PROCESS BOOK

CoderDojo

Liza de Graaf	500745856
Isabel Onrust	500756012
Remon Francois	500733483
Gian van Holt	500713631
Nicolaj Jørgensen	500793487
Maximilian Botta	500792732

Content

Team

01

Nicolaj

Currently studying at Business College Aarhus where i study applied computer science, I have experience in the languages C#, Java, Javascript, Jquery and a little bit with Python. Worked with node.js, ASP.NET MVC5 and QA tools such as Selenium and Microsoft Test Manager 2013

Max

Currently studying Business Informatics and Digital Media in Stuttgart, Germany.
Has experience in web-development, software-development, augmented reality, interaction and interface design. Currently started working with Arduinos.

Remon

Currently studying Cyber Security at the Amsterdam University of Applied Sciences. During this minor he wants to learn about the interaction between hardware and software.

Liza

Currently studying Computer Science at the Amsterdam University of Applied Sciences. Experienced in Java and app development with Java in Android Studio. Also experienced in C, C++ and a little bit of C#.

Isabel

Currently studying Game Development at the Amsterdam University of Applied Sciences. Mainly working with C#. Learned a lot of things concerning how to make a game, from design to implementation.

Gian

Currently studying System and Network Engineering at the Amsterdam University of Applied Sciences. Worked with HTML/CSS, PHP and Javascript for web developmentduring his internship and free time.

Introduction

02

Preposition

In the second semester of school year 2017-2018 we received an assignment for the Internet of Things project. Our assignment was provided by the Coderdojo organization in cooperation with the University of Applied Science in Amsterdam.

We worked in a multidisciplinary team of six students at the University of Applied Science in Amstrdam. We achieved our final prototype within five months. This process book surves the purpose of giving you some insights into our thought process, which lead to the final product.

Assignment

We received an assignment from the CoderDojo organisation. The current problem relays on the fact that hardware, that is especially designed for children to help them programming is too expensive. Our goal is to design an affordable robot that can be built and programmed by young children.

Vision

Our vision is to build an effordable robot, so that every child is able to buy one, independend from his or her background. Furthermore, the robot must enable children to get a basic understanding about programming in a fun way.

CODER(DOJO);

onderdeel van
CoderDojo

Company

CoderDojo is an organization built by a global network of volunteers, that aims to teach children to code and familiarize them with the digital technology.
At this very moment there are 85 CoderDojos in the list of Dojos on the Coder-Dojo website (https://coderdojo.nl/).

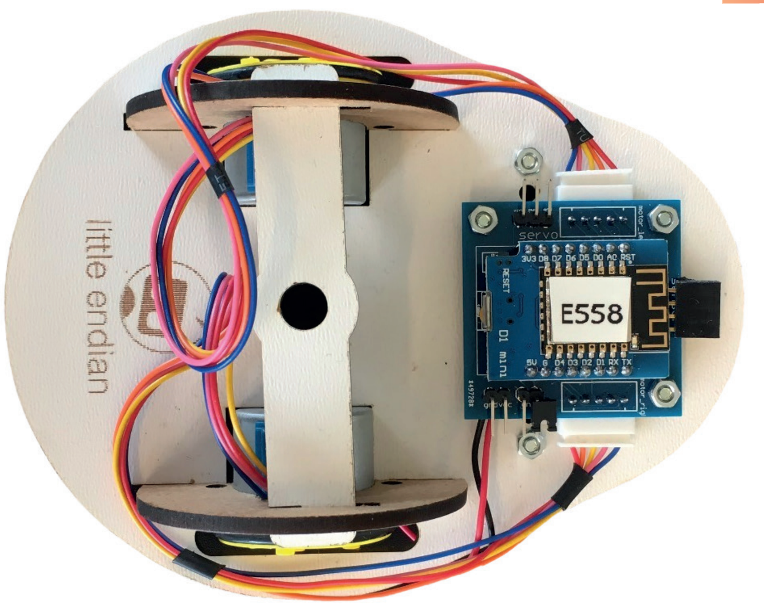
The CoderDojo organisation was founded by two individuals, James Whelton and Bill Liao. The two formed the organisation in July 2011 in the city of Cork in south-west Ireland. The first ever CoderDojo event took place a few weeks later on the 23 of July 2011. In less than a year CoderDojo had already spread to other cities across the globe, such as London in England and San Francisco in The United States. To this day their headquarters are still situated in Ireland as they are operating from Dublin.

Due to the rapid growth of CoderDojo co-founder James Whelton established a foundation to help support the CodorDojo. Established in 2013 the CoderDojo Foundation supports Dojos all around the globe. Their operations range from sourcing new partners and mentors to facilitating the communication and collaboration across the movement.

The only service CoderDojo is supplying are the Coding events that they organize. They don't profit from these and as a grass-root organisation they are dependent on the local effort of the places where the Dojos are being held. CoderDojo is also being supported by the aforementioned charity called the CoderDojo Foundation.

Product

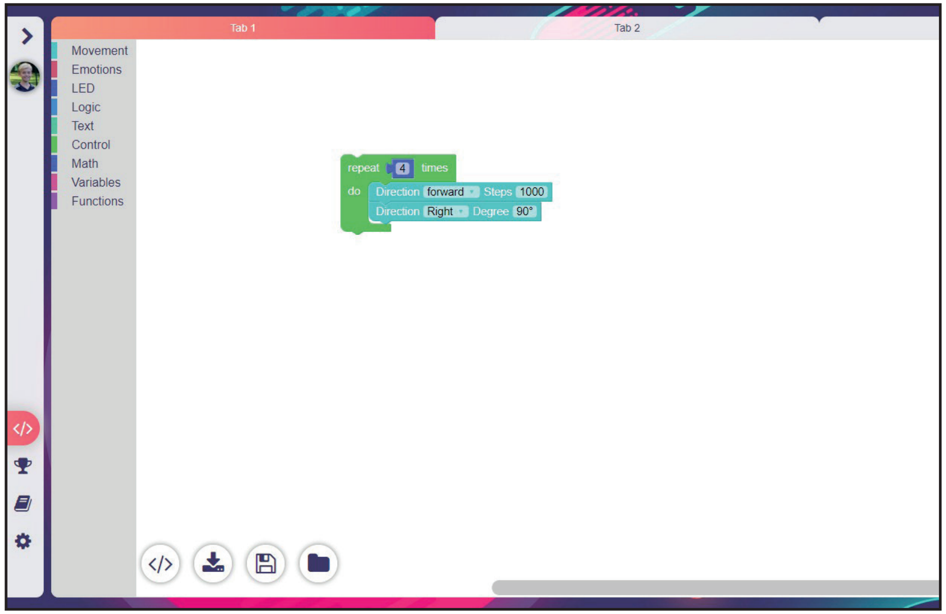
03



Product

The product is made out of MDF (wood) which we laser cut using our own design. We made it out of MDF because of the findings that met our requirements, ideas, critical probes and concept sketch.

The brain of the robot is a Wemos D1 mini microcontroller. The Wemos is extended by a PCB designed by our product owner Bas Pijls. The Wemos and extension board are powered by a battery pack on the bottom of the laser cut body. The extension board will communicate with the stepper motors, so it is able to move in the desired direction. On the top of the robot, we have a mounting plate for a pen. A pen or pencil can be placed in the hole, so the robot drive and draw at the same time.



We build a web app to manage the robot and offer challenges that can be completed. A storyline is implemented, so children can complete their challenges in a fun way. Each challenge will improve the users way into thinking like a real programmer.

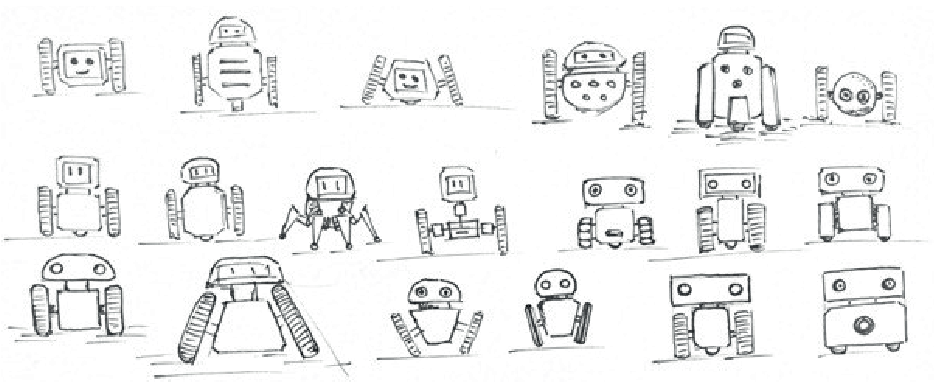
We made the product affordable, because we use inexpensive hardware. The robot can be assembled for €7,50. This makes it possible to hand out a robot for each visitor at the Coderdojo.

