taught. Due to the larger volume of content, its integration in project days is no longer suitable. Instead, the robotic club should already be used here, if it is not possible to implement the content in regular lessons.

- **year 8- 4th year of robotics lessons:** In this year it is unfortunately also necessary to switch with the robotics lessons to the robotic club at BG/BRG/BORG Köflach (for one school year), because there are also no computer science lessons in the natural science department this year. However, it is planned to change this in the future and to teach the lessons for all students of the department in regular lessons. The suggestions for alternative teaching remain the same as last year.
- **year 9- 5th year of robotics lessons:** In this school year, robotics content is going to be reintegrated into regular lessons. In Austria this is the only year in which computer science lessons are integrated into the timetable. In the BRG department in Köflach, the contents specified in the curriculum for this year are covered. If there were no lessons in robotics for the last 2 years (like at the BG department in Köflach), there would be two possibilities, which contents could be covered this year. The contents of year 3 could taught as well as the contest from year 5. The contents of year 3 or 4 could taught as well as the contest from year 5. A textual programming language can also be used, since students already have experience in abstract thinking through algebra. In the end, the decision is up to the teacher, depending on the performance of the class which contents would be taught. If it is not possible to teach the contents in the regular computer science class, a alone standing subject robotic or the robotics club could be used to teach the contents of this year. This could then be combined with a preparation for competitions.

- **year 10- 6th year of robotics lessons:** This school year is the last in which computer science is a compulsory lesson at the BG/BRG/BORG Köflach for all students of the BRG. The contents listed in the curriculum are integrated into these lessons. This is also an exception within the Austrian school system. As a rule, there is a compulsory elective subject (in Köflach for the students of the BG) computer science in which the contents can be integrated depending on progress. At the BG/BRG/BORG Köflach this year the contents of the 4th year of robotics would be taught in the compulsory elective subject in order to practice textual programming with Lego Mindstorm EV3 because of the further contents of the curriculum. The alternatives where the lessons could be integrated are the same as last year.
- **year 11- 7th year of robotics lessons:** In this school year, only students who have selected computer science as an optional course will attend the robotic lessons. In general, it is also recommended to work only with interested and talented students in the field of computer science/robotics from this point, because the contents of the curriculum grows in complexity. For a possible elective subject (students of the BG) it is recommended to cover the contents of the 6th year of learning. The other alternatives are the same as in the previous two years.
- **year 12- 8th year of robotics lessons:** The content of this year will also be taught in the elective subject of the BRG. The contents are also only taught in the elective subject of the BRG, since the students in this department have received the necessary (extra) training in mathematics to understand the contents. It is therefore not recommended to teach these contents in the elective compulsory subject of a BG, for example. For the elective subject of the BG it is recommended to deal with the contents of the 7th year of learning. The other alternatives remain the same as in previous years.

As can be seen in the above list, the contents of the curriculum can also be exchanged or omitted. However, if it is possible to deal with all contents of the curriculum or to keep the order of the contents, this is recommended.

The presented curriculum, when fully integrated, allows a continuous robotics education within the school for 8 school years. This will ensure a continuous development of skills in robotics, as well as the skills related to it (e.g. computational thinking). Like the curriculum in mathematics, it also offers, from the perspective of developmental psychology, a stringent development of skills, especially in the area of abstraction. The exact integration of the curriculum depends on the school location and the school system. In general, as mentioned above, the integration of the content into existing subjects should be pursued. This could be computer science for example, but also a subject like physics or mathematics. The contents of class 5 for example are also suitable for a laboratory in physics. The individual units can also be adapted. For example, the Ozobot can also be used for vocabulary training, so that the contents of year 1 can also be taught in a foreign language for example. A robotics club is also suitable for the implementation of the contents. However, the problem is that the attending of such a club is voluntary and therefore many students who would be reached in regular classes might not attend it.

6 Evaluation of Curriculum

This chapter serves to evaluate the curriculum. This is done in 2 parts.

Part 1 reports on experiences from the practical implementation of parts of the curriculum. The aim of this evaluation is to share best practice experiences and to determine how well each part of the curriculum works for itself in the classroom. Which parts of the curriculum have already been used was explained in Section 3.1.2.2. These experiences are mainly based on observations made by teachers during classroom instruction, but also on informal discussions between students and teachers after class. From observations we also try to establish a connection to the whole curriculum. The curriculum was not in use in his complete form until now. A first large-scale deployment was planned for the school year 2019/2020, but this was only partially possible due to the corona measures, which were in effect for almost the entire summer semester.

In part 2 of the evaluation, expert comments on the curriculum were collected. The focus was more on the curriculum as a total. The aim was to determine whether the structure of the curriculum is appropriate, whether the individual parts fit together well, and whether the curriculum is also suitable for use in other schools, especially in other countries, besides the pilot school. The robotics curriculum was sent out in combination with questions to ensure a consistent structure of the feedback. The individual questions, as well as a justification for their choice, will be explained later. The evaluation was sent to 13 experts, 5 of them wrote a feedback. The individual feedbacks are incorporated verbatim into this paper and the most important statements are summarized.

6.1 Best Practice Experiences at the Pilot School

As already mentioned, this part of the evaluation shows the experiences at the pilot school. As described in section 3.1.2.2, some parts of the curriculum have already been used in class. This was the case in projects (e.g. summer school), but also in regular lessons. The contents of the curriculum were also used by 4 different teachers. In total, about 150 different students have been taught at least one part of the curriculum in the last two school years. In the following, the experiences with the individual parts of the curriculum will be examined in more detail:

- **year 5 - 1st year of robotics lessons:** According to feedback from the teachers mentioned above and from the author's own experience it can be said that the use of Ozobot as a first contact with robots has worked well so far with small experimental groups or in the form of workshops. This experience is in all cases based on observations by the teachers and informal interviews with the students. The success with regard to the entire curriculum and thus the following years was shown by the fact that students were motivated for further instruction with robots. They indicated that they would like to continue working with robots in the future. As far as the organization of the lessons is considered, it can be said based on personal experience and from the feedback of other teachers that, it is recommended that enough worksheets are available. Furthermore, the pens, with are included in the package of the Ozobot, tend to empty quite fast. Therefore it is recommended that there are enough pens in reserve and that color dots are used, especially during correction.
- **year 6 - 2nd year of robotics lessons:** The form of programming was used by the author in the context of the summer school and was experimentally tested with older students of the robotics club after the purchase until now. In both cases there were problems with the calibration of the Ozobots. The calibration of the Ozobot in screen often does not work well with the button on the Ozobot. It is better to calibrates the Ozobot using the App[1] provided. It is also recommended

[1]https://play.google.com/store/apps/details?id=com.evollve.evo&hl=de_AT [Online; Last call 19.August 2020]

that the teacher calibrate the Ozobot before the lessons to save time. As far as programming is concerned, it turned out that the students at the summer school, who are the target group of this form of programming, were able to solve the tasks well and quickly. This shows that an understanding for this form of programming was built up. Whether these successes in the context of the summer school are also long-term and whether the entry of the LegoMindstorms EV3 in class 7 is now easier for the students can unfortunately not be said due to the short time gap since the summer school.

- **year 7 - 3rd year of robotics lessons:** The study (see Section 4) showed a significant improvement in the students' performances in robotics (and computational thinking). In class, the procedure was the same as in the curriculum. This also showed that the materials used are suitably structured within the context of the presented curriculum and that the desired learning success is achieved. From the author's experience, it can be reported that students who have already made first experiences with the graphical form of programming were able to solve the tasks faster with textual programming in the following year than students who had not worked with LegoMindstorm before.
- **year 8 - 4th year of robotics lessons:** In the area of textual programming a significant increase in the performance using the proposed curriculum could be shown. Furthermore, three first places could be achieved in the KNAPP Robo League in the school year 2018/2019 (the competition did not take place in the school year 2019/2020). In the school year 2018/2019 the preparation took place in the setting of the robotics club. It should be noted, however, that these first places were achieved by students of the 9th and 10th grade. The reason for this was that the school was equipped with the Mindstorms in school year 2018/2019 and the level at RoboCup would have been too high. However, the students were prepared with the curriculum/materials mentioned above.
- **year 9 - 5th year of robotics lessons:** Based on practical experience, the use of the Bluetooth sensor is not recommended because of the many signals in the classroom, which caused problems due to interference and the identification of the appropriate signals. Therefore this sensor should only be used when working with very small groups. Unfortunately, nothing can be said at this point about the effects on

the curriculum as a whole, since further observations in this area were not yet possible.

- **year 10 - 6th year of robotics lessons:** As already mentioned, only parts of the curriculum were dealt with in class. The line tracking was not content of the last years. It should also be mentioned that the Bluetooth sensor (remote control of the car) should not be used for the robot car either. The reasons for this can be seen in the previous year. Again, no statements can be made at this point about the effects on the following years.
- **year 11 and year 12 - 7th and 8th year of robotics lessons:** The contents of school years 11 and 12 have not yet been covered in class. Therefore nothing can be reported here from practice.

6.2 Qualitative Research

As part of the development of the curriculum, feedback in the form of "expert-feedback" was collected from experts in the field of teaching robotics and computer science. With Mr. Elias and especially Mr. Jump, experts with experience in curricula were selected. In the case of Mr. Elias this is for example the implementation and support of other schools with the implementation of the curriculum presented in Section 2.2.2.2, in the case of Mr. Jump this is the self development of the curriculum presented in Section 2.2.1.2. All other people do research in the ER, Mrs. Amy Eguchi is even one of the leading researchers in this field. In order to accomplish the elevation the experts received first information about the author, as well as informations about the goals of the project. To give the experts a better overview of the project, an "Executive Summary" (see Appendix C) was written. This contained the most important information about the curriculum. Detailed informations could be found in the context of this thesis, which was also send to the experts.

In order to facilitate the writing of the review for the experts and to make the review consistent and comparable, they were asked to answer the following questions:

- **Is the curriculum suitable for use in schools?**: The idea behind this question is that experts first think in general terms about whether the curriculum is suitable for use at school. This includes, for example, the kits used and the learning objectives presented. In this context, experts should also think about didactic aspects and whether these have been appropriately implemented in the curriculum.
- **Is it well structured?**: The next question focuses on the structure of the curriculum. The question is whether the individual development steps are comprehensible, or whether development of skills is too slow or too fast. It is also important to determine whether it is comprehensible for the experts when which goals are pursued and when which content is covered.
- **Are the requirements appropriate?**: This question is mainly about the requirements within the curriculum. The experts should consider whether the content covered is appropriate for students in the respective school levels.
- **Could you imagine that the curriculum could also be used in your country?**: The last question is to determine whether the presented curriculum can also be used in the countries of the experts. This is of great importance because it is an goal of the curriculum to serve as a template and to be used in other schools and countries.

In the course of the survey, the following feedback could be collected. Where possible, this feedback is taken over literally:

- **Timothy E. Jump, Director, Quadrivium Engineering and Design, Hill School, USA:** *"This curriculum represents a well thought out series of manageable learning steps, suitable for the limited number of available coverage hours per year. The progression of devices and code forms should work well to engage younger students and develop their comfort level working with technology beyond the basic end-user applications. Comfort with, and exposure to "under the hood" concepts and terminologies is critical to expanded growth of code and control learning through life.*

  The progression in complexity of the challenges is aligned well with child brain development and readiness to transition from the concrete to the abstract. The thorough review of graphical vs text level code as related to learning supports the curriculum appropriateness for student readiness to learn.

Reflecting on student experiences in their first college course where code was discussed, if they've never had any exposure to programming they can feel lost. Often, they drop the course. Comfort and recognition of code processes is essential to minimize "code shock". This curriculum maintains exposure to code (and various forms of code and problem solving) across multiple levels of school, and should build adequate comfort with embedded thinking for students.

Finding ways to integrate computer science and engineering learning and thinking into cramped curricular structures is a challenge for many education systems. The approach presented in this curriculum is appealing, and should be interesting to teachers that also have a reluctance to engage with code. Outcomes are easy to visualize, which helps to diffuse teacher angst when adding an unknown into their teaching day. The curriculum also retains connections between years, which allows teachers across the different grades to collaborate and support each other as they can learn the various stages of the curriculum together. I can see this curriculum fitting well into the US education system.

Note: Ease of, and persistent calibration are critical for teachers without a strong sense of expected vs. actual data output. If they do not recognize the data is skewed, teachers can themselves be, and perpetrate, confusion about observed outputs.
This curriculum represents an appealing and practical integration of code and control-based learning.

Nice Job Michael! "
- **Martin Kandlhofer, Austrian Computer Society (OCG):** *"The thesis focuses on the development of an extensive Educational Robotics (ER) curriculum which covers eight years of school education (for students from age 10 to age 17). For each grade (module), Robotics activities are elaborated with detailed information regarding learning objectives, contents, applied technologies (ER platforms) and estimated amount of time.*

  The modules are aligned, with a subsequent module building on the previous one(s). Learning objectives, contents and suggested platforms of each *module are matched with the skill-sets and abilities of the corresponding age group. This is backed by relating the curriculum to other best-practice curricula and experiences from different countries all over the world as well as by insights from the field of educational sciences and didactics. Therefore, the structure of the curriculum as well as the level of educational requirement is adequate. A very promising pilot implementation and evaluation of certain modules of the curriculum was done in an Austrian school. In order to proof the effectiveness of the entire curriculum, further implementations and evaluations (of all modules) and with a larger sample are of course necessary.*

Though, a lot of ER material and activities can be found nowadays, most focus on certain age groups or deal with isolated aspects. A coherent ER curriculum, as presented in this thesis, could therefore be real asset and template for further schools - in a first step in the German-speaking countries, and subsequently - after further implementations, evaluations and translations also in further European countries. Having said that, the integration into regular school curricula still poses a challenge. Based on the well-grounded methodology applied during the development and evaluation, the integration of external experts and best-practice examples as well as the scientific educational and didactical foundation, the proposed ER curriculum is definitely suitable for school implementation. I'm looking forward to see the curriculum being implemented in the pilot school as well as in various other schools too. All the best!"

- **Gerard Elias St. Francis Xavier College, Australia:** *"Is the curriculum suitable for use in schools?*
  Very much so, it's scaffolded enough so a school that isn't experienced in robotics could pick it up and run with it. Have you thought of using 'Learning Intentions' and 'Success Criteria' instead of Learning objectives, This has been a real emphasis in the last 18 months At many school around the country. Also maybe a range of technologies that could be used to implement the objectives and skills being taught.

  Is it well structured?
  At first glance I found it not lineal in its structure in terms of equipment. However when I looked at the details the emphasis of the program in the later years focused on solving mechanical problems, and using a custom solution would not be appropriate in the timeframe allocated.

Are the requirements appropriate?
I think the requirement are appropriate if you mean that the time allocation vs the content wanting to be covered. It would be tight and there would be no room for classroom management (packing/setting up) but it could be done. We run 3x1 hr lessons a week in 7-10 and 7x1hr a fortnight in year 11/12 and it gets busy.