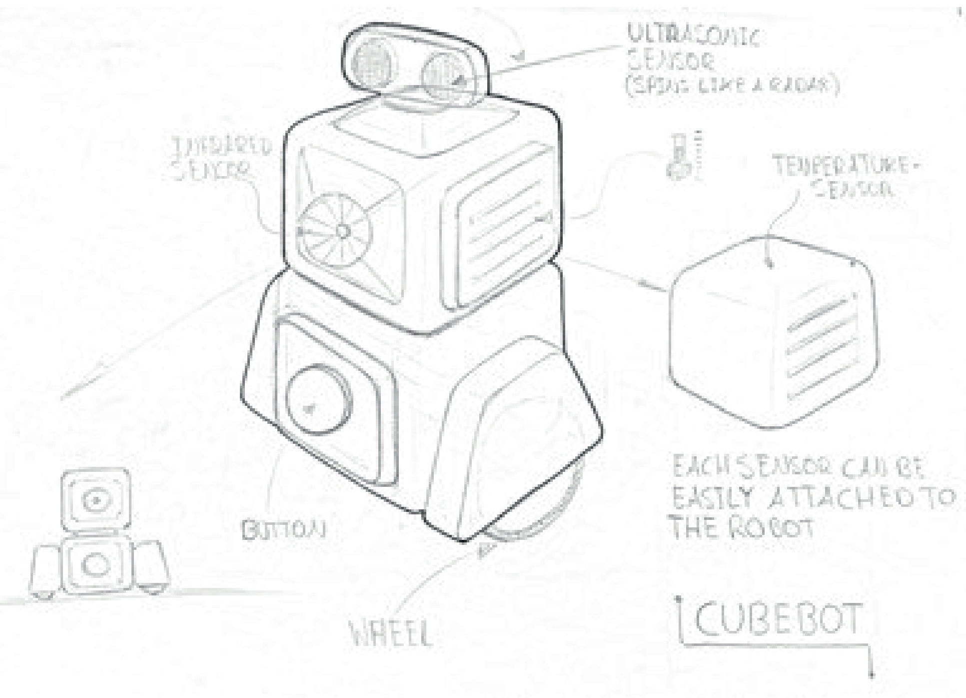
Ideation

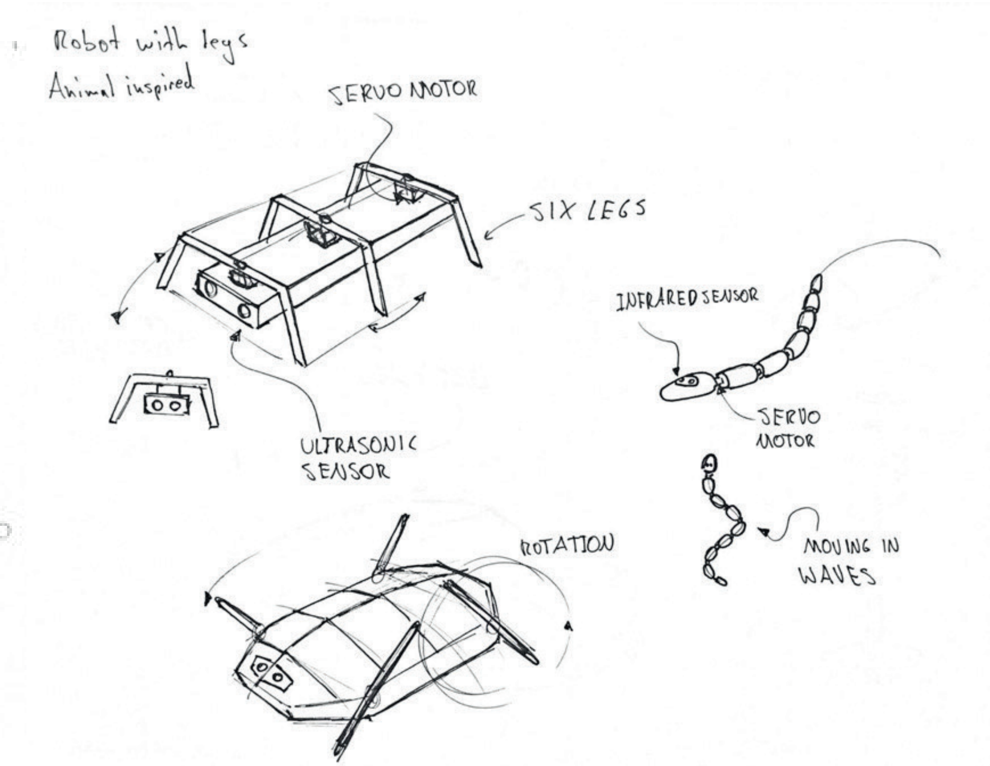
04

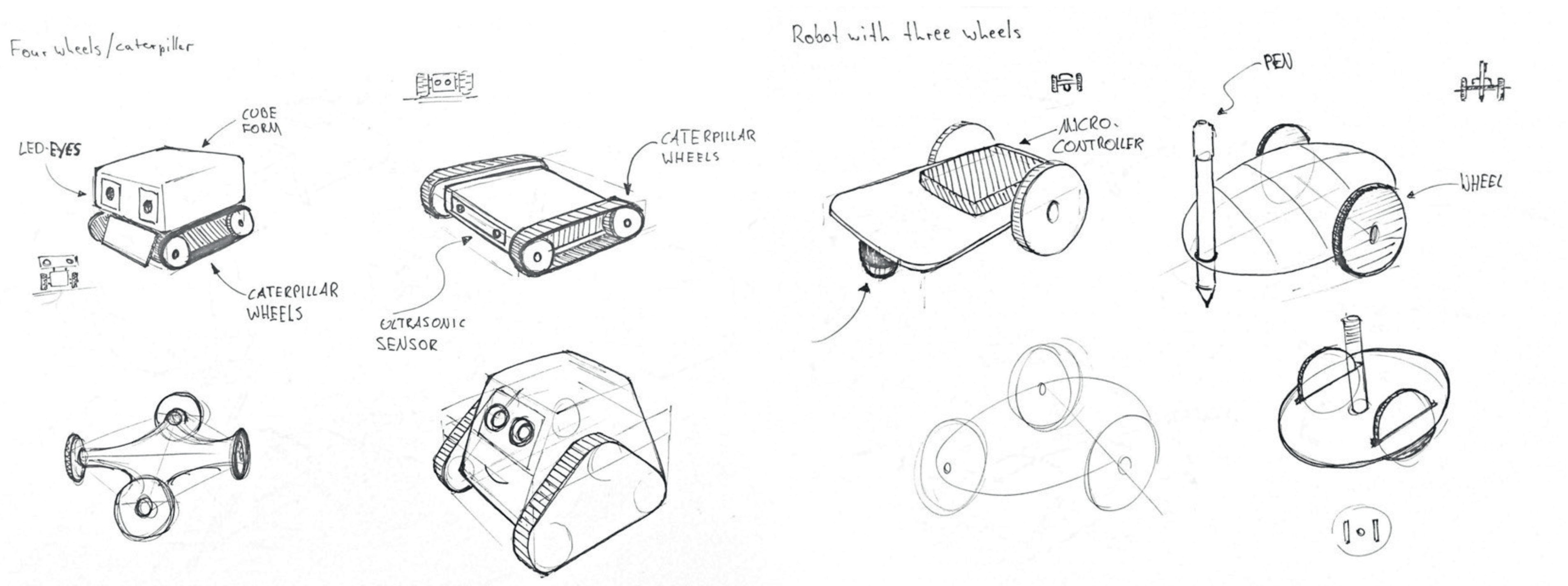
Concept Sketches

During the first weeks of the project, we had to come up with different ideas for the product. Therefore, everybody had the task to think about some ideas and visualize them by making sketches. There were a lot of different approaches to the product. Among other things, we had the idea of 3D printing modified Lego pieces and equip them with electronic components. Furthermore, we thought of a cube shaped robot, which could be attached with different sensors. Another idea was to build a robot with legs, which would use two servos to move around. At the end we came up with a more simple solution, which fits the purpose of the project a lot more.




ULTRASONIC
SENSOR
(SPINS LIKE A RADAR)
INFRARED
SENSOR
TEMPERATURE-
SENSOR
EACH SENSOR CAN BE
EASILY ATTACHED TO
THE ROBOT
BUTTON
WHEEL
CUBEBOT


Robot with legs
Animal inspired
SERVO MOTOR
SIX LEGS
INFRARED SENSOR
ULTRASONIC
SENSOR
SERVO
MOTOR
ROTATION
MOVING IN
WAVES


Four wheels/caterpiller
CUBE FORM
LED-EYES
CATERPILLAR WHEELS
CATERPILLAR WHEELS
ULTRASONIC SENSOR
Robot with three wheels
MICRO-CONTROLLER
PEN
WHEEL

Target Group

The target group we were supposed to design our product for, are children between the age of 7 and 12. The CoderDojo, which we were designing the product for, mainly teaches children in that age group. The goal was to get an insight into the thougts of children. It was quite hard for the group at the beginning, because young children differ a lot from adults, which makes it tough to think as the target audience.

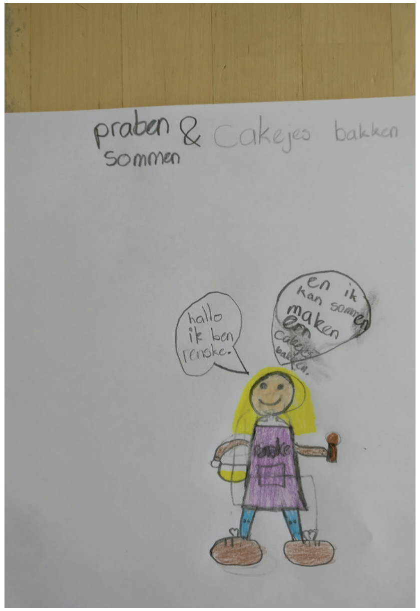
To get some further insights, on what children like and how they behave, we went to two CoderDojo workshops and spend a day at a day care for young children. Going to these events, gave us a lot of ideas, which we applied to the robot design. During the visit at the day care, we gave a workshop for the children to get an insight to their conception of robots. Therefore, we let them all draw different robot types. The children enjoyed the event a lot and we had a chance to see what they expect from a robot.

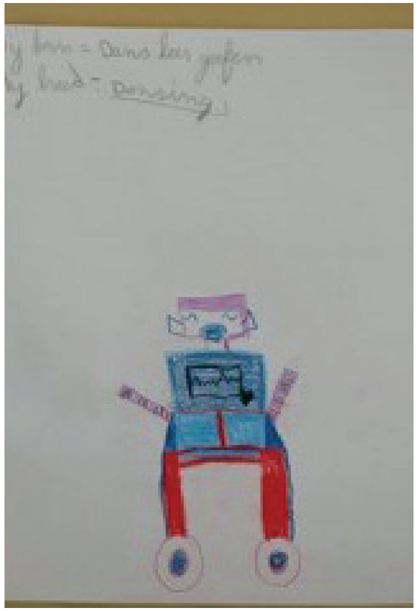

MATCH 2008


STIFTEN

Draw your


naam koekje
zingen


praben & Cakejes bakken
sommen
en ik kan sommen maken en Cakejes bakken.




naamRaket


Dancer

own robot!

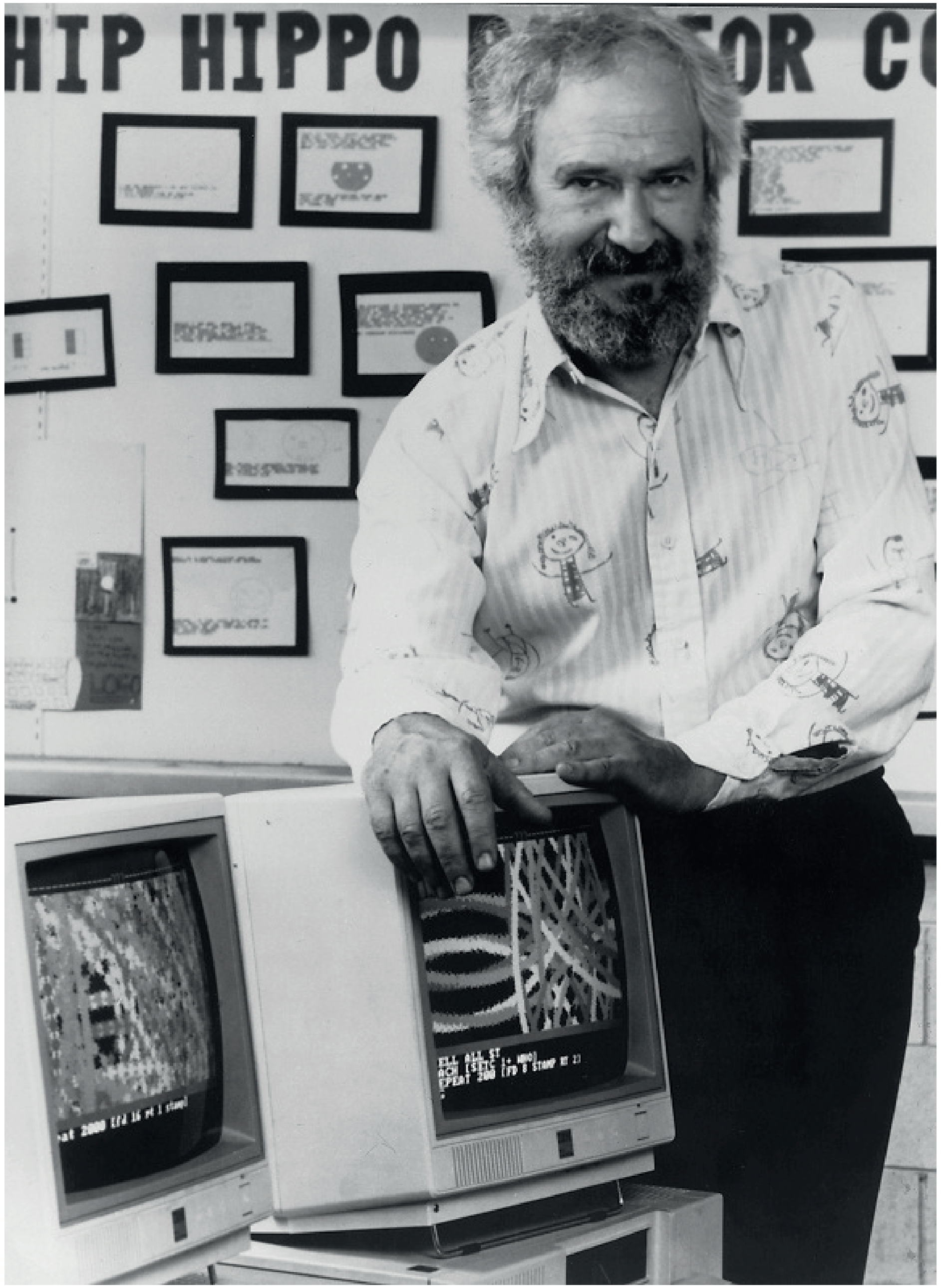
Constructionism

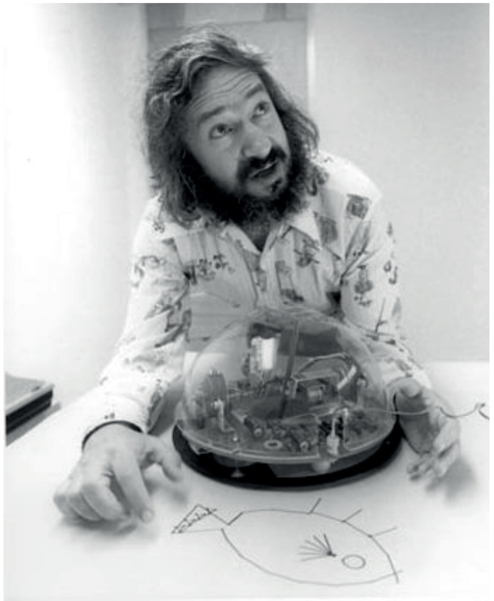
Seymour Papert introduced the idea of constructivism, which means, that children should learn by actively working on a project rather than just memorizing prepared content. Enhancing children to make failures and giving them the chance to try out different approaches will teach them how to fix problems and to make decisions rather than just following a well-known path. He also states, that children learn by themselves and develop intellectuality by themselves.

In his book Mindstorms Seymour Papert mentions, that computers can change the way of how we educate and what we can teach to children. It will enable us to adapt new learning techniques like constructivism. For example, instead of just memorizing vocabularies, people could use a software, which lets them talk with a bot or another person speaking the language. Learning by communicating in the desired language, will be much more natural, not only to children.

When Papert introduced those ideas 40 years ago, it was hard to think of a computer as an educational instrument, imagining that computer would need a whole room and weren't accessible to most people. Today, we can produce small computers very cheap and nearly every person has some sort of computer. Having the possibility of buying cheap microcontrollers or small computers like the Raspberry Pi enables us to create a low budget robot, easier than ever before.

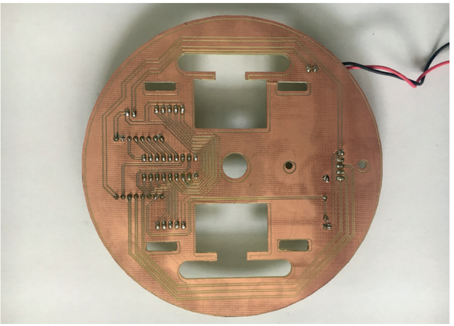
For the robot, we will use the idea of teaching math with geometry. Papert development a turtle robot, which was able to draw. We will also include this function into our robot. After talking to children, something simple like drawing geometric shaped, is still very interesting to children. It will motivate them to develop algorithms, which will draw certain shapes. The motivation therefore will come from inside rather than from outside through teachers.