Could you imagine that the curriculum could also be used in your country?
Yes, I could implement this in our school as we are setup to do so, I think for a new school it would be a huge undertaking and to purchase the equipment would be difficult, but not impossible as you have given the teachers a solid base to put together a curriculum plan that they could use to obtain the required funding to implement the courses. It would be interesting to see what the perceived cost of each unit would be with a class of 24 students and cooperative learning groups of four.

I would also love to see extensions concerts that could be covered using the equipment stated if there were students that needed to be challenged."

After a mail exchange with Mr. Elias it turned out that the last remark can be solved by using the "Robotic Club".

- **A. Fernande Ribeiro, Universidade do Minho, Dep. Electrónica Industrial – Robótica:** *"I read carefully the document "er_curricula_hubman.pdf". The plan is good and well established, but even though, I added some comments (details which the student might consider, or not) on the right side of the PDF document.*

  Is the curriculum suitable for use in schools?
  Yes, I just have doubts about the number of hours required... Maybe those are too many. Students already spend too much time in school (at least in my country)

  Is it well structured?
  Yes, definitely. I liked it very much.

Are the requirements appropriate?
Yes, although you could argue. Each person has his own opinion, and the student should follow his opinion.

Could you imagine that the curriculum could also be used in your country?
Yes, why not ? It is well planned. I just don't think that all schools would have the money to buy all these educational robots (unfortunately). "

Mr. Riberio's comments focused on the extra time that students have to spend at the school due to robotics. However, there was a misunderstanding because the executive summary did not clearly state that students do not have to spend extra time in school, because robotics are integrated into regular classes. He also made some remarks on alternative construction kits.

- **Amy Eguchi, Department of Education Studies, UC San Diego:** *I think the curriculum/unit for each year/grade is well organized and suitable for use in schools, especially at the school the author teaches and Austria. However, for the multiyear curriculum, there needs more clear continuity, especially to expand the implementation to the wider community or another country. One way to emphasize the continuity is to add a transition from the previous year to the unit of the following year (you might already have it, but it was not clear by reading the manuscript). Overall, each unit is well structured with an introduction, difficulty incrementation from the beginning to the end, and some expansion ideas.*

  Regarding the introduction of input with LEGO ev3, I would suggest introducing a touch/push sensor first than other sensors which require students to understand the sensor reading. The touch sensor is easier to understand how input works especially because it uses a binary number (0, 1). It is an important concept for students to understand. Then students can move onto other sensors that require students to understand how sensor readings are used.

  In addition, the transition from one type of robotics tool/kit to another and/or one programming environment to another seems too quick, which is also contributing to the lack of continuity, I believe. For example, from years 7 to

8, students are using the same hardware but using different programming languages/environments. It would be OK with students who are already familiar with graphic-based programming language with robotics. But if students do not have much programming experience prior to using ev3, this transition could be very confusing. Also, the curriculum suggests students to use multiple robotics hardware, and the transition seems too quick. It also suggests going back and forth between ev3 and Arduino in a short period of time. Those additional changes could confuse some students.

I would suggest using Ozobots with younger students, introduce them to robotics earlier. In the US, some teachers use Ozobots with younger students (1st -3rd graders). Move to ev3 much earlier, which will provide more time to learn with ev3 before moving to Arduino in middle school. Continue using Arduino from middle school into high school will help them learn the hardware and text-coding at a much slower pace. We use ev3 from 2nd grade to 4th grade, and in robotics after school in elementary school. Introduce motor control in 2nd grade, touch sensor in 3rd grade, and other sensors (introducing sensor reading/value) in 4th grade. This structure has been working well for a particular school. Also, 4th graders showcase their robotics creation with their kindergarten buddies, which help us introduce robotics in younger grade (less fear with robotics and excite them to see themselves creating robots when they are older).

It is not clear in the manuscript if, at the end of each year, students work on a final project, which is strongly recommended. A final project should be as open-ended as possible so that it helps students learn design thinking, engineering, and collaboration skills. Also, it will spark their creativity. For example, with the above mentioned 4th-grade robotics curriculum, students work on creating a robotic toy for their kindergarten buddies, which provides them with an application of their robotics learning in the real world. They have a target population for their robotics creations, and it is easier for them to "visualize" their audience/customer using the people in their community (in their real-life).

What will help the US teachers to implement the proposed curriculum is to tie it with learning standards, which could be quite different between Austria and the US. However, the cross-mapping will help them adapt the curriculum to *their existing lessons. In the US, it is hard to create robotics as a subject/class. Rather, teachers need to be creative and adapt it in existing lessons (could be math, science, etc.). We try to do cross-mapping with non-STEM subjects so that robotics can be a learning tool for other subjects, such as language arts, music, social studies, etc.*

To answer the last question about whether the curriculum could be implemented in the US, I need to respond in multiple steps. I think each year's unit can be implemented in the US public schools if teachers have the capacity and capability (programming experience/skills) to implement. But it will be very difficult to implement the entire curriculum in US public schools. Maybe it is possible in private schools if they have the right teachers and funding. The reason for my answer is because of the cost of the hardware, and the requirement for teachers to be able to teach multiple programming languages. In US public schools, it is very rare for a school to obtain multiple robotics kits, which are still fairly expensive for schools to purchase enough kits. In addition, because of the Computer Science Standards in the US, many high school teachers focus on teaching Java at the high school level. Especially because the Advancement Placement test is on Java (https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-a-glance.pdf?course=ap-computer-science-a), it is unlikely that teachers will take time to teach other programming languages.

Having said that, I believe the presented curriculum will make other educators realize multiple potentials and possibilities of implementing robotics in their classrooms. Even if not the entire curriculum, part of it will be ideal for those teachers willing to take on the new challenge than none.

We will now summarize the most important statements of the experts regarding the different questions:

- **Is the curriculum suitable for use in schools?**: Mr. Jump mentions in his review that the curriculum is suitable for use in schools and that its structure will help teachers to overcome their fear of dealing with the new field of robotics. Mr. Jump also wrote that the curriculum represents an appealing and practical integration of code and control-based learning. Mr. Kandlhofer pointed out that the curriculum is coherent and could therefore be a template for other schools. According to Mr. Elias, the curriculum is suitable for use in schools and can be used especially by those schools that have no experience in robotics. According to Mr. Riberio the curriculum is good and well established. Mrs. Eguchi wrote that the curriculum is well suited for use in Austrian schools, but needs more clear continuity if it is to be used internationally.

- **Is it well structured?**: Mr. Kandlhofer wrote in his review that the modules of the curriculum are well structured and each module is founded on the previous one. Mr. Elias wrote that at first glance he thought the curriculum was not linear in terms of the kits used. Nevertheless, the details helped him to better understand the structure. Mr. Riberio wrote that the curriculum is well structured. Mrs. Eguchi wrote that every unit is well structured with an introduction, difficulty incrementation from the beginning to the end, and some expansion ideas. However, she also made suggestions for restructuring. For example, she suggested an earlier use of LegoMindstorm EV3. She also suggested that students should work on a project at the end of each school year.
- **Are the requirements appropriate?**: The complexity of the tasks is mentioned positively by Mr. Jump. He also noted that they are well coordinated with the development of the children. Mr. Kandlhofer wrote that each module of the curriculum matches well with the skill-sets and abilities of the corresponding age group. According to Mr. Elias, the requirements are appropriate, even though the time schedule is very tight. According to Mr. Riberio, the requirements are suitable, but he mentioned that everyone will have a different opinion in the context of requirements.
- **Could you imagine that the curriculum could also be used in your country?**: According to Mr. Jump, the curriculum can be used in the US school system. As already mentioned Mr. Kandlhofer pointed out that the curriculum could be used in other schools. He pointed out, that it could be a first step for teaching robotics in German-speaking countries. According to Mr. Elias, the curriculum can also be used in Australia, although procuring the materials would be a challenge. Mr. Riberio (Portugal) and Mr. Eguchi(USA) also wrote that the curriculum could be used in their country, but also expressed concerns about the financing.

Analyzing the above feedback, one conclusion that can be made is that the curriculum is suitable for use in schools. The structure of the curriculum and its units are also appropriate for use in school, even if improvements could still be made in some places (e.g. projects at the end of each school year). Moreover, the requirements were described as suitable for the respective age group by the experts. Regarding the use of the curriculum in the international arena, all experts indicated that, from their point of view, the curriculum could be used. One point that was mentioned several times was the high costs for the kits. This problem also existed at the pilot school and exists in Austria in general. At the pilot school the robot kits were financed by the parents' association of the school. At other schools (e.g. Kepler) sponsors (e.g. KNAPP) also funded a part of the kits. These possibilities of financing can also be considered for schools in other countries. Although, proper funding of a good school system should be the aim of society.

In the coming school years, the curriculum need be evaluated more carefully. Especially how well the individual parts of the curriculum fit into each other must be investigated. To which extent feedback from the experts can be incorporated into the curriculum will also be analyzed in the near future.

7 Conclusion

In this chapter the most important findings of the thesis are summarized. It also provides an outlook on further studies related to the research.

The goal addressed in this thesis was the development of a holistic curriculum for robotics education for students between 10 and 17 years of age. Details of this curriculum was presented in Chapter 5 of this thesis. It is based on previous research on the area of Educational Robotics and developmental psychology. In order to ensure a suitable curriculum structure, the first step was to identify which skills should be taught with the help of ER and examine existing curricula from countries and students around the world were examined. It has been shown that ER is primarily intended to teach computational thinking, problem solving and social skills. In order to develop a strong sustainable curriculum, it was also investigated which type of programming (graphical vs. textual) is better suited for novice programmers. The results of this study were presented in Chapter 4. Based on this results the curriculum for robotics was developed. In the context of its development, knowledge from previous studies and curricula as well as aspects of developmental psychology were used. In Chapter 6 results of an evaluation of the proposed curriculum were presented. The evaluation was done with the help of experts as part of a qualitative research. In addition, best practice advice for the individual school grades has been provided in this chapter. In the following section the most important findings of the thesis are now considered in the form of research questions.

7.1 Results of the Research Questions

Now the research questions presented in section 1.1 at the beginning of the thesis will be answered:

Which skills (e.g. computational thinking) can successfully be taught using educational robotics (ER)?
In the thesis the skills computational and engineering thinking, problem solving and social skills were mentioned as those which should be taught using ER. During the research the positive influence of ER on these skills can be shown. As stated in section 3.2.3, a test was conducted to measure the success of skills in the context of ER. In the related research discussed in the context of this thesis a significant positive influence on computational thinking skills (for both groups) was found. Moreover, a significant positive influence can been seen on questions of knowledge in the context of problem solving for the group using textual programming, a significant positive influence on self-assessment questions in the context of problem solving for the group using graphical programming and again a significant positive influence on social aspects could be shown across group boundaries. In any case, a positive effect was shown in all areas considered in the conducted study. The area of engineering thinking was not investigated in this thesis.

How are curricula structured for teaching robotics on various levels around the world? How do they differ in their structure?
In the course of the thesis, curricula/learning contents from Austria, China, Australia, the USA and Canada were investigated. They differed in terms of content and duration in school years. One curriculum from the USA, for example, has a very strong focus on engineering thinking, whereas this skill plays a minor role in another curriculum from China. The remaining schools were more broad-based in their approach to knowledge transfer and tried to cover as many aspects as possible. At this point, the state defined curriculum of Canada should be mentioned. This was developed by a state institution, covers 7 school years and is designed for a very broad education in the context of robotics. This is the largest time span after the curriculum of the BRG Kepler, which covers 8 school years. They are followed by the one from the USA with 4 years, Australia (the curriculum is also

state-defined) with 2 years and the one from China with 1 year. Thus, it is clearly evident that the scope of the curricula and the extent of robotics instruction differs considerably. Another interesting fact is that only the curriculum from Austria provides obligatory contents for all students. In the other curricula all contents are optional or can be chosen as an elective class. The technologies used also differ widely. In Australia and China, established technologies are used like Arduino and the LEGO Mindstorm, while in the USA and Austria custom-built kits are used. In Austria established comercially available building kits are used in the first years of learning and the own building kits are used in further education. In summary, it can be said that curricula vary greatly across countries and that up to now robotics contents have mostly been taught in freely selectable courses.

Does modality (text-based vs. graphical programming) in the context of ER influence the learning success of the skills mentioned in question 1 for programming novices?
No significant difference could be found in any of the areas examined. It should be noted that the group using graphical programming showed a higher increase in performance at computational thinking (programming part) and the group using textual programming in the skills in problem solving. It can be stated, that both approaches are equally well suited for the area of computational thinking because of the major part of of the lesson comprises programming.

Based on questions 1 to 3, how should a curriculum be structured to ensure a holistic and broad robotics education over 8 years?
The main objective addressed this thesis was achieved with the creation of a holistic curriculum for robotics education. The goal of the curriculum is to educate students as broadly as possible in the field of robotics, as well as to give students insights into deeper concepts of robotics. The evaluation by experts has shown that the curriculum is well suited for use in schools. The structure of the curriculum was also described by the experts as appropriate. However, it was also noted that restructuring could be considered. According to experts, the requirements are also suitable for every age group. All experts also stated that the curriculum is suitable for an implementation in their country, even if some of the explanations in the context of the curriculum would have to be more precise. The cost factor was also mentioned as

a problem. In summary, however, it can be seen that a suitable curriculum with a consistent structure has been developed, which can also be used in other schools and countries.

7.2 Outlook and Future Work

The most important issue for future work is that the entire curriculum need to be evaluated in practical use. Ideally, it should also be used in various different schools. Thus, valuable objective feedback can be obtained and necessary adjustments could be made. However, such adaptations are necessary in schools and especially in teaching computer science due to technical innovations. It is also necessary to take another close look at how well the individual modules work together. This has only been partially possible up to now, as only parts of the curriculum have been used for teaching purposes.

In order to better identify possible differences between textual and graphical access to programming, it will be necessary to conduct a large study including much more individuals. Above all, it is important that the actual effect of ER is determined in such a research setting. For this purpose there would have to be another group that solves programming tasks without robots. Moreover, the study should also include students of different ages. This was also planned for the research presented in this thesis but could not be completed because of the COVID19 measures. We expect that, more exact statements regarding the comparison of graphical vs. textual programming could be made so.

Appendix

Appendix A

Test of the Group Using Graphical Programming(German)

A.1 Computational Thinking

Abschnitt 1 von 4

Abschlusstestung Robotik - Graphisch

Vortestung für den graphischer Teil

ID *

Kurzantwort-Text

Geschlecht *

- weiblich
- männlich

Klasse *

- 3.Klasse
- 5.Klasse
- Testklasse

Abschnitt 2 von 4

Technisches

In diesem Abschnitt werden dir einige Verständnisfragen zum Thema Robotik gestellt.

Im Grund gibt es zwei Typen von Fragen:
*) Typ1: Die Frage wird dir in Form einer Aufgabenstellung gestellt. Dies kann ein Bild aber auch ein Text sein. Du musst per Multiple Choice aus 4 Antwortmöglichkeiten (A, B, C, D) die passende auswählen.
*) Typ2: Hier ist der Code gegeben und du musst herausfinden, was dieser Code bewirkt. Hier musst du per Multiple Choice aus bis zu 4 Antwortmöglichkeiten die passende auswählen.

Viel Spaß und gutes Gelingen!

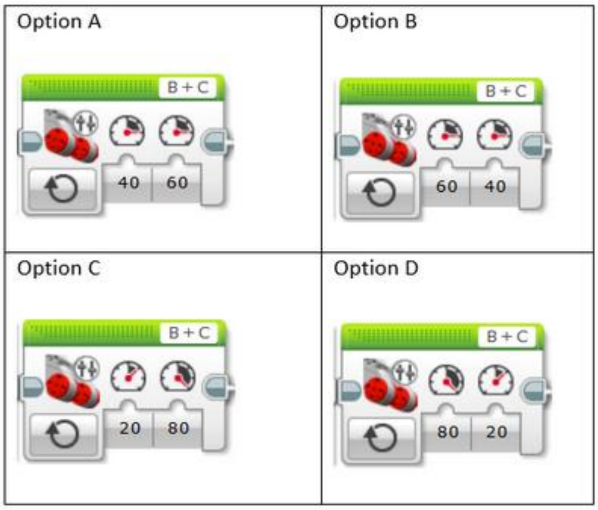
Frage 1 *

Welcher der unten abgebildeten Codeabschnitte löst die folgende Aufgabenstellung? Dein Roboter soll eine gewisse Distanz vorwärtsfahren, dann anhalten. Anschließend soll er nur die halbe Strecke wieder rückwärtsfahren und anhalten.

- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 2 *

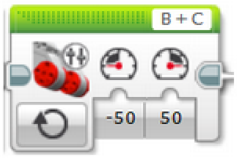
Dein Roboter soll nun eine möglichst scharfe Linkskurve machen. Welche der unten dargestellten Optionen entspricht der schärfsten Linkskurve? WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!



- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 3 *

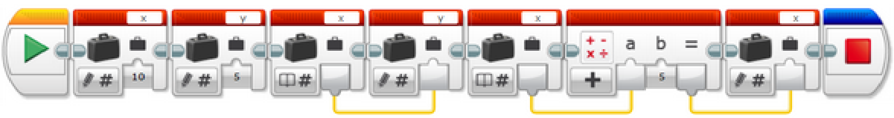
Was bewirkt der rechts dargestellte Baustein? WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!



- Der Roboter fährt einen Kreis. Der Mittelpunkt ist dabei der linke Reifen.
- Der Roboter dreht sich im Uhrzeigersinn um die eigene Achse.
- Der Roboter fährt einen Kreis. Der Mittelpunkt ist dabei der rechte Reifen.
- Der Roboter dreht sich gegen den Uhrzeigersinn um die eigene Achse.

Frage 4 *

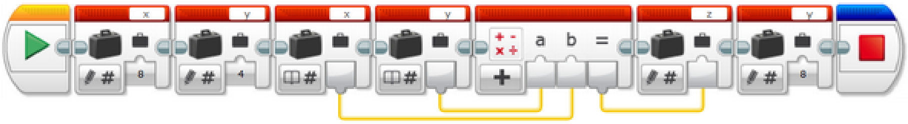
Betrachte das Programm in der unteren Abbildung. Welchen Wert haben die Variablen x und y nach Ablauf des Programmes?



- x = 15; y =15
- x = 5; y = 10
- x = 15; y =10
- x = 10; y =5

Frage 5 *

Betrachte das Programm in der unteren Abbildung. Welchen Wert haben die Variablen x,y und z nach Ablauf des Programmes?

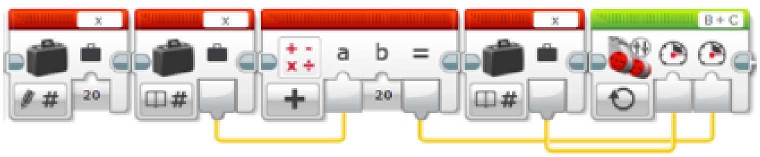


- ◯ x = 8; y = 4; z = 84
- ◯ x = 8; y = 8; z = 16
- ◯ x = 8; y = 8; z = 12
- ◯ x = 8; y = 4; z = 12

Frage 6 *

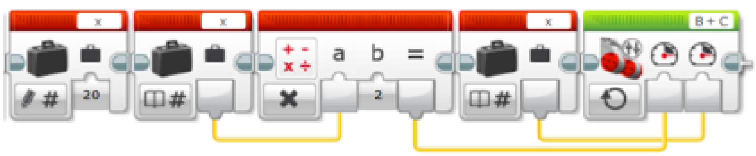
Die beliebige Geschwindigkeit des linken Rades wird in einer Variable x gespeichert. Das rechte Rad soll sich immer doppelt so schnell drehen wie das linke Rad. Welcher Option löst dieses Problem? Denke daran, dass die Geschwindigkeit x beliebig gewählt werden kann. Das Programm soll nie angepasst werden müssen und soll das Problem allgemein lösen! WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!

Option A

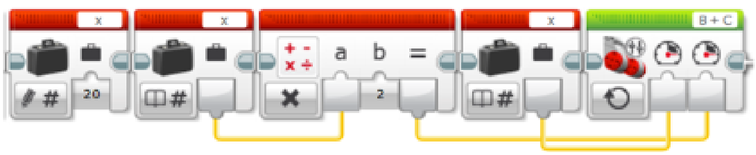


Option B

Option C



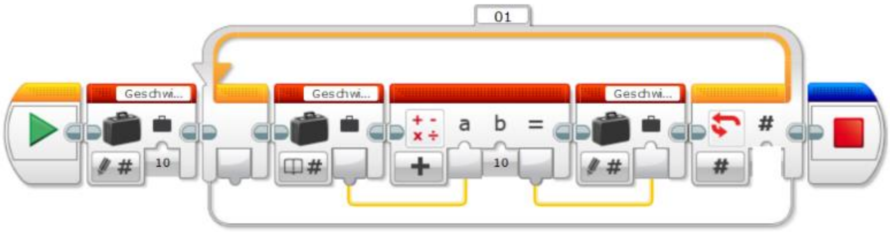
Option D



- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 7 *

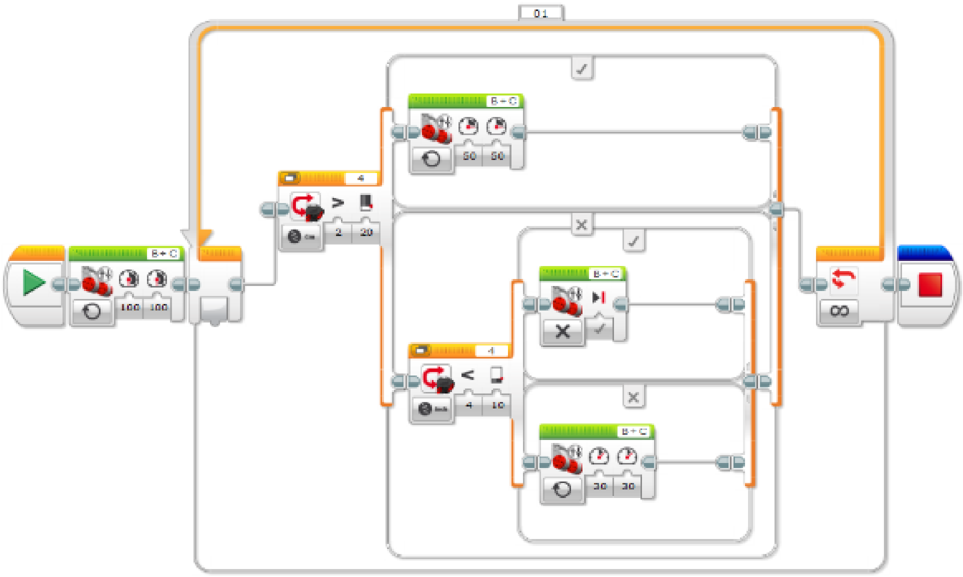
Im folgenden Programm wird innerhalb der Schleife die Variable Geschwindigkeit immer um 10 erhöht. Wie oft muss die Schleife mindestens ausgeführt werden, damit die Variable Geschwindigkeit den Wert 100 erreicht?



- 10 mal
- 11 mal
- 9 mal
- Die richtige Antwort ist nicht dabei!

Frage 8 *

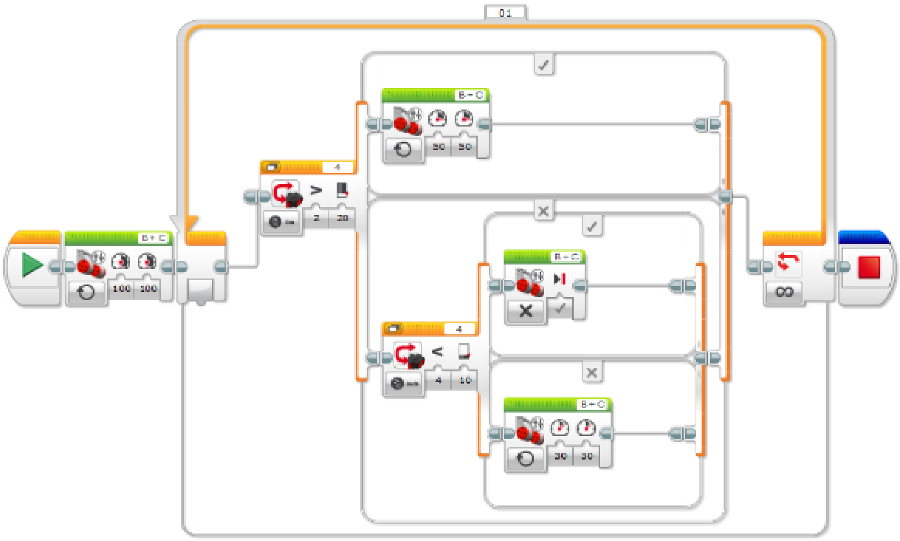
Mithilfe eines Ultraschallsensors, welcher an Port 4 angeschlossen ist, soll abhängig vom Abstand zur einem Hindernis (z.B. Mauer) die Geschwindigkeit des Roboters reguliert werden. Betrachte dazu das unten dargestellte Programm. Nimm an, dass der Roboter noch 10 cm von der Mauer entfernt ist. Wie schnell fährt der Roboter zu diesem Zeitpunkt?



- ◯ 100
- ◯ 50
- ◯ 30
- ◯ 0 bzw. Stopp

Frage 9 *

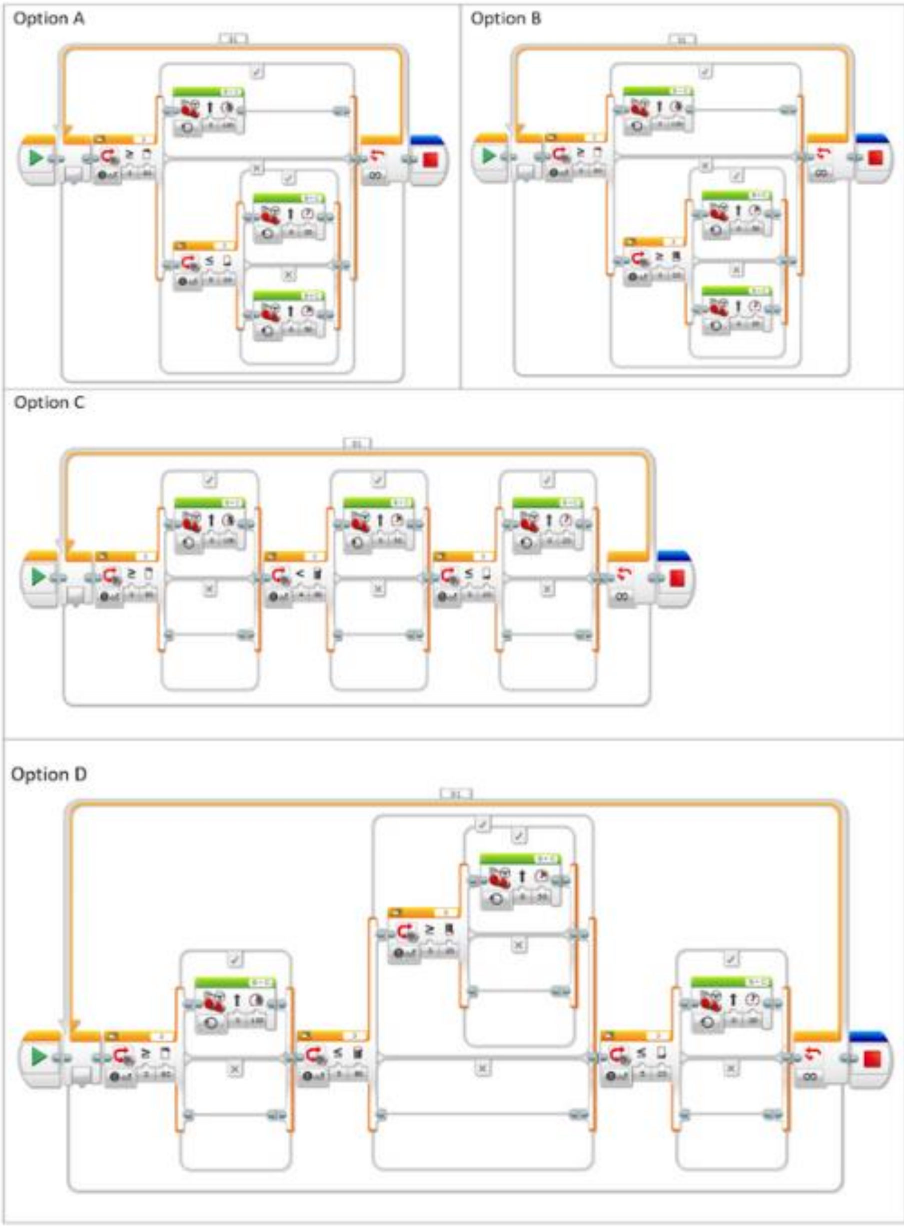
Mithilfe eines Ultraschallsensors, welcher an Port 4 angeschlossen ist, soll abhängig vom Abstand zu einem Hindernis (z.B. Mauer) die Geschwindigkeit des Roboters reguliert werden. Betrachte dazu das unten dargestellte Programm. Wie oft bzw. wie lange wird die Schleife ausgeführt?



- Die Schleife wird beendet, wenn der Abastand < 10 cm ist
- Die Schleife wird beendet, wenn der Roboter stoppt.
- Die Schleife wird unendlich lange ausgeführt.
- Die Schleife wird 8-mal durchlaufen.

Frage 10 *

Du sollst ein Programm entwickeln, sodass der Roboter abhängig von der Reflexion eine gewisse Geschwindigkeit fährt: Ist die Reflexionsstärke größer gleich 80 dann soll die Geschwindigkeit 100 sein; ist die Reflexionsstärke zwischen 80 und 20 dann soll die Geschwindigkeit 50 sein; ist die Reflexionsstärke kleiner gleich 20 dann soll Geschwindigkeit 20! Nimm an, dass der Lichtsensor an Port 3 angeschlossen ist.