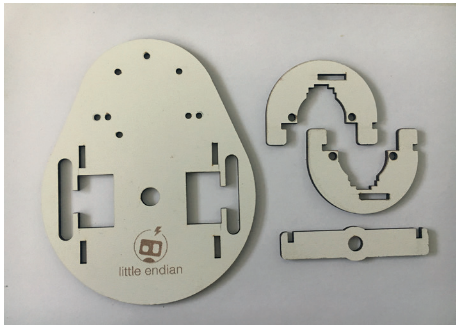




Material

The focus point of our project is to make it affordable because of this we wanted to use a material that reflects this ideal. We immediately thought to use cardboard for the majority of the robot. If done right, cardboard can be a flexible, cheap and sturdy material. So we all agreed that we would be using that. However further down the design process we ran into some problems. So design choices conflicted with our material. After a few prototypes we came to the decision that we had to let the idea of cardboard go. After this we looked into a few other options and found that wood would be a fine replacement. In the end we went with medium-density fibreboard.

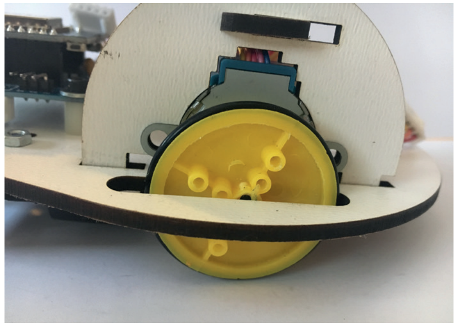


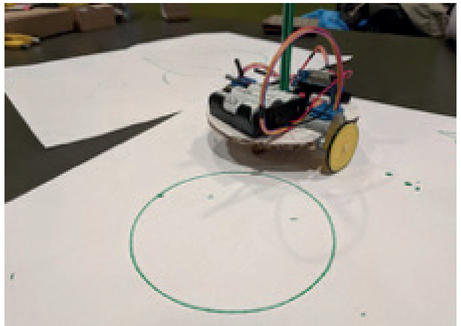


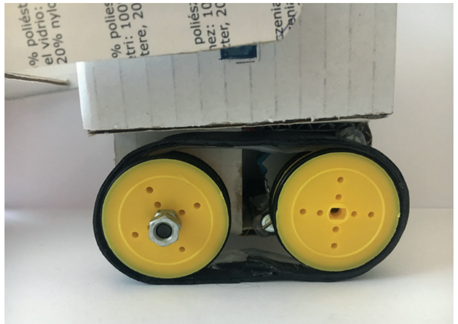
Movement

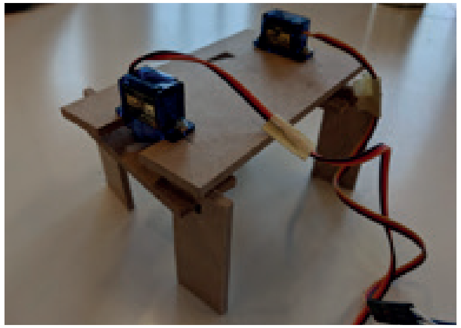
An important aspect of our robot is the movement. One of the easiest ways to reflect a child's code is by showing it in the movement of our robot. So when it came to deciding how our robot would move we had a few ideas. In the concept stages of our product we thought of a lot of different things, for example: a robot that moved like a snake, a robot that walked on spiderlike limbs, etc.
A few of these ideas stuck with us namely the use of caterpillar wheels, normal wheels and to use 4 sticklike limbs to make the robot "walk". To use the sticks we had to make use of servo motors, for the other two ideas we had to use stepper motors.

Servo motors have the advantage that they move a lot faster than stepper motors however servo motors have to compromise on movement.
Because we wanted to integrate drawing into our robot we knew we would need a lot of precise movements, so in the end we went with the stepper motors.
Then the choice came down to caterpillar wheels and normal wheels and this choice was in fact fairly easy. Normal wheels are cheaper than caterpillar wheels so in the end we ended up going with the normal wheels.



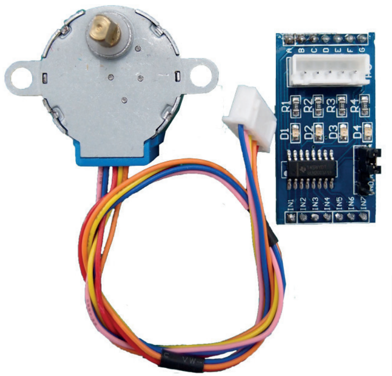




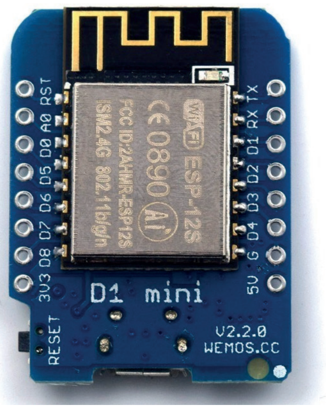


Sensors

At the start of the project we planned on using a lot of sensors but in the end we didn’t put any on our end product, most of these were scrapped due time constraints. We wanted to include a temperature sensor and have a few challenges revolve around displaying sensor data onto our little LED matrix. Seeing as we haven’t gotten around to implementing the challenge’s, this idea was also thrown away. We also planned on using an ultrasonic sensor to enable the robot to maneuver around obstacles in its way. This idea was scrapped due to the sheer complexity of the task and again the time constraint. So in the end we decided that we would not clutter our robot with unnecessary sensors if we weren’t going to use them anyways. In our end product we ended up scrapping all sensors.




1088AH


ESP-12S
D1 mini
RESET
V2.2.0
WEMOS.CC


Vcc
Trig
Echo
Gnd

Obstacles

05

Costs

One of our biggest obstacles was that we had to make the robot affordable. We've researched other products that are like ours, like the Lego Mindstorm, Anki Cozmo and Dash & Dot. But those products are all above 50-100 euros. Because we want every kid to be able to learn how to program we want to make this as affordable as possible.
Sensors can be cheap but we also want quality. We had to research some more into where we could get cheap sensors but still have some good quality. Eventually we decided to order most of the parts on Aliexpress.

Design

We also had to think about the design of our robot and website. The robot has to be child friendly and both the robot and the website need to be easy to use. In order to do this we had to do some more research. Our robot is rather slow, this is because we wanted to make the robot more precise for drawing.
For the website we had to know more about what the children like best. How will they use the website, does it have to look colorful/playful. We looked at games for children between 7 and 12 (our target group) and saw that all those games looked very colorful and the buttons are of a decent size.

Hard/Software Communication

For the communication between the robot and the web application, we decided to use the Espruino firmware on the Wemos microcontroller. The reason we chose the Espruino firmware is because, everything on the website is JavaScript. JavaScript is an interpreted language so we don't have to compile it everytime time we want to upload the code to the robot. This causes the code to be uploaded fairly fast without any delay.

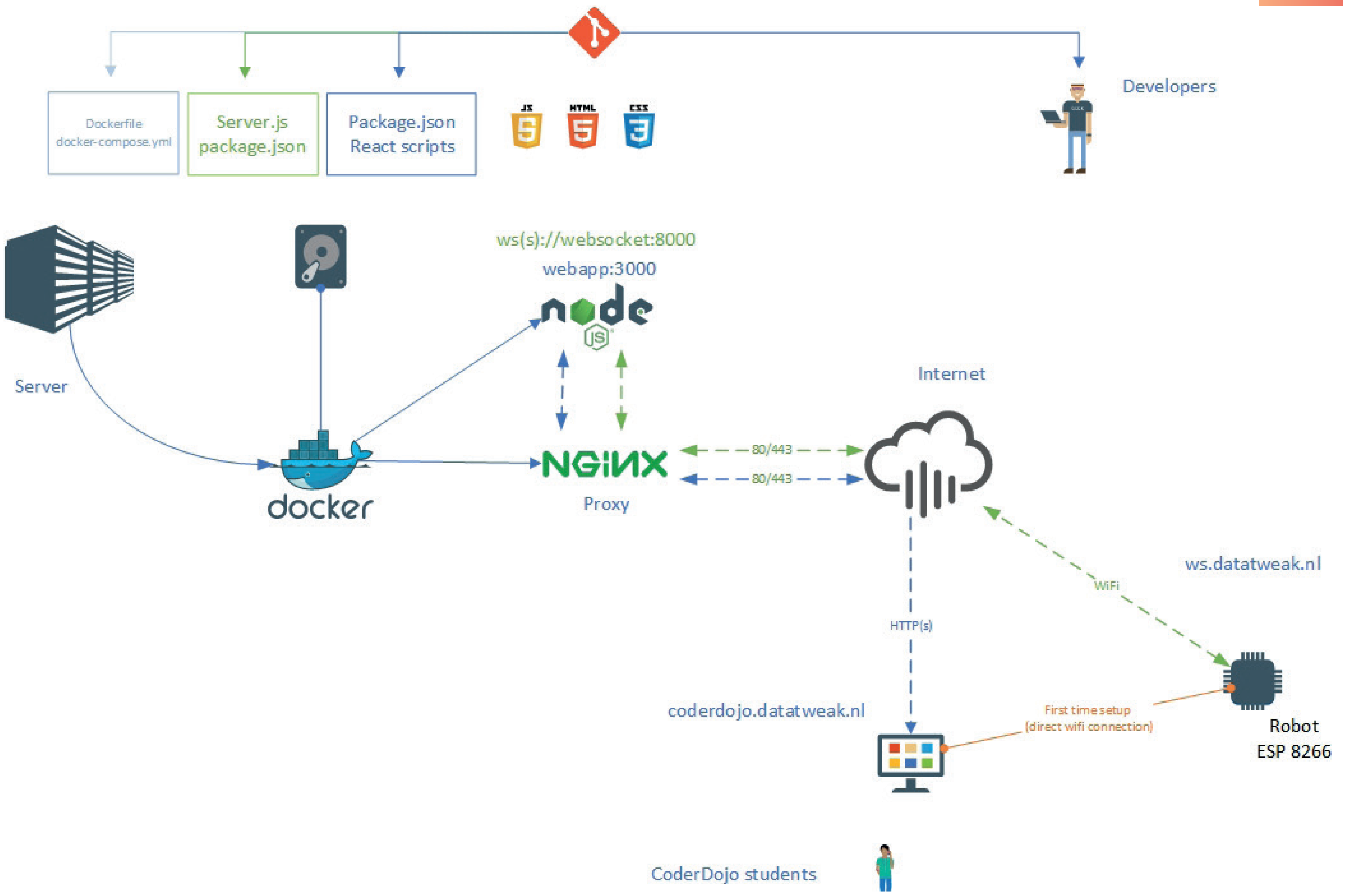
To create a connection from the board to the internet, we decided to use a Wi-Fi manager with a captive portal. The captive portal consists of html form which contains the nearby SSIDs and an input field for the password. We had to make a lot of changes to the code of the WIFI-manager. Some of the functions and procedures that we added to the Wi-Fi code are: creating an input field for the username to the form, creating a unique identifier on the captive portal that is used to identify the user on the web application, saving the unique identifier on the board, and adding a connection to the WebSocket server after gaining access to the internet.