◯ Option A

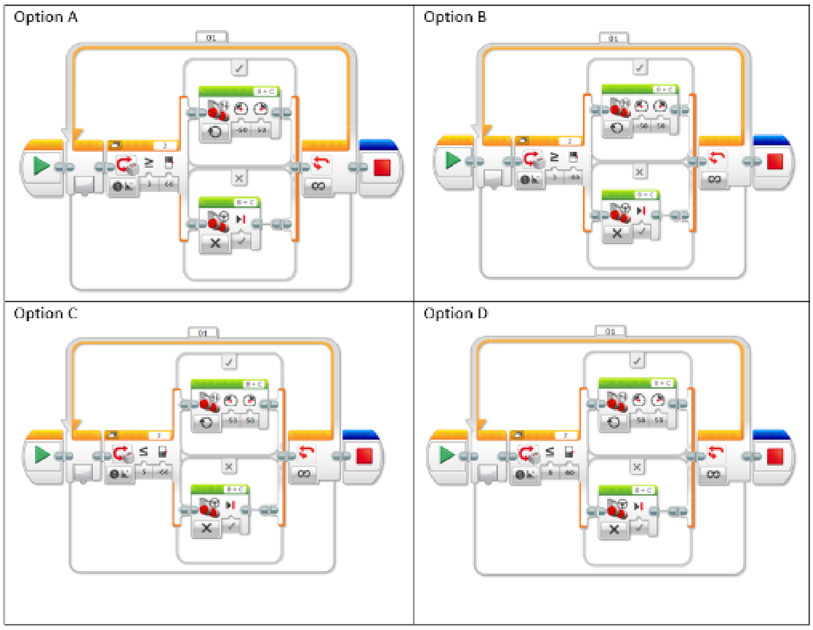
◯ Option B

◯ Option C

◯ Option D

Frage 11 *

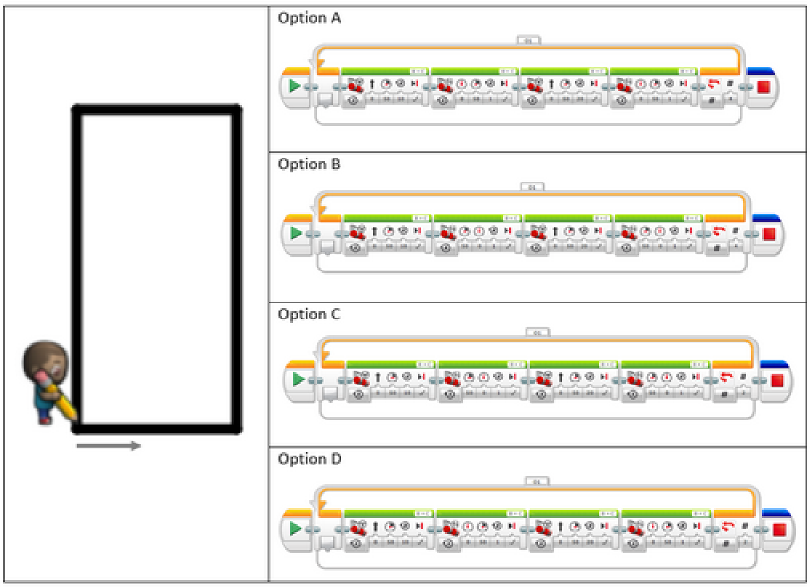
Der Roboter soll sich am Stand um 60° Grad nach links drehen. Welches der folgende Programme bzw. Optionen erfüllt diese Aufgabe korrekt? Der Drehsensor ist an Port 2 angeschlossen.



- Option A
- Option B
- Option C
- Option D

Frage 12 *

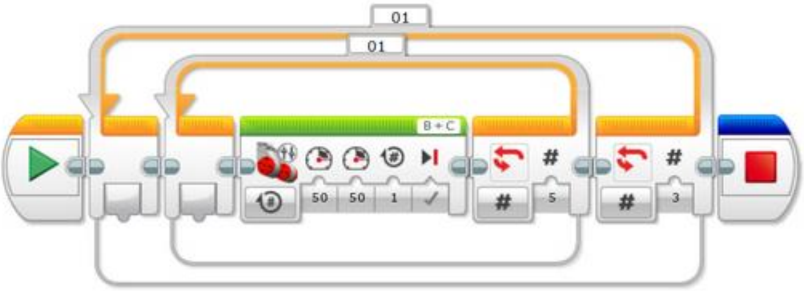
Dein Roboter wird durch die Figur links in der Abbildung dargestellt. Ziel ist es, dass dein Roboter die Figur (Rechteck) korrekt abfährt. Das Rechteck hat eine Höhe von 100 cm und eine Breite von 50 cm. Nimm an, dass dein Roboter mit einer Radumdrehung 5 cm zurücklegt. Wir nehmen auch an, dass die Kombination (50,0) bzw. (0,50) einem perfekten rechten Winkel entspricht. Ferner ist B der Linke und C der rechte Motor! Welche Option musst du wählen, damit dein Roboter das Rechteck korrekt abfährt?



- Option A
- Option B
- Option C
- Option D

Frage 13 *

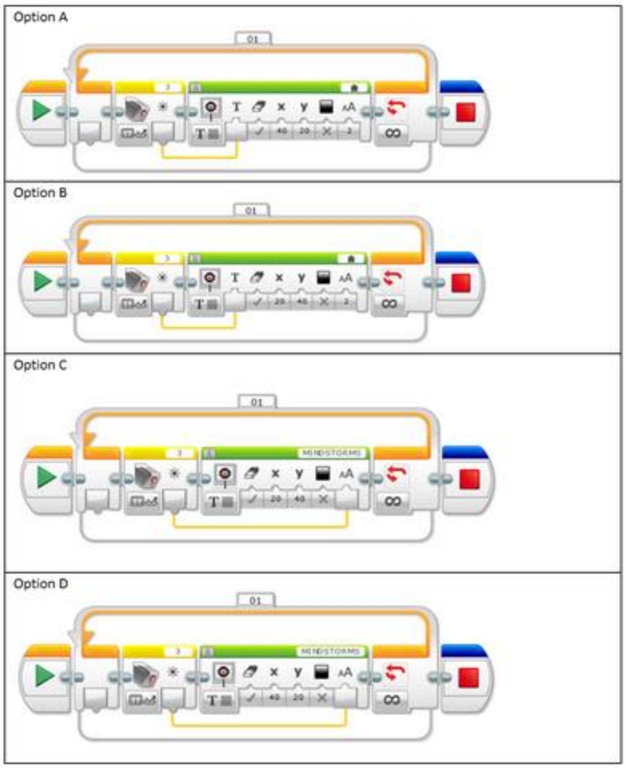
Betrachte das nebenstehende Programm. Nimm an, dass eine Umdrehung vorwärts einer Distanz von 1 cm entspricht. Wie weit fährt der Roboter nach vorne bis das Programm beendet wird?



- 5 cm
- Er bleibt nie stehen.
- 15 cm
- 8 cm

Frage 14

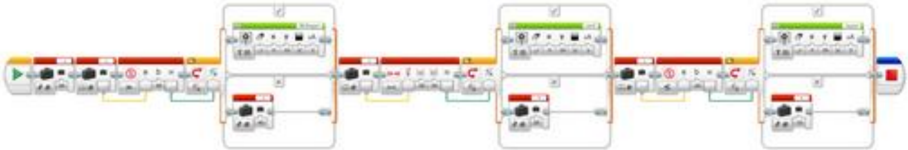
Ziel der Aufgabe ist es, dass am Bildschirm des Roboters die gemessene Lichtstärke ausgegeben wird. Dies soll in einem Abstand von 20 Pixel zur oberen Kante und in einem Abstand von 40 Pixel zur linken Kante des Bildschirm erfolgen. Welche Option liefert die gewünschte Ausgabe?



- Option A
- Option B
- Option C
- Option D

Frage 15 *

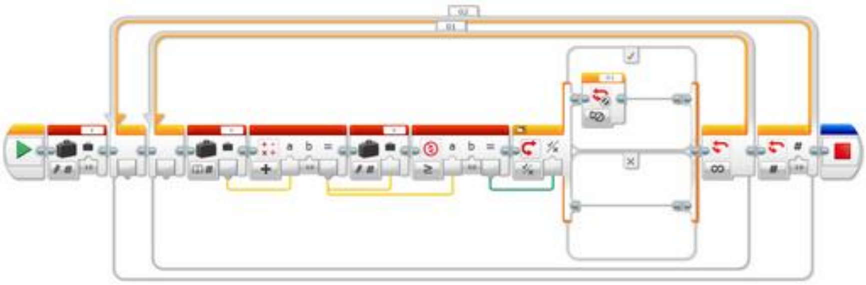
Was steht nach Ablauf des Programmes am Bildschirm des Roboter und welchen Wert hat die Variable x am Ende?



- "Abfragen sind super" und x = 60
- "Abfragen super" und x = 10
- "Abfragen" und x = 10
- "Abfragen super" und x = 20

Frage 16 *

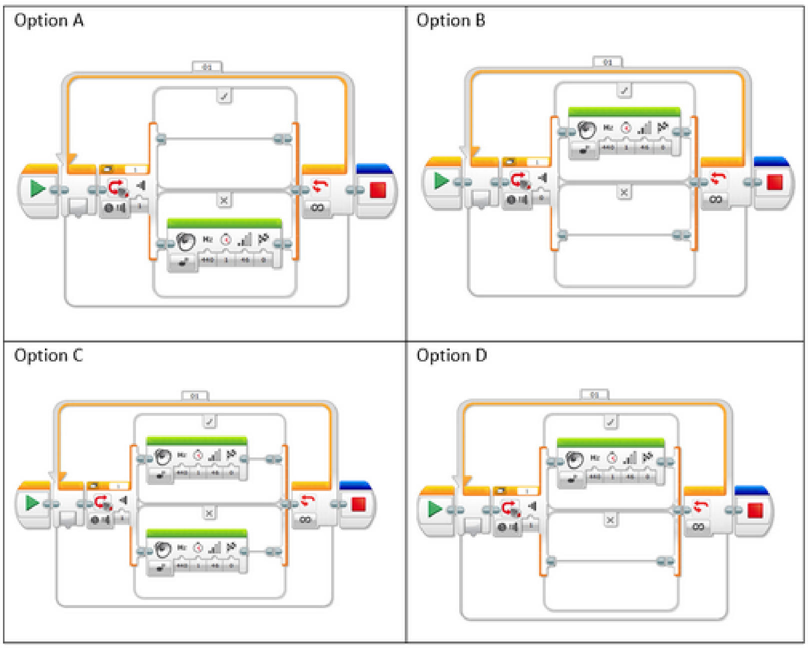
Welchen Wert hat die Variable x nach Ablauf des Programmes?



- 140
- 50
- 10
- 150

Frage 17 *

Ziel ist es, dass bei Berührung des Touch-Sensors(Port 1) ein akkustisches Singal ertönt solange der Sensor gedrückt ist. Wird er losgelassen, hört das Signal wieder auf. Welche Option löst das Problem?



- Option A
- Option B
- Option C
- Option D

Abschnitt 3 von 4

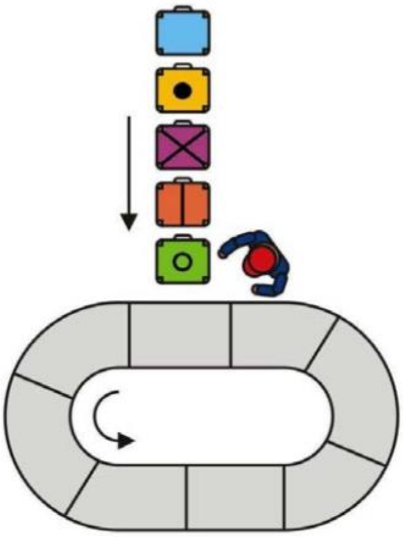
Lösen von Problemen

In diesem Abschnitt wird gemessen, wie gut deine Fähigkeiten im Rahmen der Problemlösung sind. Dies erfolgt einerseits auf Basis einer Selbsteinschätzung und andererseits auf Basis von Verständnisfragen. Kreuze bei den Selbsteinschätzungsfragen an in welchem Ausmaß die Aussage auf dich zutrifft!

A.2 Problem Solving

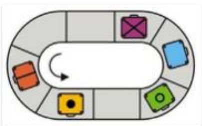
Frage1 *

Das Förderband des Flughafens hat 8 Plätze und es dreht sich im Kreis (in Pfeilrichtung). Ein Arbeiter legt 5 Koffer der Reihe nach auf das Förderband. Er legt den nächsten Koffer immer auf den drittnächsten leeren Platz. Er lässt also die schon belegten Plätze und auch zwei leere Plätze vorbeidrehen. Der Arbeiter ist fertig, wenn alle 5 Koffer auf dem Förderband liegen. Wie schaut das Förderband am Ende seiner Arbeit aus?

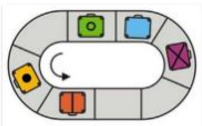


Option A

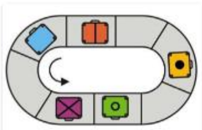
Option B



Option C

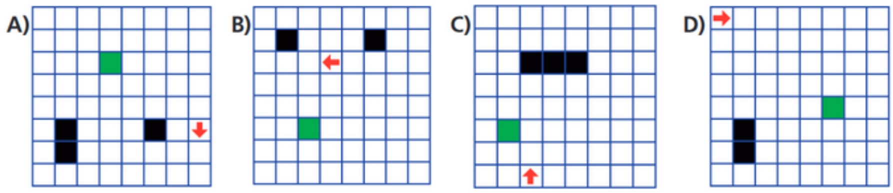


Option D



Frage 2 *

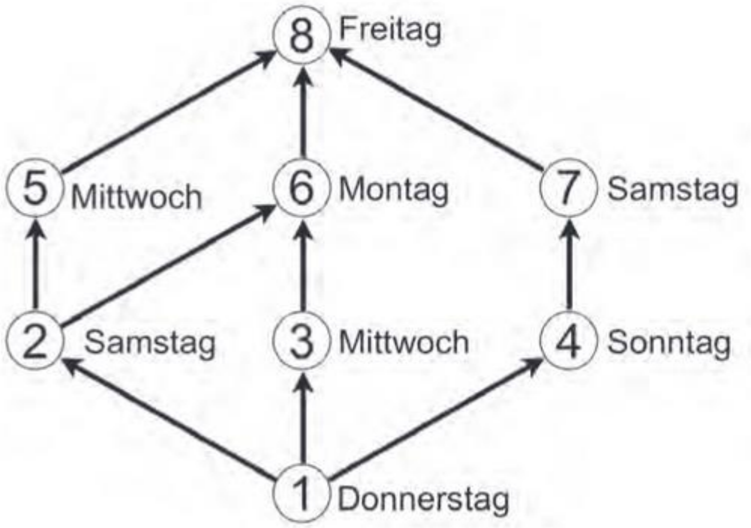
Unser Roboter soll auf einem Spielbrett sein Ziel erreichen: das grüne Feld. Das Feld mit dem Pfeil ist sein Startfeld. Die schwarzen Felder sind Hindernisse. Der Roboter ist so programmiert: Er bewegt sich in Richtung des Pfeils geradeaus, bis er auf ein Hindernis oder den Spielbrettrand trifft. Dann dreht er sich um 90 Grad nach rechts und bewegt sich wieder geradeaus so weit es geht, usw. Jedes Feld, über das der Roboter kommt, wird sofort zu einem weiteren Hindernis, auch das Startfeld. Auf welchem Spielbrett erreicht der Roboter NICHT sein Ziel?



- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 3 *

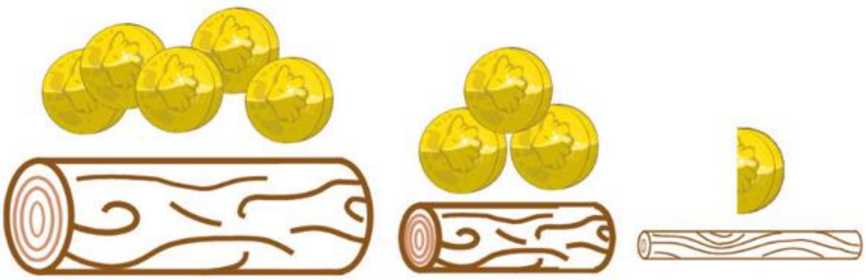
Im Wilden Westen, wo die Cowboy-Biber leben, hat die Bebras Stagecoach Company Postkutschenlinien zwischen acht Siedlungen (1 bis 8) eingerichtet. Im Fahrplan steht neben jeder Siedlung, an welchem Wochentag eine Postkutsche abfährt. Sie fährt jeweils frühmorgens los und erreicht am Abend desselben Tages die nächste Siedlung. Fährt die Kutsche also z.B. vom Punkt 7 ab, so erreicht sie am Samstag Abend Punkt 8. Der Freitag wäre hier nur der Tag der Abfahrt. Über welche Route wird ein Paket am schnellsten von Siedlung 1 nach Siedlung 8 transportiert?



- 1 - 2 - 5 - 8
- 1 - 2 - 6 - 8
- 1 - 3 - 6 - 8
- 1 - 4 - 7 - 8

Frage 4 *

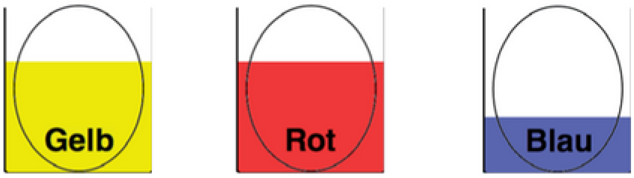
Benny soll Hölzer aus dem Wald holen. Sie werden für den Damm gebraucht. Für schwere Hölzer mit 3 Kilo Gewicht bekommt Benny am meisten ausbezahlt, nämlich 5 Münzen. Für mittelschwere Hölzer mit 2 Kilo Gewicht bekommt er 3 Münzen. Und leichte Hölzer mit 1 Kilo Gewicht sind nur eine halbe Münze wert. Benny kann nur einmal in den Wald gehen und nicht mehr als 7 Kilo tragen. Welche Hölzer wird Benny holen, damit er möglichst viele Münzen verdient?



- Ein schweres Holz und zwei mittelschwere Hölzer.
- Zwei schwere Hölzer und ein mittelschweres Holz.
- Drei mittelschwere Hölzer und ein leichtes Holz.
- Ein schweres Holz und ein mittelschweres Holz und zwei leichte Hölzer.

Frage 5 *

Lina färbt weiße Eier und benutzt dazu drei Becher mit Farben. Die Becher mit Gelb und Rot sind so voll, dass ein Ei zu zwei Dritteln in die Farbe eintauchen kann. Vom Blau ist weniger da, so dass ein Ei nur zu einem Drittel eintauchen kann. Lina taucht die Eier immer so tief wie möglich ein. Lina mischt gern zwei der Grundfarben: Gelb und Rot zu Orange, Rot und Blau zu Violett, Blau und Gelb zu Grün. Nie mischt sie mehr als zwei Farben. Wenn Lina z.B. ein Ei erst in Rot und dann in Blau taucht, es dann umdreht und wieder in Blau taucht, erhält sie ein violett-rot-blau gefärbtes Ei. Von den unteren Eiern kann nur eines von Lina gefärbt worden sein. Welches?



○ Option A

○ Option B

○ Option C

○ Option D

Frage 6 *

Ich verwende ein schrittweises Vorgehen um ein Problem zu lösen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 7 *

Bevor ich ein Problem löse mach ich einen Plan.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 8 *

Ich verwende neue Methoden zur Problemlösung, wenn alte nicht funktionieren.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 9 *

Ich analysiere ein Problem genau bevor ich es versuche zu lösen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

...

Frage 10 *

Um ein komplexes Problem zu lösen unterteile ich es in mehrere Teilprobleme.

- Starke Zustimming
- Zustimmung
- Unsicher
- Ablehung
- Starke Ablehnung

Abschnitt 4 von 4

Soziales

In diesem Abschnit musst du dich selbst einschätzen. Kreuze jeweils an in welchem Ausmaß die Aussage auf dich zutrifft.

A.3 Social Skills

Frage 1 *

Ich höre gerne auf andere, wenn ich versuche Ansätze zur Lösung eines Problems oder einer Aufgabe zu finden.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehung
- ◯ Starke Ablehung

Frage 2 *

Ich bin gerne Teil eines Teams welches versucht ein Problem zu lösen.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehung

Frage 3 *

Wenn ich im Team arbeite frage ich meine TeamkollegInnen um Hilfe, wenn ich nicht weiter weiß oder etwas nicht verstanden habe.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehung
- ◯ Starke Ablehung

Frage 4 *

Ich arbeite gerne mit anderen zusammen um Aufgaben zu lösen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 5 *

Es fällt mir leicht FreundInnen von meiner Meinung zu überzeugen

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehung

Frage 6 *

Es fällt mir leicht neue FreundInnen zu finden.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 7 *

Es fällt mir leicht FreundInnen um Hilfe bzw. einen Gefallen zu Fragen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehung

Frage 8 *

Wie schwer oder einfach ist es für dich mit anderen Menschen auszukommen?

- Sehr leicht
- Leicht
- Unsicher
- Schwer
- Sehr schwer

Appendix B

Test of the Group Using Textual Programming(German)

B.1 Computational Thinking

Abschnitt 1 von 4

Abschlusstestung Robotik - Textuell

Abschlusstestung für den textuellen Teil

ID *

Kurzantwort-Text

Geschlecht *

- weiblich
- männlich

Klasse *

- 3.Klasse
- 5.Klasse
- Testklasse

Abschnitt 2 von 4

Technisches

In diesem Abschnitt werden dir einige Verständnisfragen zum Thema Robotik gestellt.

Im Grund gibt es zwei Typen von Fragen:
*) Typ1: Die Frage wird dir in Form einer Aufgabenstellung gestellt. Dies kann ein Bild aber auch ein Text sein. Du musst per Multiple Choice aus 4 Antwortmöglichkeiten (A, B, C, D) die passende auswählen.
*) Typ2: Hier ist der Code gegeben und du musst herausfinden, was dieser Code bewirkt. Hier musst du per Multiple Choice aus bis zu 4 Antwortmöglichkeiten die passende auswählen.

Viel Spaß und gutes Gelingen!

Frage 1 *

Welcher der unten abgebildeten Codeabschnitte löst die folgende Aufgabenstellung? Dein Roboter soll eine gewisse Distanz vorwärtsfahren, dann anhalten. Anschließend soll er nur die halbe Strecke wieder rückwärtsfahren und anhalten.

Option A

```
Motor.Start("BC",50)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",50)
Program.Delay(2000)
Motor.Stop("BC","True")
```

Option B

```
Motor.Start("BC",50)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(2000)
Motor.Stop("BC","True")
```

Option C

```
Motor.Start("BC",50)
Program.Delay(2000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(4000)
Motor.Stop("BC","True")
```

Option D

```
Motor.Start("BC",100)
Program.Delay(4000)
Motor.Stop("BC","True")
Motor.Start("BC",-50)
Program.Delay(4000)
Motor.Stop("BC","True")
```

- ○ Option A
- ○ Option B
- ○ Option C
- ○ Option D

Frage 2 *

Dein Roboter soll nun eine möglichst scharfe Linkskurve machen. Welche der unten dargestellten Optionen entspricht der schärfsten Linkskurve? WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!

Option A `Motor.StartSync("BC",60,40)`	Option B `Motor.StartSync("BC",40,60)`
Option C `Motor.StartSync("BC",20,80)`	Option D `Motor.StartSync("BC",80,20)`

- ○ Option A
- ○ Option B
- ○ Option C
- ○ Option D

Frage 3 *

Was bewirkt der rechts dargestellte Baustein? WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!

```
Motor.StartSync("BC",-50,50)
```

- ○ Der Roboter fährt einen Kreis. Der Mittelpunkt ist dabei der linke Reifen.
- ○ Der Roboter dreht sich im Uhrzeigersinn um die eigene Achse.
- ○ Der Roboter fährt einen Kreis. Der Mittelpunkt ist dabei der rechte Reifen.
- ○ Der Roboter dreht sich gegen den Uhrzeigersinn um die eigene Achse.

Frage 4 *

Betrachte das Programm in der unteren Abbildung. Welchen Wert haben die Variablen x und y nach Ablauf des Programmes?

```
x = 10
y = 5
y = x
x = x + 5
```

- x = 15; y =15
- x = 5; y = 10
- x = 15; y =10
- x = 10; y =5

Frage 5 *

Betrachte das Programm in der unteren Abbildung. Welchen Wert haben die Variablen x,y und z nach Ablauf des Programmes?

```
x = 8
y = 4
z = x + y
y = 8
```

- x = 8; y = 4; z = 84
- x = 8; y = 8; z = 16
- x = 8; y = 8; z = 12
- x = 8; y = 4; z = 12

Frage 6 *

Die beliebige Geschwindigkeit des linken Rades wird in einer Variable x gespeichert. Das rechte Rad soll sich immer doppelt so schnell drehen wie das linke Rad. Welcher Option löst dieses Problem? Denke daran, dass die Geschwindigkeit x beliebig gewählt werden kann. Das Programm soll nie angepasst werden müssen und soll das Problem allgemein lösen! WICHTIG: Nimm an, dass der Motor des LINKEN REIFENS an PORT B angeschlossen ist und jener des RECHTEN REIFENS an PORT C!

Option A

```
x = 20
y = x + 20
Motor.StartSync("BC",x,y)
```

Option B

```
y = x * 2
x = 20
Motor.StartSync("BC", x, y)
```

Option C

```
x = 20
y = x * 2
Motor.StartSync("BC", y, x)
```

Option D

```
x = 20
y = x * 2
Motor.StartSync("BC", x, y)
```

- Option A
- Option B
- Option C
- Option D

Frage 7 *

Im folgenden Programm wird innerhalb der Schleife die Variable Geschwindigkeit immer um 10 erhöht. Wie oft muss die Schleife mindestens ausgeführt werden, damit die Variable Geschwindigkeit den Wert 100 erricht wird?

```
Geschwindigkeit = 10
For counter = 1 To ???
  Geschwindigkeit = Geschwindigkeit + 10
EndFor
```

- 10 mal
- 11 mal
- 9 mal
- Die richtige Antwort ist nicht dabei!

Frage 8 *

Mithilfe eines Ultraschallsensors, welcher an Port 4 angeschlossen ist, soll abhängig vom Abstand zur einem Hindernis (z.B. Mauer) die Geschwindigkeit des Roboters reguliert werden. Betrachte dazu das unten dargestellte Programm. Nimm an, dass der Roboter noch 10 cm von der Mauer entfernt ist. Wie schnell fährt der Roboter zu diesem Zeitpunkt?

```
Sensor.SetMode(4,0)
Motor.Start("BC",100)
While "True"
  Abstand = Sensor.ReadRawValue(4,0)
  If Abstand > 200 Then
    Motor.Start("BC", 50)
  Else
    If Abstand < 100 Then
      Motor.Stop("BC","True")
    Else
      Motor.Start("BC",30)
    EndIf
  EndIf
EndWhile
```

- ○ 100
- ○ 50
- ○ 30
- ○ 0 bzw. Stopp

Frage 9 *

Mithilfe eines Ultraschallsensors, welcher an Port 4 angeschlossen ist, soll abhängig vom Abstand zu einem Hindernis (z.B. Mauer) die Geschwindigkeit des Roboters reguliert werden. Betrachte dazu das unten dargestellte Programm. Wie oft bzw. wie lange wird die Schleife ausgeführt?

```
Sensor.SetMode(4,0)
Motor.Start("BC",100)
While "True"
  Abstand = Sensor.ReadRawValue(4,0)
  If Abstand > 200 Then
    Motor.Start("BC", 50)
  Else
    If Abstand < 100 Then
      Motor.Stop("BC","True")
    Else
      Motor.Start("BC",30)
    EndIf
  EndIf
EndWhile
```

- ◯ Die Schleife wird beendet, wenn der Abstand < 10 cm ist
- ◯ Die Schleife wird beendet, wenn der Roboter stoppt.
- ◯ Die Schleife wird unendlich lange ausgeführt.
- ◯ Die Schleife wird 8-mal durchlaufen.

...

Frage 10 *

Du sollst ein Programm entwickeln, sodass der Roboter abhängig von der Reflexion eine gewisse Geschwindigkeit fährt: Ist die Reflexionsstärke größer gleich 80 dann soll die Geschwindigkeit 100 sein; ist die Reflexionsstärke zwischen 80 und 20 dann soll die Geschwindigkeit 50 sein; ist die Reflexionsstärke kleiner gleich 20 dann soll die Geschwindigkeit 20 sein! Nimm an, dass der Lichtsensor an Port 3 angeschlossen ist.

Option A

```
Sensor.SetMode(3,0)
While "True"
  Reflexion = Sensor.ReadRawValue(3,0)
  If Reflexion >= 80 Then
    Motor.Start("BC",100)
  Else
    If Reflexion <= 20 Then
      Motor.Start("BC",20)
    Else
      Motor.Start("BC",50)
    EndIf
  EndIf
EndWhile
```

Option B

```
Sensor.SetMode(3,0)
While "True"
  Reflexion = Sensor.ReadRawValue(3,0)
  If Reflexion >= 80 Then
    Motor.Start("BC",100)
  Else
    If Reflexion >= 20 Then
      Motor.Start("BC",50)
    Else
      Motor.Start("BC",20)
    EndIf
  EndIf
EndWhile
```

Option C

```
Sensor.SetMode(3,0)
While "True"
  Reflexion = Sensor.ReadRawValue(3,0)
  If Reflexion >=80 Then
    Motor.Start("BC",100)
  EndIf
  If Reflexion < 80 Then
    Motor.Start("BC",50)
  EndIf
  If Reflexion <= 20 Then
    Motor.Start("BC",20)
  EndIf
EndWhile
```

Option D

```
Sensor.SetMode(3,0)
While "True"
  Reflexion = Sensor.ReadRawValue(3,0)
  If Reflexion >=80 Then
    Motor.Start("BC",100)
  EndIf
  If Reflexion <= 80 Then
    If Reflexion >=20 Then
      Motor.Start("BC",50)
    Endif
  EndIf
  If Reflexion <= 20 Then
    Motor.Start("BC",20)
  Endif
EndWhile
```

- ○ Option A
- ○ Option B
- ○ Option C
- ○ Option D

Frage 11 *

Der Roboter soll sich am Stand um 60° Grad nach links drehen. Welches der folgende Programme bzw. Optionen erfüllt diese Aufgabe korrekt? Der Drehsensor ist an Port 2 angeschlossen.