For the server we are currently using an open source server environment called Node.JS. Node.JS is build on a JavaScript engine which makes it easy to build a network application. We came up with the idea of using Node.JS, after brainstorming with the group, about the infrastructure that we would create for the robot and the web application.

To develop the web application, we used a JavaScript library named ReactJS, which is a JavaScript library that is created to build user interfaces for an application. After we killed our darling (the electron desktop application), we were brainstorming about, which framework we should use to create the web application. The 3 frameworks we had in mind where, ReactJS, AngularJS, and WUI JS. After getting advice from a teacher that was in the makers lab and doing some research about the differences in the 3 frameworks, we decided to go for ReactJS. ReactJS is simple, easy to learn and has a great testability so it wasn't that hard of a choice.

We use Docker on a physical server to serve different containers. In Docker it is possible to build containers that run software. We build three containers. The first container is running a Node.js environment which contains a websocket which is programmed to handle incoming connections from the robot. The second container is running the Node.js webapp environment. The third container is running a proxy server for security reasons. The proxy server is exposed to the internet and handles requests to the correct container, which are running in a dedicated network behind the proxy.

For continuous integration we enabled the GitLab a connection to our websocket and webapp files, so we can edit them and change when needed.

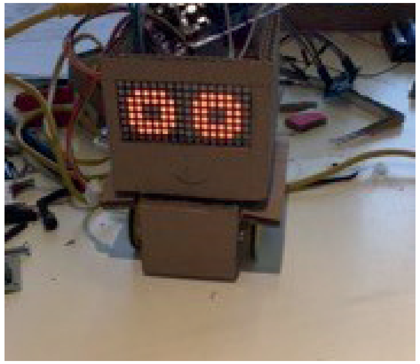

Dockerfile
docker-compose.yml
Server.js
package.json
Package.json
React scripts
JS
HTML
CSS
Developers
ws(s)://websocket:8000
webapp:3000
Server
Internet
80/443
80/443
docker
Proxy
ws.datatweak.nl
WiFi
HTTP(s)
coderdojo.datatweak.nl
First time setup
(direct wifi connection)
Robot
ESP 8266
CoderDojo students

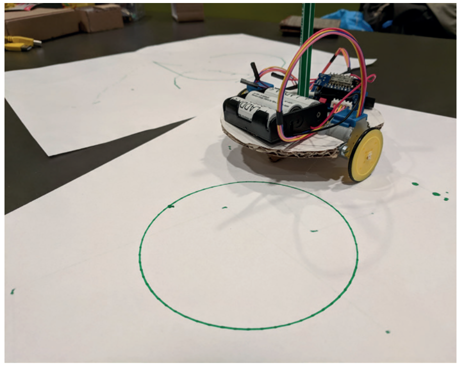
Prototyping

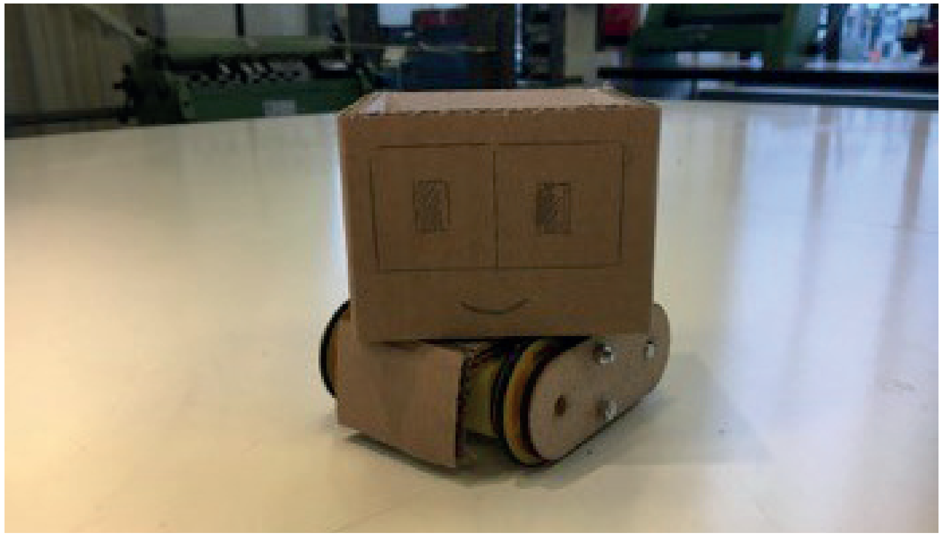
06

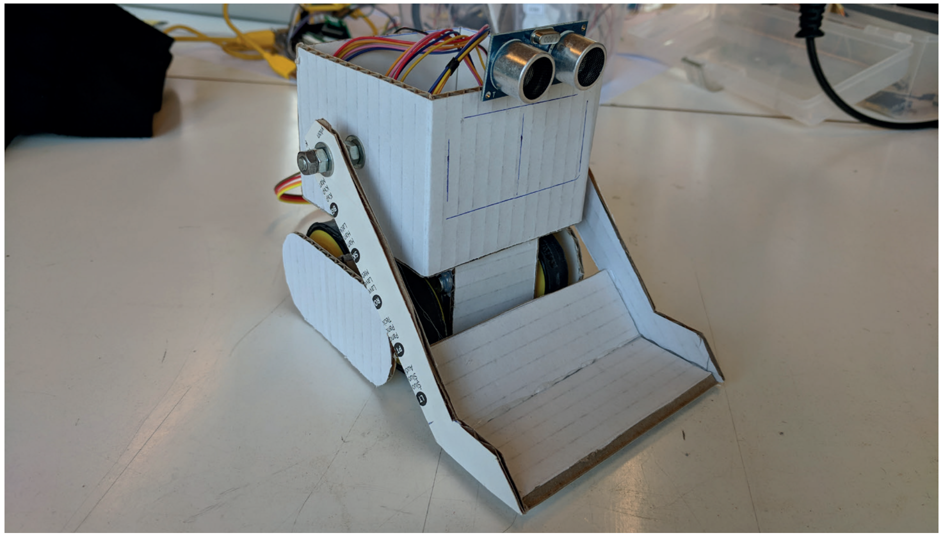
Cardboard

When we first started prototyping, we had the idea of building a robot out of cardboard. Cardboard is lightweight, stable and low priced. The idea was to lasercut the pieces out of a cardboard plate. Our first prototypes were folded and put together with hot glue. After a while we realized though, that cardboard is not as durable as we hoped it to be. Carrying it around in a backpack, damages the material. Therefore, we decided not to use only cardboard, but instead building a frame out of a solid material like wood.



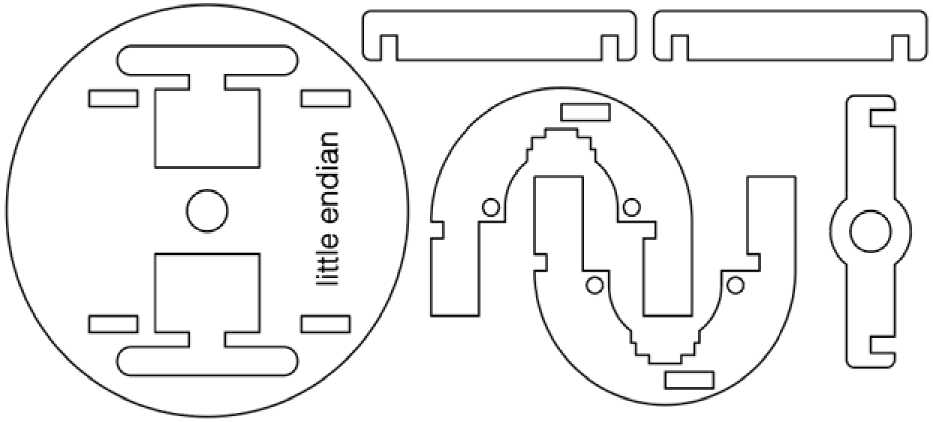


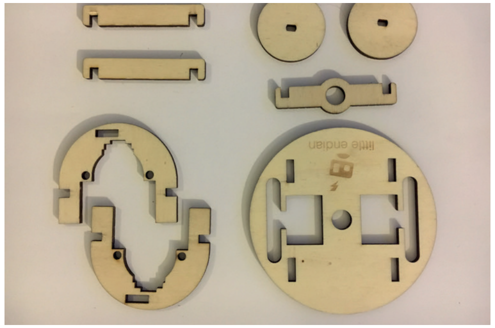


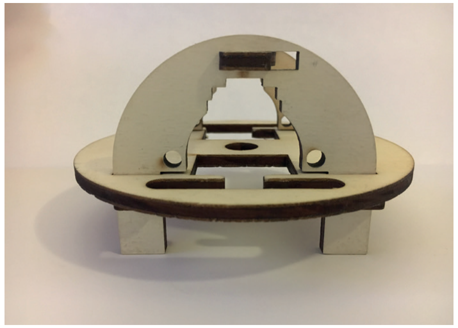


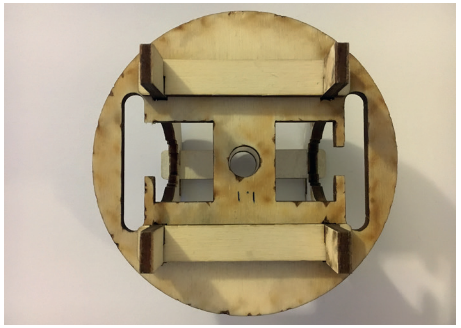
Wood

After we decided to use a solid material instead of cardboard, we decided to try out wood and mdf (medium-density-fibreboard). First we started out by using a multiplex wood material with a thickness of 4mm. The material has a very clean and natural look to it. Sadly the material is not as stable as expected. Some parts were breaking due to high pressure, when connecting the pieces to eachother.

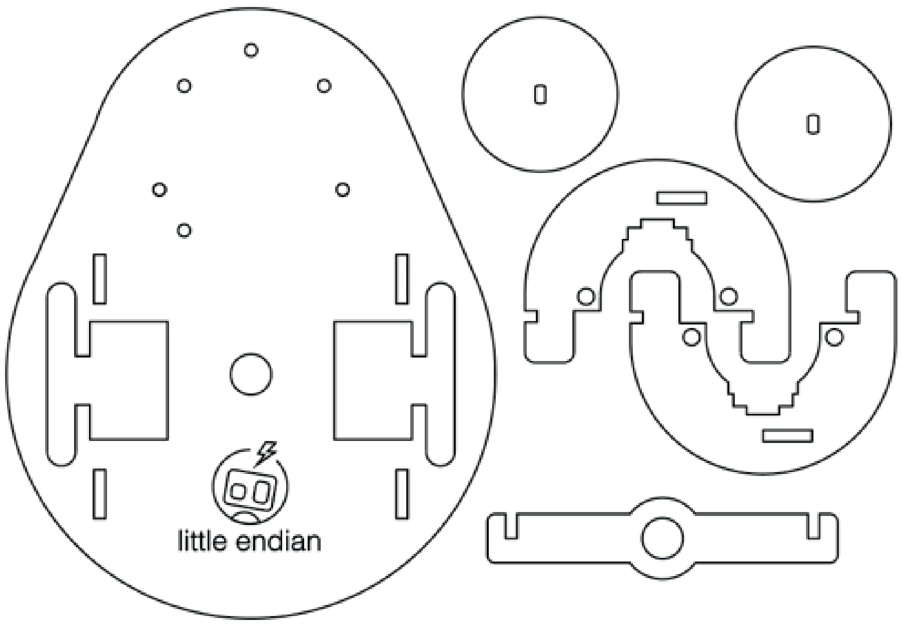




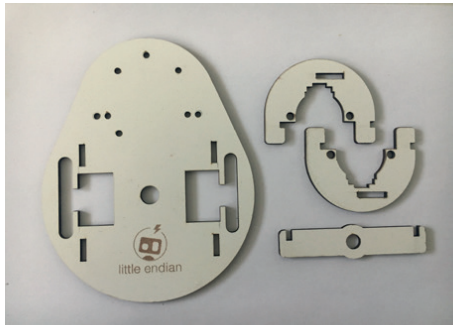


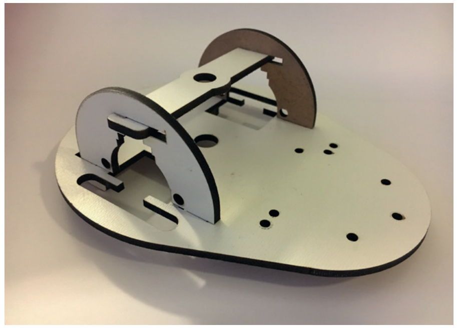


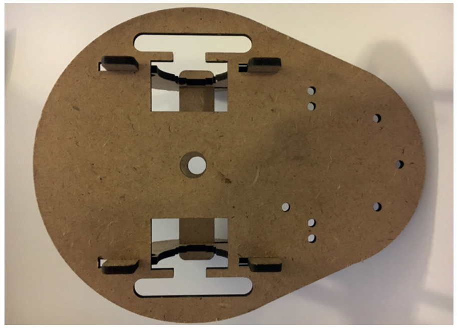
MDF

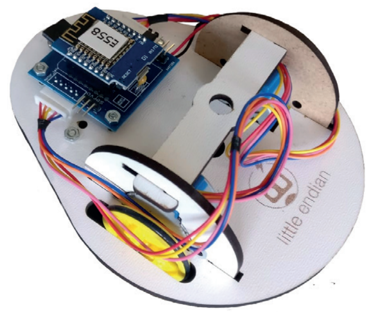


The next version of the robot frame was amde out of Medium-Density-Fibre-board, also called MDF. The material is low priced and has a high stability. The MDF plate had a thickness of 3mm, which is optimal for the robot frame. The top side of the material is white, which gives it a appealing look. After we lasercutted the prototypes, we decided to stick with MDF, due to its durability. In Addition to the new material, we also made the bottom piece of the frame larger, so that the battery package would fit. Furthermore, we also reduced the amounts of parts needed to assemble the robot. We could do that, by using a sliding technic.