Option A

```
Sensor.SetMode(2,0)
Motor.Start("BC",100)
While "True"
  Winkel = Sensor.ReadRawValue(2,0)
  If Winkel >= 60 Then
    Motor.StartSync("BC",-50,50)
  Else
    Motor.Stop("BC","True")
  EndIf
EndWhile
```

Option B

```
Sensor.SetMode(2,0)
Motor.Start("BC",100)
While "True"
  Winkel = Sensor.ReadRawValue(2,0)
  If Winkel >= -60 Then
    Motor.StartSync("BC",-50,50)
  Else
    Motor.Stop("BC","True")
  EndIf
EndWhile
```

Option C

```
Sensor.SetMode(2,0)
Motor.Start("BC",100)
While "True"
  Winkel = Sensor.ReadRawValue(2,0)
  If Winkel <= -60 Then
    Motor.StartSync("BC",-50,50)
  Else
    Motor.Stop("BC","True")
  EndIf
EndWhile
```

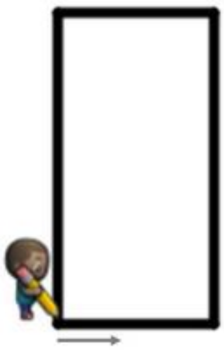
Option D

```
Sensor.SetMode(2,0)
Motor.Start("BC",100)
While "True"
  Winkel = Sensor.ReadRawValue(2,0)
  If Winkel <= 60 Then
    Motor.StartSync("BC",-50,50)
  Else
    Motor.Stop("BC","True")
  EndIf
EndWhile
```

- ○ Option A
- ○ Option B
- ○ Option C
- ○ Option D

Frage 12 *

Dein Roboter wird durch die Figur links in der Abbildung dargestellt. Ziel ist es, dass dein Roboter die Figur (Rechteck) korrekt abfährt. Das Rechteck hat eine Höhe von 50 cm und eine Breite von 25 cm. Nimm an, dass ein Delay von 1000ms einer Distanz von 5 cm entspricht. Wir nehmen auch an, dass die Kombination (50,0) bzw. (0,50) einem perfekten rechten Winkel entspricht. Darüberhinaus ist B der linke und C der rechte Motor! Welche Option musst du wählen, damit dein Roboter das Rechteck korrekt abfährt?



Option A

```
For counter = 1 To 4
  Motor.Start("BC", 50)
  Program.Delay(5000)
  Motor.StartSync("BC",0,50)
  Program.Delay(1000)
  Motor.Start("BC", 50)
  Program.Delay(10000)
  Motor.StartSync("BC",0,50)
  Program.Delay(1000)
EndFor
```

Option B

```
For counter = 1 To 4
  Motor.Start("BC", 50)
  Program.Delay(5000)
  Motor.StartSync("BC",50,0)
  Program.Delay(1000)
  Motor.Start("BC", 50)
  Program.Delay(10000)
  Motor.StartSync("BC",50,0)
  Program.Delay(1000)
EndFor
```

Option C

```
For counter = 1 To 2
  Motor.Start("BC", 50)
  Program.Delay(5000)
  Motor.StartSync("BC",50,0)
  Program.Delay(1000)
  Motor.Start("BC", 50)
  Program.Delay(10000)
  Motor.StartSync("BC",50,0)
  Program.Delay(1000)
EndFor
```

Option D

```
For counter = 1 To 2
  Motor.Start("BC", 50)
  Program.Delay(5000)
  Motor.StartSync("BC",0,50)
  Program.Delay(1000)
  Motor.Start("BC", 50)
  Program.Delay(10000)
  Motor.StartSync("BC",0,50)
  Program.Delay(1000)
EndFor
```

- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 13

Betrachte das nebenstehende Programm. Nimm an, dass während einer Unterbrechung von 1000ms der Roboter 1 cm vorwärts fährt. Wie weit fährt der Roboter nach vorne bis das Programm beendet wird?

```
For Counter= 1 To 3
  For Counter= 1 To 5
    Motor.Start("BC",50)
    Program.Delay(1000)
  EndFor
EndFor
```

- ◯ 5 cm
- ◯ Er bleibt nie stehen.
- ◯ 15 cm
- ◯ 8 cm

Frage 14 *

Ziel der Aufgabe ist es, dass am Bildschirm des Roboters die gemessene Lichtstärke ausgegeben wird. Dies soll in einem Abstand von 20 Pixel zur oberen Kante und in einem Abstand von 40 Pixel zur linken Kante des Bildschirm erfolgen. Welche Option liefert die gewünschte Ausgabe?

Option A

```
Sensor.SetMode(3,0)
While "True"
  LCD.Clear()
  Reflexion = Sensor.ReadRawValue(3,0)
  LCD.Write(40,20,Reflexion)
  Program.Delay(500)
EndWhile
```

Option B

```
Sensor.SetMode(3,0)
While "True"
  LCD.Clear()
  Reflexion = Sensor.ReadRawValue(3,0)
  LCD.Write(20,40,Reflexion)
  Program.Delay(500)
EndWhile
```

Option C

```
Sensor.SetMode(3,0)
While "True"
  LCD.Clear()
  Reflexion = Sensor.ReadRawValue(3,0)
  LCD.Write(Reflexion,40,20)
  Program.Delay(500)
EndWhile
```

Option D

```
Sensor.SetMode(3,0)
While "True"
  LCD.Clear()
  Reflexion = Sensor.ReadRawValue(3,0)
  LCD.Write(Reflexion,20,40)
  Program.Delay(500)
EndWhile
```

- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Frage 15 *

Was steht nach Ablauf des Programmes am Bildschirm des Roboters und welchen Wert hat die Variable x am Ende?

```
x = 65
LCD.Clear()
If x > 50 Then
  LCD.Write(0,10,"Abfragen")
Else
  x = 40
EndIf

If x >= 25 Then
  If x <= 50 Then
    LCD.Write(0,20,"sind")
  Else
    x = 20
  EndIf
EndIf

If x <= 20 Then
  LCD.Write(0,30,"super")
Else
  x = 10
EndIf
```

- ◯ "Abfragen sind super" und x = 60
- ◯ "Abfragen super" und x = 10
- ◯ "Abfragen" und x = 10
- ◯ "Abfragen super" und x = 20

Frage 16 *

Welchen Wert hat die Variable x nach Ablauf des Programmes?

```
x = 10
For counter = 1 To 10
  run = 1
  While run = 1
    x = x + 10
    If x >=50 Then
      run = 0
    EndIf
  EndWhile
EndFor
```

- ◯ 140
- ◯ 50
- ◯ 10
- ◯ 150

Frage 17 *

Ziel ist es, dass bei Berührung des Touch-Sensors(Port 1) ein akkustisches Signal ertönt solange der Sensor gedrückt ist. Wird er losgelassen hört das Signal wieder auf. Welche Option löst das Problem?

Option A

```
Sensor.SetMode(1,0)
While "true"
  Touch = Sensor.ReadPercent(1)
  If Touch = 100 then

  Else
    Speaker.Tone(46,440,1000)
  Endif
EndWhile
```

Option B

```
Sensor.SetMode(1,0)
While "true"
  Touch = Sensor.ReadPercent(1)
  If Touch = 0 then
    Speaker.Tone(46,440,1000)
  Else

  Endif
EndWhile
```

Option C

```
Sensor.SetMode(1,0)
While "true"
  Touch = Sensor.ReadPercent(1)
  If Touch = 0 then
    Speaker.Tone(46,440,1000)
  Else
    Speaker.Tone(46,440,1000)
  Endif
EndWhile
```

Option D

```
Sensor.SetMode(1,0)
While "true"
  Touch = Sensor.ReadPercent(1)
  If Touch = 100 then
    Speaker.Tone(46,440,1000)
  Else

  Endif
EndWhile
```

- ○ Option A
- ○ Option B
- ○ Option C
- ○ Option D

Abschnitt 3 von 4

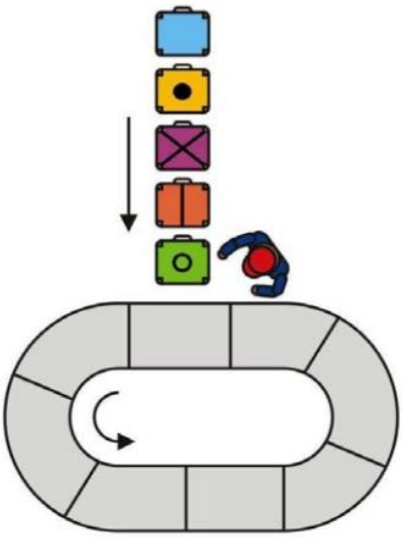
Lösen von Problemen

In diesem Abschnitt wird gemessen, wie gut deine Fähigkeiten im Rahmen der Problemlösung sind. Dies erfolgt einerseits auf Basis einer Selbsteinschätzung und andererseits auf Basis von Verständnisfragen. Kreuze bei den Selbsteinschätzungsfragen an in welchem Ausmaß die Aussage auf dich zutrifft!

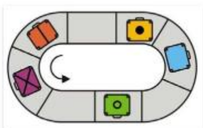
B.2 Problem Solving

Frage1 *

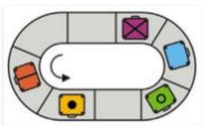
Das Förderband des Flughafens hat 8 Plätze und es dreht sich im Kreis (in Pfeilrichtung). Ein Arbeiter legt 5 Koffer der Reihe nach auf das Förderband. Er legt den nächsten Koffer immer auf den drittnächsten leeren Platz. Er lässt also die schon belegten Plätze und auch zwei leere Plätze vorbeidrehen. Der Arbeiter ist fertig, wenn alle 5 Koffer auf dem Förderband liegen. Wie schaut das Förderband am Ende seiner Arbeit aus?



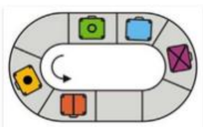
Option A



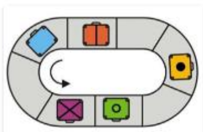
Option B



Option C

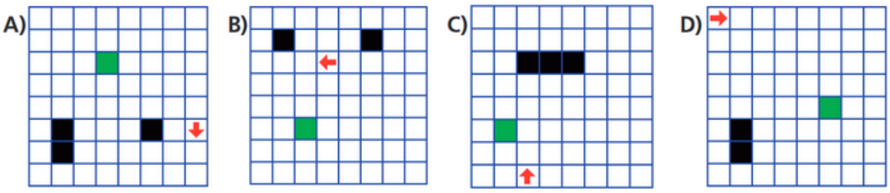


Option D



Frage 2 *

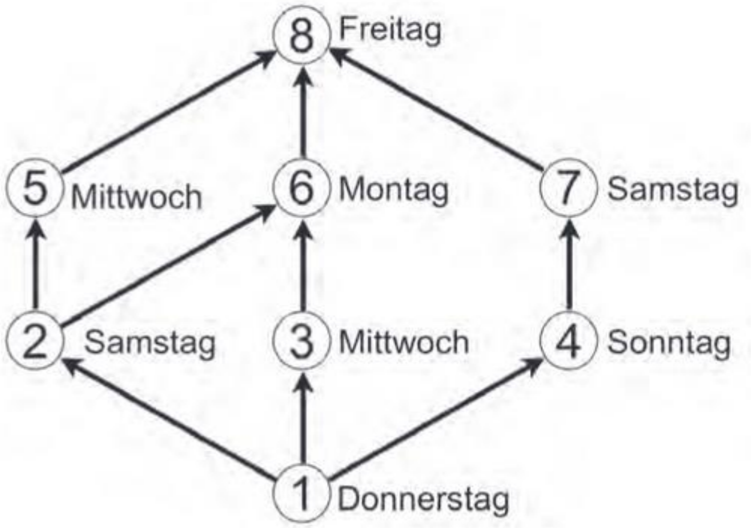
Unser Roboter soll auf einem Spielbrett sein Ziel erreichen: das grüne Feld. Das Feld mit dem Pfeil ist sein Startfeld. Die schwarzen Felder sind Hindernisse. Der Roboter ist so programmiert: Er bewegt sich in Richtung des Pfeils geradeaus, bis er auf ein Hindernis oder den Spielbrettrand trifft. Dann dreht er sich um 90 Grad nach rechts und bewegt sich wieder geradeaus so weit es geht, usw. Jedes Feld, über das der Roboter kommt, wird sofort zu einem weiteren Hindernis, auch das Startfeld. Auf welchem Spielbrett erreicht der Roboter NICHT sein Ziel?



- Option A
- Option B
- Option C
- Option D

Frage 3 *

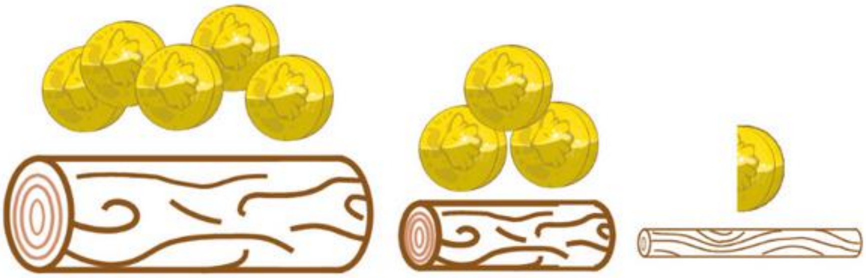
Im Wilden Westen, wo die Cowboy-Biber leben, hat die Bebras Stagecoach Company Postkutschenlinien zwischen acht Siedlungen (1 bis 8) eingerichtet. Im Fahrplan steht neben jeder Siedlung, an welchem Wochentag eine Postkutsche abfährt. Sie fährt jeweils frühmorgens los und erreicht am Abend desselben Tages die nächste Siedlung. Fährt die Kutsche also z.B. vom Punkt 7 ab, so erreicht sie am Samstag Abend Punkt 8. Der Freitag wäre hier nur der Tag der Abfahrt. Über welche Route wird ein Paket am schnellsten von Siedlung 1 nach Siedlung 8 transportiert?



- 1 - 2 - 5 - 8
- 1 - 2 - 6 - 8
- 1 - 3 - 6 - 8
- 1 - 4 - 7 - 8

Frage 4 *

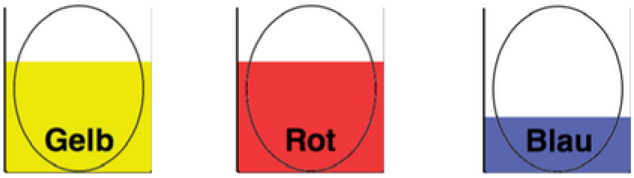
Benny soll Hölzer aus dem Wald holen. Sie werden für den Damm gebraucht. Für schwere Hölzer mit 3 Kilo Gewicht bekommt Benny am meisten ausbezahlt, nämlich 5 Münzen. Für mittelschwere Hölzer mit 2 Kilo Gewicht bekommt er 3 Münzen. Und leichte Hölzer mit 1 Kilo Gewicht sind nur eine halbe Münze wert. Benny kann nur einmal in den Wald gehen und nicht mehr als 7 Kilo tragen. Welche Hölzer wird Benny holen, damit er möglichst viele Münzen verdient?



- ○ Ein schweres Holz und zwei mittelschwere Hölzer.
- ○ Zwei schwere Hölzer und ein mittelschweres Holz.
- ○ Drei mittelschwere Hölzer und ein leichtes Holz.
- ○ Ein schweres Holz und ein mittelschweres Holz und zwei leichte Hölzer.

Frage 5 *

Lina färbt weiße Eier und benutzt dazu drei Becher mit Farben. Die Becher mit Gelb und Rot sind so voll, dass ein Ei zu zwei Dritteln in die Farbe eintauchen kann. Vom Blau ist weniger da, so dass ein Ei nur zu einem Drittel eintauchen kann. Lina taucht die Eier immer so tief wie möglich ein. Lina mischt gern zwei der Grundfarben: Gelb und Rot zu Orange, Rot und Blau zu Violett, Blau und Gelb zu Grün. Nie mischt sie mehr als zwei Farben. Wenn Lina z.B. ein Ei erst in Rot und dann in Blau taucht, es dann umdreht und wieder in Blau taucht, erhält sie ein violett-rot-blau gefärbtes Ei. Von den unteren Eiern kann nur eines von Lina gefärbt worden sein. Welches?



- Option A

- Option B

- Option C

- Option D

Frage 6 *

Ich verwende ein schrittweises Vorgehen um ein Problem zu lösen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 7 *

Bevor ich ein Problem löse mach ich einen Plan.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 8 *

Ich verwende neue Methoden zur Problemlösung, wenn alte nicht funktionieren.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehnung

Frage 9 *

Ich analysiere ein Problem genau bevor ich es versuche zu lösen.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehnung

Frage 10 *

Um ein komplexes Problem zu lösen unterteile ich es in mehrere Teilprobleme.

- ◯ Starke Zustimming
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehung
- ◯ Starke Ablehnung

Abschnitt 4 von 4

Soziales

In diesem Abschnit musst du dich selbst einschätzen. Kreuze jeweils an in welchem Ausmaß die Aussage auf dich zutrifft.

B.3 Social Skills

Frage 1 *

Ich höre gerne auf andere, wenn ich versuche Ansätze zur Lösung eines Problems oder einer Aufgabe zu finden.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehung
- ◯ Starke Ablehung

Frage 2 *

Ich bin gerne Teil eines Teams welches versucht ein Problem zu lösen.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehung

Frage 3 *

Wenn ich im Team arbeite frage ich meine TeamkollegInnen um Hilfe, wenn ich nicht weiter weiß oder etwas nicht verstanden habe.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehung
- ◯ Starke Ablehung

Frage 4 *

Ich arbeite gerne mit anderen zusammen um Aufgaben zu lösen.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehnung

Frage 5 *

Es fällt mir leicht FreundInnen von meiner Meinung zu überzeugen

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehung

Frage 6 *

Es fällt mir leicht neue FreundInnen zu finden.

- ◯ Starke Zustimmung
- ◯ Zustimmung
- ◯ Unsicher
- ◯ Ablehnung
- ◯ Starke Ablehnung

Frage 7 *

Es fällt mir leicht FreundInnen um Hilfe bzw. einen Gefallen zu Fragen.

- Starke Zustimmung
- Zustimmung
- Unsicher
- Ablehnung
- Starke Ablehung

Frage 8 *

Wie schwer oder einfach ist es für dich mit anderen Menschen auszukommen?

- Sehr leicht
- Leicht
- Unsicher
- Schwer
- Sehr schwer

Appendix C

Executive Summary

Executive Summary

The present curriculum is intended to ensure continuous education in the field of robotics for children between 10 and 17 years of age. The curriculum should also be a template for other schools. A grammar school in Austria, the BG/BRG/BORG Köflach, is a pilot school for the use of the curriculum. The contents of robotics will be integrated into the computer science lessons of this school.

One goal of the curriculum is a continuous development of skills in the field of robotics. A focus is also placed on developmental psychology as well as previous knowledge of research. This should ensure that learning contents do not overstrain students and taught at the right time. The learning plan in mathematics was used, among other things, as a benchmark. Another goal of the curriculum is to get as many students as possible interested in robotics and computer science, especially in the beginning. In the following years, especially in the last two years, the focus is more on the talented students. In addition, the curriculum is designed to teach important skills such as computational thinking, problem solving and social skills. For two school levels (years 7 and 8) an increase in performance in these areas could be demonstrated with the learning contents of the curricula.

The two tables on the following pages give an overview of the learning content, goals, technologies used, and the calculated amount of time per school grade.

grade/ age	contents	technologies	learning objectives	amount of time
year 5 / age 10	-) Making first experiences with robots. -) Observe the behavior of the robot. -) Using color codes to solve tasks with increasing difficulty.	Ozobots with color sensors	-) Students first fears of contact with robots are taken away. -) Students know important color codes and the basic behavior of the robot. -) Students combine color codes to solve simple tasks.	2 school hours of 50 min
year 6 / age 11	-) Get to know alternative forms of programming in the form of blocks. -) Transferring programs to the Ozobot. -) Online provided tasks are solved with Ozoblockly. -) Students solve physical tasks (on paper) like the "Slot Car Challenge" and "Find the black Box" with Ozoblockly.	Ozobots with Ozoblockly	-) Students know that there is an alternative programming language in the form of blocks. -) Students know how to transfer a program to the Ozobot. -) Students can solve practical tasks of varying degrees of difficulty using Ozoblockly.	2-3 school hours of 50 min
year 7 / age 12	-) Theoretical basics and application areas of robotics, as well as historical information about robotics. -) Theoretical information about the Lego Mindstorm EV3 like background knowledge about the brick and knowledge about the function of the sensors. -) Programming using a graphical language. Solving tasks with an increasing degree of difficulty. -) Application of all available sensors in different practical situations. -) Students learn basic programming concepts such as variables and control structures.	LEGO Mindstorm EV3 with EV3-G	-) Students know what a robot is, where they are used and roughly their development path. -) Students know how each sensor works and can use it in different practical situations. -) Students can solve practical tasks of varying degrees of difficulty using a graphical programming language. -) Students know the most important concepts of programming such as variables and control structures.	9 - 10 school hours of 2x50 min
year 8 / age 13	-) Students learn the same concepts as in the previous school year. This time they work with a textual programming language. -) Furthermore, tasks of the KNAPP Robo League are solved.	LEGO Mindstorm EV3 with EV3-Basic	-) Students now using familiar methods of solution within the framework of a textual programming language. -) Students solve complex competition tasks using a textual programming language and LEGO Mindstorm EV3 (-) Students take part in the KNAPP Robo League competition.)	12 school hours of 2x50 min, >12 school hours of 2x50 min for students who participate at the competition (students join robotic club)

year 9 / age 14	-) Theoretical basics about the microcontroller and in particular the Ardunino UNO. -) Students get to know the development environment and take their first steps in the Arduino C language. -) Students work with different sensors from the experimental box and solve different application tasks in the fields of measuring, control and regulation.	-) Arduino UNO with experimental box and Arduino C	-) Students know what a microcontroller is and where it is used. They have knowledge about the Arduino UNO, such as its construction. -) Students can solve various practical application tasks in the fields of measuring, control and regulation with sensors and use suitable components of the experimental box. They are also able to establish connections to the subject of physics.	6 school hours of 2x50 min
year 10 / age 15	-) Introduction to the Smart v3.0 Robotic Car in combination with the Arduino UNO. -) Solving different motion tasks with the help of the car and ArduinoC. -) Working with the ultrasonic and line tracking sensor (with (PID-algorithm) in the context of practical tasks	Smart v3.0 Robot Car with ArduinoUNO and ArduinoC.	-) Students know the construction of the Smart v3.0 Robtic Car including the most important connections. -) Students perform basic movement with the car using ArduinoC. -) Students use the available sensors to solve practical tasks such as tracking a black line and know the concepts used.	6 school hours of 2x50 min
year 11 / age 16	-) Students learn the most important basics about artificial intelligence, such as what artificial intelligence is and where it is used. -) Students get to know different chatbots and program their own chatbot in succession. -) Students connect the AI with robotics. With the help of an EV3, the fastest way out of a maze is determined	LEGO Mindstorm EV3 with EV3-Baisc for the maze and the GaitoBot for programming the ChatBot.	-) Students can define what artificial intelligence is and know important areas of application. -) Students know what a chatbot is and how it works. -) Students independently solve a project on the topic of machine learning in connection with robotics.	8 school hours of 2x50 min
year 12 /age 17	-) Getting to know different robot gripper arms. -) Acquisition of theoretical (mathematical) basics necessary for programming such as matrices, homogenous transformations, translated frame base (TFB) approach and the geometrical model. -) Solving practical tasks using the programming language Python in a given framework.	Python and framework of EDLRIS- Course	-) Students know how gripper arms of robot's work and how they are constructed. -) Students know theoretical basics to be able to solve and understand practical tasks. -) Students know the most important structures and commands of the programming language Python to be able to solve tasks of the given framework in connection with the learned theories.	8 school hours of 2x50 min
year 8 – 12/ age 13 – 17 robotic club	-) In the free subject "Robotics Club" students are specifically prepared for competitions. These are for example the KNAPP Robo League (year 8) or the Robo Cup Junior (year 9 - 12).	LEGO Mindstorm EV3 with EV3-Basic	-) Students succeed in solving the problems of the competitions independently and take part in competitions in consequence.	2x 50 min every two weeks for one school year

Bibliography

Absolventa (2019) Soziale Kompetenz: Für diese Jobs brauchst du Teamfähigkeit und Co. URL https://www.absolventa.de/karriereguide/kommunikation/soziale-kompetenz, [Online; Last call 02.August 2019]

Ainslee J (2018) Digitization Of Education In The 21st Century. URL https://elearningindustry.com/digitization-of-education-21st-century, [Online; Last call 11.July 2019]

Barker BS, Ansorge J (2007) Robotics as means to increase achievement scores in an informal learning environment. Journal of research on technology in education 39(3):229–243

Bayrisches Staatsministerium für Unterricht und Kultus (2000) Methodiküberlegungen für den mathematisch-naturwissenschaftlichen Unterricht. Wiederholenals bewusstes Unterrichtselement. URL https://www.isb.bayern.de/download/6767/wiederholen_als_bewusstes_unterrichtselement.pdf

Bedford, Mike and Schischka, Sabine (2018) Programmieren lernen: Das sind die besten Sprachen. URL https://www.pcwelt.de/ratgeber/Programmieren-lernen-Das-sind-die-besten-Sprachen-9958738.html, [Online; Last call 05.August 2019]

Benitti FBV (2012) Exploring the educational potential of robotics in schools: A systematic review. Computers & Education 58(3):978–988

Biber der Informatik (2009) Biber der Informatik 2009- Aufgaben und Lösungen. URL https://www.ocg.at/sites/ocg.at/files/medien/pdfs/BiberAufgaben2009.pdf, [Online; Last call 14.August 2020]

Biber der Informatik (2011) Biber der Informatik 2011- Aufgaben und Lösungen. URL https://www.ocg.at/sites/ocg.at/files/medien/pdfs/BiberAufgaben2011.pdf, [Online; Last call 14.August 2020]

Biber der Informatik (2012) Biber der Informatik 2012- Aufgaben und Lösungen. URL https://www.ocg.at/sites/ocg.at/files/medien/pdfs/Biber-Aufgaben2012-mitLoesungen-AT-web.pdf, [Online; Last call 14.August 2020]

Biber der Informatik (2013) Biber der Informatik 2013- Aufgaben und Lösungen. URL https://www.ocg.at/sites/ocg.at/files/medien/pdfs/Biber2013_Aufgaben-Loesungen.pdf, [Online; Last call 14.August 2020]

Blanca MJ, Alarcón R, Arnau J, Bono R, Bendayan R (2017) Non-normal data: Is ANOVA still a valid option? Psicothema 29(4):552–557

Booth T, Stumpf S (2013) End-user experiences of visual and textual programming environments for arduino. In: Dittrich Y, Burnett M, Mørch A, Redmiles D (eds) End-User Development, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 25–39

Brandhofer G, Baumgartner P, Ebner M, Koeberer C Nina Trueltzsch-Wijnen, Wiesner C (2018) Bildung im Zeitalter der Digitalisierung. Nationaler Bildungsbericht 2018 pp 307–362, DOI 10.17888/nbb2018-2-8, URL https://www.bifie.at/wp-content/uploads/2019/03/NBB_2018_Band2_Beitrag_8.pdf

British Columbia - Ministery of Education (2018a) Applied Design, Skills, and Technologies— Electronics and Robotics. URL https://curriculum.gov.bc.ca/sites/curriculum.gov.bc.ca/files/curriculum/adst/en_adst_10_electronics-and-robotics_elab.pdf, [Online; Last call 06.August 2020]

British Columbia - Ministery of Education (2018b) Applied Design, Skills, and Technologies— Electronics and Robotics. URL https://curriculum.gov.bc.ca/sites/curriculum.gov.bc.ca/files/curriculum/adst/en_adst_11_robotics_elab.pdf, [Online; Last call 06.August 2020]

British Columbia - Ministery of Education (2018c) Applied Design, Skills, and Technologies— Electronics and Robotics. URL https://curriculum.gov.bc.ca/sites/curriculum.gov.bc.ca/files/curriculum/adst/en_adst_12_robotics_elab.pdf, [Online; Last call 06.August 2020]

British Columbia - Ministery of Education (2019) Applied Design, Skills, and Technologies K-9 – Content. URL https://curriculum.gov.bc.ca/sites/curriculum.gov.bc.ca/files/curriculum/continuous-views/en_ADST_k-9_content.pdf, [Online; Last call 28.July 2020]

British Columbia - Ministery of Education (2020a) British Columbia's Education Program. URL https://www2.gov.bc.ca/assets/gov/education/administration/kindergarten-to-grade-12/diverse-student-needs/educationwelcomeletter_jan2016_english_final.pdf, [Online; Last call 28.July 2020]

British Columbia - Ministery of Education (2020b) Introduction to Applied Design, Skills, and Technologies. URL https://curriculum.gov.bc.ca/curriculum/adst/core/introduction, [Online; Last call 28.July 2020]

BSS Australian Capital Territory (2019) Robotics & Mechatronics A/T/M/V. Accredited from 2020 - 2024. URL http://www.bsss.act.edu.au/curriculum/courses, [Online; Last call 25.July 2020]

Cavas B, İsmail Kesercioğlu T, Holbrook JB, Rannikmae M, Ozdoğru E, Gokler F (2012) The Effects of Robotics Club on the Students' Performance on Science Process & Scientific Creativity Skills and Perceptions on Robots, Human and Society