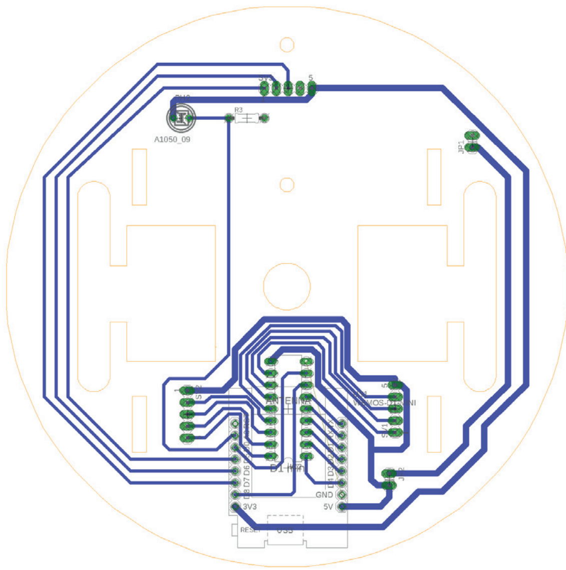


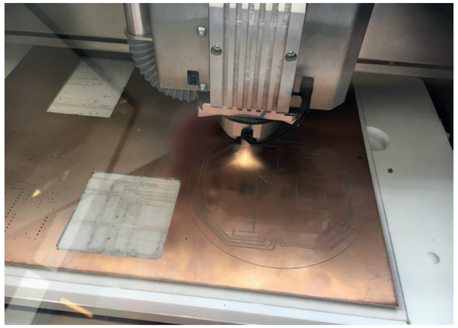


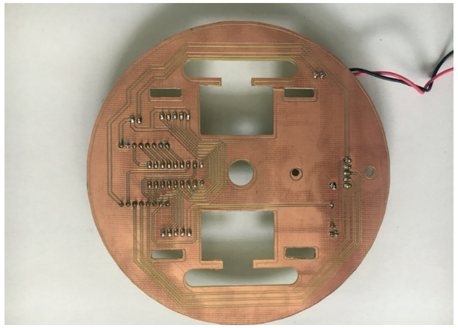


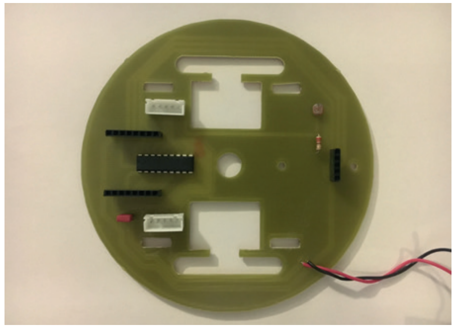
Circuit Board

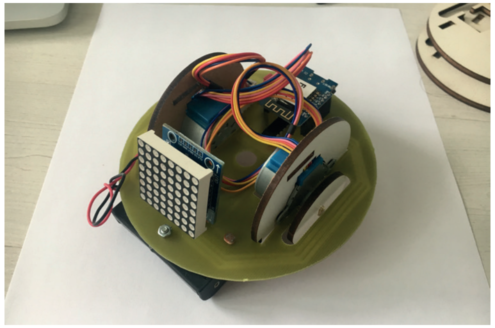
During the process of prototyping with different materials, we had the idea of making our own circuit board, which would be a part of the robot. This means, that all the necessary hardware parts would be already integrated into the pieces. Therefore, we tested different circuit boards, starting out by manualy soldering a circuit board to actually making one with a CNC machine. After tweo to three attemps, we had a working circuit boards, which was the bottom plate of the robot. Due to a limited number of pins available on the microcontroller, we could only attach a 8x8 LED Matrix.





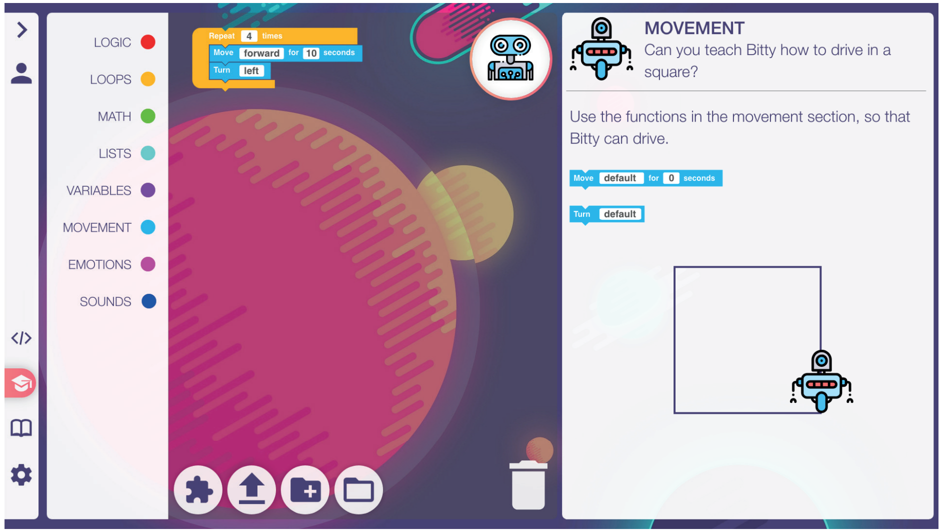


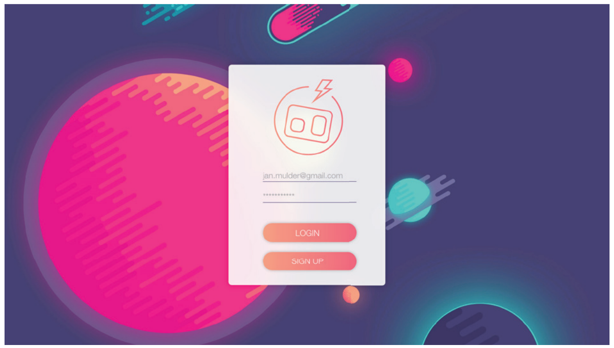




Web-App Mockup

The second part of our project is the web-application, which will be used to program the robot. We are using the programming language Blockly, which was developed by Google in cooperation with the MIT. Blockly is a visual programming language, consisitng out of blocks, that have to be attached to each other. Furthermore, Blockly was created especially for younger people or people, who are completely new to programming. The purpose is to teach the basics of programming and to enable an understanding for programming and the logic behind it.





Login Screen

Programming Environment

Challenges

Programming

07

Robot

The Microcontroller, which we are using for the robot is a Wemos D1 Mini. The programming language we are using to program the chip is JavaScript. Using JavaScript is possible due to the firmware Espruino, which we had to flash the microcontroller with. Using JavaScript made it possible, to easily send and evaluate code on the robot. To give the robot functions like driving, we had to prepare all the functions and save them on the board. This way, we can send code, which executes functions that are already on the microcontroller.

```
/*This function handles the queue for
all the queued up movement. functi-
ons.*/

function callNextFunction(){
  if(queue.length > 0){
    console.log("running: " + queue[0]);
    eval(queue[0]);
    //queue[0]();
    queue.shift();
  }
  else{
    console.log("done");
  }
}
```

```
/*Here is the move function, we also-
have a turn function that works in the
same way*/

function move(direction, steps){
  if(direction == 'forward'){
    queue.push('forward('+steps+')');
  }
  else if(direction == 'backward'){
    queue.push('backward('+steps+')');
  }
}
```

```
/*The move forward function, we also
have backwards, right and left which all
works pretty much in the same fashion.*/

function forward(steps) {
  console.log(steps);
  print(„forward“);
  motorRight.moveTo(motorRight.getPo-
sition()+steps, undefined, callNextFunc-
tion);
  motorLeft.moveTo(motorLeft.getPosi-
tion()-steps);
}
```

Web-Application

We started out making a web app in plain JavaScript. We didn‘t know what exactly we wanted to do on the web app or how it would work so the web app was mostly just being used for trying stuff out such as different ways to send commands to the robot. We knew we wanted to use Blockly and have challenges so that was the first things we added to the web app.
The first version of the web app send commands to the robot by sending a long url, it made the robot move but it was not the right approach, url’s can for example get too long and the robot are locked to the few commands that are coded on the ESP.

Right before the midterm we discussed creating a new web app because the first one we did was starting to become too hard to build on and because we did not use a well known framework we just made it in plain JavaScript. After the midterm, we agreed that we would make a new web app using a well known library.

We researched different library/frameworks such as Vue.js, angular and react js. We had some talks with some other students and they really liked React and got good reasons to why it’s a good framework to use. We went with React because it has a virtual DOM which creates high performance, it has its own language JSX which enables you to write JavaScript in the “html code”, it adds a structure to the webapp, it has a lot of libraries that can be reused, and we can use 3rd party modules such as blockly or firebase.
The biggest reasons that we chose React was because of the big community and that you are able to use React native, so in the future it‘s very easy to take the web app code and make a phone app using the same code.
In the end it didn‘t really matter which one we chose, we think either of them would have been a good choice.

Connection

At first we just used a usb cable but we talked with our product owner and he really liked the idea of being able to program wirelessly, we thought about using bluetooth but since not all laptops have bluetooth we stuck with wifi instead. We found Espruino and after that we were able to do it over wifi, we wanted to connect directly to the robot over wifi but our product owner did not like that idea. He wanted us to connect the robot to a web app and he suggested using websockets, we researched websockets and we liked the idea so we went with it.

On the server we have a websocket server. We connect both the robot and the web app to the websocket server and then when you send code from the web app the server finds the connected robot and sends the code to the robot.

We handle the connection between the two by having 2 arrays on the server. The objects in each of the arrays contains the username of the user and the websocket connection. So when you send something from the web app the server will search the robot array and find the matching username and send the code using the websocket.

We are also using a heartbeat (ping, pong) to keep track of which connection are alive. Each 5th second the server will send a ping thereafter the client should send a pong. If the client miss 3 pong's it will be disconnected and removed from the servers array. We do this to make sure there are no lingering connections in the arrays.

Usability Test

08

Robot

The usability test took place at a CoderDojo workshop in Bussum. We just randomly picked different children to test our product. We tested the product with 7 children, to get an insight, what has to be improved. The goal was to see if the children could assemble the robot in less then 10 minutes.

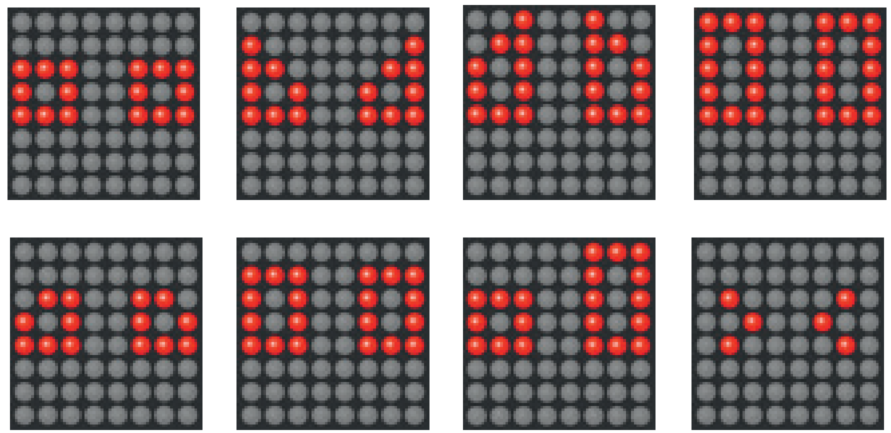
We gave each participant all the pieces, which are needed to assemble the robot. We asked different questions before the test started and our observer filled out a form, which was prepared before the test.