Citrin W, Doherty M, Zorn B (1995) The design of a completely visual object-oriented programming language. Visual Object-Oriented Programming: Concepts and Environments Prentice-Hall, New York

codakid (2020) Top 9 Kids Coding Languages of 2020. URL https://codakid.com/top-7-kids-coding-languages-of-2018/, [Online; Last call 16.November 2020]

Cohen J (1988) Statistical power analysis for the behavioral sciences. Academic press

Constantinou V, Ioannou A (2018) Development of Computational Thinking Skills through Educational Robotics. In: EC-TEL (Practitioner Proceedings)

Council NR (2012) A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. The National Academies Press, Washington, DC, DOI 10.17226/13165, URL https://www.nap.edu/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts

Dai L, Martins P (2020) Does vocational education pay off in China? Instrumental-variable quantile-regression evidence, DOI 10.13140/RG.2.2.21715.32807

Deitelhoff F (2017) Programmiersprachen für LEGO-Robotik-Sets. URL https://www.brickobotik.de/programmiersprachen-fuer-lego-robotik-sets/, [Online; Last call 10.August 2020]

Department of Education Austria (2015) Gesamte Rechtsvorschrift für Lehrpläne der Höheren technischen und gewerblichen Lehranstalten 2015 sowie Bekanntmachung der Lehrpläne für den Religionsunterricht, Fassung vom 21.08.2020. URL https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20009288, [Online; Last call 21.August 2020]

Department of Education Austria (2017a) Mathematik. URL https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568&FassungVom=2017-08-31, [Online; Last call 21.August 2020]

Department of Education Austria (2017b) Physik. URL https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568&FassungVom=2017-08-31, [Online; Last call 21.August 2020]

Department of Education Austria (2019) Digitale Grundbildung. URL https://bildung.bmbwf.gv.at/schulen/schule40/dgb/index.html, [Online; Last call 11.July 2019]

ebotics (2019) What is Educational Robotics? URL https://www.ebotics.com/what-is-educational-robotics/, [Online; Last call 02.August 2019]

Eguchi A (2014a) Learning experience through RoboCupJunior: Promoting STEM education and 21st century skills with robotics competition. In: Society for Information Technology & Teacher Education International Conference, Association for the Advancement of Computing in Education (AACE), pp 87–93

Eguchi A (2014b) Robotics as a learning tool for educational transformation. In: Proceeding of 4th international workshop teaching robotics, teaching with robotics & 5th international conference robotics in education Padova (Italy)

Eguchi A (2016) RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. Robotics and Autonomous Systems 75:692–699

Engineering^3 (2020) E3 FAQs. URL https://engineering3.org/about/e3-team/, [Online; Last call 21.July 2020]

Engineering^3 FAQ (2020) E3 faqs. URL https://engineering3.org/e3-faqs/, [Online; Last call 21.July 2020]

EV3 Basic (2020) EV3Basic- Introduction . URL https://sites.google.com/site/ev3basic/, [Online; Last call 10.August 2020]

Fojtik R (2017) The Ozobot and education of programming. New Trends and Issues Proceedings on Humanities and Social Sciences 4, DOI 10.18844/prosoc.v4i5.2666

Franklin D, Skifstad G, Rolock R, Mehrotra I, Ding V, Hansen A, Weintrop D, Harlow D (2017) Using Upper-Elementary Student Performance to Understand Conceptual Sequencing in a Blocks-based Curriculum. pp 231–236, DOI 10.1145/3017680.3017760

Fredericks A (2010) The Teacher's Handbook: Strategies for Success. R&L Education, URL https://books.google.at/books?id=10RqquryDUIC

French JH, Crouse H (2018) Using Early Intervention to Increase Female Interest in Computing Sciences. J Comput Sci Coll 34(2):133–140

Futschek G (2016) Computational Thinking im Unterricht. Schule Aktiv 10

García-Peñalvo F, Gerrard A, Hughes J, Jormanainen I, Toivonen T, Vermeersch J (2016) A survey of resources for introducing coding into schools. pp 19–26, DOI 10.1145/3012430.3012491

Geier G, Ebner M (2017) Einsatz von OZOBOTs zur informatischen Grundbildung. Erziehung & Unterricht 167:109–113

Guter Unterricht (2020) Unterrichtsphasen. URL https://www.guterunterricht.de/unterrichtsphasen, [Online; Last call 20.August 2020]

Hill School (2020) Team RoboBlues. URL https://www.thehill.org/academics/academic-departments/quadrivium-engineering-and-design/team-roboblues/robocup-blog-clone, [Online; Last call 21.July 2020]

HTL Kaindorf (2020) Wahlfach Artifical Intelligence & Visualisation. URL https://www.htl-kaindorf.at/abteilungen/informatik/artificial-intelligence, [Online; Last call 21.August 2020]

Hunsaker E (2018) Ozobot Bit. URL https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?filename=6&article=1007&context=ipt_projects&type=additional

Hussain S, Lindh J, Shukur G (2006) The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. Journal of Educational Technology & Society 9(3):182–194

Ioannou A, Makridou E (2018) Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. Education and Information Technologies 23(6):2531–2544, DOI 10.1007/s10639-018-9729-z, URL https://doi.org/10.1007/s10639-018-9729-z

Jang Y, Lee W, Kim J (2015) Assessing the Usefulness of Object-based Programming Education Using Arduino. Indian Journal of Science and Technology 8:89, DOI 10.17485/ijst/2015/v8iS1/57701

Jump TE (2015) Engineering[3]. URL http://e3textbook-v5index-jspid-20213498-lng-en.engineering3.org/

JUNIOR L, GUERRA F, BRAVO L, Hernandez M, Neto O, Martins P (2013) A Low-Cost and Simple Arduino-Based Educational Robotics Kit. Journal of Selected Areas in Robotics and Control (JSRC) 3:1–12

Kan Q (2019) A brief introduction to the Chinese education system . URL https://www.open.edu/openlearn/education/brief-introduction-the-chinese-education-system, [Online; Last call 27.July 2020]

Kandlhofer M (2017) Development and evaluation of innovative concepts in educational robotics and education in artificial intelligence. PhD thesis

Kandlhofer M, Steinbauer G, Lassing J, Menzinger M, Baumann W, Ehardt-Schmiederer M, Bieber R, Winkler T, Plomer S, Strobl-Zuchtriegl I, Miglbauer M, Ballagi A, Pozna C, Miltenyi G, Alfoldi I, Szalay I (????) EDLRIS: A European Driving License for Robots and Intelligent Systems, not puphlished on 21.November 2020

Köberl L (2020a) Einführung. URL http://msr.leo-edv.com/index.php?option=com_content&view=article&id=169&Itemid=2, [Online; Last call 21.August 2020]

Köberl L (2020b) Thermischer Regelkreislauf. URL http://msr.leo-edv.com/index.php?option=com_content&view=article&id=97&Itemid=82, [Online; Last call 30.August 2020]

Kepler IT (2020) BRG Kepler Informationstechnologie. URL http://www.brgkepler.at/~crlo_it/, [Online; Last call 25.July 2020]

Kepler Robotik (2020) RoboCup Erfolge. URL http://www.keplerrobotik.at/www/index.php?seite=erfolge, [Online; Last call 25.July 2020]

KNAPP (2020) Teilnahmebedingungen. URL https://www.roboleague.at/teilnahmebedingungen/, [Online; Last call 20.August 2020]

LEGO (2020) Roboter für Kinder. URL https://www.lego.com/de-at/categories/robots-for-kids, [Online; Last call 19.August 2020]

LEGO Education (2020) LEGO MINDSTORMS Education EV3. URL https://education.lego.com/de-de/product/mindstorms-ev3, [Online; Last call 10.August 2020]

Lucas B, Claxton G, Hanson J (2014) Thinking Like an Engineer: Implications for the education system.

Maheshwari, VK (2017) The Problem –solving Method in Education. URL http://www.vkmaheshwari.com/WP/?p=2375, [Online; Last call 21.July 2019]

Marcelo Rovai (2020) Maze Solver Robot, Using Artificial Intelligence With Arduino. URL https://www.instructables.com/id/Maze-Solver-Robot-Using-Artificial-Intelligence-Wi/, [Online; Last call 30.August 2020]

Martín-Ramos P, Lopes MJ, Lima da Silva MM, Gomes PE, Pereira da Silva PS, Domingues JP, Ramos Silva M (2017) First exposure to arduino through peer-coaching: Impact on students' attitudes towards programming. Computers in Human Behavior 76:51 – 58, DOI https://doi.org/10.1016/j.chb.2017.07.007, URL http://www.sciencedirect.com/science/article/pii/S0747563217304193

May B (2020) Programmieren für Kinder – 11 tolle Ideen. URL https://www.codingkids.de/machen/programmieren-fuer-kinder-11-tolle-ideen#codeorg, [Online; Last call 19.August 2020]

Mayerová K, Kubincová Z, Veselovská M (2019) Creating Activities for After School Robotic Workshop with Ozobot Evo. In: 2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET), pp 1–5

Merkel P (2020) Visual Programming im Überblick. URL http://www.plippo.de/dwl/study/pit-ausarbeitung.pdf, [Online; Last call 16.November 2020]

Norbert-Brendan K, Cristian Marius T (2019) Autonomous Line Maze Solver Using Artificial Intelligence. In: 2019 15th International Conference on Engineering of Modern Electric Systems (EMES), pp 133–136

Nourbakhsh IR, Crowley K, Bhave A, Hamner E, Hsiu T, Perez-Bergquist A, Richards S, Wilkinson K (2005) The robotic autonomy mobile robotics course: Robot design, curriculum design and educational assessment. Autonomous Robots 18(1):103–127

Nuffic (2018) The Australian education system described and compared with the Dutch system. URL https://www.nuffic.nl/en/publications/education-system-australia/, [Online; Last call 22.July 2020]

Nugent G, Barker B, Grandgenett N, Adamchuk V (2009) The use of digital manipulatives in k-12: robotics, GPS/GIS and programming. In: 2009 39th IEEE Frontiers in Education Conference, IEEE, pp 1–6

Nugent G, Barker B, Grandgenett N, Adamchuk V (2010) Impact of Robotics and Geospatial Technology Interventions on Youth STEM Learning and Attitudes. Journal of Research on Technology in Education 42:391–408, DOI 10.1080/15391523.2010.10782557

Owens G, Granader Y, Humphrey A, Baron-Cohen S (2008) Lego®therapy and the social use of language programme: An evaluation of two social skills interventions for children with high functioning autism and asperger syndrome. Journal of Autism and Developmental Disorders 38(10):1944, DOI 10.1007/s10803-008-0590-6, URL https://doi.org/10.1007/s10803-008-0590-6

Ozobot (2020a) Color Codes. URL https://ozobot.com/create/color-codes, [Online; Last call 10.August 2020]

Ozobot (2020b) FAQs. URL https://ozobot.com/support/faq, [Online; Last call 19.August 2020]

Ozobot (2020c) Ozoblockly. URL https://ozobot-deutschland.de/code-mit-ozobot/, [Online; Last call 10.August 2020]

Ozobot Deutschland (2020) Was ist der Ozobot? URL https://ozobot-deutschland.de/ozobot/, [Online; Last call 10.August 2020]

Pagano RR (2012) Understanding statistics in the behavioral sciences, vol 1. Cengage Learning

Page BR, Ziaeefard S, Moridian B, Mahmoudian N (2017) Learning autonomous systems — An interdisciplinary project-based experience. In: 2017 IEEE Frontiers in Education Conference (FIE), pp 1–7

Pahl A, Stadler-Altmann U (2019) MINT-Didaktik und Allgemeine Didaktik im Gespräch: Problemlösung und Differenzierung als Planungsprinzipien. Verlag Barbara Budrich

Papert S (1980) Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

lern psychologie (2020) Das Entwicklungsstufenmodell nach Piaget. URL http://www.lern-psychologie.de/kognitiv/piaget.htm, [Online; Last call 21.November 2020]

QuickStart Computing (2019) Computational Thinking. URL https://community.computingatschool.org.uk/files/8221/original.pdf, [Online; Last call 14.July 2019]

Ralph B McNeal J, Hansen WB, Harrington NG, Giles SM (2004) How all Stars Works: An Examination of Program Effects on Mediating Variables. Health Education & Behavior 31(2):165–178, DOI 10.1177/1090198103259852, URL https://doi.org/10.1177/1090198103259852, pMID: 15090119, https://doi.org/10.1177/1090198103259852

Román-González M (2015) COMPUTATIONAL THINKING TEST: DESIGN GUIDELINES AND CONTENT VALIDATION. DOI 10.13140/RG.2.1.4203.4329

Rose M (2020) K-12. URL https://whatis.techtarget.com/definition/K-12, [Online; Last call 13.November 2020]

Salkind N (2010) Encyclopedia of Research Design. URL https://methods.sagepub.com/reference/encyc-of-research-design, [Online; Last call 17.November 2020]

Shi, Jiaxiang (2020) Robot design and programming, provided by Mr. Jiaxiang Shi. Translated from Chinese into English

St Francis Xaver College (2020) SFX hosts Robocup. URL https://sfx.act.edu.au/sfx-hosts-robocup/, [Online; Last call 25.July 2020]

Standards NGS (2013) Science and Engineering Practices in the NGSS. URL https://www.nextgenscience.org/sites/default/files/Appendix%20F%20%20Science%20and%20Engineering%20Practices%20in%20the%20NGSS%20-%20FINAL%20060513.pdf, [Online; Last call 17.July 2019]

Study in the USA (2019) Understanding the American Education System. URL https://www.studyusa.com/en/a/58/understanding-the-american-education-system, [Online; Last call 20.July 2020]

Sunitha K (2013) Compiler construction

SurveyMonkey (2020) Einsatzmöglichkeiten für Fragen mit Likert-Skala. URL https://www.surveymonkey.de/mp/likert-scale/, [Online; Last call 17.November 2020]

Tarkowski P (2019) Digitalisierung in Schule und Ausbildung: wie sich der Nachwuchs auf die Industrie 4.0 vorbereitet. URL https://digital-magazin.de/digitalisierung-schule-ausbildung-industrie-4/?cn-reloaded=1, [Online; Last call 11.July 2019]

University of the People (2020) Understanding The Canadian Education System. URL https://www.uopeople.edu/blog/understanding-the-canadian-education-system/, [Online; Last call 28.July 2020]

Weintrop D, Wilensky U (2015) Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. DOI 10.1145/2787622.2787721

Weintrop D, Wilensky U (2017) Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE) 18(1):3

Weintrop D, Afzal A, Salac J, Francis P, Li B, Shepherd D, Franklin D (2018) Evaluating CoBlox: A Comparative Study of Robotics Programming Environments for Adult Novices. pp 1–1, DOI 10.1145/3170427.3186599

Wikipedia (2020a) BRG Kepler. URL https://de.wikipedia.org/wiki/BRG_Kepler, [Online; Last call 25.July 2020]

Wikipedia (2020b) The Hill School. URL https://en.wikipedia.org/wiki/The_Hill_School, [Online; Last call 20.July 2020]