Findings

It seems to be quite complicated and not understandable to most children, on how to properly attach the stepper motors. Most participants first assembled the frame and tried afterwards to attach the motors. We will probably have to work with some sort of icons to indicate where the parts belong, or create a manual to display the steps to assemble the robot.
The pen holder got often attached, after the robot was almost build together. It is easier tough, to attach the pen holder as one of the first steps.
Some of the children broke mainly the part to hold the motor in place. There was a failure in the thickness of the part at a certain point. This need to be redisgned to guarantee stability. One of the volunteers gave us the idea of including icons onto the robot pieces. This would make it more understandable for the children to assemble the robot.

Faces

The second test was to see how children would react on different facial expressions on a 8x8 LED-Matrix. Therefore we showed each of the following faces on an LED Matrix and ask the children how they feel, when they see it.



Findings

The joy and disgust face were hardly recognized by the children. Those need to be redesigned.
There was a high interest in the facial expressions and it seems, that the children would highly enjoy this function.
One of the volunteers and some of the children mentioned, that we should animate the eyes, to make it more interactive.

Future Prospects

09

Storymode

Sadly we didn’t get to implement the story mode, the story mode is a very important part of the project. It’s what we think makes the product a lot more fun for the children and the main part of our product that teaches the children the basic (and maybe even also advanced) concepts of programming.

Wifi Manager

As it is right now the product is not functional enough because you have to hardcode the username and wifi credentials you have for the robot you are currently using. This is not something kids can do and also you need an adult with actual coding skills to use the robot as it is right now.

Sensors

We encountered an annoying problem when we tried to implement the speaker, LED matrix and the ultrasonic sensor. We simply didn‘t have enough pins on our microcontroller board. We thought we did but it turned out we could not use a lot of the pins. The things are important since they add a lot of new functionality to the robot so you can make even more cool challenges for the robot.

Change Motors

Right now we are using stepper motors and they are very precise but also very slow. We fear that the speed of the motors might affect the experience when using the robot, we wanted to test out if we could use AC motors instead because from what we read they are still pretty precise and a lot faster.

Save/load Code

For now we can’t save and load the code in the web application. The save and load button are done, but we still have to add the code for the save and load functions. This idea would make the web application more convenient because you can simply save your blocks and load it up when you want to continue to work in the web application.

Profile Page

At the moment, the profile page on the web application is not working. We do have a hardcoded profile page that works as a mock-up for what would be the dynamic profile page. The hardcoded profile page as for now consists of: a static profile picture, a static username, a static rank, it also misses information about the challenges. We are planning to finish a big part of the profile page before the project ends. With at least the username that is linked to the profile, a profile picture that is linked to the profile and a timer in which you can see how long you have been logged in on the web application.

Security (TLS)

During development we came to the conclusion that the required websockets can’t make use of TLS. This extra layer of security is needed for two reasons. One of the reasons is that the data between the websocket on the server and the websocket on the client (robot) is unencrypted, so the data can be exposed by a man in the middle attack.

The second reason is related to our website. Because the websocket is linked to the website we can’t make use of HTTP(S). Secure HTTPS websites require secure websockets. The reason why we can’t use secure websockets is related to the microcontroller. The microcontroller don’t have enough memory to handle TLS. This will be critical because we expose the website to the internet and process sensitive data like email addresses and names.
We didn’t have enough time to fully investigate what we would need to change to enable the use of TLS, but we think you need to switch the microcontroller to one with more ram or maybe use something else than websockets.

Team Contribution

10

Remon

Did the backend of our infrastructure. Made the web app and websocket accessible from the internet. Worked a bit on the websocket software to make communication possible between the robot and the server. Did some coding for the first time robot setup and applied design thinking methods at the Coderdojo events, which are applied in the prototype.

Reflection

During this minor I want to get familiar with the concepts of design thinking. I would like to apply the concepts on the project we are working on. This must result in a deliverable prototype.

During the Creation And User Design course I get familiar with different tools to make design thinking possible. The product we had to make relies mainly on user interaction. Because the robot has a job, to learn children the programming logic. To make this possible we applied and combined a lot of tools from the mediaLAB Amsterdam in our findings to improve the final prototype. We presented our product at different events and people liked it very much!

Nicolaj

Did the research about Espruino, coded most of the Espruino stuff we use for our robot, set up the firebase database for the login and authentication, wrote most of the code for handling the websocket. Acted as the scrum master and project leader.

Reflection

I want to get better at prototyping since it is not something I have done much and learn about the different design methods and how they can help me to improve a product.

I think I learned a lot about prototyping, I learned how it helps you come up with new ideas and that you should not be afraid to try out things that I otherwise would think was a bad idea, from this I learned that sometimes bad ideas brings good ideas.

Gian

Did research about the Espruino WIFI-manager, Electron desktop application, Node JS local server and a little bit about React. Worked on changing code for the WIFI-manager and testing this out on both the Wemos and NodeMCU, did a little bit of work for the first web application(sidebar), imported the old web application into Electron, put up a simple local Node.js server for testing purposes that could receive messages from the microcontroller and worked on the Profile page for the React web application.

Reflection

During the project Coderdojo, i wanted to focus on making an internet of things product that is useful and meaningful to users. I wanted to focus on usability of the product and the design of the product.

I learned alot about making a product that is useful and meaningful to users, specially because the product that we made are for kids, so we really had to design everything in a perspective of the target audience. We had to take into account that we don't save to much personal information on our server, that we designed the robot to be safe and that we kept everything simple. To do that we interviewed children and spoke to other teachers about how we could accomplish creating a safe and simple product while maintaining the functions that it needs.

Max

Mainly worked on the prototyping of the robot and was responsible for the user interaction design. Was developing different ideas and designs for the web-application as well as different robot prototypes. Spend most of his time in the makerslab lasercutting pieces or making circuit boards.
Besides the hardware prototyping he worked on the web-application side, by setting uo the environment and programming the navigation. Furthermore, he worked on the freecoding part which inherits the implementation of the Blockly programming language.

Reflection

During the process of devloping new ideas I learned about killing ideas and creating new solutions. At the beginning I had a pretty solid idea, of how the robot should look like and what functions it should have. The idea was to expensive and didn‘t fit the purpose of the project. Because I was spending a lot of time in the makerslab, I learned a lot about the machines I was using. I never used a lasercutter before or made a circuit board, so I had to teach it myself, with a little help from the people in the makerslab.

In addition, I learned something about analyzing the target audience, which was one of my goals for this semester. We spend quite some time with the target audience and got an idea of how they think. I also believe, that we created a meaningful IOT product, which will help children to learn about robotics and programming in a fun way and making is accessible for everybody.

Liza

Did research on Blockly and integrated it into the website together with Remon. Made a website prototype from scratch with Blockly integrated which was an important part in the process of creating our final website.
I've given advice about hardware on the robot, and I've worked on the story and challenges on the final website.

Reflection

By the end of this subject, I will have learned how to gather more information. I will have feedback documents from at least five different people.

I don't have feedback documents but from the children at the daycare we've "interviewed", we received a lot of drawings (some with text) showing their ideas on how they would like a robot to look like and function. With this information we could continue building our robot concepts.

By the end of this subject, I will have made multiple prototypes for our IoT project. Pictures will be placed inside of my process book.

I can't say I've achieved this. There are pictures of our prototypes in the process book but I wasn't the one who made them. The prototypes we've made are really neat and I've only contributed to them through the thinking process. But the actual assembly wasn't done by me.

Isabel

In the beginning we split the group in two teams, one for the web application and one for the robot. I worked together with Liza and Remon on the web application. I learned how to make a web application. After that I mainly worked on documentation and presentations and supported other team members with their tasks. Then I researched into teaching methods and how you should structure assignments to make the challenges for our application. I've also written story ideas for the challenges and the final story. Towards the end I also worked on the design of our robot and looked into how one can do such a task.

Reflection

I learned a lot about the thought and work that goes into making an IoT. I‘ve had an up close experience with the development process. Even though i‘m a lot wiser than i was at the start of the project I must admit I wish i‘d gotten myself a bit more involved with the internal process of the robot. I prioritized concern of other task within in the project over the making of the robot. I lost sight of what I wanted to achieve with this project. However, I was able to come back from this I bit towards the end. I took a long look at the project and said to myself that I still wanted to have something to do with the robot and I got involved again. For future projects I‘m actively going to remind myself of what I want to achieve. This way I hopefully can prevent what happened during this project in the future.